Context-Aware Stress Detection in the AWARE Framework*

Marija Trajanoska
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
1000 Skopje, Macedonia
marijatrajanoska@gmail.com

Marko Katrašnik
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
marko.katrasnik@gmail.com

Junoš Lukan
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
junos.lukan@ijs.si

Martin Gjoreski
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
martin.gjoreski@ijs.si

Hristijan Gjoreski
Faculty of Electrical Engineering and Information Technologies
Ss. Cyril and Methodius University
1000 Skopje, Macedonia
hristijang@feit.ukim.edu.mk

Mitja Luštrek
Jožef Stefan Institute
Jamova cesta 39
1000 Ljubljana, Slovenia
mitja.lustrek@ijs.si

ABSTRACT

Physiological signals are good predictors of stress, which can be thought of as part of a user's context. In this work, an option to combine the user's stress level with other contextual factors is presented. This is done in the form of two AWARE plugins – Android applications that can be incorporated into a smartphone monitoring setup. In the first part, the stress detection method is described, which consists of a lab stress detector, an activity classifier, and a context-aware stress model. In the second part, two plugins are described. One streams the data from the Empatica E4 wristband and the other one uses this physiological data to predict stress. Finally, some possibilities to improve this work are presented.

Keywords

AWARE, plugin, stress detection, Empatica E4, physiology

1. INTRODUCTION

Mental stress is most often researched because of its negative health consequences when it is chronic. The ability to detect stress from physiological signals collected with a wearable device is thus valuable for research in situations when stress occurs, as well as to trigger stress-relief interventions. In addition, stress affects one's short-term psychological state and behaviour, which makes it a part of the user's context as understood in ambient intelligence. Detecting stress is therefore also valuable for adapting intelligent services to the user (e.g., a mobile application may postpone non-essential notifications when the user is stressed). In this paper we present a stress-detection plugin (and its prerequisite – the plugin for the Empatica sensing wristband) for the AWARE framework [3]. This makes stress detection easily accessible to researchers and other interested parties.

AWARE is an Android framework, used to capture the phone sensors' data to infer context. Its modular nature enables it to be extended by plugins. There are already several plugins available such as a Google activity recognition plugin, which captures the users' mode of transportation, and a Fitbit plugin, which enables collecting data such as heart rate and sleep duration from a Fitbit device.

In this work, a state-of-the-art stress detection method is implemented as an AWARE plugin. To make this possible, the method was adapted to real-time operation, being previously only used offline. The plugin classifies the user's physiological data as representing a stressful or a non-stressful condition, after receiving the data from the Empatica wristband via another plugin. Both plugins are planned to be released publicly, so that other researchers will be presented with a ready-made solution for the first time.

In Section 2, we first present the stress-detection method and what data is needed for it. In Section 3 our implementation in the AWARE framework is presented: a plugin for the stress detection model itself (Section 3.2) and a plugin for data collection (Section 3.1). Finally, some possible improvements are outlined in Section 4.

2. CONTEXT-AWARE STRESS DETECTION METHOD

The stress-detection AWARE plugin is based on a real-life stress detection method as described by Gjoreski et al. [7], and the more general context-based reasoning framework introduced by Gjoreski et al. [5]. It consists of three separate machine learning components: a laboratory stress detector, an activity recognition classifier, and a context-based (real-life) stress detector. Each is presented in its own subsection.

2.1 Lab Stress Detection

The lab stress-detection model was trained using data obtained in a laboratory experimental setup during a standardized stress-inducing experiment [7]. The main stressor in this experiment was solving a mental arithmetic task under time and evaluation pressure. The laboratory data was then labelled taking into account both the difficulty of an

*The work presented in this paper was carried out as a part of a project funded by the Slovenian Research Agency (ARRS, project ref. N2-0081).

equation-solving session (easy, medium or hard) and short STAI-Y anxiety questionnaires [8] filled out by the participants. According to this information, the data was classified into three degrees of stress: no stress, low and high. Additionally, baseline no-stress data was recorded on a separate day when the participants were relaxed.

For the creation of the laboratory stress-detection classifier, the machine-learning pipeline involves segmentation, signal filtering, feature extraction, model learning, and evaluation of the models.

Segmentation refers to the partitioning of the data into windows for the purposes of feature extraction. According to the windowing experiments, which provide a performance comparison between models for varying data window sizes, the optimal data window size was 5 min with 2.5 min overlap.

The signals obtained from the Empatica sensors are: blood volume pulse (BVP), interbeat intervals (IBI), heart rate (HR), electrodermal activity (EDA) and skin temperature (TEMP). After filtering the signals to reduce noise, numerical features are extracted from each data window using statistical functions, regression analysis, and frequency and time analysis, depending on the type of signal. A total of 70 features are extracted from these signals.

The best performing classifier on this dataset proved to be the WEKA implementation of the support vector machine algorithm. Its final output is a stress level prediction of 0 (no stress), 1 (low stress) or 2 (high stress), which is then used as input to the context-based stress detector.

2.2 Activity Recognition

It is important for a stress-detection system to be aware of the user's physical activity, since physical activity elicits physiological arousal similar to psychological stress. For this reason, we used the 3-axis accelerometer provided by the Empatica wristband, which has proven to be successful in recognising activities, according to Gjoreski et al. [6]. The activity recognition model was trained on 60 minutes of real-life Empatica data from one person, with nearly 10 minutes of labelled data per class.

The machine-learning pipeline for the acceleration data is similar to the one used in the lab stress detector. Here, data segmentation involves an overlapping sliding-window technique, which divides the continuous stream into 4 s windows with a 2 s overlap.

Feature extraction produces 52 features: seven represent body posture, while the remaining represent body motion. The extracted feature vectors are fed into a machine-learning algorithm to build an activity-recognition classifier.

The best performing algorithm on multiple acceleration datasets was Random Forest [6], so this is the final algorithm used to build the activity-recognition model. The final output of the activity recognition model is a numeric activity level on a scale from 1 to 5, where each number corresponds to an everyday activity as follows: 1 = lying, 2 = sitting, 4 = walking or standing, 5 = cycling or running. Finally, the activity recognition classifier's output is input into the context-based stress detection model.

2.3 Context-Based Stress Detection

The context-based stress detection classifier was trained using the data obtained as part of the real-life experimental setup described in Gjoreski at al. [7] The data duration totalled to 1327 h and involved 5 participants who wore Empatica E4. The labelling process involved a combination of a stress log and Ecological Momentary Assessment (EMA) prompts implemented on a smartphone. For the stress log, the participants logged the start, duration, and intensity (on a scale from 1 to 5) of everyday stressful situations. The EMA prompts were additionally displayed randomly 4 to 6 times throughout the day, with at least 2 hours between consecutive prompts.

The labelled data was then windowed using non-overlapping windows lasting 10 min, since aggregation experiments showed that most algorithms perform better for smaller aggregation windows (10 min to 17.5 min) as compared to larger ones.

The context-based stress detector's input is four-fold:

- context features,
- features extracted from the output of the activity-recognition model,
- features extracted from the output of the lab stress-detection model, and
- a subset of the lab stress detector features.

The whole stress-detection method, including lab stress detection and activity recognition, is illustrated in Figure 1.

The context features refer to the hour of the day (1 to 24) and the type of day (a weekday or weekend).

The output from the activity recognition model gives an estimate of the reliability of the lab model's prediction. Aside from features extracted from the activity level predictions themselves, the activity level is also taken into account as a modifier to the lab stress predictions prior to performing feature extraction on them. The lab model's prediction is discarded if it is made in an unsteady environment, which is defined as the occurrence of an average activity level above 4 (high) in one of the (5-minute long) instances within the last 30 min. Additionally, the stress prediction is decreased (its class is changed to a lower one, or left unmodified if its already zero) if the subject exhibited an average activity level between 2 and 4 (moderate) within the last 20 min.

There are a total of 15 features extracted from the modified lab stress prediction. A subset of lab stress detector features is also used as input to the context-based stress detector.

The best performing algorithm for making a binary ("stress" or "no stress") prediction using the outlined windowing parameters on the labelled real-life data was a Decision Tree [7], so this algorithm was used to build the final context-based real-life model. Using event-based windowing, this algorithm achieved an F-score of 0.9 using leave-one-subject-out cross-validation.

3. AWARE IMPLEMENTATION

AWARE is a mobile instrumentation toolkit which had the initial purpose of inferring users' context [4]. Extensibility, however, was a primary requirement when developing the framework. Specifically, extending the context by using external sensors was explicitly envisioned, as was using the gathered data and machine-learning techniques for "creat[ing] new higher-level context"[4, p. 4].

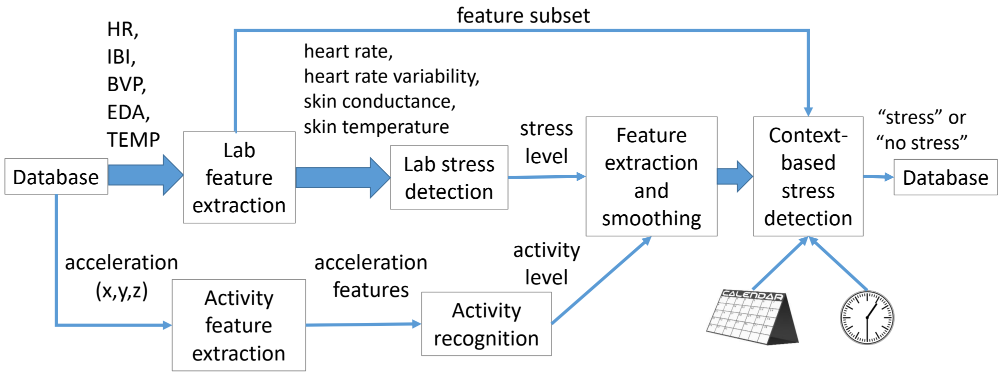

Figure 1: An overview of the context-based stress detection model. Features are extracted from physiological signals from Empatica and input to a lab stress detector. Its predictions are used in the context-based stress detector, which in addition takes activity features from acceleration data into account and considers the time and the type of day.

In the next two sections, two such extensions are presented. The first one gathers physiological data from Empatica, saving it in the standard AWARE format. The second implements the stress detection method presented in Section 2 as an AWARE plugin.

3.1 Empatica Data Streaming Plugin

The AWARE framework already offers plugins for acquiring data from Fitbit and Android Wear wristbands, but it does not have one for the more research-oriented Empatica E4. Our goal was to create an AWARE plugin that enables users to easily connect the Empatica E4 wristband to an Android smartphone.

Figure 2 shows the overview of the processes implemented in this plugin. The physiological data is first transmitted over a Bluetooth connection. It is then available to other plugins via broadcasts and written to a database for later use.

The data from Empatica is transmitted over a low-energy Bluetooth connection, so that the impact on the smartphone's battery is minimal. This enables Empatica E4 to stream up to 24 hours on a single charge [1]. When a sensor reading is successfully transmitted from the wristband to the smartphone, specific functions (`didReceiveAcceleration`, `didReceiveBVP` etc.) are called automatically.

In our case, these functions were expanded to include code to send broadcasts and to save the reading to the database. In this we follow the logic of other AWARE sensors or plugins, in which received data is accessed via broadcasts to display and handle in real time, via content providers (through database) for more complex analysis (the middle part of Figure 2).

Broadcasts are inter-app messages that are sent when a specific event happens, in our case triggered by the data transmission. These messages can be read with observers (broadcast receivers) from any app or plugin that is installed on the phone. Since broadcasting happens in real time, other plugins can use the data from Empatica without including code for communication with the wristband (an example of such a plugin is described in the following section). The data can also be displayed in real time in the native AWARE application.

The collected physiological data is also written to a database (the lower part of Figure 2) by using content providers. Because of different sampling rates, each Empatica sensor has its own content provider. A separate SQL table for each sensor contains columns for `id`, `timestamp`, `device id` and a `value` from the sensor. Columns are defined in this way, so that the database tables maintain the standard AWARE format. Again, each content provider has its own "content URI", a unique address used by other plugins to identify providers for specific sensors.

When broadcasts are sent and the data is written to a database, it is up to other plugins to use it. Even though the main purpose of this plugin is to provide the data that will be handled by other plugins, it also offers a basic user interface. It has its own activity (a user interface that most Android apps have) from which users can export and clear the data in the database.

Contrary to most other Empatica data acquisition applications, our plugin is meant to run in the background. Therefore, it was anticipated that the connection is lost without users noticing. In an effort to solve this problem, our plugin uses a notification to inform the user about the current state of connection.

3.2 Stress Detection Plugin

The goal of the stress detection plugin is to provide real-time stress predictions based on the context-based stress detection method described in Section 2.

In the original stress-detection study [7], the sensor data was recorded in the Empatica E4 wristband's internal memory and later transferred to a computer for further processing. The novelty in our AWARE plugin is that the data is streamed to an Android device via Bluetooth in real time and stored in the phone's database. This allows for real-time processing and classification.

The machine-learning pipeline for the stress detection plugin mirrors the original pipeline used to train and test the context-based stress detector. The three models (lab stress

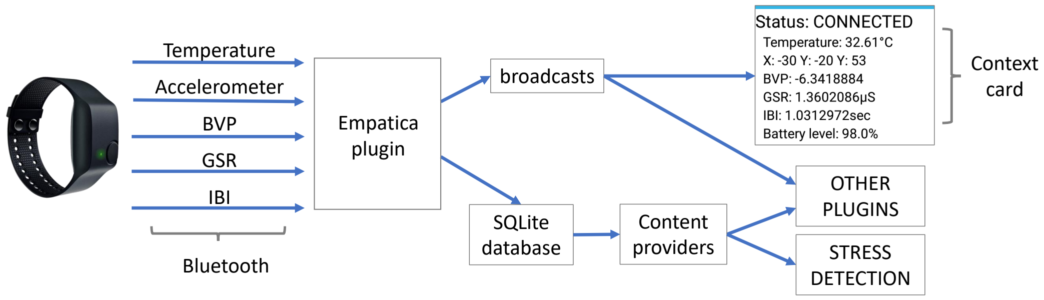

Figure 2: An overview of the Empatica data streaming plugin. The data is first received via a Bluetooth connection and can then be broadcast to other plugins or written to a database.

detector, activity recognition classifier and context-based stress detector) are independent and saved locally in the plugin's assets. The models are triggered periodically using the optimal time intervals discussed in Section 2.

As discussed in Section 3.1, the Empatica data streaming plugin writes the raw data from the E4 in SQL tables in real time. The stress detection plugin then has access to this data through the former's content providers. The plugin reads the last 5 min of raw Empatica data every 2.5 min and provides this data to the models for processing. The context-based model gets its context features using the phone's current date and time.

The features from the lab stress detector, the activity recognition classifier and the context-based stress detector are, both broadcasted and saved in the phone's database. The same is true for each lab stress prediction, activity level prediction and context-based prediction. The stored data is further accessible through content providers for any other application to use, as is the case with other AWARE plugins. In this way, both the Empatica data and the stress prediction method are easily available for other researchers to use.

4. FURTHER WORK AND CONCLUSIONS

The plugins described in the previous section offer a ready-made solution which researchers could use to add a stress level to the user's context. There are some limitations in their current implementation which we aim to amend.

Currently, the standardization (normalization) of some of the features is arbitrary. It is done by subtracting a "typical" value of a given signal and divided by a "typical" standard deviation. To account for inter-individual physiological differences, means and variability could be calculated on a person-specific basis. This would, at the very least, require keeping track of a user ID and then calculating signal mean and variability over a longer time-period when a new user would start using the application. If baseline values would be needed, this would also require the user to indicate when they are not under stress and calculate their specific physiological values in that time-window. This type of user interaction has not been accounted for in the current implementation.

An evaluation of this method is also planned. The models described in Section 2 have been evaluated as outlined in related work, but they have been used in different experimental scenarios. Online real-life use of the method would merit its own evaluation.

To understand causes of stress and the situation where a physiological stress response arises, it is helpful to know as much as possible about a user's context. The plugins described in this work will simplify combining physiological data with other contextual data the AWARE framework already provides. Additionally, stress predictions can be used as context themselves and inform other interactions with users, such as offering them prompts at certain stress levels by using the AWARE Scheduler [2] and using stress predictions as broadcast triggers.

5. REFERENCES

[1] E4 wristband for developers. `http://developer.empatica.com/`, 2018. Accessed: 2018-08-29.

[2] D. Ferreira. AWARE: Scheduler. `http://www.awareframework.com/scheduler/`, 2015. Accessed: 2018-08-29.

[3] D. Ferreira and V. Kostakos. AWARE: Open-source context instrumentation framework for everyone. `http://www.awareframework.com/what-is-aware/`, 2018. Accessed: 2018-08-29.

[4] D. Ferreira, V. Kostakos, and A. K. Dey. AWARE: Mobile context instrumentation framework. *Frontiers in ICT*, 2(6):1–9, 2015.

[5] H. Gjoreski, B. Kaluža, M. Gams, R. Mili, and M. Luštrek. Context-based ensemble method for human energy expenditure estimation. *Applied Soft Computing*, 37:960970, 2015.

[6] M. Gjoreski, H. Gjoreski, M. Luštrek, and M. Gams. How accurately can your wrist device recognize daily activities and detect falls? *Sensors*, 16(6), 2016.

[7] M. Gjoreski, M. Luštrek, M. Gams, and H. Gjoreski. Monitoring stress with a wrist device using context. *Journal of Biomedical Informatics*, 73:159–170, 2017.

[8] C. D. Spielberger and S. J. Sydeman. *State-Trait Anxiety Inventory and State-Trait Anger Expression Inventory*. Lawrence Erlbaum Associates, Inc, 1994.