xR4DRAMA

Extended Reality For DisasteR management And Media planning

H2020-952133

D3.10

Stress level detection techniques v2

Dissemination level:	Public
Contractual date of delivery:	Month 24, 31/10/2022
Actual date of delivery:	Month 25, 11/11/2022
Work package:	WP3
Task:	T3.4 Stress level detection
Type:	Demonstrator
Approval Status:	Final version
Version:	1.0
Number of pages:	23
Filename:	D3.10_xR4Drama_StressLevelDetectionTechniquesV2_20221111_v1.0.pdf

Abstract

This deliverable describes the advanced versions and outcomes of the stress level detection component of xR4DRAMA developed in T3.4 of WP3. This component is responsible for developing body sensor-based and audio signal-based technologies for the assessment of the stress level experienced by actors in a situation. The results of the audio and sensor modules are then merged to obtain one unique stress prediction.

co-funded by the European Union

History

Version	Date	Reason	Revised by
0.1	03-05-2022	First draft voice stress module	Mónica Domínguez
0.2	14-10-2022	Second draft voice stress module	Jens Grivolla
0.2.1	24-10-2022	Additional revisions, corrections and formatting	Jens Grivolla
0.3	31-10-2022	Fusion module	Vasilis Xefteris
0.4	04-11-2022	Conclusions and formatting. Draft for internal review by Smartex.	Jens Grivolla
1.0	11-11-2022	Final based on review comments, adjust metadata	Jens Grivolla

Author list

Organization	Name	Contact Information
UPF	Mónica Domínguez	monica.dominguez@upf.edu
UPF	Jens Grivolla	jens.grivolla@upf.edu
CERTH	Vasilis Xefteris	vxefteris@iti.gr
CERTH	Georgios Tzanetis	tzangeor@iti.gr
CERTH	Emmanouil Michail	michem@iti.gr

Executive Summary

This deliverable describes the advanced versions and outcomes of the stress level detection component of xR4DRAMA developed in T3.4 of WP3. This component is responsible for developing body sensor-based and audio signal-based technologies for the assessment of the stress level experienced by actors in a given situation. The results of the audio and sensor modules are then combined in the fusion module to obtain one unique stress prediction.

The focus of this deliverable is on the audio-based stress level estimation as well as the fusion module, with the sensor-based analysis detailed separately in D3.7.

Abbreviations and Acronyms

CCC	Concordance Correlation Coefficient
FRs	First Responders
kNN	k-Nearest Neighbors
LLD	Low-Level Descriptors
ML	Machine Learning
MSE	Mean Squared Error
MuSe	Multimodal Sentiment
SVM	Support Vector Machines
XGB	eXtreme Gradient Boosting

Table of Contents

List of Figures

List of Tables

1 INTRODUCTION

As described previously in D3.4, the stress level detection component of xR4DRAMA, developed in T3.4 of WP3, is responsible for developing body sensor-based and audio signal-based technologies for the assessment of the stress level experienced by actors in a situation.

The audio-based stress detection system is designed to work with voice recordings from different sources, in particular phone calls (from citizens to emergency numbers) and voice messages from first responders (FRs). In addition, the stress level of the first responders is also assessed through physiological signals measured through a smart sensing vest developed to collect electrocardiograph, inertial measurement unit and respiration measurements data. The results of the audio and sensor modules are then combined to obtain one unique stress prediction.

The position of the stress level detection component in the xR4DRAMA architecture is depicted in Figure 1.

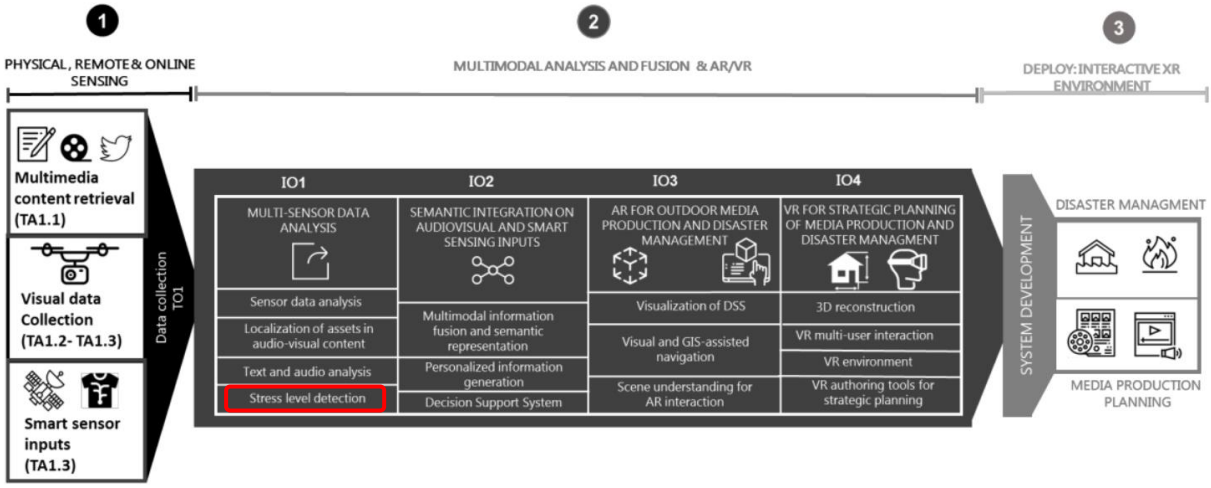

Figure 1: The stress level detection component in the xR4DRAMA architecture.

In this deliverable we describe an improved system for audio-based stress detection, and the fusion technique that is used in the second prototype for combining the results of the audio and sensor modules regarding stress.

2 AUDIO-SIGNAL BASED TECHNIQUES TO DETECT STRESS

Detecting stress using audio-signal based techniques requires two main processes: i) acoustic feature processing and extraction and ii) prediction (or classification) of the estimated level of stress. At a high level, these tasks can be described as audio signal processing, data processing and statistical inference via machine learning algorithms.

A brief report was done in the previous deliverable (D3.4) on the first attempts to tackle the task of stress detection in speech and version 1 of the stress detection module was integrated in the xR4Drama platform. Such preliminary implementation was essentially used to carry on basic integration and communication testing in operational terms. The functionalities of the module as such were quite limited in several respects. Hence, a full report on the tasks carried out to improve and develop a functional stress detection module from speech is presented in the current document. These activities include: i) to analyse the output of version 1 in the domain of xR4Drama, ii) to explore possible source of errors/limitations of version 1 and find remedial actions, iii) to gather in-domain training data, iv) to test new feature extraction techniques and experiment with different classification techniques, v) to implement and deploy new model as version 2, and vi) to evaluate the improvements of version 2 in comparison to version 1.

The following sections target tasks from i) to vi) in organised three blocks. Section 2.1 accounts for the analysis of the preliminary stress detection module and exploration of limitations to find remedial actions. Section 2.2 reports the collection and annotation of in-domain data and the experiments carried out for features extraction, data processing and classification. Finally, Section 2.3 describes the final implementation and integration of the functional module in the xR4Drama platform as well as the evaluation comparing version 2 to the baseline from version 1.

2.1 Testing the stress module

This section reports the evaluation of the version 1 of the stress detection module which had not been carried out at the time D3.4 was submitted. A brief description of the module is furthermore provided as context for the reader to better understand the improvements carried out in the current reporting period.

2.1.1 Description of version 1 of the stress detection module

The baseline system (version 1) to detect stress is based on standard open-source software both for the signal processing (i.e. Praat[1]) and the classification (i.e. Weka). A total of 18 acoustic features of the spectral envelope of voice are extracted that serve as features for the classification of stress. The data used to train this preliminary version were job interviews in German (from the ULM-TSST dataset (Stapen, et al. 2021)). The elicitation method of the speech samples is by means of an oral presentation under a job interview

[1] https://www.fon.hum.uva.nl/praat/

scenario in front of two silent interviewers. The length of the interviews is around 5 minutes, and the total recorded time of the dataset is 6 hours. 69 participants (out of which 49 were female) ranging from 18 to 39 years old took part in the recordings. Three annotators provided continuous dimensional ratings of valence and arousal (within a scale from 0 to 1). All three annotations were fused to construct the gold standard using the RAAW method and concordance correlation coefficient (CCC) of arousal reported is 0.186(+- 0.23). A total of 73 speech samples, segmented into 0,5 second fragments with their corresponding gold annotation values are used for training version 1 of the stress detection module.

2.1.2 Testing the stress detection module

The Stress Detection service is available at https://xr4drama.upf.edu/xr4drama-services/ under api/stress/estimate/. The service can be called using the following command:

curl -X POST "https://xr4drama.upf.edu/xr4drama-services/api/stress/estimate" -H "accept: application/json" -H "Content-Type: multipart/form-data" -F "user_id=" -F "timestamp=" -F "model_id=" -F "file=@spk2m_010.wav;type=audio/x-wav"

Testing the whole stress pipeline with the data available from the physiological and psychological experiment described in D3.4 and stored in Drive is carried out and reported in the current deliverable. In the experiment, participants reaction to changes in colour, relaxing music, mathematical operations, telling a stressful situation of people's life, etc was recorded and annotations of stress were provided from the participants at specific points in time.

Only files from the folders Stress_detection_data_2 and 3 are processed. The first folder seems to contain preliminary recordings, so files from Stress_detection_Data_1 have been discarded for this experiment. Figure 2 shows the distribution of output values from the stress module. At the very first sight, data shows that there are several incongruencies in the estimated values of stress. First of all, values outside the 0 to 1 range are found in all speakers resulting in a range from -0.5 to 1.7. As an immediate measure, we capped the numbers that were out of the 0 to 1 range, to at least produce a logical output that would not mess up with the rest of the pipeline. Moreover, average values (represented by the yellow horizontal line) show a concentration of values between 0 and 0.5, which would mean that the majority of samples (except for those belonging to speaker 5) are classified with very low values of stress. The overall accuracy of the stress module on the specific timestamps used for evaluation of the whole stress pipeline was 35%.

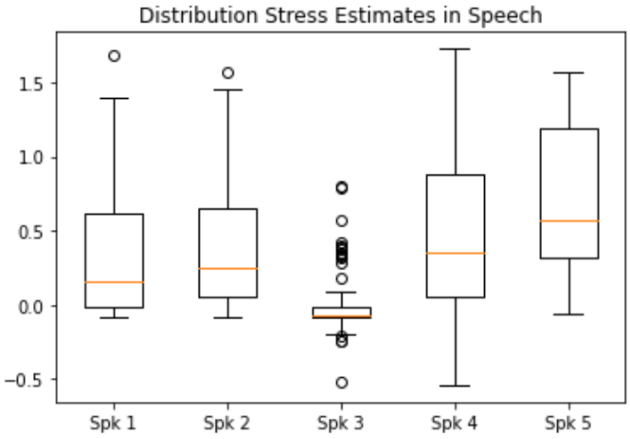

Figure 2: Distribution of stress values from audio files (output version 1).

A detailed analysis of each speaker was carried out to compare the estimated output of the stress module compared to the gold annotation at the specific timestamps (namely, at minutes 5, 17, 19, 21, 24, 26, 29, 31, 32 as shown by the vertical dotted black lines) used for the evaluation. Figures 2 to 6 show the results of this comparison for speakers 1 to 5.

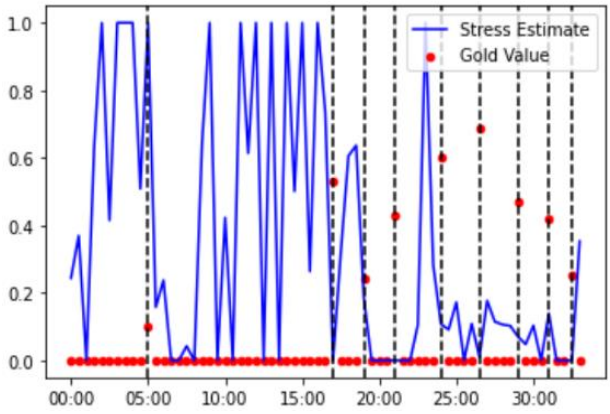

Figure 3: Comparison estimated versus gold stress value for speaker 1.

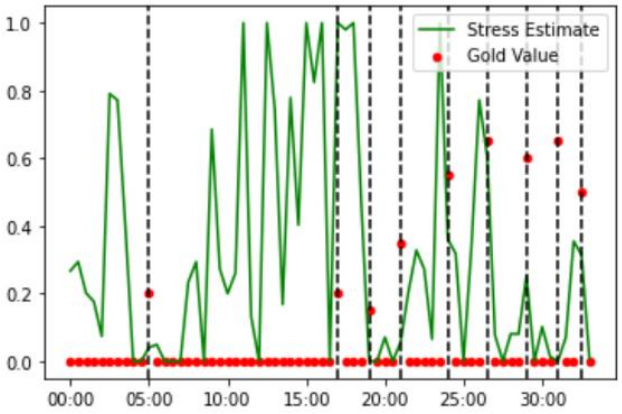

Figure 4: Comparison estimated versus gold stress value for speaker 2.

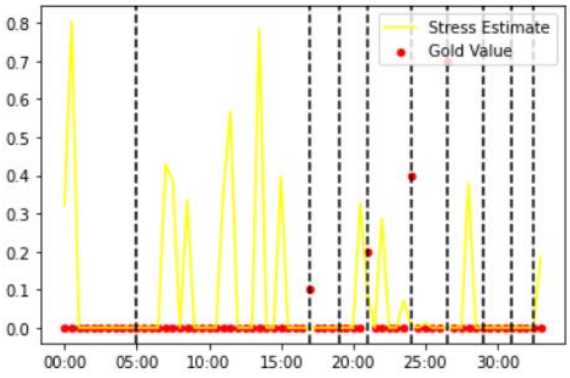

Figure 5: Comparison estimated versus gold stress value for speaker 3.

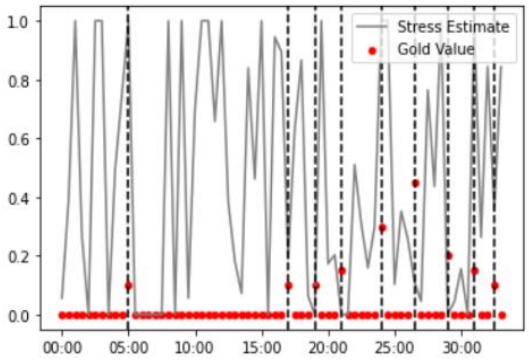

Figure 6: Comparison estimated versus gold stress value for speaker 4.

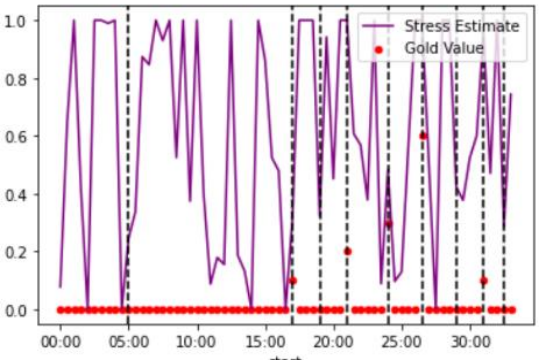

Figure 7: Comparison estimated versus gold stress value for speaker 5.

Limitations from the current model were made obvious after this test which served basically to devise an action plan to remedy these deficiencies. The very first deficiency is that the module is practically unable to estimate zero values of stress, with the exception of speaker 3 where the model is able to match value 0 of stress at evaluated timestamps at minutes 5,

29, 31, and 32. It also fails to even approximate values of stress in the majority of speakers with the exception of speaker 2 at timestamp 24 and 26 and speaker 1 at minute 19.

A step by step methodology was devised to actually test all techniques of the module as well as acquiring data that could better serve the goals of the project. Next section describes this process.

2.2 Towards version 2 of stress detection from voice

The results from testing version 1 of the stress detection module from voice indicated that the whole system required a thorough revision. Two key aspects were identified as fundamental to achieve an improvement: the data used to train the model, and the feature extraction technique. Once these two key areas are looked into, classification experiments were done to evaluate the process. The following sections report the activities carried out in these three areas.

2.2.1 Collection of in-domain annotated data

Material from AAWA had been provided in the domain of citizens' phone calls reporting emergencies. The dialogues were simulated but closer to real-PUC1 contexts than any other material previously used for developing the stress module. A total of 20 phone dialogues between citizens and operators were provided including mostly male voices usually performing different roles (as citizen and as operator). The task of annotating this material to gather training data for the stress module consisted in several steps: i) choosing an annotation tool, ii) developing annotation guidelines, iii) preparing the material for annotation, iv) follow-up the annotators completion of the task and v) processing the material for training.

We chose as annotation tool the open source software NOVA[2] (Baur et al, 2020) due to its user-friendly interface and compatibility with Windows OS. NOVA allows framewise labelling for a precise coding experience, and value-continuous annotations for labelling e.g emotions or social attitudes, including perception of stress in voice. The interface is customizable and allows loading and labelling data of multiple persons. The resulting continuous annotation can be exported as a csv file with timestamps.

Step-by-step annotation guidelines were provided to user partners from AAWA who kindly helped out in the annotation task including a demonstration video on how to use the NOVA software. In order to have a minimum amount of material for training a model, three annotations from different people are needed. We segmented the 20 dialogues into dialogue turns to isolate each speaker utterance for the annotation task. A total of 262 audio files were used for the annotation of stress. Three rounds of annotations were carried out and a total of 11 annotators took part in the process to split the amount of material and thus efficiently distribute the task. Thus, we obtained the minimum required amount of three annotations for each audio file.

[2] https://github.com/hcmlab/nova

Post-processing of annotated material was needed. We processed inconsistent file naming and computed the mean average score for each audio both as continuous values of stress in each audio file (at 40 milliseconds frames) and as one overall score per audio file. We also split the material into training and testing sets for machine learning experiments and validation. Table 1 summarises the audio files and average gold annotations for each file.

Table 1: Test set using for in-domain evaluation of the stress detection module

File name	Recording	Turn	Role	Speaker	Gold Annotation
rec01_009_op1m	rec01	9	operator	1m	0.29
rec02_010_ci3m	rec02	10	citizen	3m	0.23
rec03_004_ci2m	rec03	4	citizen	2m	0.73
rec03_008_ci2m	rec03	8	citizen	2m	0.85
rec04_004_ci3m	rec04	4	citizen	3m	0.37
rec05_002_ci4m	rec05	2	citizen	4m	0.20
rec06_001_op1m	rec06	1	operator	1m	0.43
rec07_004_ci1f	rec07	4	citizen	1f	0.56
rec08_011_op1m	rec08	11	operator	1m	0.26
rec09_010_ci1m	rec09	10	citizen	1m	0.59
rec10_001_ci1m	rec10	1	citizen	1m	0.38
rec11_006_ci1m	rec11	6	citizen	1m	0.70
rec11_008_ci1m	rec11	8	citizen	1m	0.70
rec11_017_op3m	rec11	17	operator	3m	0.33
rec11_027_op3m	rec11	27	operator	3m	0.32
rec13_001_op2m	rec13	1	operator	2m	0.59
rec14_001_op1m	rec14	1	operator	1m	0.23
rec14_013_op1m	rec14	13	operator	1m	0.23
rec15_009_op1m	rec15	9	operator	1m	0.19
rec16_007_op1m	rec16	7	operator	1m	0.29
rec17_010_ci3m	rec17	10	citizen	3m	0.10
rec17_019_op1m	rec17	19	operator	1m	0.21
rec18_010_ci1f	rec18	10	citizen	1f	0.45
rec19_011_op1f	rec19	11	operator	1f	0.45
rec20_004_ci4m	rec20	4	citizen	4m	0.22

2.2.2 **Feature extraction techniques**

The version 1 of the stress detection module worked extracting 18 acoustic features from speech using the software Praat. Experiments replicating MuSE Challenge feature extraction techniques have been carried out to see whether a larger set of features improved the result of the classifier.

In the MuSE Challenge (Stappen et al., 2021), several feature sets from several modalities (i.e., heart rate, face movements, etc.) are used to predict stress annotations. The feature set that outperforms the task of stress detection is precisely the acoustic feature set.

Specifically, they use the open-source software OpenSMILE[3] (Eyben et al., 2010) with a predefined set of 88 acoustic features, known as the eGeMAPs feature set (Eyben et al., 2016). Prediction of continuous values of stress at 500 ms windows is performed and an overall accuracy of 0.44 on the MuSE challenge's test set is reported.

We tested the OpenSMILE API both locally and calling the available Python library in two different scenarios: i) using the predefined configuration extracting 88 acoustic features from the whole audio file and ii) using the predefined configuration extracting 10 Low-Level-Descriptor (LLD) features at 25 ms windows with 10 ms steps. Two sets of acoustic features were derived and post-processed for classification experiments to match the gold annotations described in the previous section.

2.2.3 Classification experiments

Version 1 of the stress detection module used the Weka ML Toolkit [REF] with a neural perceptron as classifier. We have conducted classification experiments using the same Weka Toolkit and a variety of classifiers to compare their performance. 10 cross-fold validation was used as classification technique computing the correlation coefficient and mean absolute error as evaluation metrics.

The top three best classifiers on 10-fold cross validation are:

- Gaussian Processes with a correlation coefficient 0.74 and mean absolute error 0.09
- Bagging with correlation coefficient 0.70 and mean absolute error 0.09
- Additive regression with correlation coefficient 0.60 and mean absolute error 0.10

Multilayer perceptron which is the current classifier in version 1 of the stress modules achieved a correlation coefficient of 0.55 and mean absolute error of 0.13 on the same dataset and classification task. Any of the two best classifiers seem adequate for the per audio classification.

Results on the LLD dataset (annotated with continuous values) best results were achieved by the smoReg classifier, attaining a SoTA performance in this task (continuous prediction of values) of 0.38 correlation coefficient.

2.3 Development of stress detection module version 2

2.3.1 Training an in-domain stress detection model

In order to properly train a model for deployment, we carried out a final experiment splitting the data into training and testing sets. Such splitting is a process which might be bound to bias due to the relatively small amount of training material and characteristics of the recordings.

- There are only 262 audio files (accounting for speaker turns in a total of 20 citizen-operator dialogues)

[3] https://audeering.github.io/opensmile/index.html

- Some speakers perform the roles of both citizen and operator
- There is only one female voice (acting as operator in 2 dialogues and as citizen in 1)
- Annotations are normally distributed with a skewness towards the left, which means there are considerably more annotations in quartile 1 (that is stress level around 0.3 in a scale from 0 to 1) see Figure 7.

Results from this analysis implies that the normal distribution of stress should also be considered around level 0.3 in the output of the model.

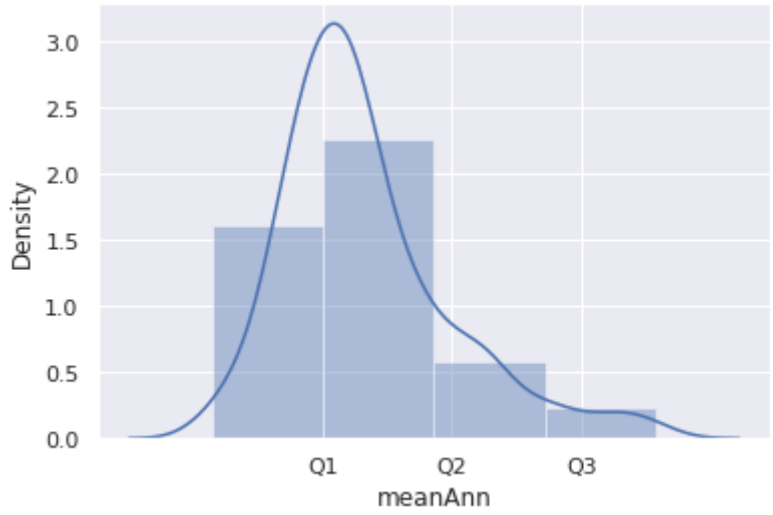

Figure 8: Distribution of mean stress annotations per audio file

A random split trying to select a representative sample of files was done considering the above mentioned characteristics. Results show that the best classifier in this case is the bagging classifier that obtains a correlation coefficient of 0.88 on the test set.

2.3.2 **Comparison of stress modules versions 1 and 2**

A comparison between version 1 and version 2 of the stress detection module was carried out using the test set described in the previous sections including 25 audio samples. Table 2 reports the results of this evaluation including the distance from each version of the module to the gold annotation.

Table 2: Comparison of version 1 and version 2 of the stress detection module

File name	Version 1	Version 2	Gold	Dist_v1_gold	Dist_v2_gold
rec01_009_op1m	0.49	0.36	0.29	0.20	0.06
rec02_010_ci3m	0.35	0.29	0.23	0.12	0.06
rec03_004_ci2m	0.01	0.55	0.73	-0.71	-0.18
rec03_008_ci2m	0.32	0.74	0.85	-0.53	-0.10

rec04_004_ci3m	0.08	0.46	0.37	-0.29	0.09
rec05_002_ci4m	0.93	0.28	0.20	0.74	0.08
rec06_001_op1m	0.19	0.29	0.43	-0.23	-0.14
rec07_004_ci1f	0.13	0.40	0.56	-0.43	-0.16
rec08_011_op1m	0.19	0.28	0.26	-0.08	0.02
rec09_010_ci1m	0.50	0.37	0.59	-0.09	-0.21
rec10_001_ci1m	0.18	0.39	0.38	-0.19	0.01
rec11_006_ci1m	0.70	0.50	0.70	0.01	-0.20
rec11_008_ci1m	0.82	0.55	0.70	0.12	-0.16
rec11_017_op3m	0.19	0.28	0.33	-0.15	-0.05
rec11_027_op3m	0.29	0.31	0.32	-0.03	-0.01
rec13_001_op2m	0.67	0.58	0.59	0.09	-0.01
rec14_001_op1m	0.10	0.29	0.23	-0.14	0.06
rec14_013_op1m	0.21	0.27	0.23	-0.02	0.04
rec15_009_op1m	0.19	0.31	0.19	0.00	0.12
rec16_007_op1m	0.41	0.28	0.29	0.12	-0.01
rec17_010_ci3m	0.09	0.18	0.10	0.00	0.09
rec17_019_op1m	0.37	0.27	0.21	0.16	0.05
rec18_010_ci1f	0.61	0.38	0.45	0.15	-0.08
rec19_011_op1f	0.41	0.30	0.45	-0.04	-0.16
rec20_004_ci4m	0.47	0.27	0.22	0.25	0.05

Results show that 18 out of 25 scores (72% of samples) are closer to the gold annotation from version 2 (numbers on the table are highlighted). The overall improvement computing the mean squared distance (to avoid negative numbers) is 0.01 for version 2 compared to 0.07 for version 1 in this test set, which implies an improvement of 0.06 points over the baseline reducing considerably the distance to the gold annotations.

2.3.3 Integration and deployment of version 2

The new algorithm is implemented in Java[4], using the Weka[5] framework for underlying the machine learning algorithms, and then packaged and deployed as a Docker[6] container, running on Docker Swarm[7]. It is accessible as a REST-like[8] web service, with Swagger[9]-based

[4] https://www.java.com/

[5] https://www.cs.waikato.ac.nz/ml/weka/

[6] https://www.docker.com/

[7] https://docs.docker.com/engine/swarm/

[8] https://en.wikipedia.org/wiki/Representational_state_transfer

[9] https://swagger.io/

documentation and an interactive web-based test interface available at https://xr4drama.upf.edu/xr4drama-services/.

The service currently exposes three endpoints:

- `/api/stress/estimate` is the fundamental service that takes an audio file as an input and returns the estimated stress level (within a JSON structure) to the caller
- `/api/stress/estimate_to_kb` is identical to the former, but additionally sends its output to the stress fusion engine to be combined with other sources, such as the sensor-based predictions
- `/api/stress/estimate_from_url` is again identical to the previous one, but instead of receiving the audio directly within the call to the service it is given a URL pointing to the audio file, which is then retrieved and analysed.

From the viewpoint of the overall system integration, the flow is as follows (illustrated in Figure 9 below):

- First responders in the field wear a "smart" shirt with physiological sensors which communicates with the AR app used by First Responders (which also provides situational awareness features, etc.).
- This same app is used to gather audio recordings from FRs, whenever they use the app to record a voice note to be attached to a task, etc. These audio recordings are stored in the XR4Drama backend where they are accessible through a URL pointing to the audio file.
- Whenever audio is recorded by a FR, the backend calls the `/api/stress/estimate_from_url` endpoint, which is a convenience wrapper around the core (audio-based) stress estimation functionality.
- The stress estimation service retrieves the audio file from the backend storage and analyses it, resulting in a numeric prediction value.
- The analysis result is sent to the stress fusion module which combines this estimation with the sensor-based estimations also received from the AR app.
- The combined result is made available to end users for decision support.

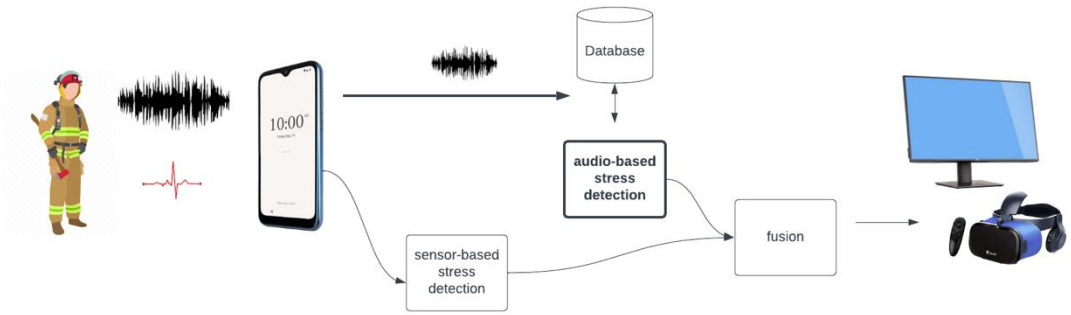

Figure 9: Stress analysis flow

3 FUSION MODULE

In the context of the xR4DRAMA project, the stress detection requirement is carried out using both sensor-based and audio-based methods. Since two different methods have been developed for the stress detection, the fusion of the results of the different methods is necessary. Fusion of different modalities can take advantage of the unique attributes of each modality and combine them to a unified outcome, improving the overall performance of the stress detection. The following sections describe the main methods for the fusion module including the different models tested, the results of the evaluation and the integration process of the fusion module.

3.1 Evaluation process

The fusion process of sensor-based and audio-based stress detection can be depicted in Figure 10. The different detections of audio-based and sensor-based results are fed into a model in order to perform a decision level fusion in the context of stress detection.

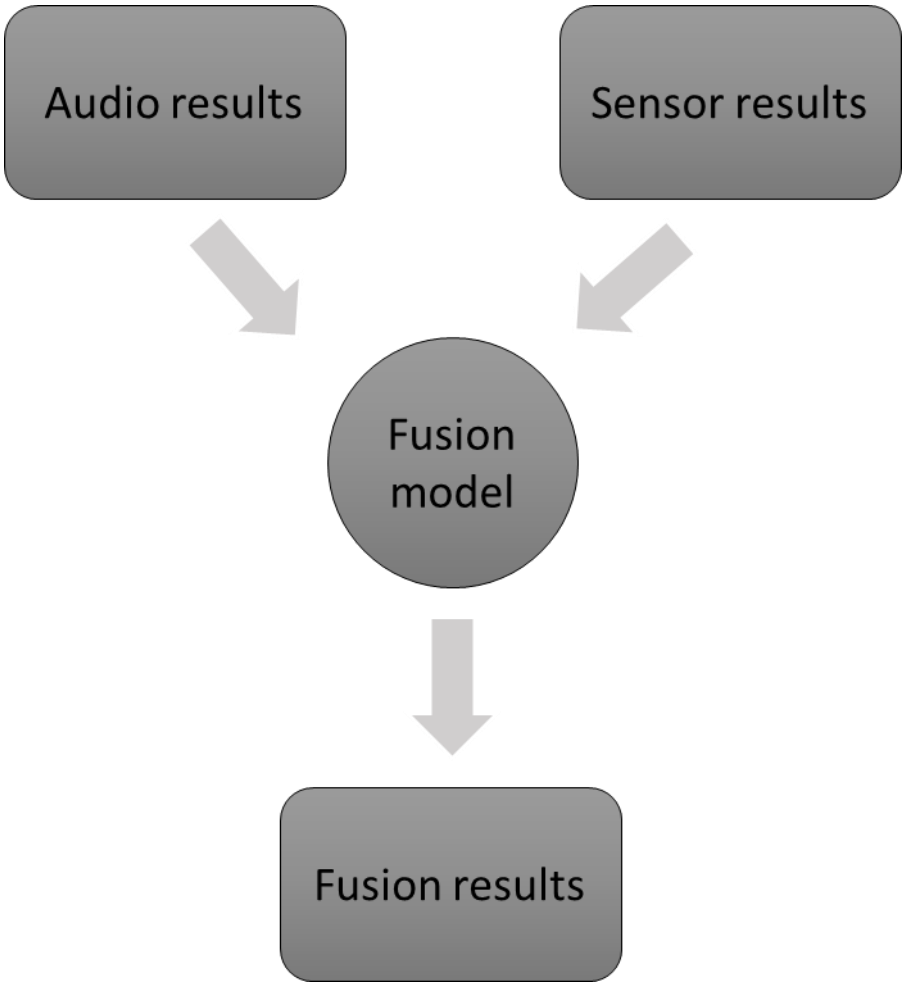

Figure 10: Fusion process

For the decision level fusion of audio-based and sensor-based stress detections five different classifiers were tested. Those are Support Vector Machines (SVM) with linear and radial basis function, k Nearest Neighbours (kNN), Random Forest (RF) and eXtreme Gradient Boosting (XGB) decision trees. Apart from the different regressors, also a weighted average technique was tested. The weights were optimized using a Genetic Algorithm (GA) optimization method using the mean squared error (mse) as the fitness function.

We evaluated the different methods using the mse and using a 10-fold cross validation method. The stress levels were normalized into the 0-1 range. The results of the different methods tested are presented in Table 3.

Table 3: Results of the different fusion methods tested

Regressor	**Fusion results**
SVM - Radial	**0.0062**
SVM – Linear	0.0085
kNN	0.0077
RF	0.0083
XGB	0.0098
Weighted average	0.0192

From Table 3 it can be seen that the SVM with radial basis function achieves the best performance out of all different fusion methods tested with an mse score of 0.0062. Since audio-based results achieve an mse score of 0.01 and sensor-based results achieve an mse score of 0.0567, it is clear that the fusion of sensor and audio results improves the overall performance of the stress detection module.

3.2 Integration of fusion module

The trained algorithm has been implemented in Python[10], using the sklearn package[11] for the machine-learning algorithm, and is deployed using a virtual environment. The stress detection results from the fusion module are accessible through the swagger found in **https://xr4drama.iti.gr:5200/**, where there are endpoints to retrieve results based on the user id, the project id and the timestamp, or a combination of the previous.

[10] **https://www.python.org/**

[11] **https://scikit-learn.org/stable/**

4 CONCLUSIONS

In this deliverable we have presented the improved methods for audio-based stress detection, as well as the fusion techniques to combine these with sensor-based estimations.

Compared to the earlier iteration described in D3.4, both of these aspects have improved significantly. Additionally, the integration in the project workflow has also been completed, going from an initial prototype implementation for testing to having the stress detection integrated with the applications for First Responders, able to obtain data from FRs in the field, analysing the data and making the resulting predictions available for decision making.

5 REFERENCES

Aigrain, J., Spodenkiewicz, M., Dubuisson, S., Detyniecki, M., Cohen, D., & Chetouani, M. (2016). *"Multimodal stress detection from multiple assessments"*, IEEE Transactions on Affective Computing, Vol.*9* (4), p. 491-506.
https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=7752842

Baur, T., Heimerl, A., Lingenfelser, F., Wagner, J., Valstar, M., Schuller, B., André, E. 2020. *"eXplainable Cooperative Machine Learning with NOVA"*, Künstliche Intelligenz , p. 109-115.
https://doi.org/10.1007/s13218-020-00632-3

Bobade P., Vani, M. (2020).*"Stress Detection with Machine Learning and Deep Learning using Multimodal Physiological Data"*, Proceedings of Second International Conference on Inventive Research in Computing Applications (ICIRCA-2020), p. 51-17.
https://ieeexplore.ieee.org/document/9183244

Eyben, F., Wöllmer, M., Schuller B. 2010. *"Opensmile: the munich versatile and fast open-source audio feature extractor"*, Proceedings of the 18th ACM international conference on Multimedia, p. 1459–1462. **https://doi.org/10.1145/1873951.1874246**

Eyben, F., Scherer, K., Schuller, B., Sundberg, J., Andre, E., Busso, C., Devillers, L., Epps, J., Laukka, P., Narayanan, S., Truong, K. 2016. *"The Geneva Minimalistic Acoustic Parameter Set (GeMAPS) for Voice Research and Affective Computing"*, IEEE Transactions on Affective Computing, vol. 7 (2), p. 190-202.

Giakoumis, D., Drosou, A., Cipresso, P., Tzovaras, D., Hassapis, G., Gaggioli, A., Riva, G. (2012). *"Using activity-related behavioural features towards more effective automatic stress detection"*, Plos One, Vol. 7 (9), p. e43571.
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0043571

Indikawati, F. I., Winiarti, S. (2020). "*Stress detection from multimodal wearable sensor data"*, *I*OP Conference Series: Materials Science and Engineering, Vol. 771 (1), p. 012028.
https://iopscience.iop.org/article/10.1088/1757-899X/771/1/012028

Schmidt P., Reiss A., Duerichen R., Marberger C., Laerhoven K. 2018. "*Introducing WESAD, a Multimodal Dataset for Wearable"*, Proceedings of the 20th ACM International Conference on Multimodal Interaction (ICMI'18). **https://doi.org/10.1145/3242969.3242985**

Siirtola, P. (2019). *"Continuous stress detection using the sensors of commercial smartwatch",* Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, p. 1198-1201.
https://dl.acm.org/doi/pdf/10.1145/3341162.3344831

Stappen, L., Baird, A., Christ, L., Schumann, L., Sertolli, B., Meßner, E., Cambria, E., Zhao, G,. Schuller, B. 2021. *"The MuSe 2021 Multimodal Sentiment Analysis Challenge: Sentiment, Emotion, Physiological-Emotion, and Stress"*, Proceedings of the 2nd on Multimodal Sentiment Analysis Challenge, p. 5–14.
https://dl.acm.org/doi/10.1145/3475957.3484450

Walambe, R., Nayak, P., Bhardwaj, A., Kotecha, K. (2021). *"Employing Multimodal Machine Learning for Stress Detection", Journal of Healthcare Engineering,* Vol. 2021. **https://www.hindawi.com/journals/jhe/2021/9356452/#materials-and-methods**