

# **The Realtime Assessment of Mental Workload by Means of Multiple Bio-Signals**

## **Masterthesis Report**

Methodology and Statistics for the Behavioral, Biomedical and  
Social Sciences

Utrecht University

Bart-Jan Boverhof, 6000142

## **Thesis Supervisor**

Prof.dr.ir. B.P. Veldkamp

## **Date**

January 7, 2021

# 1 Introduction

The topic of mental workload is a widely studied phenomenon across a variety of different fields, amongst others the field of ergonomics (Young, Brookhuis, Wickens, & Hancock, 2015), human factors (Pretorius & Cilliers, 2007) and cognitive neurosciences (Shuggi, Oh, Shewokis, & Gentili, 2017). A commonly utilized definition of mental workload, hereafter referred to as simply "workload", is the demand placed upon one whilst one carries out a particular task. As rightfully pointed out by de Waard and te Groningen (1996), the aforementioned definition is lacking, for it defines workload solely as a phenomenon external to the individual. Workload, however, requires to be recognized as a person-specific construct, for the amount of perceived workload ushered by a given task may vary substantially across individuals. (de Waard & te Groningen, 1996).

A commonly employed method for the assessment of workload is the well established NASA-Task Load Index questionnaire, hereafter referred to as "NASA-TLX". This method inquires into the amount of perceived workload, embodying six different dimensions (Hart, 2006). Due to practical considerations such an assessment is usually conducted post-experiment, which can in certain situations be deemed suboptimal. Consider for example an experiment in which it is aimed to assess the perceived workload of a pilot in flight. An insightful approach towards such an experiment would be to measure the degree of perceived workload during different phases of the flight, such that can be determined which specific manoeuvres tend to cause an increase in perceived workload. However, a questionnaire-based assessment such as the NASA-TLX can only be administered after the flight is concluded. In such a situation, utilizing a post-experiment assessment well after the action took place is prone to generate biases. One example of such a bias is the observer-bias, dictating that participants of an experiment tend to overexaggerate the treatment effect, i.e. the amount of perceived workload in our case, when having to report it post-experiment (Mahtani, Spencer, Brassey, & Heneghan, 2018).

An alternative approach to the assessment of workload is to collect physiological bio-signals during the experiment, and utilize these inputs to predict the degree of experienced workload. A considerable advantage of such an approach is that it renders possible to cater the method towards the individual, hereby acknowledging the between-personal perceptual differences inherent to workload, the importance of which was discussed in the first paragraph of this paper and stressed by de Waard and te Groningen (1996). But

taking a physiological bio-signal-approach poses more advantages: e.g. one may utilize multiple complementary bio-signals, each stemming from a different physiological source simultaneously (Ramachandram & Taylor, 2017). Doing so poses the theoretical potential of yielding a rich and multifaceted assessment of a mental construct such as workload. In addition, with such an approach one may select a variety of the most suitable physiological bio-signals, hereafter referred to as "modalities", and combine these information streams in order to yield a multifaceted prediction.

With the current research endeavor we explore the feasibility of an approach to workload assessment by means of multiple modalities simultaneously, hereafter referred to as "multi-modular approach". We do so by combining data-streams stemming from three different modalities, being electroencephalography, photoplethysmography and galvanic skin response. Electroencephalography, hereafter referred to as "EEG", is often utilized in the assessment of workload (Craik, He, & Contreras-Vidal, 2019) (Berka et al., 2005) and found to be amongst the most adequately performing techniques within the field (Hogervorst, Brouwer, & Van Erp, 2014). Galvanic skin response, hereafter referred to as "GSR", equally so is a widely adopted approach towards workload assessment (Nourbakhsh, Wang, Chen, & Calvo, 2012) (Zhou, Jung, & Chen, 2015). Lastly, heart-rate is a widely recognized indicator of workload, mostly obtained through photoplethysmography, hereafter referred to as "PPG" (Zhang et al., 2018) (Jimenez-Molina, Retamal, & Lira, 2018).

A deep learning approach towards modeling is adopted. A range of four different deep neural networks, hereafter referred to as "DNN's", are constructed and contrasted in performance: three of which are single-modular DNN's, i.e. networks each utilizing data from only a single modality. Thus, one DNN solely utilizes the EEG modality, one solely utilizes GSR and one solely utilizes PPG. The fourth network is a multi-modular network combining all three modalities into one. With this research we firstly sought to explore which modalities constitute the most adequate predictors of workload in an experimental setting, and secondly whether a multi-modular combination is preferable over the three simpler (but much less computationally demanding) single-modular DNN's. This assessment will be propelled by contrasting network performance, however when drawing conclusions, considerations regarding computational costs will be taken into consideration as well. All networks were constructed in Python, by utilizing the deep-learning toolbox PyTorch (Paszke et al., 2017).

## 2 Data & Methods

### 2.1 Data

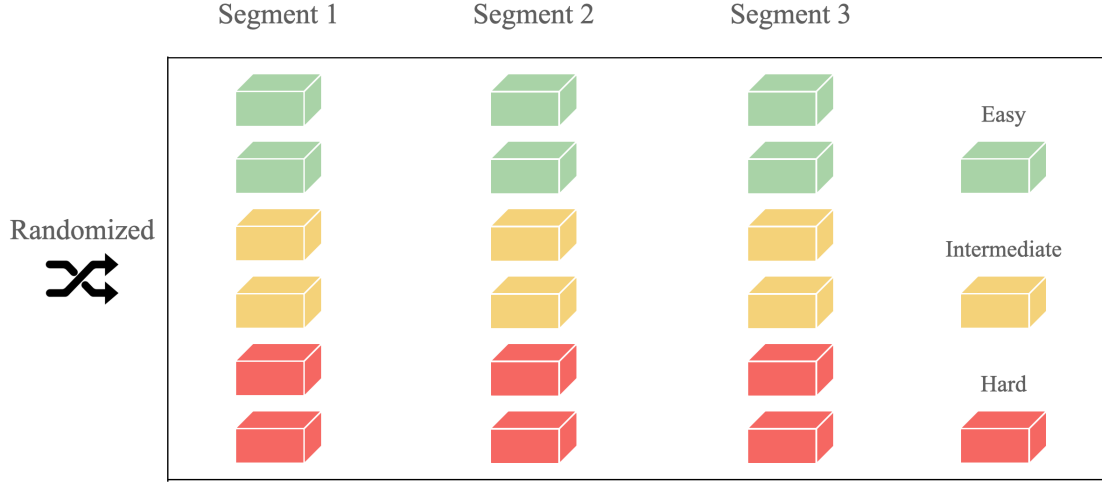
The subsequent section will provide an overview of the utilized data. Attention is placed upon the experimental setup, the description of the respondents, the utilized devices for data collection and the synchronization process.

#### 2.1.1 Experimental Setup

The experimental setting in which data collection befell is the open-source spaceship video-game Empty Epsilon, in which partakers carry out tasks on a virtual spaceship bridge (Daid & Nallath, 2016). This experiment was instituted by the Brain Computer Interfaces testbed, hosted by the University of Twente and carried out in cooperation with Thales group Hengelo. The experiment constituted three different segments, in each of which the respondent had to carry out tasks. These tasks were designed to evoke varying degrees of workload. Each segment consisted of six small sessions, lasting roughly five to ten minutes. These sessions varied in difficulty, including two easy, two intermediate and two hard sessions per segment. Within each segment, the order in which the sessions were presented was randomized. Between every three sessions, respondents were given a short two minute break. A schematic overview of the experimental structure is depicted as Figure 1.

After each of the 18 sessions, respondents were asked to fill out the NASA-TLX, resulting in 18 filled in questionnaires per respondent. Each questionnaire consists out of 6 questions, inquiring upon the degree to which the respondent experienced workload during the preceding session. The mean of a subset of four of these items have been utilized to label the data for training purposes. Two items were not incorporated into the scale, for they inquired into physical demand specifically, which given the experiment took place on a computer whilst sitting behind a desk was considered to be a redundant factor. Each item, and hence the newly constructed scale, ranges from 0-20, wherein 0 reflects low workload and 20 reflects high workload. In order to encourage numerical stability, label scores were normalized to reside in-between 0-1.

The first segment emulated a scenario in which hostile spaceships approached the respondent's spaceship. The respondent was required to quickly respond by defusing the

**Figure 1***Schematic Depiction of the Experimental Setup*

hostiles in order to survive. The increment in difficulty caused the process of defusing hostile spaceships to become more challenging, hereby aiming to cause an increase in workload. The second segment emulated a scenario in which the respondent had to navigate their spaceship through space, gathering as many way-points as possible. Obstacles around which the respondent had to navigate carefully, as well as hostile spaceships the respondent had to decimate, were introduced in the higher difficulty sessions. The third and final segment emulated a machine room, in which respondents had to control the power based on randomly generated requests. In the increased difficulty settings, variables that could overheat the spaceship were introduced, demanding the respondent to multi-task and aiming to increase workload as a consequence.

### 2.1.2 Participants

After having to omit 7 participants due to hardware failure, 27 respondents were included in the final analysis. 18 participants are female whereas 9 are male. The mean age and standard deviation are  $\mu = 26, \sigma = 10.31$  respectively. The participants are students recruited from the University of Twente, as well as several employees of Thales group Hengelo.

### 2.1.3 Devices and Synchronization

The Shimmer3 GSR+ sensor was used for both PPG and GSR measurements. This device is worn on the wrist. Signals are communicated wirelessly via Bluetooth. An ear-clip is utilized for measuring PPG. The Shimmer3 GSR+ automatically converts measured PPG signals to heart-rate. Skin conductivity, or GSR, is measured by two electrodes attached to the fingers (Shimmer-research, n.d.). EEG measurement was conducted with the Muse 2, which is a commercially offered multi-sensor headband that provides feedback on brain activity (Muse-incorporation, n.d.). The Muse 2 headband constitutes five sensors, each monitoring a different brainwave frequency.

The Shimmer3 GSR+ measures signals on a sampling rate of 256 Hz, whereas the Muse 2 measures at a sampling rate of 220 Hz. Given that multiple devices are utilized, data streams are required to be properly synchronized. This was accomplished by means of an application called Lab-Streaming Layer, hereafter referred to as "LSL", developed by Kothe, Medine, and Grivich (2018). The three data streams stemming from the two devices were all streamed to LSL during the experiment. LSL subsequently properly synchronizes all data streams in real-time, such that they are parallel, hence all referring to equivalent points in time.

## 2.2 Related Work

The following section provides an overview of preceding research regarding the assessment of workload by means of physiological bio-signals, with a focus on deep learning approaches. Attention is predominantly placed upon the most feasible network architectures, in addition to a range of model optimization techniques and to the data fusion strategy for the multi-modular DNN. The final part of this section will be dedicated to hyperparameter optimization.

### 2.2.1 Electroencephalography (EEG)

An overview of the complete literature on EEG applications with deep-learning was presented by Craik et al. (2019), who reported a total of 16 % of all publications to consider workload assessment specifically. The lion's share of these publications utilized either a deep belief networks or convolutional neural network, hereafter referred to as "ConvNet". One of these approaches is encouraged by the authors consequently (Craik et al., 2019).

Tabar and Halici (2016) proposed a hybrid of a ConvNet with a stacked auto-encoder network, hereafter referred to as "SAE network". Inputs were specified to feed into the convolutional-part with the objective of learning extracting features. The output of this part was subsequently specified to feed into the SAE part of the network, which encompasses a stack of dense-layers, designed specifically as an input layer, six hidden layers and an output layer. A classification accuracy of 90 % was acquired with this network (Tabar & Halici, 2016).

Research by Schirrmeister et al. (2017) contrasted the performance of several ConvNets against the widely utilized baseline method for EEG classification, filter bank common spatial pattern, hereafter referred to as "FBCSP". A deep ConvNet, a shallow ConvNet, a deep-shallow hybrid ConvNet and a residual ConvNet were contrasted with an FBCSP. Both the deep and shallow ConvNets were found to reach at least similar, and in some regards better classification results as compared with the FBCSP baseline approach. Altogether, a deep ConvNet with four convolutional-max-pooling blocks was found to perform best, exhibiting an accuracy of 92.4 % (Schirrmeister et al., 2017).

### 2.2.2 Galvanic Skin Response (GSR)

Sun, Hong, Li, and Ren (2019) explored the most suitable approach to the classification of several emotional states with GSR. Various models were explored, amongst others a support vector machine, a ConvNet and a long-short-term-memory, hereafter referred to as "LSTM", network. Additionally, the feasibility of a hybrid DNN, combining both the ConvNet and LSTM approaches, was explored. This aforementioned hybrid model was found to perform best, exhibiting an accuracy of 74% (Sun et al., 2019).

Dolmans, Poel, van 't Klooster, and Veldkamp (2020) took, amongst other approaches to workload prediction, a GSR approach, and designed a variant based on the previously delineated CovNet-LSTM structure. The performance of this model was contrasted with a network consisting solely of fully connected dense layers. Conform with findings by Sun et al. (2019), the hybrid model was found to perform best, displaying an absolute difference between predicted and true label of 0.197 (scaled on 0-1). The model architecture as utilized deployed two convolutional max-pooling blocks and two LSTM layers (Dolmans et al., 2020).

### 2.2.3 Photoplethysmography (PPG)

Research by Biswas et al. (2019) explored a deep learning approach towards PPG classification, with the objective to perform both bio-metric identification and obtain heart rate information. Exceptional results were realized with a DNN, attaining an average accuracy of 96 % (Biswas et al., 2019). This performance was realized with a ConvNet-LSTM hybrid, incorporating two convolutional max-pooling blocks followed by two LSTM layers.

### 2.2.4 Multi-modular Fusion Strategy

When conducting a multi-modular deep learning approach, information streams stemming from the different modalities are required to be combined, i.e. "fused", at a certain point in the network in order to ultimately result in a single prediction of workload. Fusion can be done conforming several strategies. Three strategies as proposed by Ramachandram and Taylor (2017) have been considered.

Early, or data-level, fusion constitutes an approach that fuses data sources before being fed into the network. Early fusing is usually proven to be quite challenging, residing in the fact that data streams stemming from different modalities often differ in their dimensionality and sampling rate. In addition, when taking an early fusion approach, the oversimplified assumption of conditional independence is made implicitly. This assumption is unrealistic in nature, for data stemming from different modalities are expected to be correlated in practice (Ramachandram & Taylor, 2017).

Late, or decision-level, fusion refers to the process of aggregating the decisions of multiple separate networks, each applied towards every modality separately. In case the data sources stemming from the various modalities are either correlated or ultimately differ in their dimensionality, late fusion is often a more feasible approach as opposed to early fusion (Ramachandram & Taylor, 2017).

Lastly, intermediate fusion is the most widely employed fusion strategy for multi-modular deep-learning problems. Data streams are usually fused by a concatenation layer, joining the outputs of the separately defined network parts of each modality. This results in a single joint deep neural network. Several "higher-order layers" are usually defined in between the concatenation layer and the ultimate classification. The depth of the fusion, i.e. the specified number of higher-order layers, can be chosen conform to the specific circumstances, posing intermediate fusion to be the most flexible and therefore



the most widely adopted fusion strategy (Ramachandram & Taylor, 2017).

Indeed, when consulting the literature, one is prone to conclude that intermediate fusion strategies are the most prevailing for multi-modular deep learning approaches. When taking an intermediate fusion approach, the higher-order part of the network needs to be designed and established, for which several previous research endeavors are considered. Rastgoo, Nakisa, Maire, Rakotonirainy, and Chandran (2019) utilized a multi-modular ConvNet approach, and fused the modalities by concatenation, followed with two LSTM layers and two dense layers. A simpler approach is adopted by Han, Kwak, Oh, and Lee (2020), who utilized an intermediate fusion approach solely consisting of several fully connected dense layers. Lastly, Dolmans et al. (2020) took a relatively deep intermediate fusion approach, consisting of two dense layers, two convolutional layers followed by another two dense layers.

### 2.2.5 Model Optimization Strategies

The technique of batch normalization was initialized by Ioffe and Szegedy (2015), and is widely utilized in DNN’s with the objective of stabilization. Especially ConvNets benefit from this technique. It is beneficial to incorporate a batch normalization layer subsequent to a convolutional layer, but before feeding into the activation function (Ioffe & Szegedy, 2015). All previously delineated networks architectures utilize batch normalization layers, in most cases specified subsequent to each convolutional layer (Dolmans et al., 2020) (Tabar & Halici, 2016) (Schirrmeister et al., 2017) (Biswas et al., 2019) (Sun et al., 2019).

Pooling layers are commonly employed in ConvNets, usually succeeding a convolutional layer with the purpose of reducing dimensionality. The objective of such layers are to down-sample features into a more compact space, hereby only retaining indispensable information. For a more extensive elaboration on pooling, please consult LeCun, Bengio, and Hinton (2015). All previously delineated networks architectures utilize max-pooling layers, usually specified subsequent to the activation layer (Dolmans et al., 2020) (Tabar & Halici, 2016) (Schirrmeister et al., 2017) (Biswas et al., 2019) (Sun et al., 2019).

### 2.2.6 Hyper Parameter Optimization

Hyper parameter optimization, hereafter referred to as "HPO", is a technique that can be utilized to optimize network hyper-parameters, such as learning rate and dropout

rate. In theory, one could optimize the entire DNN architecture, including the number of neurons/filters in (convolutional) layers, the number of layers in general, whether to use certain layers etc. There is, however, a strong relationship between the amount of parameters to be optimized and the computational resources required to do so, in where many parameters to optimize tends to inflate computational costs by a substantial amount. Substantial advancements in DNN performance have been attained by utilizing HPO, especially for ConvNets (Bergstra & Bengio, 2012). The Optuna toolbox provides a method for creating a parameter search space, from which values for the hyper-parameters can be sampled and optimization can be performed (Akiba, Sano, Yanase, Ohta, & Koyama, 2019).

## 2.3 Deep Neural Network Architectures & Training

Before elaborating on the chosen DNN architectures, we will firstly touch upon several universal truths for all networks. Given the person-specific character inherent to physiological data, networks for each respondent have been trained independently. Despite this, all network architectures across persons are similar, except for the specified hyperparameters: these were optimized for each of the four networks and for each respondents individually as well. As a consequence, four DNN’s are trained for each of the 27 respondents separately, resulting in a total of 108 trained networks, each one utilizing one of 108 sets of optimized hyperparameters. All networks were

### 2.3.1 Single-modular Network

All single-modular network architectures took inspiration from previously delineated research (Schirrmester et al., 2017) (Dolmans et al., 2020) (Sun et al., 2019) (Biswas et al., 2019). In contrast with most of these previous endeavors, we opted for a ConvNet-only approach, i.e. without LSTM-layers, for the objective of the current research is not of a time-series-forecasting-kind. An experimental scenario wherein the course of actions over the duration of the experiment always unfold in a similar manner are inclined to benefit from acknowledging this time-series-like nature, hence benefiting from LSTM. An example of such a scenario would be a pilot in flight: whilst workload arousing phenomena that occur during a flight could differ across takes, the general chronological structure of the scenario always adheres to a similar blueprint. I.e., starting with ascend, likely

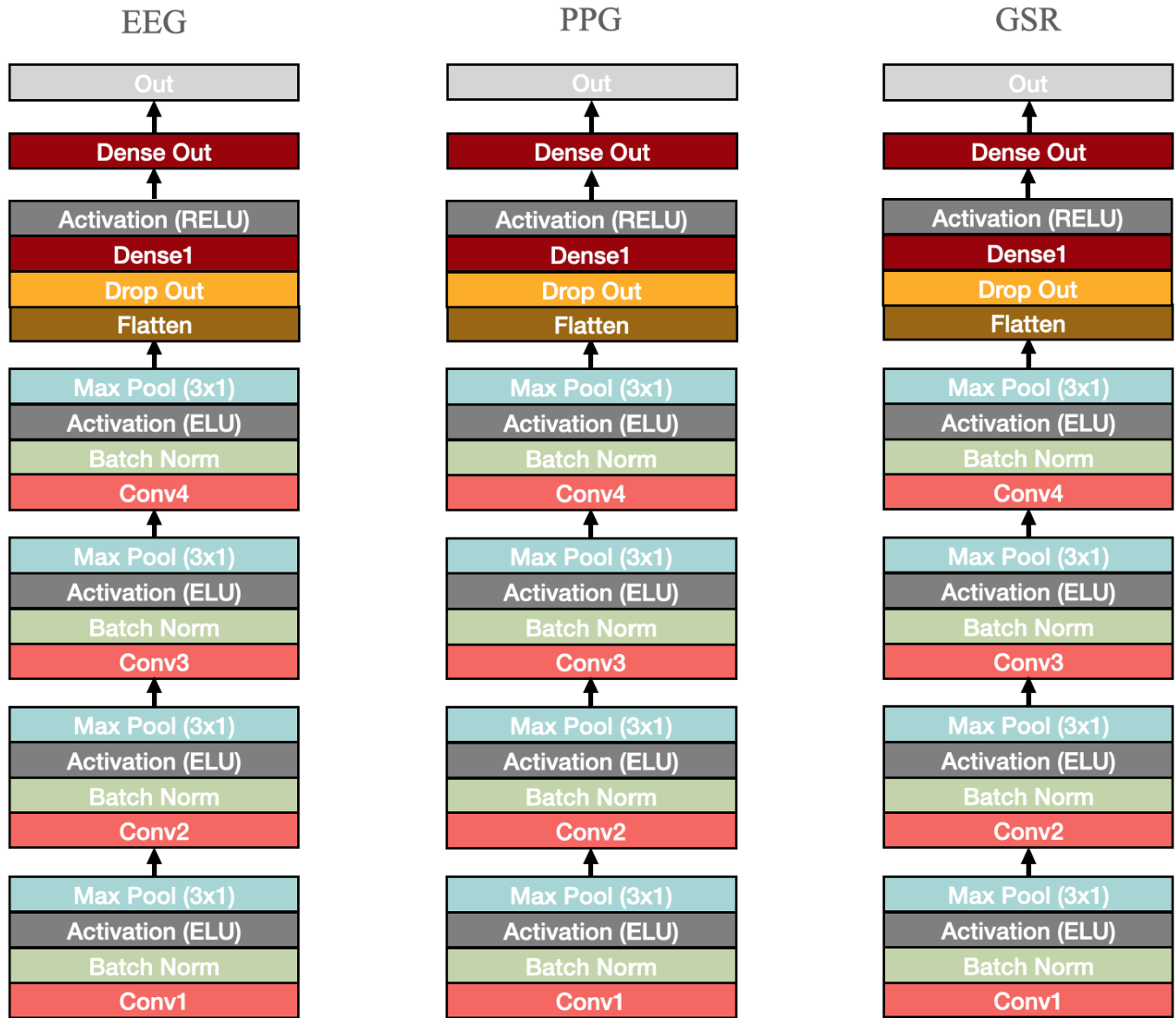
to arouse a peak in workload, and closing with touch down, equally so likely to arouse a peak in workload. Our experimental scenario doesn't adhere to such a set in stone unfolding, for the order in which the scenario's with variable difficulty were presented was entirely randomized. Hence, for there is no chronologically consistent development of the experimental scenario, we opted for a ConvNet approach without opting to learn from the development over time, thus without LSTM-layers.

The three single-modular DNN architectures are depicted alongside one another as Figure 2. Each of the single-modular networks can be distinguished by two separate part, being the convolutional part and the prediction part. The objective of the convolutional part is to extract features from the data. For each network, this part consists of four convolutional blocks, each compromising a convolutional layer, a batch normalization layer, the Exponential Linear Unit, hereafter referred to as "ELU", activation function, and a max-pooling layer respectively. The prediction part compromises a flatten-layer, with the objective of representing all input into one dimensional shape, followed by a dropout layer, a dense layer, a Rectified linear activation, hereafter referred to as "RELU", closed by another dense layer. An overview of the amount of utilized filters / neurons for each layer of each network is provided in Table 1.

Consistent with training, HPO was equally so done for each respondent individually, resulting in 27 optimized sets of parameters for the EEG networks, 27 for the PPG networks and 27 for the GSR networks. We optimized the learning rate, dropout rate and the amount of neurons for the first dense layer in 50 trials per network. For training, a total of 600 epochs have been made through the data for each network independently. All training and HPO of the single-modular networks has been done on a V100 GPU, offered by the GPU cloud service Google Collab. Total running time for the single-modular networks was about four days, the most of which was absorbed by HPO.

**Figure 2**

*The Three Single-modular Network Architectures*



*Note:* Each network constitutes a convolutional part of four convolutional blocks, and a classification part of two fully connected dense layers.

**Table 1***Network Sizes*

	EEG-Net/Part	GSR-Net/Part	PPG-Net/Part	Multi-Modular Head-Net
Single-Modular-Nets	Dense Out: 1	Dense Out: 1	Dense Out: 1	
	Dense 1: <i>hpo</i>	Dense 1: <i>hpo</i>	Dense 1: <i>hpo</i>	
	Drop out: <i>hpo</i>	Drop out: <i>hpo</i>	Drop out: <i>hpo</i>	
	Conv4: 200	Conv4: 128	Conv4: 128	
	Conv3: 100	Conv3: 64	Conv3: 64	
	Conv2 50	Conv2: 32	Conv2: 32	
	Conv1: 25	Conv1: 16	Conv1: 16	
Multi-Modular-Net	Dense 1: <i>input</i>	Dense 1: <i>input</i>	Dense 1: <i>input</i>	Dense Out: 1
	Conv4: 200	Conv4: 128	Conv4: 128	Dense 3: <i>hpo</i>
	Conv3: 100	Conv3: 64	Conv3: 64	Dense 2: <i>hpo</i>
	Conv2 50	Conv2: 32	Conv2: 32	Drop out: <i>hpo</i>
	Conv1: 25	Conv1: 16	Conv1: 16	

*Notes:* For all convolutional layers the depicted number reflects the amount of utilized filters, whereas for dense layers it reflects the amount of nodes. All depicted values are the amount of filters/nodes that a respective layer outputs. *input* refers to fully connected dense layers, wherein the number of outputting nodes equals the number of inputting nodes. *Hpo* implies that the value for the respective layer was optimized (see first paragraph of section 2.3)

### 2.3.2 Multi-modular Network

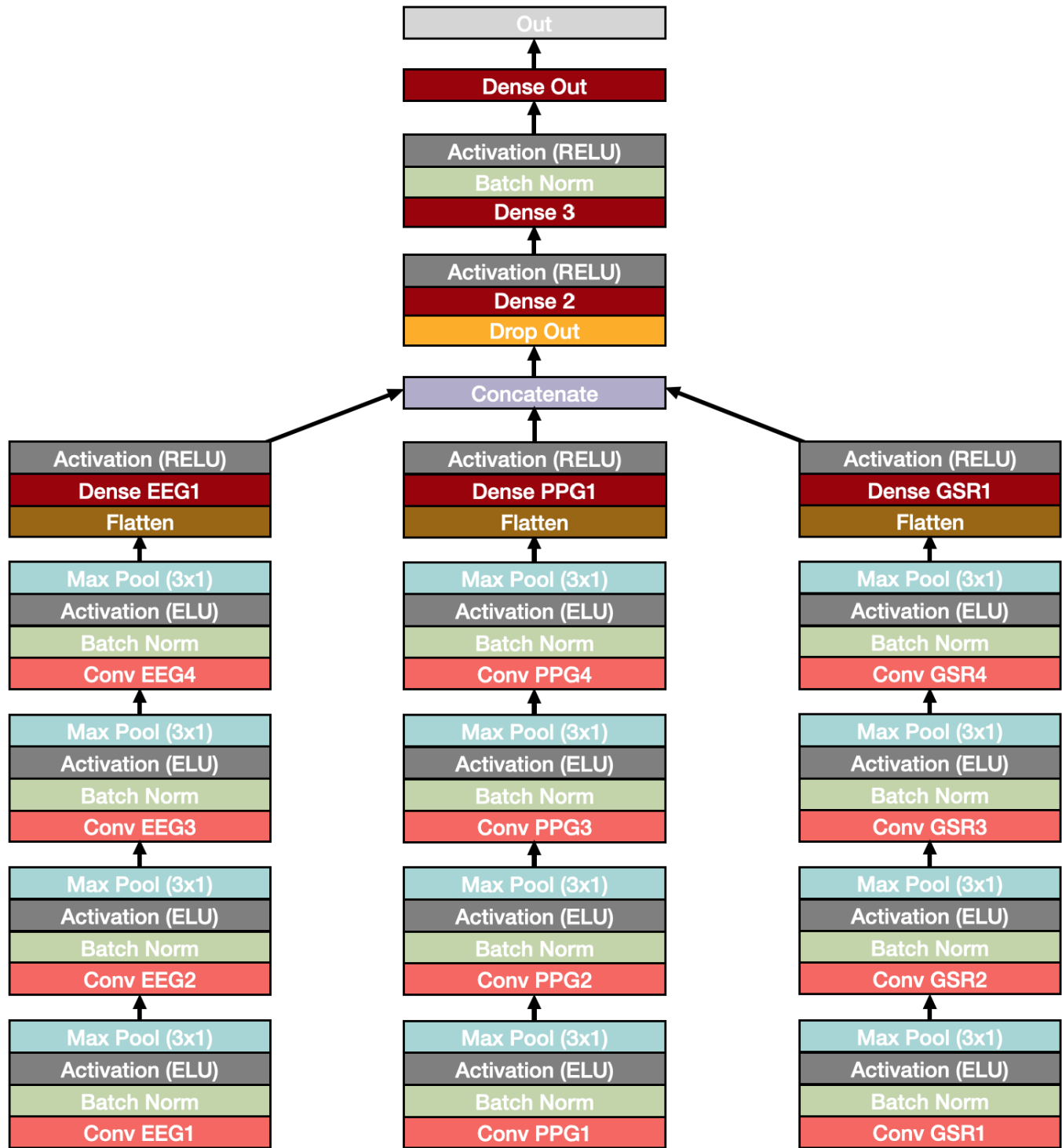
The network architecture utilized for the multi-modular approach was determined by combining the previously characterized single-modular networks with insights drawn from previous research, in particular (Han et al., 2020). A visual representation of the multi-modular network is depicted as Figure 3. An overview of the amount of utilized filters and neurons for each layer of the multi-modular network is provided in Table 1.

The network architecture for the single-modular parts of the network are separately defined entities of the multi-modular network, but are highly similar as compared with the previously delineated single-modular networks. An intermediate fusion strategy is adopted due to its highly flexible nature as compared with alternative fusion strategies. The output of all three distinct parts in the network were flattened, followed by a dense layer and RELU activation. Flattening was necessary such that all inputs were represented in one-dimensional space, after which concatenation was possible. The prediction part, also referred to as "Head-Network", consists of three dense layers, alternated with dropout, batch normalization and RELU activation.

Again, consistent with training, HPO was equally so done for each respondent individually, resulting in 27 optimized sets of parameters each of the 27 multi-modular networks. We optimized the learning rate, dropout rate and the amount of neurons for the first and second dense layers in 18 trials per network. For training, a total of 250 epochs have been made through the data for each network independently. All training and HPO has been done on a Cloud TPU v2, offered by the GPU cloud service Google Collab Pro. Total running time for the multi-modular networks was about 12 days, the most of which absorbed by HPO.

**Figure 3**

*The Multi-modular Network Architecture*



*Note:* The network constitutes a convolutional part of four convolutional blocks for each of distinct parts of the networks. All three distinct parts were flattened, after which concatenation was possible. The prediction part of the network constitutes several dense layers.

## 2.4 Model Evaluation

The performance of the networks has been assessed and contrasted by means of several performance metrics. When assessing performance in deep/machine-learning, it is of importance to realize that each performance metric may favor one approach over the other, simply and solely due to the mathematical nature in which both the metric and the model are defined (Gunawardana & Shani, 2009). For this reason, we consider a conglomerate of different metrics.

Mean Absolute Error, hereafter referred to as "MAE", of a DNN is the first utilized performance metric, defined as Equation 1,

$$\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i| \quad (1)$$

where  $n$  refers to the amount of windows utilized for testing,  $x_i$  to the predicted value for window  $i$  and  $y_i$  to the true label of window  $i$ . The MAE constitutes the most straightforward metric for the assessment in performance of a DNN, especially since the value can simply be interpreted as the average amount by which the prediction was off.

Root Mean Squared Error, hereafter referred to as RMSE, is the second utilized performance metric, defined as Equation 2,

$$\sqrt{\frac{1}{n} \sum_{i=1}^n |f(x_i) - y_i|^2} \quad (2)$$

where  $n$  refers to the amount of windows utilized for testing,  $x_i$  to the predicted value for window  $i$  and  $y_i$  to the true label of window  $i$ . The RMSE, which is strongly related to the MAE, differs in that it punishes big absolute differences more severely.

Finally, the Pearson Correlation Coefficient, hereafter referred to as simply "correlation", constitutes the third utilized performance metric defined as Equation 3,

$$\rho_{X,Y} = \frac{cov(X,Y)}{\sigma_X \sigma_Y} \quad (3)$$

where  $x$  refers to the predicted values for the test windows,  $y$  to the true labels of test windows and  $\sigma_X$  &  $\sigma_Y$  refer to the standard deviation of  $x$  and  $y$  respectively. Conceptually, correlation indicates the coherence between predictions and labels, hereby posing an alternative model performance metric.



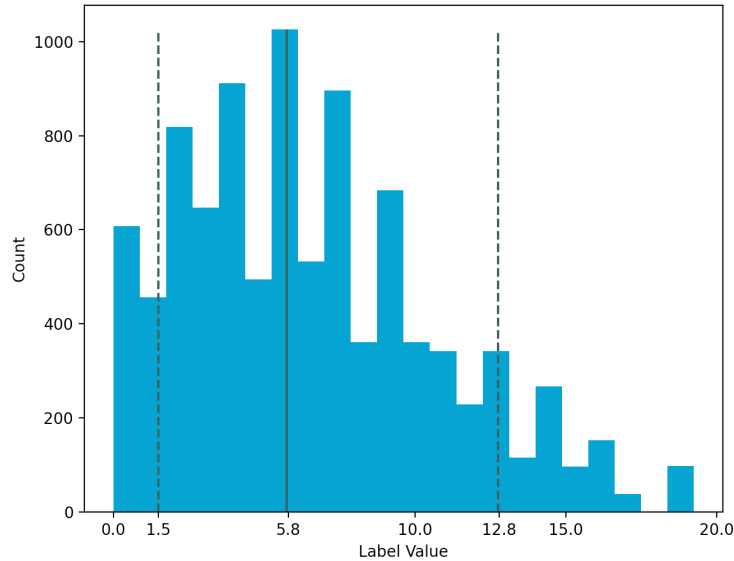
### 3 Results

Before illustrating results, a short elaboration on the testing procedure will be provided. Windows of eight seconds each were cut, resulting in a total of 9474 windows over all respondents, i.e. on average about 351 windows per respondent. For each network for each respondent individually, 80% of all windows have been used for training, 10% for validation within the training process and 10% for assessing performance of the trained DNN's. This results in an average of 35 windows for assessing DNN performance per respondent, i.e. a total of 969 windows in total. Partitioning windows into train-, validation- and testing-sets was done systematically by selecting every  $n$ -th window, such that windows over the entire duration of the experiment were represented equally in each partition.

The distribution of all 9474 window labels, aggregated for all respondents, is depicted as Figure 4. It becomes readily apparent that distribution of the window labels displays a noticeably right-skewed tendency, with a median of 5.75 and a 12.75 90% upper-bound, implying that higher workload windows are relatively uncommon within both train and test partitions.

**Figure 4**

*Distribution Window Labels*

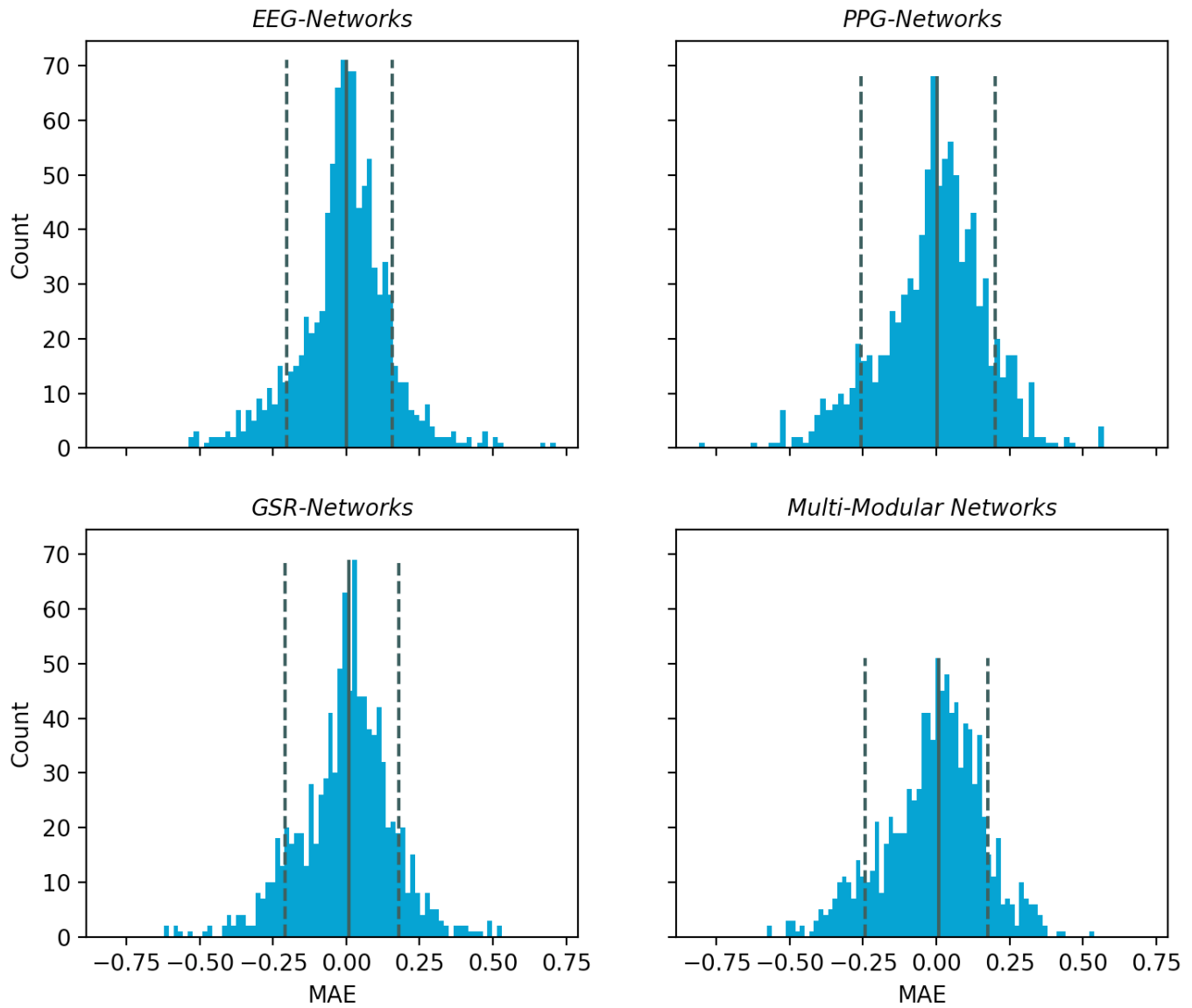


*Note:* The solid line represents the median, whereas the two dashed line represent the 10% and 90% quantiles respectively.

### 3.1 Single-modular Network Performance

**Figure 5**

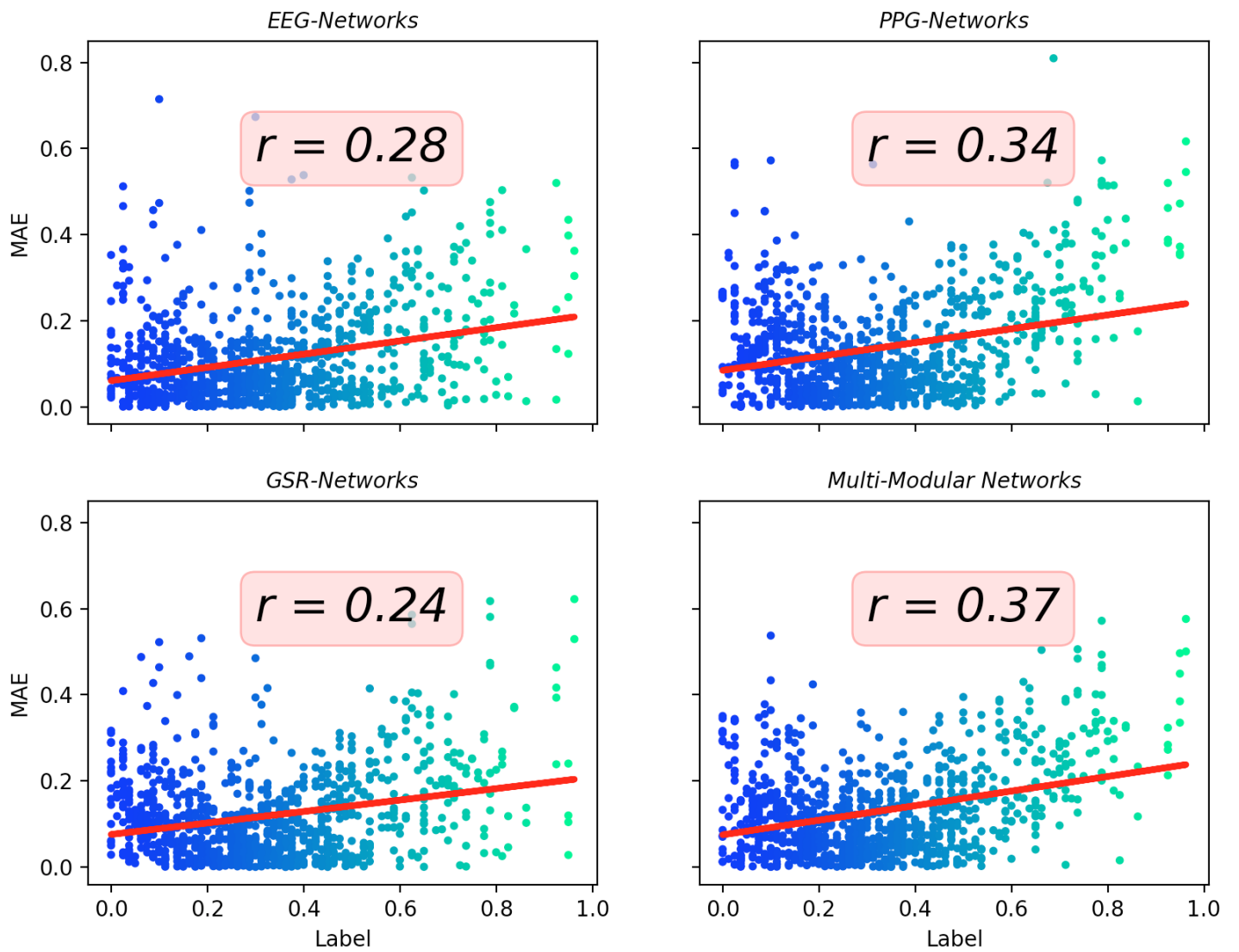
*Mean Absolute Error (MAE) per modality*



*Note:* The solid line represents the median, whereas the two dashed line represent the 10% and 90% quantiles respectively.

**Figure 6**

*Mean Absolute Error (MAE) against label size*



## 3.2 Multi-modular Network Performance

### Results EEG networks

- Correlation predictions and labels: 0.68788.
- Mean absolute error: 0.11029.
- Mean absolute error on the original scale: 2.20586.

### Results PPG networks

- Correlation predictions and labels: 0.52998.
- Mean absolute error: 0.13587.
- Mean absolute error on the original scale: 2.71747.

### Results GSR networks

- Correlation predictions and labels: 0.65339.
- Mean absolute error: 0.11883.
- Mean absolute error on the original scale: 2.37665.

### Results Multi-Modality networks

- Correlation predictions and labels: 0.60864.
- Mean absolute error: 0.12848.
- Mean absolute error on the original scale: 2.56964.

## 4 Limitations

- More HPO could have been done, to get even better performing models but we had limited computational power.
- This approach costs a lot of computational power to both train and do HPO. Especially since person-specific models are fitted. This upholds mostly for the multi-modal network.
- Due to the sheer size of the trained model and the fact that models are trained person specific, storage sizes of the models are huge. This could be impractical. This upholds mostly for the multi-modal network.
- Possibly different window sizes per modality could lead to better results. \* Insert some of the sources here that suggest different window sizes per modality\* . This was beyond the scope of this project.
- How generalizable is this to other situations for the same person (since things such as PPG are context dependent (like for example regarding how one ate)).
- Long runtimes, even when testing. This upholds only for the multi network. The reason is likely the sheer size of these networks, which for some persons are blown up due to device lagging.
- hardware failure?
- The fact that we use TLX as labels.

## References

- Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th acm sigkdd international conference on knowledge discovery & data mining* (pp. 2623–2631).
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, 13(1), 281–305.
- Berka, C., Levendowski, D. J., Ramsey, C. K., Davis, G., Lumicao, M. N., Stanney, K., ... Stibler, K. (2005). Evaluation of an eeg workload model in an aegis simulation environment. In *Biomonitoring for physiological and cognitive performance during military operations* (Vol. 5797, pp. 90–99).
- Biswas, D., Everson, L., Liu, M., Panwar, M., Verhoef, B.-E., Patki, S., ... others (2019). Cornet: Deep learning framework for ppg-based heart rate estimation and biometric identification in ambulant environment. *IEEE transactions on biomedical circuits and systems*, 13(2), 282–291.
- Craik, A., He, Y., & Contreras-Vidal, J. L. (2019). Deep learning for electroencephalogram (eeg) classification tasks: a review. *Journal of neural engineering*, 16(3), 031001.
- Daid, & Nallath. (2016). *Empty epsilon multiplayer spaceship bridge simulation*. URL: <https://github.com/daid/EmptyEpsilon>. GitHub.
- de Waard, D., & te Groningen, R. (1996). *The measurement of drivers' mental workload*. Groningen University, Traffic Research Center Netherlands.
- Dolmans, T., Poel, M., van 't Klooster, J.-W., & Veldkamp, B. (2020). Percieved mental workload detection using intermediate fusion multi-modal networks. manuscript submitted for publication.
- Fujita, D., & Suzuki, A. (2019). Evaluation of the possible use of ppg waveform features measured at low sampling rate. *IEEE Access*, 7, 58361–58367.
- Gunawardana, A., & Shani, G. (2009). A survey of accuracy evaluation metrics of recommendation tasks. *Journal of Machine Learning Research*, 10(12).
- Han, S.-Y., Kwak, N.-S., Oh, T., & Lee, S.-W. (2020). Classification of pilots' mental states using a multimodal deep learning network. *Biocybernetics and Biomedical Engineering*, 40(1), 324–336.
- Hart, S. G. (2006). Nasa-task load index (nasa-tlx); 20 years later. In *Proceedings of the*

- human factors and ergonomics society annual meeting* (Vol. 50, pp. 904–908).
- Hogervorst, M. A., Brouwer, A.-M., & Van Erp, J. B. (2014). Combining and comparing eeg, peripheral physiology and eye-related measures for the assessment of mental workload. *Frontiers in neuroscience*, 8, 322.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jimenez-Molina, A., Retamal, C., & Lira, H. (2018). Using psychophysiological sensors to assess mental workload during web browsing. *Sensors*, 18(2), 458.
- Kothe, C., Medine, D., & Grivich, M. (2018). Lab streaming layer (2014). *URL: <https://github.com/sccn/labstreaminglayer> (visited on 02/01/2019)*.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436–444.
- Mahtani, K., Spencer, E. A., Brassey, J., & Heneghan, C. (2018). Catalogue of bias: observer bias. *BMJ Evidence-Based Medicine*, 23(1), 23.
- Muse-incorporation. (n.d.). *Muse 2 brainwave activity headband*. Retrieved from: <https://choosemuse.com/muse-2/>.
- Nourbakhsh, N., Wang, Y., Chen, F., & Calvo, R. A. (2012). Using galvanic skin response for cognitive load measurement in arithmetic and reading tasks. In *Proceedings of the 24th australian computer-human interaction conference* (pp. 420–423).
- Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., ... Lerer, A. (2017). Automatic differentiation in pytorch.
- Pretorius, A., & Cilliers, P. (2007). Development of a mental workload index: A systems approach. *Ergonomics*, 50(9), 1503–1515.
- Ramachandram, D., & Taylor, G. W. (2017). Deep multimodal learning: A survey on recent advances and trends. *IEEE Signal Processing Magazine*, 34(6), 96–108.
- Rastgoo, M. N., Nakisa, B., Maire, F., Rakotonirainy, A., & Chandran, V. (2019). Automatic driver stress level classification using multimodal deep learning. *Expert Systems with Applications*, 138, 112793.
- Schirrmester, R. T., Springenberg, J. T., Fiederer, L. D. J., Glasstetter, M., Eggersperger, K., Tangermann, M., ... Ball, T. (2017). Deep learning with convolutional neural networks for eeg decoding and visualization. *Human brain mapping*, 38(11), 5391–5420.
- Shi, Y., Ruiz, N., Taib, R., Choi, E., & Chen, F. (2007). Galvanic skin response (gsr) as an index of cognitive load. In *Chi'07 extended abstracts on human factors in*



- computing systems* (pp. 2651–2656).
- Shimmer-research. (n.d.). *Shimmer3 gsr unit*. Retrieved from: <https://www.shimmersensing.com/products/shimmer3-wireless-gsr-sensor>.
- Shuggi, I. M., Oh, H., Shewokis, P. A., & Gentili, R. J. (2017). Mental workload and motor performance dynamics during practice of reaching movements under various levels of task difficulty. *Neuroscience*, 360, 166–179.
- Srinivasan, R., Tucker, D. M., & Murias, M. (1998). Estimating the spatial nyquist of the human eeg. *Behavior Research Methods, Instruments, & Computers*, 30(1), 8–19.
- Sun, X., Hong, T., Li, C., & Ren, F. (2019). Hybrid spatiotemporal models for sentiment classification via galvanic skin response. *Neurocomputing*, 358, 385–400.
- Tabar, Y. R., & Halici, U. (2016). A novel deep learning approach for classification of eeg motor imagery signals. *Journal of neural engineering*, 14(1), 016003.
- Weiergraeber, M., Papazoglou, A., Broich, K., & Mueller, R. (2016). Sampling rate, signal bandwidth and related pitfalls in eeg analysis. *Journal of neuroscience methods*, 268, 53–55.
- Young, M. S., Brookhuis, K. A., Wickens, C. D., & Hancock, P. A. (2015). State of science: mental workload in ergonomics. *Ergonomics*, 58(1), 1–17.
- Zhang, X., Lyu, Y., Hu, X., Hu, Z., Shi, Y., & Yin, H. (2018). Evaluating photoplethysmogram as a real-time cognitive load assessment during game playing. *International Journal of Human-Computer Interaction*, 34(8), 695–706.
- Zhou, J., Jung, J. Y., & Chen, F. (2015). Dynamic workload adjustments in human-machine systems based on gsr features. In *Ifip conference on human-computer interaction* (pp. 550–558).