iStress: Stress Classification from Heart Rate Variability

Luke Plewa, *Student, Cal Poly San Luis Obispo*

Abstract—iStress is an Internet of Things solution to the busybodys everyday healthcare issue: stress. Through Heart Rate Variability (HRV) we can indicate level of stress. Furthermore, advancements in wearable technology and popularity have given us the tool to monitor our stress levels all the time. This project utilizes the Microsoft Band to send heart rate sensor data to an iPhone and eventually to a Google Cloud Server. Machine learning classifiers, implemented in Python and hosted on this cloud, are then used to indicate stress into binary classes.

Keywords—Machine Learning, Internet of Things, Stress, Boosting, LogitBoost, Classification

1 Introduction

This system sends 2-minute packets of heart rate data to a Google Cloud Server. This packet is collected on the Microsoft Band, sent to an iPhone via Bluetooth which hosts the iOS app for display purposes, and sends it to the server via HTTP. The machine learning code hosted there contains a LogitBoost classifier with Decision Trees, SVMs, Naive Bayes, and Stochastic Gradient Descent weak learners. The classifier trains on the existing, cloud hosted dataset and returns the stress classification. The goal of this project is to provide a system to collect heart rates and classify stress into binary classes with comparative or better results than existing systems. The algorithms in this paper provide better results than related work.

June 11, 2015

2 Background Information

2.1 Stress

The motivation for this paper is to provide users with a tool to track their stress [1]. Being able to display level of stress can be a powerful tool for those interested in monitoring their health. For example, someone who is at high risk for cardiac arrest may try to avoid high levels of stress because of the potential cardiac impact. Being able to measure this stress could be the deciding factor between choosing to take the bus over driving to work. An indicator is particularly useful because stress is not always recognizable by the individual. Lowering levels of stress can have benefits for business, communications, and creativity.

- *L. Plewa is with the Department of Computer Science, Cal Poly San Luis Obispo, San Luis Obispo, CA, 93405.*
  E-mail: lplewa@calpoly.edu

Manuscript received June 11, 2015.

2.2 Heart Rate Variability

This project uses the heart rate sensor of a wearable device to derive heart rate variability features [1]. The statistical analysis of heart rate is called heart rate variability (HRV) analysis, where the heart rate is determined to be the time between the peaks of the R-wave in the electrical signal from the heart: electrocardiogram (ECG). The differences in these beats can be used to derive time domain and frequency features. Through these features, the status of the autonomic nervous system can be derived.

2.3 Machine Learning

The techniques in this paper are all supervised learning techniques. The data is all labelled either by hand (in the case of the Microsoft Band physical dataset). This provides labelled classes for the algorithms. Classification is a machine learning problem where a data sample can belong to one of two or more classes. While classification algorithms can deal with more than two classes, this paper focuses on a binary classification problem where the two possible classes are either stress or non-stress samples. Machine learning classifiers use the feature vectors derived from the data samples to learn how these features help identify what class future data samples belong to. The classifiers in this paper are the most popular in terms of stress classification and are compared against the proposed LogitBoost classifier.

2.4 Support Vector Machines

A Support Vector Machine (SVM) is a machine learning algorithm for classification proposed by Cortes and Vapnik in 1995 [2]. The main idea of SVMs is to learn a nonlinear function by a combination of linear mappings in high dimensional feature space. The number of features to the dimensionality of the input space. The desired function would be able to map all training data within a certain margin of error. Support vectors are especially good in high dimensionality spaces because SVMs do not depend upon the dimensionality of the input space. Fig. 1 shows the 2d space case.

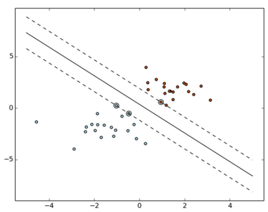

Fig. 1. The SVM line and margin in 2d space.

2.5 Naive Bayes

This classifier uses the theorem to derive the probability that a given feature vector is associated with a specific class. The algorithm naively assumes that there is an independence between every pair of features. This assumption creates a weakness in the algorithm, because there will almost never been an indepdendence between every pair of features given enough features.

$$P(\text{Class} \mid \text{Features}) = \frac{P(\text{Class}) * P(\text{Features} \mid \text{Class})}{P(\text{Features})}$$

Fig. 2. The Bayes Theorem

2.6 Boosting

This paper announces the use of LogitBoost as a classifier for stress using HRV features. LogitBoost is of the Boosting family of classifiers [8]. The main idea of boosting is to take an ensemble of weak learners and weight them based on their ability to train on a dataset. LogitBoost takes this one step further and includes a loss function while training these weak learners. The sum of these weak learners and their weights creates a strong learner.

$$f = \sum_t \alpha_t h_t$$

Fig. 3. The Generic Boosting Algorithm. The derived hypothesis is the sum of all weak learner hypotheses times their respective weights.

$$\sum_i \log\left(1 + e^{-y_i f(x_i)}\right)$$

Fig. 4. The LogitBoosting Algorithm loss function.

3 Related Work

There are four systems comparable to this project. Each one incorporates GSR, HRV, or ECG in order to determine some persons state (stress, emotions, or activity) through machine learning.

The oldest system, from 2010, uses ECG, GSR, and accelerometer data to derive three activities: sitting, standing, and walking [10]. The system utilized 60 second samples as input into a WEKA-based classifier. The classifier used J48 Decision Trees, Bayes Net, and SVM algorithms to achieve 92.4% accuracy for classes on one individual and 80.9% accuracy across individuals.

Villarejo et. al. use GSR exclusively to achieve 76.56% in determining user state [11]. These states are staying relaxed, performing mathematical operations, breathing deeply, and reading as fast as possible. Their data is separated into ten second samples as input for a WEKA-based classifier using BayesNet, J48, and SMO classifiers.

Sharma et. al. use HRV, GSR, breathing rate, blood volume pulse, brain waves, temperature, and muscle tension to determine stress [9]. They use a Naive Bayes classifier with 5 minute samples and emotional states of anxiety, joyful, calm, and illness as classes. The authors do not disclose their accuracy with the classifier, but are able to determine value mappings for GSR, BVP, and temperature to emotion.

Liu et. al. use ECG and GSR data to predict stress [3]. In this case, the authors determine stress to be a binary value (stressed or not stressed). They use a linear SVM with an F1 score of 0.85.

4 System Specifications

Fig. 5 demonstrates the proposed and implemented system. The individual components are a Microsoft Band, an iPhone, and a Django server hosting the classifiers and labelled dataset.

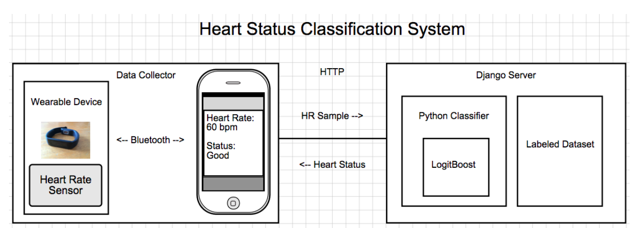

Fig. 5. The Proposed System

4.1 Microsoft Band

Fig. 6 shows the Microsoft Band in use by this system. The metal piece hosts a green laser sensor that uses light to collect the heart rates. This data is sent via Bluetooth to a smart phone, using the SDK provided by Microsoft.

Fig. 6. The Microsoft Band

4.2 iOS Application

Fig. 7 illustrates the iOS application. The text box shows the instantaneous beats as well as the skin temperature. Two minute samples are stored on the application before being sent to the server via HTTP PUT request. The implementation is made available via GitHub [7].

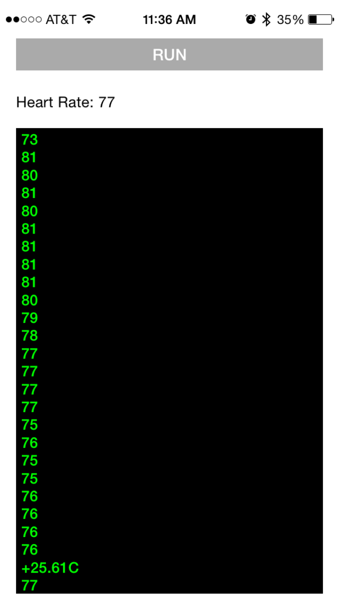

Fig. 7. The iOS Application

4.3 Django Server

The server is on a Google Cloud Storage system, seen in Fig. 8 and made publicly available at https://console.developers.google.com/storage/browser/heart_rates/. Each sample is stored in the bucket on the server, and can be classified via the associated Python code. The implementation is made available via GitHub [5].

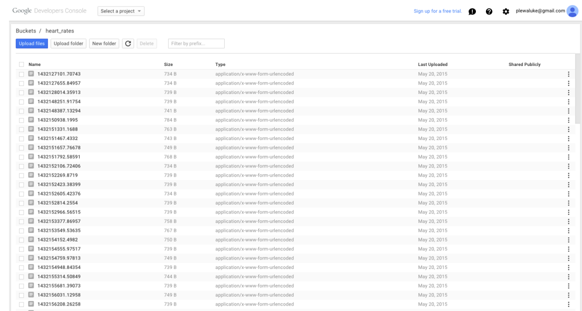

Fig. 8. The Google Cloud Storage Interface.

5 Prospective Study

The study in this paper uses five product evaluators. The evaluators are outlined in Fig. 9. For data collection purposes, the product evaluators are told to breathe and move normally. The recording is a two minute sample that is started and ending during the activity, and not before or after. Each sample is from a normal workday hours (9 am to 5 pm) and not during any exhausting timeslot (after or before waking hours). Each activity is sampled with at least a two minute resting period between each activity so that there is no bleeding between activities. This bleeding may have an effect on the heart rates and is avoided. This connection of the strap to the evaluators wrist is checked before and after each sample to make sure that the connectivity of the sensor to the wrist is strong.

Subject ID	Gender	Age	Height	Weight	Weekly Exercise Amounts
0	Male	23	5 Feet 8 Inches	150 lbs.	1 times per week
1	Male	23	5 Feet 8 Inches	160 lbs.	6 times per week
2	Male	23	6 Feet 2 Inches	200 lbs.	6 times per week
3	Male	21	6 Feet 0 Inches	180 lbs.	4 times per week
4	Male	21	6 Feet 2 Inches	220 lbs.	2 times per week

Fig. 9. The Product Evaluators.

6 Experimental Setup

The classifier parameters are as follows. The LogitBoost classifier uses five of each weak learner for a heterogeneous setup: SVM, Naive Bayes, Stochastic Gradient Descent, and Decision Trees. The weak learners and comparative classifiers in this chapter use the default settings from the Scikit-Learn Python library [4] unless stated otherwise. The SVM classifier uses a linear kernel with the C parameter set to 0.1 and gamma set to 0.0 (defaulting to 1 / number of features). The SGD has an alpha parameter of 0.0001, no penalty parameter set, and a hinge loss function. The Naive Bayes implementation has an alpha parameter of 1.0 with uniform prior probabilities. The Decision Tree implementation has no max features set, no max depth set, and the minimum number of samples for an internal node split set to two. The implementations are available via GitHub [6], [5].

The features for this experiment are outlined in Fig. 10. The activities performed by the product evaluators is seen in Fig. 11. These activities are divided into stressful or non-stressful categories. Fig. 12 shows the parallel coordinates graph of these derived features for each sample.

Number	Features
1	Mean HR
2	Minimum Instantaneous HR
3	Maximum Instantaneous HR
4	Standard Deviation of the HRs
5	Standard Deviation of the Beats per Minute
6	Median of the HRs
7	Root Mean Square of the Standard Deviation of the HRs
8	The Number of Outliers (greater than 50 ms difference)

Fig. 10. The Derived HRV Features.

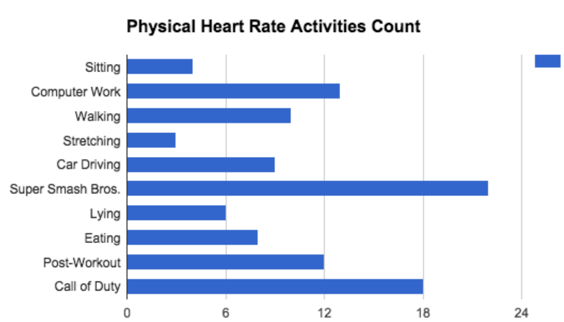

Fig. 11. The Dataset Activities.

7 EVALUATION

The classifiers in this project are evaluated in terms of accuracy, precision, recall, and f-score. The positive class for these evaluations is the stress class. The LogitBoost classifier that this paper proposes is compared against the most popular classifiers and best performers of related works: SVM and Naive Bayes. Overall, the LogitBoost approach using heterogeneous weak learners seems to be the better implementation except for in terms of precision, where Naive Bayes is slightly ahead.

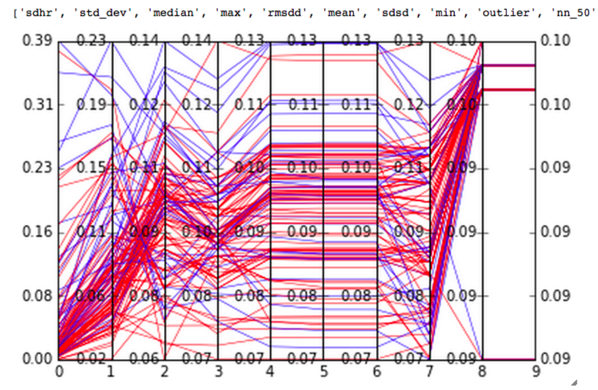

Fig. 12. The Parallel Coordinates of Features. Stressful samples are in red and non-stressful are in blue.

7.1 Accuracy

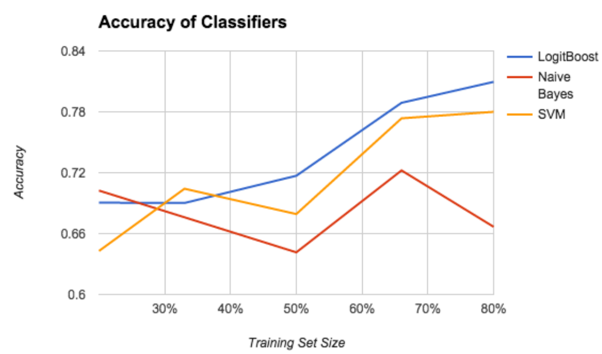

Fig. 13. The accuracy of each classifier across different training-testing dataset splits.

Fig. 13 shows the accuracies of each classifier across different training-testing dataset splits. The LogitBoost algorithm is a more accurate classifier with a peak 80.95% accuracy compared to the other learners.

7.2 Precision

Fig. 14 shows the precision scores of each classifier across different training-testing dataset splits. The Naive Bayes algorithm is a more precise classifier with a peak 0.8117 score compared to the other learners.

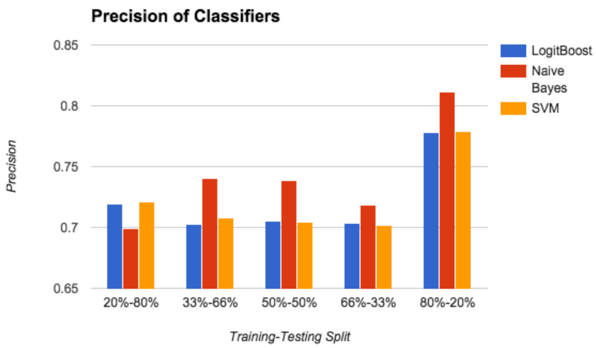

Fig. 14. The precision of each classifier across different training-testing dataset splits.

7.3 Recall

Fig. 15 shows the recall scores of each classifier across different training-testing dataset splits. The LogitBoost algorithm is a better recall classifier with a peak 0.9857 score compared to the other learners.

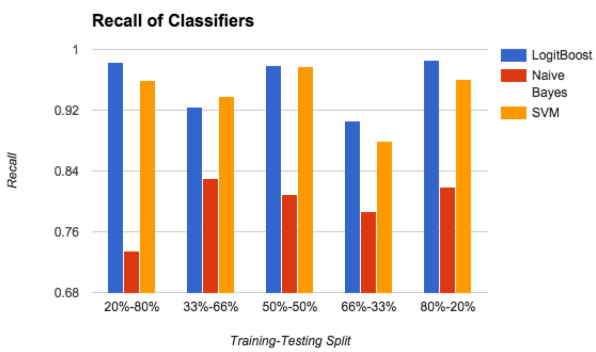

Fig. 15. The recall of each classifier across different training-testing dataset splits.

7.4 F-score

Fig. 16 shows the F-scores of each classifier across different training-testing dataset splits. The LogitBoost algorithm is a better F-score classifier with a peak 0.8888 score compared to the other learners.

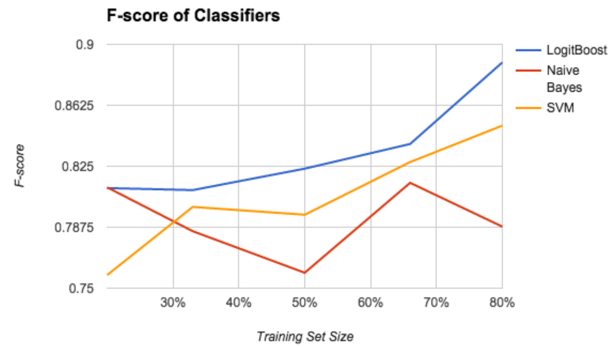

Fig. 16. The F-score of each classifier across different training-testing dataset splits.

8 Discussion

This paper introduces an Internet of Things system for collecting heart rates as well as providing an indicator of stress. In terms of accuracy, recall, and F-score, the LogitBoost classifier outperforms the classifiers of existing systems on this dataset.

The difficulties of this paper include the availability of labelled activity data for stress. In terms of implementation, the connectivity of all the system components is also a challenge. The system is modular, but the connections between each component are difficult in terms of time to implement.

9 Future Work

A larger dataset may be useful to derive more accurate evaluations of classifiers. This study features only five project evaluators and could incorporate more. Also, a broader amount of activities could be covered. Implementations could be made for multiple smart phones or multiple wearable devices to provide a broader user base.

References

[1] Ahuja N. et. al. GSR and HRV: Its Application in Clinical Diagnosis. Indian Institute of Technology. 2003. http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=1212802

[2] C. Cortes and V. Vapnik. Support-vector networks. Machine learning, 20(3):273297, 1995.

[3] Liu D. and Ulrich M. Listen to Your Heart: Stress Prediction Using Consumer Heart Rate Sensors. Stanford. 2014. http://cs229.stanford.edu/proj2013/LiuUlrich-ListenToYourHeart-StressPredictionUsingConsumerHeartRateSenso pdf

[4] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:28252830, 2011.

[5] Plewa, L. *Stress Endpoint*. Github repository. 2015. https://github.com/luke-plewa/stress_endpoint

[6] Plewa, L. *Zagreus LogitBoost HRV Classifier*. Github repository. 2015. https://github.com/luke-plewa/zagreus

[7] Plewa, L. *HRCollector iOS Application*. Github repository. 2015. https://github.com/luke-plewa/stress_endpoint

[8] R. Schapire and Y. Freund. Boosting: Foundations and algorithms. 2014.

[9] Sharma T. and Kapoor B. Data Analysis By Using Machine Learning Algorithm on Controller for Estimating Emotions. Chitkara University. 2014. http://airccse.org/journal/ijcsa/papers/4614ijcsa02.pdf

[10] Sun F. et. al. Activity-aware Mental Stress Detection Using Physiological Sensors. CMU. 2010. http://repository.cmu.edu/cgi/viewcontent.cgi?article=1011&context=silicon_valley

[11] Villarejo M. et. al. A Stress Sensor Based on Galvanic Skin Response (GSR) Controlled by ZigBee. 2012. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3386730/

Luke Plewa Luke Plewa is a student at Cal Poly pursuing a joint BSMS degree in Computer Science. His other works include a cardiac arrest prediction algorithm through HRV and HVAC energy consumption prediction through weather data.