xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.1

Sensor data analysis for situation awareness V1

Dissemination level:	Public
Contractual date of delivery:	Month 12, 31 October 2021
Actual date of delivery:	Month 13, 1 November 2021
Work package:	WP3 Analysis and fusion of multi-modal data
Task:	T3.1 Sensor data analysis
Type:	Demonstrator
Approval Status:	Final Version
Version:	0.7
Number of pages:	34
Filename:	D3.1_xR4Drama_SensorDataAnalysisFor SituationAwarenessV1_20211101_v0.7.pdf

Abstract

The current deliverable describes the relevant literature and the work conducted so far on the analysis of physiological signals, extracted from sensors of a smart vest, with the primary goal to detect stress and secondarily to recognize activities.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	06/09/2021	Initial version with Table of Contents	Stamatis Samaras
0.2	06/10/2021	State of the art, Data Integration, Methods, Deep Learning experiments	Stamatis Samaras
0.3	10/10/2021	Added content in Section 4	Stamatis Samaras
0.5	24/10/2021	Circulation for internal review by Rita Paradiso (Smartex)	Stamatis Samaras
0.6	27/10/2021	Addressing review's comments	Stamatis Samaras
0.7	01/11/2021	Final formatting	Stamatis Samaras

Author list

Organization	Name	Contact Information
CERTH	Stamatis Samaras	sstamatis@iti.gr
CERTH	Sotiris Diplaris	diplaris@iti.gr
CERTH	Spyridon Symeonidis	spyridons@iti.gr
CERTH	Haralampos Papadopoulos	chapapadopoulos@iti.gr
CERTH	Elissavet Batziou	batziou.el@iti.gr
CERTH	Paraskevi Kritopoulou	pakrito@iti.gr

Executive Summary

This deliverable explains the analysis of physiological sensors employed in XR4Drama so far. These sensors are embedded on a smart vest and are used in the disaster management scenario in order to predict the stress levels of first responders. The physiological sensors are suitable for stress detection and activity recognition. The purpose of the deliverable is to present the related work in the field of analysis of physiological sensors, especially focused on stress detection. Furthermore, this deliverable describes the work done on sensor data analysis during the first year of the project. The experiments and relative methodologies included, are focused on the first pilot use case, namely a disaster management scenario, where the analysis of physiological sensors will 'track' the emotional and physical state of the first responders.

The technical part of the sensor data analysis can also be found in the current deliverable. This includes the sensors data integration, the creation of databases to store results and some details about the communication between components. Experimental results from deep learning and machine learning applications reveal the preliminary work on training models that will later be deployed in the first prototype. The experimental results included here, are a part of the analysis conducted up to the time of the deliverable's submission.

Abbreviations and Acronyms

ACC X-Y-Z	Accelerometer measurements on X-Y-Z axis
API	Application Programming Interface
CNN	Convolutional Neural Network
DB	Database
DL	Deep Learning
DNN	Deep Neural Network
ECG	Electrocardiograph
EMG	Electromyogram
GSR	Galvanic Skin Response
GYRO X-Y-Z	Gyroscope measurements on X-Y-Z axis
HR	Heart Rate
HRV	Heart Rate Variability
IMU	Inertial Measurement Unit
IOT	Internet of Things
JSON	JavaScript Object Notation
kNN	k-Nearest Neighbors
MAG X-Y-Z	Magnetometer measurements on X-Y-Z axis
ML	Machine Learning
MLP	Multilayer Perceptron
PUC	Pilot use case
RF	Random Forest
SC	Skin Conductance
SOTA	State-Of-The-Art
SVM	Support Vector Machine
SVM	Support Vector Machines
TSST	Trier Social Stress Test
VR	Virtual Reality
XGB	eXtreme Gradient Boosting trees

Table of Contents

1 INTRODUCTION

The emergence of technology and the availability of a wide variety of wearable sensors allow the monitoring of potentially harmful or dangerous situations, offering the opportunity to reduce the risk that might occur and increase safety. Distant monitoring through Internet-of-Things (IoT) systems applies to different domains, like the health domain, where the health or mobility of patients is monitored and alerts can inform medical personnel for changes in the physical condition of patients or dangerous events like falls. In construction sites, wearable sensors are used to monitor the movement of workers and are able to predict and prevent unsafe occurrences. Another application of monitoring with wearable sensors is the recognition of the emotional state of a subject. This may apply in psychological tests or in situations that cause increased stress that needs to be observed. In this deliverable we describe the relevant application of xR4DRAMA, where physiological sensors embedded in a smart vest, will be used to monitor the stress of first responders in a disaster management scenario and recognize their activities.

The disaster management scenario (PUC1) takes place in Vicenza, Italy and is organized by AAWA. PUC1 intends to monitor a flood scenario, where first responders are usually not adequately trained to face such real life scenarios, thus they undergo under a lot of stress. Wearable smart vests are employed in this pilot to receive the physiological signals of the first responders and try to detect their stress levels and the activities they perform. The stress of the first responders will be detected both through the physiological sensors and audio signals. Here we describe the work conducted on sensor signals' analysis. In a nutshell, training samples of physiological signals were collected by different subjects acting as first responders, while performing controlled physical and mental activities indoors. The best performing models will be deployed in the first prototype where the first version of the xR4DRAMA system will be tested.

2 RELATED WORK

This section reports the current SOTA on stress detection methods based on physiological sensors and provides an analysis on the most commonly employed experimental protocols for the dataset creation. The first sub-section focuses on the experimental protocol that is employed to create stressor tests in order to produce the relevant data for the stress detection task. The second sub-section focuses on the best performing methods to tackle the task at hand.

2.1 Experimental protocol on dataset creation

The affective computing community lacks commonly used standard datasets for wearable stress detection. As a result, recent works focus on contributing publicly available dataset for stress detection or emotion recognition. These datasets often follow a specific experimental protocol which allows for different stress level states to be monitored and captured during the data recording.

Schimdt et al. [1] produced a multimodal, publicly available dataset for stress detection. The data has been recorded using two different devices (one chest-based and one wrist-based), each including high resolution physiological and motion modalities. This dataset contains three different affective states (neutral, stress, amusement). In addition, the dataset features self-reported values on the perceived affective state of the subjects, which were obtained using several established questionnaires. The experimental protocol behind the dataset creation was conducted using two different versions including variations of the following stressor states: a baseline condition (neutral affective state), amusement affective state, stress inducing state via the well-studied Trier Social Stress Test (TSST) [2] (public speaking and mental arithmetic tasks), meditation to 'de-excite' and finally a recovery state where sensors were removed.

Moreover, Healey et al. [3] published a dataset on driver stress. This dataset features electrocardiogram (ECG), electromyogram (EMG), skin conductance (SC) and respiration data, recorded continuously while drivers followed a set route through open roads in the greater Boston area. Although stressful events could not be specifically controlled on the open road, the route was planned to take the driver through situations where different levels of stress were likely to occur, specifically, the drive included periods of rest, highway and city driving that were assumed to produce low, medium and high levels of stress. The stress evaluation was verified by a driver questionnaire and a two-score derived from observable events and actions coded from video tape taken during the drives.

More recently, Koestra et al. [4] published DEAP, a database for emotion analysis using physiological signals. The dataset contains electroencephalogram, facial videos and peripheral physiological signals. The data was recorded while the subjects watched 40 one-minute excerpts from music videos. The final 40 clips were chosen from a larger pool of videos, by asking volunteers to rate the clips in terms of their valence and arousal value and then choosing the ones that had the strongest rating with the smallest variance.

Sierra et al. [5] created a dataset for stress detection based on common physiological signals, namely EMG, ECG, respiration rate, heart rate (HR) and Galvanic Skin Response (GSR). They recorded data from 80 female students. The employed experimental protocol included variations of two main stress inducing tasks, hyperventilation (HV) and talk preparation (TP), and two calm inducing baseline tasks (one before the stress test and one after).

Furthermore, Pourmohammadi et al. [6] recorded a dataset based on two physiological signals (ECG and EMG) for stress detection from 34 participants. The participants self-reported their stress levels by filling the state-trait anxiety inventory (STAI) [7] questionnaire. The STAI questionnaire is one of the most popular tools for measuring state-trait anxiety. The proposed experimental protocol included two rest states at the start and at end of the acquisition and three mental arithmetic tasks in between that had their difficulty increased as the time was going on.

Yuan Shi et al. [8] created a dataset for stress detection based on ECG, GSR, respiration and skin temperature. They collected data from 22 participants that were exposed to an experimental protocol containing four stressors and six rest periods. The four stressors were: one public speaking stressor, two mental arithmetic stressors, and one cold presser stressor. These stressors represent the social, mental, or physical challenges that might lead to either mental or physical stress.

Saskia et al. [9] published the multimodal SWELL knowledge work (SWELL-KW) dataset for research on stress and user modelling. The dataset includes both physiological signal data (ECG and skin conductance) as well as facial expression through a webcam and Kinect 3D sensor data. The dataset was collected in an experiment, in which 25 people performed typical knowledge work (writing reports, making presentations, reading e-mail, searching for information). The authors manipulated the working conditions with the stressors: email interruptions and time pressure. The experimental protocol included a calm state where the participant could work without interruption and two stress inducing tasks via time pressure and interruptions during work.

Finally, Feng-Tso, et al. [10] presented an activity-aware mental stress detection scheme based on ECG, GSR, and accelerometer data gathered from 20 participants across three activities: sitting, standing, and walking. For each activity, the authors gathered baseline physiological measurements and measurements while users were subjected to mental stressors. The employed stressors were Stroop Color-word interference tests and mental arithmetic problems. The experimental protocol included a 10-minute segment of either baseline or the two stressor tasks for each of the three activities.

2.2 Stress detection based on physiological wearable sensors

The most common stress detection methods based on physiological signals include a feature extraction step derived from statistical knowledge over the properties of the signals that attempt to describe the various affective states. The extracted features are used to train a state-of-the-art machine learning classifier which eventually learns to detect the stress levels of the subjects. A more recent approach attempts to omit the feature extraction step by utilizing a Deep Neural Network (DNN) which can do the representation learning of the different affective states directly from the physiological signals.

Schimdt et al. [1] created a benchmark for their publicly available dataset for stress detection using a large number of well-known features (extracted from physiological and motion signals) and common machine learning methods (Decision Tree (DT), Random Forest

(RF), AdaBoost (AB), Linear Discriminant Analysis (LDA) and k-nearest neighbour (kNN)). The authors validated their methods on a three-class problem (neutral, stress, amusement) achieving 80.34 % accuracy with the AB classifier and on a two-class problem (stress, no stress) achieving 93.12 % accuracy with the LDA classifier.

Rusell Li et al. [11] proposed a novel Deep Learning (DL) based method for stress detection, which was trained and evaluated on the same dataset as [1]. This work attempts to address the limitation of the handcrafted features that traditional machine learning methods rely upon and their potential decrease in accuracy due to misidentification of features. The authors designed a novel 1D Convolutional Neural Network (CNN) and a Multi-Layer Perceptron (MLP) that take as input the raw physiological signals and do not require hand-crafted features but instead extract features from raw data through the layers of the neural networks. The authors validated their classifiers on both the three and two class problems of [1] achieving 97.48 % for the three-class and 99.14 % for the two-class problem.

Sriramprakash et al. [12] proposed a method for detecting stress during working conditions based on feature extraction and machine learning. The authors trained and validated their data on the SWELL-KW dataset [9]. They utilized a set of 17 statistical features derived from ECG and GSR signals and evaluated on which of them are the most dominant in order to increase accuracies. They trained a k-nearest neighbour (kNN) classifier and a Support Vector Machine (SVM) classifier. The SVM classifier trained on the dominant selected features achieved the highest classification accuracy of 92.75 % for the stress vs no-stress classification task. Another work based on feature extraction and SVM was reported by Yuan Shi et al. [8]. The authors proposed a set of 26 handcrafted features derived from ECG, GSR, skin conductance and temperature and respiration. They reported an 80% recall over the binary classification of stress vs no stress problem.

Feng-Tso, et al. [10] extracted statistical features from ECG, GSR and accelerometer and trained a Decision Tree, Bayes Net, and support vector machine (SVM) classifier for stress detection inference combined with physical activities (sitting, standing, and walking). The best classification accuracy (92.4%) was obtained from using the decision tree classifier with the all-feature combination.

Keshan et al. [13] proposed an ECG based feature extraction scheme for driver stress detection. They trained and evaluated their data on [3]. They utilized a set of 14 statistical features derived from ECG signal and found that stress levels can be successfully detected from ECG signals alone; with random tree classifier allowing for identification of the three classes of stress, low, medium and high, with 88.24% accuracy, and Naïve Bayes for two stress levels, low and high, with 100% accuracy.

3 SENSOR DATA ANALYSIS

The sensor data analysis task includes physiological and motion signals as well as processing techniques in time and frequency domain. The RUSA device is a portable device for the acquisition, treatment and transmission of physiological signals measured through a smart sensing vest manufactured by xR4DRAMA's partner Smartex and is employed in xR4DRAMA in order to collect electrocardiograph (ECG) measurements, Inertial Measurement Unit (IMU) measurements and respiration measurements data. By being placed on the chest it is measuring the participant's acceleration, the angular rate, and the magnetic field surrounding the body in X-Y-Z axis using accelerometer, gyroscope and magnetometer, respectively. Thus, the recorded data include both physiological signals and motion analysis signals that could be taken into consideration for the estimation of situation awareness.

The main idea is to analyze the different types of signals by applying signal processing techniques aiming to filter noise and extract informative features. The extracted features can model the relation between physiological parameters and states of stress, having as a goal the estimation of situation awareness and the instantaneous feedback about the subject's activity. As a result, an analysis of the current state-of-the-art (SOTA) on the feature extraction methods on physiological and motion signals for the stress detection task is required.

One important issue for the stress detection task is the creation of a dataset recorded with the RUSA device which will include different affective states so that the stress detection module of xR4DRAMA is trained upon to perform the stress detection. In related work, this dataset creation commonly follows some specific experimental protocols which aim to create different stressor tests for different stress levels to be monitored. An analysis of the most commonly employed experimental protocols for dataset creation is also evaluated.

3.1 Experimental protocol

The experimental protocol used to collect the pre-training and training samples, includes some mental and physical activities that may cause different feelings, from calmness to anxiety, to the subjects. The instructions for these activities are described through a 30 minutes video, which is shown to the subjects and gives them time to perform the tests. The protocol will be explained in more detail in Deliverable 3.4.

3.2 Sensor data integration

3.2.1 Sensor data description

The RUSA data logger device (Figure 1) is a processing unit that is connected to the sensorised garment (smartvest) (Figure 2) which is manufactured by Smartex, an xR4DRAMA partner. The smartvest is equipped with a series of sensors which provide useful measurements for the first responders who are wearing the garment. The sensors are connected to the RUSA data logger which processes the physiological signals and transfers them to another device (e.g. mobile phone) via a Bluetooth connection. The physiological signals can be grouped in the following categories:

1. **ECG measurements**: The main measurement of this category is the value of the ECG signal. Using the ECG signal, the vest software calculates useful physiological measurements such as Heart Rate, Heart Rate Variability, and R-R interval distance in ECG signal.
2. **IMU measurements**: The strap (WWS) is equipped with a "light" IMU measuring only the participant's acceleration in X-Y-Z axis (using an accelerometer), while he/she is wearing the strap. The new vest (WWBS) is equipped with one IMU placed on the chest which is measuring the participant's acceleration, the angular rate, and the magnetic field surrounding the body in X-Y-Z axis using accelerometer, gyroscope and magnetometer, respectively. There are additional two IMUs placed on the arms which just send extracted quaternions. These measurements might not be directly connected with physiological parameters; however, they can be used in order to run Activity Classification algorithms.
3. **Respiration measurements**: The strap is also equipped with a piezo resistive point placed on the thorax, which is used to measure the strain on the thorax caused by the participant's breathing. The strap uses this measurement to calculate the Breathing Rate, and the Breathing Amplitude of the participant.
4. **Activity attributes**: Additionally, some measurements are provided about the activity the participant performs while wearing the strap. There is a simple activity recognition (lying, standing, walking, running). Also, there is a counter measuring the number of steps the participant has done while wearing the strap, and the step period which shows how fast/slow the steps are being done

These measurements are summarized in Table 1.

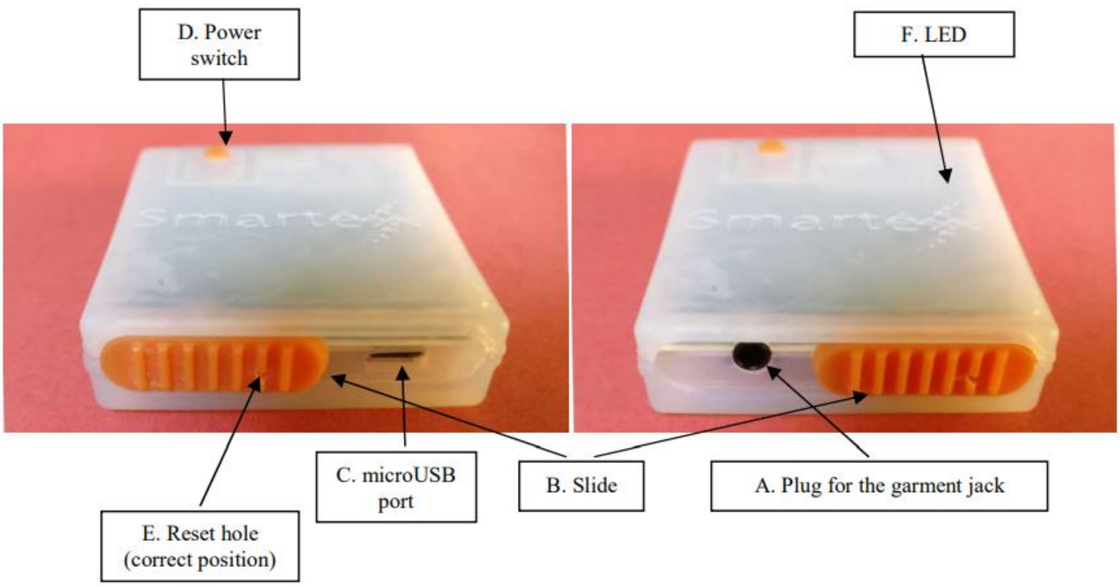


Figure 1. RUSA device.

Figure 2. Smartex sensorised garment (smartvest).

Table 1. RUSA device smartvest recorded parameters.

Recorded parameter	Description	Values (per 1 unit metric)	Sampling rate
ECG Value	Electric signal measuring the ECG	0.8 mV	250Hz
ECG quality Value	ECG signal quality	0-255 (0=poor, 255=excellent)	1/5sec
ECGHR Value	Heart rate	Beats/minute	1/5sec
ECGRR Value	R-R intervals	number of samples between R-R peaks	1/5sec
ECGHRV Value	Heart rate variability	ms	1/60sec
AccX-Y-Z Value	Accelerometer in X-Y-Z axes	0.97 10-3 g	25Hz
GyroX-Y-Z Value	Gyroscope in X-Y-Z axes	0.122 °/s	25Hz
MagX-Y-Z Value	Magnetometer in X-Y-Z axes	0.6 µT	25Hz
RespPiezo Value	Electric signal measuring the chest pressure on the piezoelectric point	0.8 mV	25Hz
BR Value	Breathing rate	Breaths/minute	1/5sec
BA Value	Breathing Amplitude	logic levels	1/15sec
Activity energy Value	estimation of energy activity	is just an estimation (0=no activity, 255=max of activity)	1/5sec
Activity class Value	Activity performed	0=other, 1=lying,	1/5sec

		2=standing/sitting, 3=walking, 4=running	
Activity1Pace Value	Step period	ms	1Hz
Activity Pace Value	Pace	steps/min	1/5sec
Q0-Q1-Q2-Q3 values	Quaternions from main electronic device (Q0, Q1, Q2, Q3 components)	Q14 format	25 Hz

3.2.2 **Integration architecture**

The RUSA device transmits the measurements described in

Table 1 via a Bluetooth connection. The Bluetooth connection takes place with the Mobile App which runs on first responders' mobile devices. As soon as the Mobile App receives the data from the RUSA device it transmits them to the physiological signals database (DB) via the internet. The physiological signals database is developed using the MongoDB software. MongoDB is a source-available cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with optional schemas.

The physiological signals database is connected with an Application Programming Interface (API) that can receive requests and can control the DB in order to provide the corresponding data to a specific endpoint. These requests are performed by the Physiological Signals Stress Detection Module of xR4DRAMA which in turn predicts the first responder's stress level at a specific timestamp. The stress level results are saved to a different database, the Stress Level Results DB, and are also reported to the Knowledge Base (KB) server of xR4DRAMA. The Stress Level Results DB is another MongoDB database that is connected with an API that can receive query requests per Person ID (unique identifier of first responder wearing the RUSA device) and per timestamp (both for a specific timestamp with accuracy in seconds or based on a time window with specific starting and ending timestamp). The complete high-level architecture of the sensor data integration is presented in Figure 3.

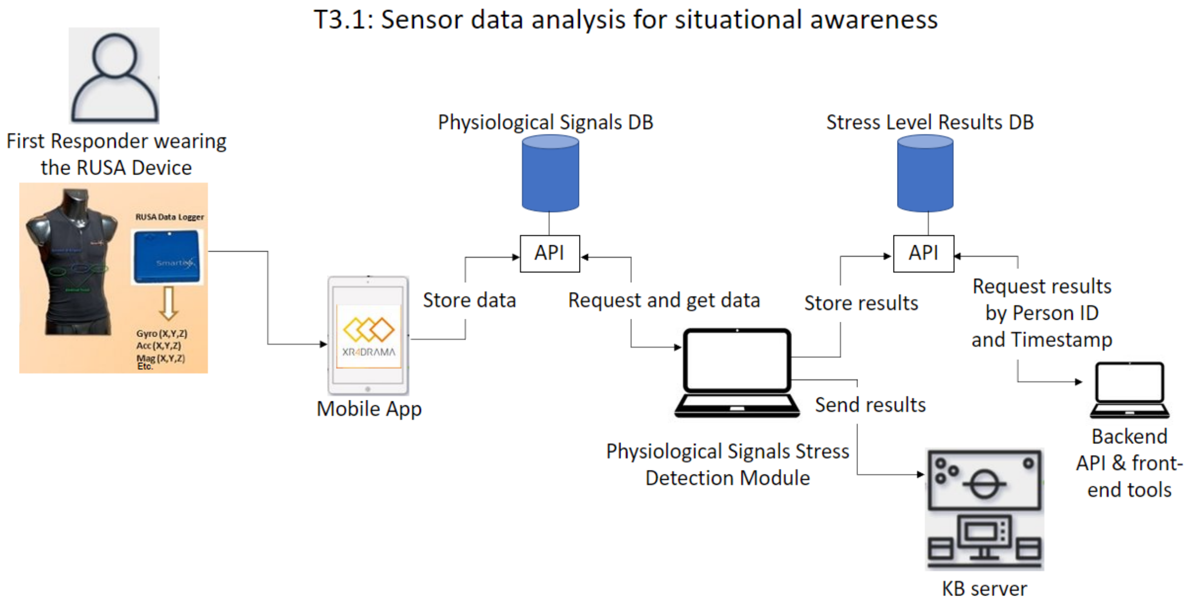


Figure 3. Sensor data analysis system architecture.

Integration of Mobile App with RUSA device

The data transmission between the RUSA device and the mobile application is based on the Bluetooth protocol. For Bluetooth-enabled devices to connect and exchange data between each other, firstly, have to initialize a channel of communication using a pairing process. A discoverable device makes itself available for incoming connection requests. A **BluetoothDevice,** a class of **android.bluetooth,** let's you create a connection with the respective device and receive information about it, such as the name, class, and bonding state. In our case the **SewDevice** from the **SewAndroidLibrary**, creates a new instance of device for the RUSA to establish a connection threat with our mobile device.

By the moment that the connection is established we have the following device states:

Idle: It may automatically start to record a new session.

Streaming: In this state, the sensor device shall send all active channels at a regular interval to the master. All sample vectors within a frame should correspond to a same time-frame.

Recording: The recording state may be active in parallel with the streaming state. While recording, the device may keep the Bluetooth module as ready to accept an incoming connection, but this mode could be altered by the user.

The next major functionality is data transmission. The **OnDataReceived** function handles the channels arriving and visualizes the values of the samples to the application interface. The main classes here is the **SewDataChannel** and **SewSample** with their functions **getSamples()** which gets the list of samples, **getChannel()** for getting the data channel to which samples belong to , **getValue()** gets the sample value and **getTimestamp()** gets the sample timestamp (in ms). For test the accuracy of the transmission these values are visualized with a *textView* in the app GUI.

Integration of Mobile App with the Physiological signals DB

Once the data from the RUSA device have been received by the Mobile App, they need to be forwarded to the physiological signals DB via the internet. This functionality is implemented through a direct TCP/IP connection between the Mobile App and the PC server hosting the physiological signals DB. The PC server binds a specific IP and PORT for the Mobile App to connect to. When the parameters of

Table 1 are available they are forwarded through the TCP/IP connection. The collection and organization of the entries for the MongoDB take place on the server hosting the DB. This can be done by utilizing the Python Library Mongo engine. Mongo engine allows the creation and handling of a MongoDB collection by creating a model of a database with all the fields that need to be stored. The Mongo engine model can then be called and fill the necessary fields with the received physiological signals to create the entries to the DB.

The Physiological Signals Stress Detection Module of xR4DRAMA evaluates the stress level of the first responder that is wearing the garment with the RUSA device at a specific timestamp. However, in order for the module to make a prediction, a certain timeframe of data needs to be collected beforehand. This timeframe is adjustable and regarding the xR4DRAMA's requirements it has been set to 5s. This timeframe does not hinder the real time stress detection capabilities. Thus, the Physiological Signals Stress Detection Module produces a result regarding the stress level of the first responder every 5s. The sensor data are being analysed at this timeframe. The analysis is performed in real time so no delay is imposed on the system. The data measurements described in

Table 1 are being organized so that they will correspond to this timeframe before they are stored in the physiological signals database. For example, 5s of ECG data for the RUSA device which utilizes a single lead ECG sensor (singular value per sample) means that the message will include 1250 ECG values that match the sensor's sampling rate (250 Hz) for the specified timeframe. The same notion applies to all recorded parameters described in Table 1 each according to the sensor sampling rate. The shapes of data for every entry in the physiological signals DB according to their sampling rate for the timeframe of 5s are presented in Table 2.

Table 2. Input shapes of recorded parameters in the physiological signal DB.

Recorded Parameter	Sampling Rate	Input shape (per 1 entry in DB)
ECG data	250 Hz	1 x 1250
AccX-Y-Z data	25 Hz	1 x 3 x 125
GyroX-Y-Z data	25 Hz	1 x 3 x 125
MagX-Y-Z data	25 Hz	1 x 3 x 125
RespPiezo data	25 Hz	1 x 125
Q0-Q1-Q2-Q3 data	25 Hz	1 x 4 x 125
ECGHR data	1/5 sec	1 x 1
ECGRR data	1/5 sec	1 x 1

BR data	1/5 sec	1 x 1
Activity Pace data	1/5 sec	1 x 1

Integration of Physiological signals DB with the Physiological Signals Stress Detection Module

Every request to the DB API returns a package of the sensor data that correspond to the specified timeframe. Whenever the collected data from the RUSA device reach the timeframe mark an entry is included in the signals DB and their unique DB entry ID is being posted on a specific endpoint (e.g., http://160.40.53.24:5100/ids/) which is being requested by the Physiological Signals Stress Detection Module. When the Physiological Signals Stress Detection Module gets the ID that has already been posted on the endpoint it creates another query request to the DB based on the ID. This is a different request to a different endpoint (e.g., http://160.40.53.24:5100/entry/sbj1_1_0) which returns the physiological signals that are stored in the DB to the Physiological Signals Stress Detection Module in the form of a JSON message.

The development of the API that controls the physiological signals DB and responds to the requests made by the Physiological Stress Detection Module has been done in Python via the Flask web framework. Flask is classified as a micro framework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where pre-existing third-party libraries provide common functions. Flask provides functionality for building web applications, including managing HTTP requests and developing the API behind the application.

Integration of the Physiological Signals Stress Detection Module with the KB server

Once the Physiological Signals Stress Detection Module receives data from the physiological signals DB the stress detection algorithm evaluates the stress level of the first responder and reports the result by posting a JSON message to the KB server. This is done by utilizing the Python request library which enables the posting of a JSON file to a specific endpoint. This message includes the person’s unique identifier, the current timestamp, the stress level and the probability of the predict value (confidence score of the prediction). An example of such a message is show bellow:

{"personid": "sbj1", "timestamp": "14/09/2021 13:56:46", "stresslevel": "20", "probability": "0.9"}

The stress levels are defined on a scale with 10 levels within the concept of xR4DRAMA (e.g. Figure 4). Hence the Stress Detection Module reports the predicted stress level with a value representing one of those 10 scales.

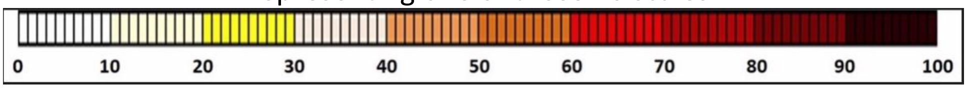


Figure 4. Stress level scales within xR4DRAMA.

The Physiological Signals Stress Detection Module utilize the mongo engine tool to create and call a model for the creation of a database that will store all the results produced by the Physiological Signals Stress Detection Module. The stress level results DB is connected with an API which responds to requests made by person ID (unique identifier of the first responder wearing the vest) or by specific timestamp or by timestamp range (with start and end). Here are the three endpoints for the aforementioned tasks:

- https://160.40.53.24:5200/sensors/stresslevel/id/ <entry_id> . This endpoint returns the stress results for the unique identifier of the person wearing the smartvest. Note that the entry_id value needs to be given by the system component that is making the request. It is a search request by ID.
- https://160.40.53.24:5200/sensors/stresslevel/result/<timestamp> . This endpoint returns the person ID, the timestamp and the stress level and the probability of the predicted stress level for the specific timestamp that is being requested. Note that the timestamp value needs to be given by the system component that is making the request. It is a search request by timestamp.
- https://160.40.53.24:5200/sensors/stresslevel/<fromTimestamp=X&toTimestamp=Y> . This endpoint returns the person ID, the timestamp, the stress level and the probability of the predicted stress level, for a specific range of timestamps.

These three endpoints provide the sensor data analysis results to the backend API and the frontend tools of the xR4DRAMA project.

3.3 Sensor data analysis

The analysis of the physiological sensors aims to a) detect the stress levels of the subject and b) recognize their activities. Both machine learning and deep learning algorithms were tested for the stress detection, while for the activity recognition only ML was used. During the pilot trials in the project, the need is to report the stress near real time and perform activity recognition when needed, offline. The methodologies followed are described below, separately for activity recognition and stress detection.

3.3.1 Activity recognition

Only the inertial sensors of the smart vest, namely accelerometer, gyroscope and magnetometer, were utilized to recognize the activities conducted. The smart vest itself provides an automatic recognition of activity classes based on the energy of the activity. This labelling was used as the ground truth.

Activity recognition applications from wearable sensors have become quite popular in the last years in everyday life as well as in monitoring systems for medical use. Inertial sensors, especially accelerometer, have proven to be very effective in recognizing daily activities like standing, walking, moving upstairs etc, although their performance may be affected by their placement or the type of the activity conducted.

Sensors produce signals that usually contain noise. These initial signals are used to extract features, categorized as time and frequency domain, that reduce noise and improve the classification results especially when used in machine learning applications.

The typical procedure of an activity recognition pipeline consists of the following steps:

1. Filtering: this is an optional step that aims to reduce noise

2. Feature extraction: the initial signals are used to extract time and frequency domain signals that optimize the performance of the later applied classifier
3. Feature selection: this step is important when many features are extracted. Only the variables that contain the most valuable information will be used in the classification algorithm.
4. Application of a classification algorithm: the initial sample is usually split to train and test sets. Various algorithms are trained on the training sample and then applied on the test set to explore their performance.

3.3.2 Stress detection from sensors

For the task of stress detection both ML and DL algorithms were tested. All the physiological sensors embedded on the smart vest were used separately and in combination to improve the detection of stress levels. In relevant literature, ECG is found to be the most effective sensor to measure stress. This was also confirmed by our experimental results.

In machine learning applications for stress detection from physiological sensors, features need to be extracted, similar to the activity recognition applications. Extracted features summarize the information included in the signal and make it more exploitable.

4 EXPERIMENTAL RESULTS

In this section we mention some of the experiments performed in samples of physiological signals. The valuable information for the end-users is stress detection, however we also include some experiments using only the inertial sensors in order to recognize activities. As already mentioned, stress will be detected using sensor and audio signals, while for the activity recognition only sensor signals will be used. In the current deliverable we include the results of deep learning applications on physiological sensors for stress detection and machine learning application for activity recognition.

4.1 Deep learning application on pre-training sample

4.1.1 Technology

Deep Learning is the technology employed by the Physiological Signals Stress Detection Module to tackle the stress level detection requirements of T3.1. Deep Learning attempts to mimic the activity in layers of neurons in the neocortex, the wrinkly 80 percent of the brain where thinking occurs. The software learns, in a very real sense, to recognize patterns in digital representations of sounds, images, and other data. Deep learning is a subset of artificial intelligence and machine learning that uses multi-layered artificial neural networks to deliver state-of-the-art accuracy in tasks such as object detection, speech recognition, language translation and others. The basic idea—that software can simulate the neocortex's large array of neurons in an artificial "neural network"—is decades old (Figure 5), and it has led to as many disappointments as breakthroughs. But because of improvements in mathematical formulas and increasingly powerful computers, lately it is possible to model many more layers of virtual neurons than ever before. Those highly flexible architectures can learn directly from raw data and can increase their predictive accuracy when provided with more data.

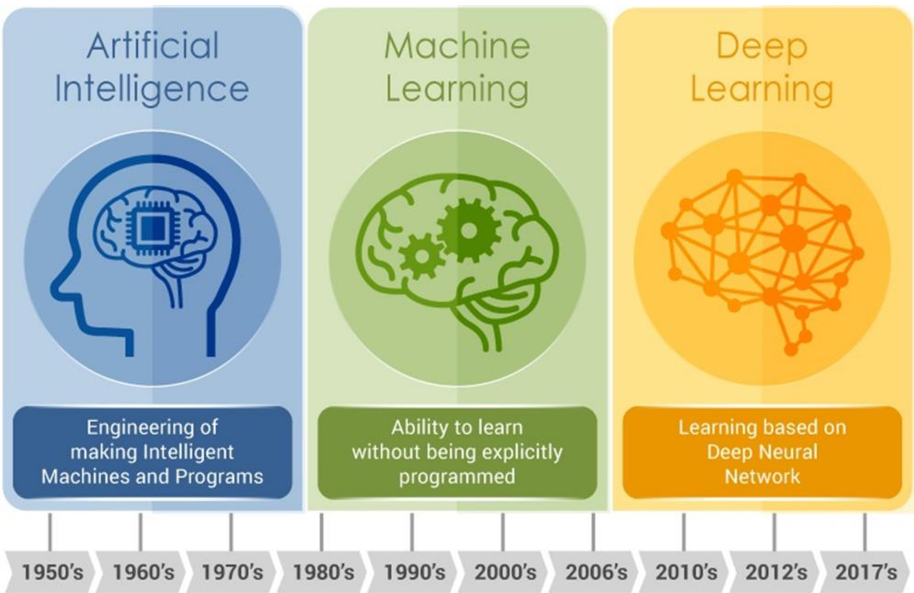


Figure 5. Evolution of deep learning.

Recent Deep Learning architectures are based on the Convolutional Neural Networks (CNN), which is a new neural network type, usually used with 2D or 3D inputs. As in conventional neural networks, CNNs are also composed of interconnected neurons, the output of which depends on their inputs and a set of trainable parameters. Those parameters (often referred as “weights”) are multiplied with the inputs and the products are added together to calculate the output value of a specific neuron. Neurons are usually organized in successive layers, with the neurons of each layer being interconnected with all neurons of the previous one. Figure 6 presents a simple 3-layer conventional neural network.

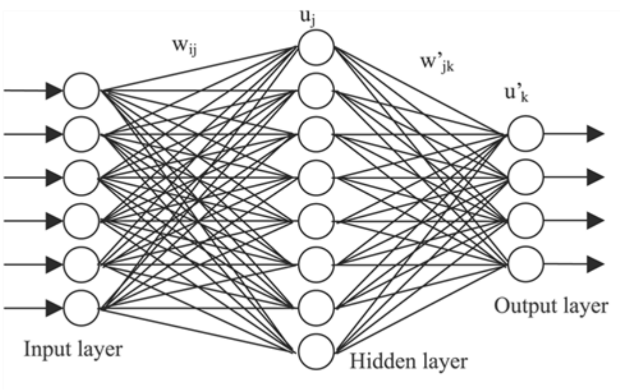


Figure 6. Conventional 3-layer neural network.

In CNNs, neurons of convolutional layers are connected only with a subset of the precious layer’s neurons, thus having a limited Field of View (FoV). Consequently, the output value of each neuron is affected only by the corresponding region of the input. Another key difference is the sharing of trainable parameters. In conventional neural networks, each neuron has its own distinct set of trainable parameters. In contrast, the neurons of each convolutional layer share a common trainable parameter set. This leads to the CNN processing all input regions in the same way. As a result, when the input is an image, the CNN can detect an object in any place within the image. An example of a simple classification CNN is presented in Figure 7.

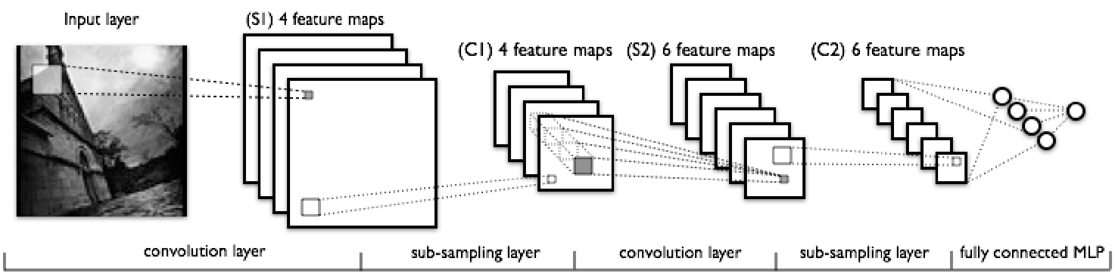


Figure 7. Convolutional Neural Network (CNN) example.

Prior research on the field of affection recognition has shown that analysing physiological signals is a reliable predictor of stress ([1], [5], [6], [8], [11]). T3.1 of xR4DRAMA includes the processing of such signals in time and frequency domain. In particular, electrocardiograph (ECG), heart rate, heart rate variability, RR peaks, respiration rate and breathing amplitude are the types of signals that are taken into consideration for the estimation of situation awareness. Furthermore, T3.1 also includes motion type signals measuring the participant's acceleration, the angular rate, and the magnetic field surrounding the body in X-Y-Z axis using accelerometer, gyroscope and magnetometer, respectively. These are also analysed in terms of estimation of the situation awareness by analysing the personalised reaction of the first responders to the situation. The personalised reaction will be measured by predicting the stress level of the first responders. The stress detection tool can assist during a training phase or in a simulation of a real scenario by providing data on how the first responders react against an emergency and how to deal with the risk.

The main task of T3.1 is to analyse these different types of physiological and motion signals by applying signal processing techniques aiming to filter noise and extract informative features that will help to model a stress detection algorithm. However, this goal does not necessarily need to rely on the commonly employed combination of hand-crafted features with a traditional machine learning method for stress detection analysis, but instead utilize the latest deep learning technology to learn directly from the raw data through the layers of the neural networks. After all, traditional machine learning methods may show decreases in accuracy if features are misidentified.

For this reason, a custom deep neural network that analyses physiological data collected from the RUSA device to perform two tasks is developed within xR4DRAMA. The first task is a 10-class classification problem for stress detection, in which the developed network classifies the input physiological signals between the aforementioned 10 scales of stress levels (see Figure 4) that have been identified within xR4DRAMA. The second task is a regression problem for the prediction of the final stress level score from an input of physiological signals, which is a value from 0-100 (see Figure 4). Both neural networks were trained and tested on the dataset that was collected within xR4DRAMA.

Functionality of Physiological Signals Stress Level Detection Module

The purpose of the Physiological Signals Stress Detection Module is to utilize machine learning and deep learning methodologies to automatically analyse the captured data of xR4DRAMA's smartvest in order to estimate the stress level of a first responder wearing the vest. Promising results have been measured on the xR4DRAMA dataset on the task of classifying the first responders stress level. A DL architecture that will model the physiological signal signatures for 10 different stress scale levels (see Figure 4) is proposed. This handles the stress detection task as a 10-class classification problem. The goal of this approach is to narrow the error margins by having less targets to predict. Instead of having to predict correct in at least 100 different cases which would apply to regressing the correct score, the proposed approach has to predict correct in 10 cases. Thus, this approach is expected to yield better accuracy.

Nevertheless, since the stress levels are identified on a scale from 0-100 within xR4DRAMA, a different DL architecture that can regress the final stress level score is also proposed. Regression is one of the most important and broadly used machine learning and statistics tools out there. It allows you to make predictions from data by learning the relationship between features of your data and some observed, continuous-valued response. However, regression is not only limited to machine learning methods but can also be applied with deep learning.

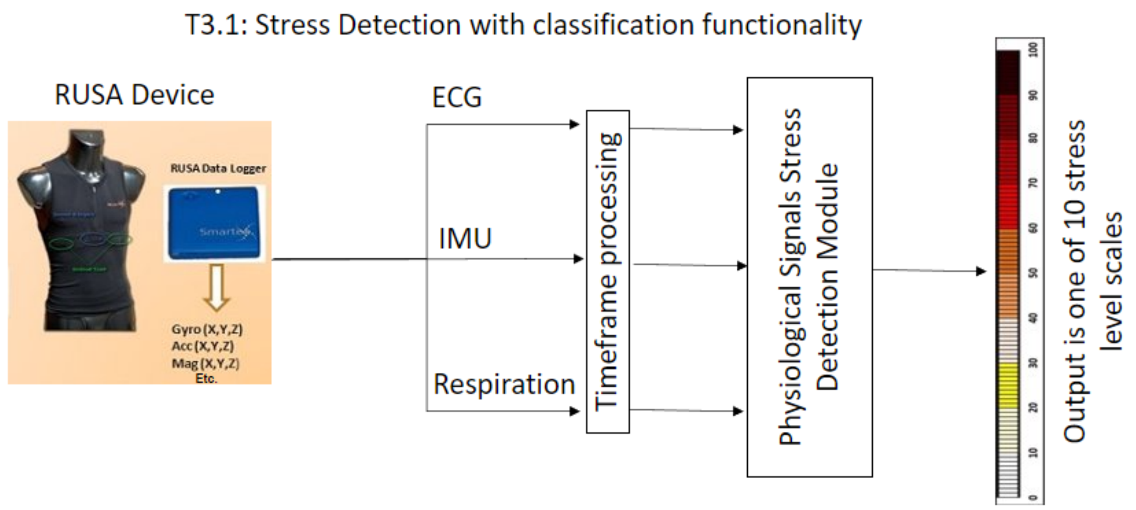
The best performing method will be utilized in the xR4DRAMA's final version. Both architectures will learn from the data without any feature extraction step that stems from specialized training in physiological signal processing. To achieve such a feat, information analysis on the physiological signals is evaluated. In order to approach the stress level detection analysis of T3.1, the proposed module will make use of all information available from the RUSA device. The available information would be the parameters described in

Table 1. In particular, the **ECG measurements**, including the ECG signal and the extracted parameters that the RUSA calculates such as the Heart Rate, Heart Rate Variability, and R-R interval distance in ECG signal are utilized as input to the proposed module. Furthermore, the **IMU measurements** placed on the chest which is measuring the first responder's acceleration, the angular rate, and the magnetic field surrounding the body in X-Y-Z axis using accelerometer, gyroscope and magnetometer, respectively are also utilized as input. Finally, the **respiration measurements** such as the Breathing Rate, and the Breathing Amplitude of the participant are also utilized as input.

The Physiological Signals Stress Detection Module of xR4DRAMA evaluates the stress level of the first responder that is wearing the RUSA device at a specific timestamp. However, in order for the module to make a prediction, a certain timeframe of data needs to be collected beforehand. This timeframe is adjustable and regarding the xR4DRAMA's requirements it has been set to 5s. The reason why the timeframe is chosen to be 5s is because it matches most of the RUSA device parameters sampling rates (see

Table 1) meaning that most of the input data can be processed to meet this timeframe criterion before they can be used as input. Furthermore, this timeframe selection is close to what is recommended as a standard procedure according to the literature. In most recent works the proposed stress detection methods evaluate the stress levels every 5s, 10s or 30s ([1], [5], [6], [8], [11]). This timeframe does not hinder the real time stress detection capabilities. Thus, the Physiological Signals Stress Detection Module produces a result regarding the stress level of the first responder every 5s.

The two DNNs that will be trained based on such samples will output whether stress level of the first responder either as one of the 10 classes identified or as a score from 0-100. The two modules functionalities are presented in Figure 8.



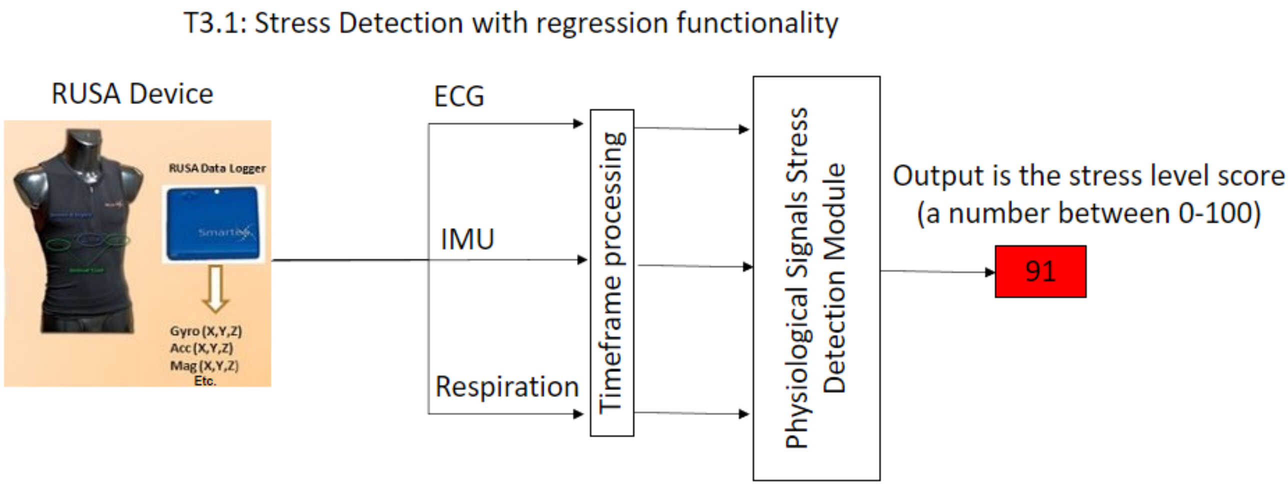


Figure 8. Physiological Signals Stress Detection Module functionality description.

Architecture of developed Deep Neural Networks

The basic building block of the Physiological Signals Stress Detection Module is the Convolutional Neural Network (CNN) [14]. Driven by the unprecedented success of the image recognition, object detection architectures (2D CNNs) ([15], [17], [18],) and signal classification (1D CNNs or Recurrent Neural Networks) [16], the proposed networks are designed from scratch based on a similar logic. One recommended DNN architecture based on literature that solve the problem at hand is [11]. We differentiate our method based on the input data which are different with [11], but the logic behind the DNN architecture is similar. Consequently, the proposed method is considered novel and is tailored to the xR4DRAMA's needs and requirements.

The most important factors that determine the network architecture are the input data shapes and the number of inputs. The number of physiological and motion signals that are used as input and their input shapes per the selected timeframe of 5s is reported in Table 2. The optimal 1D CNN architecture is presented in Figure 9. The ECG signal of size 1250x1 is currently used as the single input vector. The input signal goes through three 1D CNN layer with 3x3 convolution filters and [8,16,32] feature maps respectively. All three CNN layers are followed by a Batch Normalization layer, a Rectified Linear Unit (ReLU) activation [20] and Max Pooling layer resulting in an output shape of Nx36x32 at the end of the third CNN block. Every CNN block is followed by a dropout layer that closes off half of the activation neurons to prevent overfitting [22]. The output feature map of shape Nx36x32 describes the list of input signals. This block represents the feature extraction step and those features are followed by a fully connected layer with 10 outputs and a softmax activation function which is performing the classification task. The result of the fully connected layer will be the output of the Physiological Signals Stress Detection Module to the rest of the xR4DRAMA's system components, classifying the input signals between the 10 available stress levels with a certain probability asserted by softmax activation function.

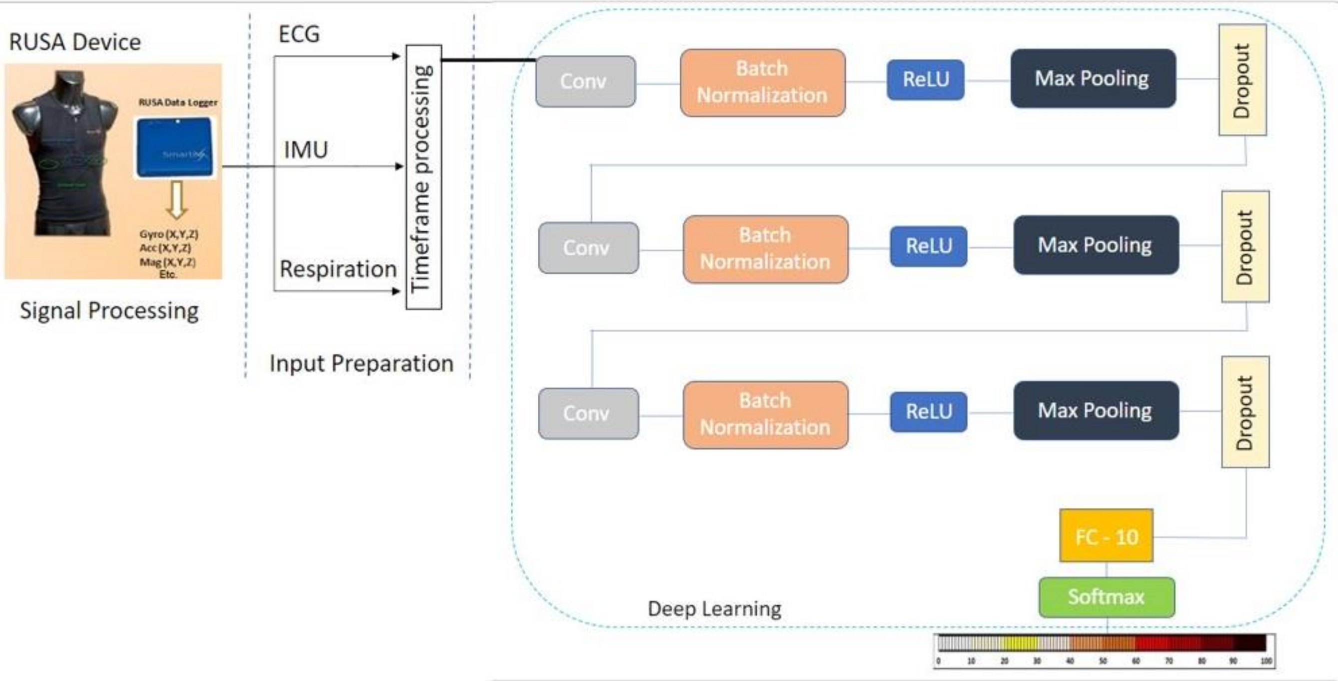


Figure 9. Proposed architecture for classification using only the ECG 1D input signal.

Experimental results prove that those dimensions are the optimal for higher classification accuracies. The proposed DNN is trained from scratch and we use the Xavier initialization [21] for the initial weight values. The dropout layers [22] probability is p = 0.5 and is applied to prevent overfitting effects. The objective function is set to be the categorical cross-entropy. We utilize the Adam [19] optimizer with an initial learning rate of 0.001. The network was trained for 100 epochs with a batch size of 128.

Experimental results

The proposed DNNs for the classification and regression methods were trained on a subset of recordings acquired from the experiments performed by xR4DRAMA's partner AAWA first responders. The participants followed the experimental protocol described in Section 3.1 while wearing the smart vest. A total of 7 end users took part in this data capturing campaign.

In Table 3 the number of samples for a timeframe of 5s per stress scale (10 classes problem) are presented. These are the results that the first responders reported after completing each stress level test as part of the experimental protocol. It can be seen the last three stress level scales does not have any reported samples; thus, the system cannot be trained to predict those stress level scales due to the lack of appropriate training data. As a result, the 10-class classification problem should be handled as a 7-class classification problem.

Table 3. Number of samples for a timeframe of 5s reported by end users per stress level scale.

	Stress level Scales										
	0-10	11-20	21-30	31-40	41-50	51-60	61-70	71-80	81-90	91-100	SUM
User1	-	-	27	128	27	135	68	-	-	-	385
User2	121	206	26	27	-	-	-	-	-	-	380
User3	60	-	37	-	78	180	30	-	-	-	385
User4	-	228	-	24	30	66	54	-	-	-	402
User5	296	24	-	36	-	-	30	-	-	-	386
User6	246	78	36	-	30	-	-	-	-	-	390
User7	299	24	36	-	-	30	-	-	-	-	389
SUM	1022	560	162	215	165	411	182	-	-	-	2717

In Table 4 the number of samples for a timeframe of 5s per reported stress level scores (stress level regression problem) are presented. These are the results that the first responders reported after completing each stress level test as part of the experimental protocol. It can be seen that not all scores are represented by the dataset. Thus, it should be expected that the trained model will have difficulties predicting all stress level scores (0-100). Nevertheless, it should not be impossible to see even scores outside of the reported scores since this regression model is trained to predict any value from 0-100.

Table 4. Number of samples for a timeframe of 5s reported by end users (reported stress level score).

Stress level Scales																							
0	10	15	20	24	25	30	35	40	42	43	47	45	50	53	55	60	65	69	70	70-80	80-90	90-100	SUM
-	-	-	-	-	-	27	-	128	-	-	-	-	27	-	-	135	-	-	68	-	-	-	385
-	121	144	62	-	26	-	-	27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	380

385	402	386	390	389	2717
-	-	-	-	-	-
-	-	-	-	-	-
-	-	-	-	-	-
-		30	-	-	98
30	-	-	-	-	30
-	54	-	-	-	54
36	30	-	-	30	231
-	36	-	-	-	36
144	-	-	-	-	144
-	30	-	-	-	57
-	-	-	30	-	30
30	-	-	-	-	30
24	-	-	-	-	24
24	-	-	-	-	24
-	-	36	-	-	191
-	24	-	36	-	60
-	-	-	-	36	63
13	-	-	-	-	39
24	-	-	-	-	24
-	204	24	30	24	342
-	24	-	48	-	216
60	-	144	246	168	739
-	-	152	-	131	283

For this version of D3.1 the DNN were trained and tested using only the User1 data from Table 3. This recording was available from July 2021 (M9) in order to proceed with development and have a prototype version ready by early November 2021 (M13) for the 1st project use case. The rest of the data recordings were available on October 2021 (M12) which is the due date for D3.1, thus there was not enough time to include them in the document. Nevertheless, the new data will be added to refine the available models and the results will be reported in D3.7 – Sensor data analysis for situational awareness v2.

The results of the DNN classifier are summarized in Table 5 and Table 6. Per class precision, recall and F1 score metrics are presented in both Tables. Table 5 presents an experiment testing on the entire User 1 recording. Table 6 presents an experiment on the test set that has been separated from the training procedure. The overall accuracy of the classifier on the annotated User 1 recording is 85%. While the overall accuracy of the classifier on the test set (20% of the User 1 samples that have been kept away from the training phase) is 43%.

Table 5. Per class precision, recall and F1 score for the User 1 recording.

Stress level	precision	recall	F1 score	samples
0-10	0.87	0.92	0.89	121
11-20	0.95	0.83	0.88	206
21-30	0.64	0.88	0.74	26
31-40	0.59	0.81	0.69	27

Table 6. Per class precision, recall and F1 score for the User 1 recording.

Stress level	precision	recall	F1 score	samples
0-10	0.61	0.58	0.6	24
11-20	0.62	0.34	0.46	41
21-30	0.15	0.4	0.22	5
31-40	0	0	0	5

By testing the classifier accuracy on all 380 samples of User 1 recording we can see that the results are looking promising. However, when we test on the remaining 20% of the data that was kept outside of the training procedure, we can see that the accuracy drastically drops leading into an overfit problem. As we can see the problem lies specifically on the medium stress level classes which have the lowest amount of data.

4.1.2 Issues and Future work

One of the biggest issues was the lack of availability of data representing all stress level scales. In particular User 1 data only had 4 different stress levels (0-10, 11-20, 21-30, 31-40) reported which were mostly low to medium stress levels, so the initial model had limited data to work with. This issue was alleviated by the addition of other 6 users who reported additional 3 stress levels (41-50, 51-60, 61-70) and the dataset was enriched. However, the dataset still lacks data representing the highest stress level scales (71-80, 81-90, 91-100) which makes the model unable to predict such high stress level scales. The activities performed by the first responders during PUC1 will be recorded when wearing the and due to the fact that this will be on field simulation it may trigger higher stress responses to the participants. Thus, more relevant data will be utilizing to fine tune the trained stress detection model.

Furthermore, the developed model is to be trained and tested on the complete dataset of xR4DRAMA with more participants and more samples for all stress levels. Finally, working on the general improvement of the model, it is expected to add more data channels from the RUSA device as input and not only the ECG signal.

4.2 Machine learning algorithms for activity recognition from inertial sensors

In this section, we refer to the results of the activity recognition based on the inertial sensors, namely accelerometer, gyroscope, magnetometer, and quaternion, embedded in the smart vest. The activity recognition can be divided into two categories, namely recognition of the activity class and the activity energy. The activity class has 5 levels ranging from 0 to 4 when the activity's energy is a continuous value ranging from 0 to 255. For both of these categories, we performed feature extraction on the sensors' data. We applied a time window of 5 seconds since the activity labels are reported with a frequency of 0.2 Hz. We extracted 48 simple statistical time and frequency domain features resulting in a total of 192 features. We tested the performance of 4 machine learning algorithms, those being Support Vector Machines (SVM), k-Nearest Neighbors (kNN), Random Forest (RF), and eXtreme Gradient Boosting trees (XGB), along with fusion and feature selection techniques.

4.2.1 Recognition of activity class

For the task of recognizing activity classes, we split the data into training and testing data having a testing size of 0.2. We compared the performance of each modality alone and various early and late fusion strategies. For early fusion, we performed concatenation, and for late fusion, we tested averaging, product of probability, maximum probability, and majority voting.

The averaging technique is described by equation 1:

$$p_i = \frac{1}{N}\sum_{j=1}^{N} p_{i,j} \qquad (1)$$

where $p_{i,j}$ is the probability of the i class of the j modality, N is the total number of modalities, and p_i is the average probability of the i class across all modalities. The final decision is performed by locating the class with the maximum probability.

The product technique is described by equation 2:

$$p_i = \prod_{j=1}^{N} p_{i,j} \qquad (2)$$

where $p_{i,j}$ is the probability of the i class of the j modality, N is the total number of modalities, and p_i is the product probability of the i class across all modalities. The final decision, again, is performed by locating the class with the maximum probability.

For the maximum probability technique, we find the class with the highest probability across all modalities, and for the majority voting, we find the class that has been predicted more times across all modalities.

The results of the fusion techniques can be seen in Table 7. From the Table it can be seen that the classifier performing the best is the XGB classifier, achieving an accuracy score of 85,11% when using the concatenation fusion technique. The best performing single modality is the accelerometer, achieving an accuracy score of 84,56%.

Table 7. Accuracy score of the activity class prediction using fusion techniques.

	Acc	Gyro	Mag	Quat	Concat	Averaging	Prod	Max prob	Maj. Vote
SVM	0.7960	0.7996	0.8051	0.7960	0.8051	0.7996	0.8015	0.7941	0.7960
kNN	0.8051	0.7886	0.7904	0.7868	0.7904	0.8015	0.8015	0.7941	0.7960
RF	0.8272	0.8088	0.8162	0.8162	0.8235	0.8033	0.8015	0.7960	0.7960
XGB	0.8456	0.7996	0.8088	0.7978	**0.8511**	0.8015	0.8033	0.7941	0.7960

It is worth mentioning that from a total of 544 samples, the 433 had the same label, that being 2, which stands for sitting/standing. So an accuracy score of 79,6%, even though it seems high, it predicts everything as class 2. In the Table, the numbers colored in red represent such cases where a classifier predicts every single class as 2.

For the feature selection technique, we applied three methods, being Recursive Feature Elimination (RFE), Principal Component Analysis (PCA), and Genetic Algorithm (GA) based feature selection. Results of the feature selection techniques can be seen in Table 8. From the Table, we can conclude that XGB is again performing the best having an accuracy score of 87,13% when using the GA technique, while the RFE technique resulted in all classifiers predicting every class as 2.

Table 8. Accuracy score of the activity class prediction using feature selection techniques.

	RFE	PCA	GA
SVM	0.7960	0.8107	0.8051
KNN	0.7960	0.8107	0.8070
RF	0.7960	0.8107	0.8419
XGB	0.7960	0.8107	**0.8713**

4.2.2 Recognition of activity energy

Since the activity class labels are imbalanced (most of the cases are labeled as 2), we also performed an activity energy recognition. Since the activity energy is not a discrete value, we performed regression analysis in the activity energy levels. For fusion techniques, we again used concatenation for early fusion, while we performed two late fusion techniques, mean and median of the prediction of each modality. For evaluation, we used the mean squared error evaluation metric, after normalizing the activity energy level values to the range from 0 to 1.

Results of the fusion techniques can be seen in Table 9. The combination of concatenation and XGB classifier is again having the best result, with a mean squared error of 0,0074. Also, the best performing single modality is again the accelerometer, having a mean squared error of 0,0081 when using the RF classifier.

Table 9. Mean squared error of the activity energy prediction using fusion techniques.

	Acc	Gyro	Mag	Quat	Concat	Mean	Median
SVM	0.0291	0.0298	0.0332	0.0336	0.0332	0.0283	0.0315
KNN	0.0261	0.0231	0.0255	0.0335	0.0255	0.0214	0.0220
RF	0.0081	0.0186	0.0225	0.0231	0.0077	0.0150	0.0166
XGB	0.0084	0.0183	0.0239	0.0271	**0.0074**	0.0142	0.0154

For feature selection, we applied the same methods. The results are presented in Table 10. From the Table, it can be seen that the GA technique outperforms the rest of the feature selection techniques, with XGB again being the best performing classifier, with a mean squared error of 0,0048.

Table 10. Mean squared error of the activity energy prediction using feature selection techniques.

	RFE	PCA	GA
SVM	0.0309	0.0311	0.0218
KNN	0.0077	0.0255	0.0185
RF	0.0067	0.0217	0.0076
XGB	0.0076	0.0244	**0.0048**

5 CONCLUSIONS

This deliverable described the experiments and findings of the first period of the project regarding the analysis of sensor data, aiming to detect stress and recognize activities. For the task of stress detection, all physiological sensors embedded in the smart vest were used. The experiments conducted showed better performance of the ECG signal alone, which is in line with relevant literature. A subset of the sensors was used for the activity recognition, which in that stage of the project was not expected to provide good results, since most of the activities performed were static.

In order to train well performing models, data need to be collected from as much subjects as possible, so the training sample will have a variety of signals addressed to different stress levels. Currently, more data have been collected for sensor analysis and will be used to train and find the best performing models that will be deployed in the first prototype.

6 REFERENCES

[1] Schmidt, Philip, et al. "Introducing wesad, a multimodal dataset for wearable stress and affect detection." Proceedings of the 20th ACM international conference on multimodal interaction. 2018.

[2] Kirschbaum, Clemens, Karl-Martin Pirke, and Dirk H. Hellhammer. "The 'Trier Social Stress Test'–a tool for investigating psychobiological stress responses in a laboratory setting." Neuropsychobiology 28.1-2 (1993): 76-81.

[3] Healey, Jennifer A., and Rosalind W. Picard. "Detecting stress during real-world driving tasks using physiological sensors." IEEE Transactions on intelligent transportation systems 6.2 (2005): 156-166.

[4] Koelstra, Sander, et al. "Deap: A database for emotion analysis; using physiological signals." IEEE transactions on affective computing 3.1 (2011): 18-31.

[5] de Santos Sierra, Alberto, et al. "A stress-detection system based on physiological signals and fuzzy logic." IEEE Transactions on Industrial Electronics 58.10 (2011): 4857-4865.

[6] Pourmohammadi, Sara, and Ali Maleki. "Stress detection using ECG and EMG signals: A comprehensive study." Computer methods and programs in biomedicine 193 (2020): 105482.

[7] Spielberger, Charles D. "State-Trait anxiety inventory." The Corsini encyclopedia of psychology (2010): 1-1.

[8] Shi, Yuan, et al. "Personalized stress detection from physiological measurements." International symposium on quality of life technology. 2010.

[9] Koldijk, Saskia, et al. "The swell knowledge work dataset for stress and user modeling research." Proceedings of the 16th international conference on multimodal interaction. 2014.

[10] Sun, Feng-Tso, et al. "Activity-aware mental stress detection using physiological sensors." International conference on Mobile computing, applications, and services. Springer, Berlin, Heidelberg, 2010.

[11] Li, Russell, and Zhandong Liu. "Stress detection using deep neural networks." BMC Medical Informatics and Decision Making 20.11 (2020): 1-10.

[12] Sriramprakash, Senthil, Vadana D. Prasanna, and OV Ramana Murthy. "Stress detection in working people." Procedia computer science 115 (2017): 359-366.

[13] Keshan, N., P. V. Parimi, and Isabelle Bichindaritz. "Machine learning for stress detection from ECG signals in automobile drivers." 2015 IEEE International Conference on Big Data (Big Data). IEEE, 2015.

[14] LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. "Deep learning." nature 521.7553 (2015): 436.

[15] Trabelsi, Chiheb, et al. "Deep complex networks." arXiv preprint arXiv:1705.09792 (2017).

[16] Guberman, Nitzan. "On complex valued convolutional neural networks." arXiv preprint arXiv:1602.09046 (2016).

[17] He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[18] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.

[19] Kingma, Diederik P., and Jimmy Ba. "Adam: A method for stochastic optimization." arXiv preprint arXiv:1412.6980 (2014).

[20] Nair, Vinod, and Geoffrey E. Hinton. "Rectified linear units improve restricted boltzmann machines." Proceedings of the 27th international conference on machine learning (ICML-10). 2010.

[21] Glorot, Xavier, and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks." Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010.

[22] Srivastava, Nitish, et al. "Dropout: a simple way to prevent neural networks from overfitting." The journal of machine learning research 15.1 (2014): 1929-1958.