Efficient Multi-modal Transformer Hyper-Parameters Optimization for Stress Detection

by

Merey Orazaly

Submitted to the Department of Data Science
in partial fulfillment of the requirements for the degree of

Master of Science in Data Science

at the

NAZARBAYEV UNIVERSITY

April 2025

Author .....
Department of Data Science
April 28, 2025

Certified by .....
Jurn Gyu Park, PhD
Assistant Professor
Thesis Supervisor

Accepted by .....
Yelyzaveta Arkhangelsky
Dean, School of Engineering and Digital Sciences

Efficient Multi-modal Transformer Hyper-Parameters Optimization for Stress Detection

by

Merey Orazaly

Submitted to the Department of Data Science
on April 28, 2025, in partial fulfillment of the
requirements for the degree of
Master of Science in Data Science

Abstract

Transformers demonstrate great potential for physiological signal analysis. However, their use of multi-class stress classifications is limited, especially in terms of deployment on constrained resource platforms. In this work, we present Efficient-HusFormer, a novel transformer-based architecture developed with hyper-parameter optimization (HPO) for multi-class stress classification using the WESAD [16] and CogLoad [5] dataset. The main contributions of this work are: (1) the design of a structured search space and local optimization strategy based on a priority assumption, targeting effective hyperparameter optimization of the number of layers (L), heads (H), dimension (d_m), and feed-forward network dimension (FFN); (2) a comprehensive ablation study evaluating the impact of architectural decisions across combinations of pairwise, triplet, and four-module configurations; (3) consistent performance improvements over the original HusFormer [24], with the best configuration achieving an accuracy of 88.41% and F1-score of 0.8815, corresponding to absolute gains of 9.73 percentage points in accuracy and 8.64 points in F1-score. The best-performing configuration is achieved with the $\{L + d_m\}$ modality combination on WESAD dataset, using single layer, 3 attention heads, a model dimension of 18, and FFN dimension of 120, resulting in a compact model with only ~30k parameters and 575MB of memory. These results imply HPO is an important part of developed transformer-based solutions for physiological computing. The full implementation of Efficient-HusFormer is publicly available on GitHub to support reproducibility and further research in physiological computing.

Thesis Supervisor: Jurn Gyu Park, PhD
Title: Assistant Professor

Acknowledgments

I would like to express my heartfelt gratitude to Professor Jurn Gyu Park for his expertise and guidance for the entire duration of my thesis. The Data Mining and Data-Driven Innovation courses he has taught greatly contributed to my academic development and furnished a necessary scope of thinking for this work. Special thanks to my family, my beloved one, and my friends for their ongoing support, encouragement, and understanding during my time as an academic. Their presence has been a continuous source of strength and motivation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The transformer-based architecture found a wide scope of research areas in the processing of physiological signals. It has the capacity to model long-range dependencies and extract features from complex data.

Since conventional recurrent neural networks (RNN) and convolutional neural networks (CNN) [10] process sequences in a local manner and are not capable of processing sequences in their entirety, these self-attention-based models are therefore well-suited for performing stress detection tasks.

This type of modeling yielded a considerable improvement for applications of health such as ECG classification, emotion recognition [23], and monitoring based on wearables [13]. The recent literature has dealt with transformer extensions like Time-Series Transformers (TST) and Physiological Signal Transformers (PST) [25] [22] demonstrating that these models can effectively manage multi-modal biosignals and enhance the accuracy of classification.

However, a big challenge is the optimization of transformer architectures toward real-time and resource-efficient implementation. Although the transformers were successful in physiological signal analysis, the challenge that exists for the design of efficient transformer models towards the multi-class stress detection is linked to their immense computational load and memory requirements coupled with trade-offs between accuracy and efficiency. Standard transformer architectures need a considerable amount of computational resources; thus, they are impractical for performing real-

time operations and resource-constrained platforms, for instance, mobile and wearable devices [11]. Moreover, for the multi-class stress detection, the model has to generalize over various states of physiological signals while having low latency and energy consumption [1].

Existing approaches primarily concentrate on performance, ignoring the practicalities of transformer deployment in real-world contexts with hardware limitations. Addressing those problems requires systematic research on transformer optimization, specifically focusing on layers (L), attention heads (H), dimension (d_m) and FFN dimension (FFN) in order to provide a practical and scalable way of stress detection.

In this paper, we propose a HusFormer-based [24] optimized model, a highly efficient transformer-based model applied to multi-class stress detection using the WESAD [16] and CogLoad [5] datasets. We introduce an optimization strategy that adjusts systematically the number of parameters to reach an optimal trade-off between accuracy and efficiency.

The paper specifically highlights the following contributions:

- Define a structured search space and apply local optimization based on a priority assumption to perform effective hyperparameter optimization.
- Conduct a comprehensive ablation study to evaluate the impact of architectural decisions from the combination of pairwise, triplet, and four-module configurations.
- Demonstrate an improvement in results compared to the original Husformer model, achieving at least a 4.74% increase in accuracy, and provide an open-source implementation of the proposed Efficient-Husformer model on GitHub[1].

The rest of the paper is organized as follows: Section II deals with the background, motivation and summary of related works. The optimization space, optimization plan and experimental setup are outlined in Section III. Section IV contains results and analysis. Section V presents discussion and future work, and the paper ends with Section VI, giving its conclusions.

[1] `https://github.com/Merey1508/Efficient-Husformer`

Chapter 2

Motivation and related work

2.1 Background

The architecture based on Husformer is defined by three major components, which are illustrated in Figure 2-1: feature extraction, cross-modal transformer, and self-attention transformer for final prediction.

Before Cross-modal: The input consists of six physiological signals — GSR (chest), BVP (wrist), EMG (chest), ECG (chest), RESP (chest), GSR (wrist). These signals are first preprocessed in order to extract salient features from the **raw signals** using **1D convolutional layers** (Conv1D). Each extracted feature vector is mapped to a d_m representation with added positional encoding to preserve their temporal de-

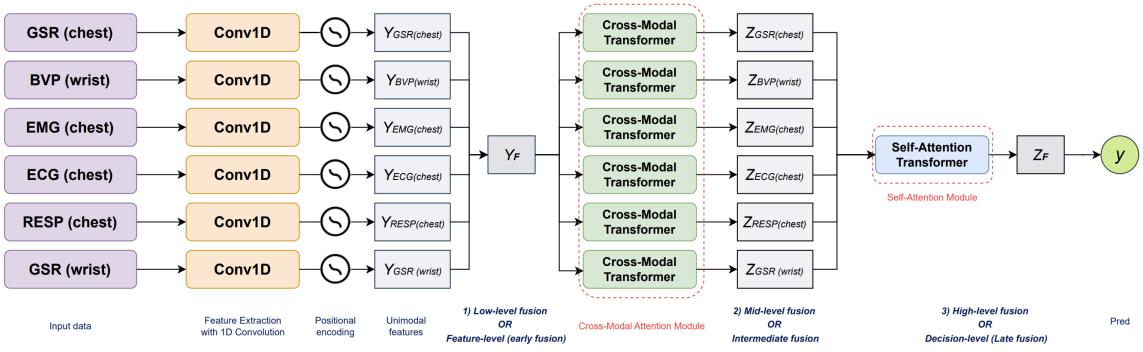

Figure 2-1: Husformer Architecture on WESAD dataset (Deployed from [24] and modified).

pendencies. After that, the unimodal features ($Y_{\text{GSR (chest)}}$, $Y_{\text{BVP (wrist)}}$, $Y_{\text{EMG (chest)}}$, $Y_{\text{ECG (chest)}}$, $Y_{\text{RESP (chest)}}$, and $Y_{\text{GSR (wrist)}}$) perform different levels of fusion:

- **Low-level fusion (Early Fusion)** — Where features are directly concatenated before passing to the cross-modal transformer.
- **Mid-level fusion (Intermediate Fusion)** — Inside the multiple cross-modal transformers, each transformer has L' layers with H' attention heads, which process independent adjacent representations of each modality and talk to each other without interfering with independent characteristics.
- **High-level fusion (Late Fusion)** — Decision-level fusion which aggregates after transformed modality-specific outputs ($Z_{\text{GSR (chest)}}$, $Z_{\text{BVP (wrist)}}$, $Z_{\text{EMG (chest)}}$, $Z_{\text{ECG (chest)}}$, $Z_{\text{RESP (chest)}}$, and $Z_{\text{GSR (wrist)}}$) for the final classification step.

Self-Attention Transformer: The fused representations are then inputted to a self-attention transformer with L layers, H attention heads, d_m dimensional input embeddings, FFN feed-forward network dimension size. The transformer leverages global dependencies among different modalities to refine learned features. The final latent representation Z_f introduces the multi-class stress classification step. The architecture allows HusFormer to effectively learn feature representation across various physiological signals, utilizing multi-head self-attention to express intermodal dependencies while assuring computational efficiency.

2.2 Motivation

The standard transformer architectures are often built with fixed hyperparameters such as **number of layers (L)**, **attention heads (H)**, **model dimension size (d_m)**, and **FFN dimension size (FFN)** which may lead to sub-optimal trade-offs between accuracy and efficiency [4]. In this study, we work toward optimizing the Husformer model by systematic tuning of these hyperparameters while still maintaining a competitive performance. We have defined the configuration of the transformer

model as:

$$\text{Husformer} = f(L, H, d_m, FFN) \tag{2.1}$$

Taking into account the multi-modal characterization of their physiologic signals, we further introduce a decoupled optimization strategy in which the intra-modal feature extraction cross-modal transformer (L, H) and the self-attention transformer (L', H') for modality fusion are separately optimized:

$$\text{Efficient-Husformer} = f(L, H) + f(L', H') + f(d_m) + f(FFN) \tag{2.2}$$

This parametric configuration thus allows independent classifications of both modality fusion and classification optimizations toward improved efficiency and interpretability. To effectively navigate this optimization space, we aim to address the following core questions:

- *How can a structured search space combined with local optimization strategies guided by priority assumptions improve the efficiency and effectiveness of hyperparameter tuning in Transformer-based models?*
- *What is the impact of different architectural configurations on the overall performance of Husformer model?*
- *what extent does the proposed Efficient-Husformer model outperform the original Husformer in terms of classification accuracy, and does its open-source implementation facilitate reproducibility?*

With these questions, we set forth a comprehensive optimization framework in balancing performance, resource efficiency, and feasibility for deployment. The study thus characterizes the influence of components of the transformer on tasks for stress detection and offers further guidance on efficient design for transformer-like models for wearable healthcare applications.

Motivating example. In the context of our optimization study, we evaluate the trade-offs between accuracy and computational resources in the Husformer model.

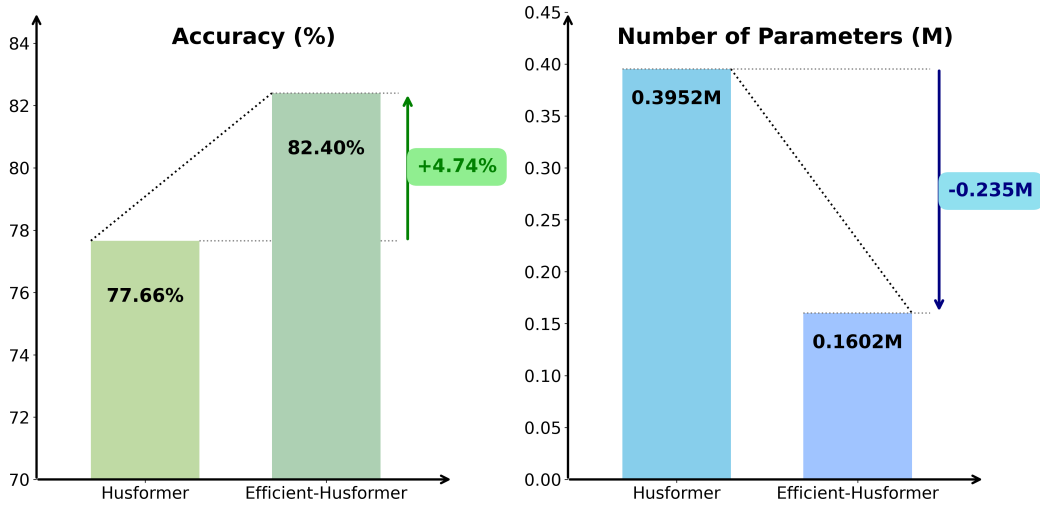

Figure 2-2: Motivating Example: Comparison of Original (L=5) and Efficient ($L = 3$) layers depth at the default heads ($H = 3$)

To assess the implications of our design decisions, we compare the original Husformer to proposed Efficient-Husformer model. Our findings in Figure 2-2 illustrates 4.74% increase in accuracy and a decrease of 0.235M in parameters, simultaneously maximizing both predictive performance and compute efficiency. These performance gains arise from the decoupled and reduction of both self-attention and cross-modal transformer layers with a default set of other hyperparameters. Using this opportunity, we can guarantee at least 82.40% accuracy, making the model a compelling choice for practical applications. This example gives strong empirical motivation in the rethink of a fixed transformer architecture for more flexible, task-adaptive, quantifiable configurations more ideally suited for multimodal tasks, such as stress detection.

2.3 Related Work

Multi-modal datasets

Multi-modal datasets have become a source of foundation in research on the detection of stress since they provide numerous sources through which complex natures of stress responses can be captured. These are normally combined by physiological signals, behavioral patterns, and contextual information that enhance the accuracy

Table 2.1: Summary of Related Works

Study	**Multi-class**	**Dataset**	**CV**	**Modalities**	**Auto. Extracted Features**	**Algorithms (Accuracy)**
Schmidt et al. (2018) [16]	✓	WESAD	✗	7	✗	DT, RF, AB, LDA, KNN (74.20%–87.74%)
Bobade et al. (2020) [3]	✓	WESAD	K-fold (k=5)	7	✗	DT, RF, AB, LDA, KNN, SVM, ANN (87.59%–95.21%)
Aqajari et al. (2021) [7]	✗	WESAD	LOOCV	1	✓	KNN, RF, SVM, NB (85%–91.60%)
Behinaein et al. (2021) [2]	✗	ECG	✗	1	✓	TF (80.4%)
Su et al. (2022) [19]	✓	WESAD	K-fold (k=10)	3	✗	RF, LR, SVM, FNN (84.62%)
Ziaratnia et al. (2023) [27]	✓	Custom	✗	1	✓	CCT-LSTM (83.2%)
Yao et al. (2021) [26]	✓	Custom	K-fold (k=5)	2	✓	MUSER Transformer (84.2%)
This work	✓	**WESAD**	**K-fold (k=10)**	**6**	✓	**Husformer (78.68% ± 2.05)** [24]

and reliability of the stress detection system.

Affective datasets: WESAD [16] is among the very few public multi-modal datasets for the detection of stress and affect. This dataset includes recordings acquired with wearable devices, specifically a chest-worn and a wrist-worn sensor to acquire physiological signals like ECG, galvanic skin response (GSR), respiration (RESP), and skin temperature. Furthermore, the dataset contains self-reported stress levels; thus, this database provides a wide view of responses to stress, both in controlled and real-life settings.

Another DEAP [12] dataset is often used as a benchmark for emotion recognition studies. It contains various physiological measurements including EEG, EMG, EOG, and GSR from 32 participants who watched 40 music video clips with the intention

of eliciting emotional responses. Subsequently, as part of the data-gathering process, participants provided self-reported emotional ratings on the Self-Assessment Manikin (SAM). Ratings for arousal, valence, liking, and dominance were all rated on a 9-point scale. Overall, DEAP is a prominent dataset within domains of affective computing and multimodal signal processing.

Cognitive load datasets: The CogLoad [5] dataset centers on cognitive load estimation. It has physiological data—specifically, EEG, GSR, and BVP—collected from 23 participants via a Microsoft Band wearable device. In the two-task scenarios, each participant completed a primary task drawn randomly from recognized psychophysiological assessment and a secondary task, usually responding to visual stimuli. Each session concluded with the participant reporting perceived mental demand on the NASA-TLX scale. Baseline (resting) data were also obtained. Furthermore, the dataset is available as an open-source resource and can be simply accessed directly from the Husformer GitHub repository. This dataset makes it possible to experiment with plug-and-play for cognitive workload modeling.

The MOCAS [9] dataset is created to facilitate cognitive load analysis in real-world surveillance situations. It includes multi-modal physiological data in the form of 5-channel EEG, EEG band powers (theta, alpha, beta, gamma), BVP, GSR, heart rate, and behavioral data that include Eye Aspect Ratio (EAR), and facial Action Units (AUs). Data were gathered from 21 participants completing monitoring tasks using Closed-Circuit Television (CCTV) to create the conditions for varying levels of cognitive load. Subjective ratings were obtained through the NASA-TLX and indexed, using weighted scores, into three classes of cognitive load (low, medium, high). The MOCAS dataset also contains a preprocessed dataset using a few features from neurokit2 thus increasing the usefulness of using in machine learning projects.

Feature extraction

Feature extraction is one of the most fundamental parts in raw data transformation to meaningful representations. There have been studies on the physiological signal time-domain and frequency-domain analyses [14] that range from the simple to com-

plex, deep learning-based feature learning. The preprocessing steps of filtering and normalizing and reducing the noise have also shown improved data quality and better performance from the models. Works, such as that of Aqajari et al. [7], have introduced automated feature extraction. Their study performed feature extraction using the pyEDA open-source library to extract features from EDA signals automatically, yielding an accuracy of 85%-91.6% across the various classifiers, namely KNN, RF, SVM, and Naive Bayes (NB), using 10-fold cross-validation, showing that automated extraction improves the overall results.

Recent advances in **automatic feature extraction** based on deep learning frameworks indeed made the process painless and relieved the researchers from burdensome feature engineering tasks [17]. For instance, CNN automatically extracts spatial features from raw sensor data. Complex patterns due to variations in heart rate or skin conductivity correlating with stress may be captured as the spatial features extracted by CNNs [15]. Similarly, RNNs and LSTM networks are used for temporal feature extraction to capture the trend and variation of physiological signals over time.

Attention mechanisms [21] have further enhanced the extraction of relevant features by assigning varying importance to different data segments. This approach helps in identifying critical moments of stress-related physiological changes. In addition, Behinaein et al. [2] presented only ECG data, using TFs for feature extraction, and reported an accuracy of 80.4% by employing the LOSO cross-validation technique.

Stress detection algorithms

Over the past years, a number of **machine learning (ML) and deep learning (DL) models** have been widely used in the domain of stress detection, considering different algorithms to analyze physiological and behavioral data.

A notable study by Schmidt et al. [16] discusses the use of multi-modal data for affect detection using multiple sensor modalities, such as ACC, BVP, EDA, TEMP, RESP, EMG, and ECG, along with a wide variety of classifiers such as Decision Tree (DT), Random Forest (RF), Ada Boost (AB), Linear Discriminant Analysis (LDA),

and K-Nearest Neighbors (KNN), with accuracy rates between 74.20% and 87.74% for LOSO.

Building on this, Bobade et al. [3] added Support Vector Machine (SVM) and Artificial Neural Network (ANN) models, hence yielding higher accuracy values of 87.59-95.21%, indicating the improvement in the current study because of the increased number of classifiers to handle the complexity and variability in the data between subjects.

In the study of Su et al. [19], four ML algorithms were developed for college students' models, by developing stress prediction models: RF, Logistic Regression (LR), SVM, and Feedforward Neural Network. The RF model resulted in the best predictive capability for stress levels, with the highest performance among all models, by reaching an accuracy of 84.62%, specificity of 96.35%, AUC of 82%, and F1 of 82%. Their work underlined the importance of specific modalities in reaching good performance while keeping the model complexity as low as possible.

Ziaratnia et al. [27] proposed a novel method on remote video-based stress estimation using a Convolutional Channel-wise Transformer combined with Long Short-Term Memory (CCT-LSTM). Their approach gave better performance about the spatial and temporal features extracted from facial cues, while yielding an accuracy of 83.2% and F1 score of 83.4%. Yao et al. [26] proposed MUSER, a transformer-based model whose performance in the task of detecting stress was facilitated by emotion recognition as an auxiliary task. Consequently, MUSER relied on the interdependence between the two variables (stress and emotion) and achieved 84.2% accuracy results in the Multimodal Stressed Emotion-MuSE dataset, anindication of the multi-task learning benefit in performing affective computing.

These studies further point out that deep learning models improve the process of stress detection by using multi-modal data and sophisticated architectures.

Table 2.1 summarizes related works, considering the modalities used, whether features were automatically extracted, cross-validation methods, and algorithms applied alongside their reported accuracies in classification. The proposed research address the limitations observed in the literature.

Chapter 3

Methodology

This section is organized into two subsections in a systematic manner. The first subsection provides a detailed account of the **optimization space** that defines all the key transformer hyperparameters, i.e., the number of layers (L, L'), attention heads (H, H'), size of model dimension (d_m) and FFN size (FFN), which are optimized for efficient functioning of the model. Secondly, we discuss **optimization plan** and procedures to adequately calibrate transformer components during the tuning stage. This provides background for our method and ensures that it can be optimized.

3.1 Optimization Space

We explore model size optimization with the most significant architectural parameters, including the number of transformer layers (L), the model dimension size (d_m), the number of attention heads (H) and FFN dimension size (FFN), as seen in Table 3.1. The default parameter settings, indicated in **bold**, serve as a reference point derived from baseline experiments.

Our design objective is to emphasize optimization speed for very efficient transformer models with minimal accuracy degradation. To realize this approach we define a **degraded search space** that only considers values up to the default configuration. This space consists of a transformer depth of 1 - 5 layers, numbers of attention heads of 1, 2 and 3, attention dimensions of 9, 18, and 30, and FFN dimension sizes of 30

Table 3.1: Optimization Space

Model size		
Transformer	Layer	{1, 2, 3, 4, **5**}
Transformer	Heads	{1, 2, **3**}
Transformer	d_m size	{9, 18, **30**}
FFN	FFN size	{30, 60, 90, **120**}

to 120. Constraining the architectural variation to this subset allows for a principled exploration of efficient transformer configurations without incurring the cost of exhaustive search. By restricting the space we are able to reduce the required number of experiments while optimizing the configurations more quickly and in a more reproducible manner. Most importantly, this setup allows for enough architectural variance to capture distinct performance profiles which can help to identify configurations that intuitively align better to the properties of the multimodal task of stress detection. This is particularly helpful when tuning models in embedded settings with constraints related to latency, memory, and energy consumption.

3.2 Optimization Plan

The optimization of the Efficient-Husformer architecture follows a structured three-stage process: defining the hyperparameter search space, identifying the most promising configuration, and conducting an ablation study to examine the contribution of individual architectural components.

The transformer models employ multi-head attention, where the **model dimensionality (d_m)** is divided over different **heads (H)**, such that partitioning is equal [8]. It ensures that each of the heads gets an equal and sensible portion of the feature space. Poorly chosen values for H (non-divisors of d_m) results in shape mismatch and computation inefficiencies. If d_m is not divisible by H, some heads would receive more features than the others and misalign multi-head self-attention computation. Besides, transformers involve attention using matrix multiplications, and uneven distribution of features across heads might introduce unnecessary computation waste. Thus, we select divisors of d_m=30 (*i.e.*, $H = 1, 2, 3$) for a balanced distribution of heads without

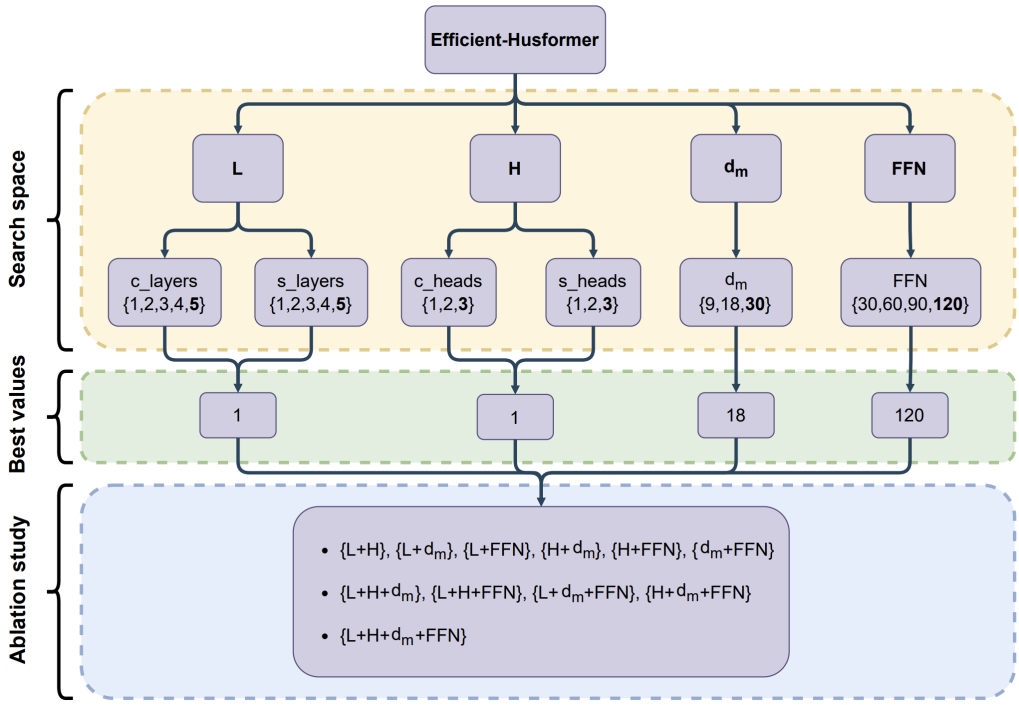

Figure 3-1: Methodology Overview

any additional padding or reshaping.

In order to achieve increased flexibility, we change the original model by taking the hard-coded **model dimension** `self.d_m` = 30, and changing it to a parameter `self.d_m` = `hyp_params.d_m`.

We also incorporated individual hyperparameters for each component of the transformer:

- **Cross-modal transformer:** `c_heads` (number of heads) and `c_layers` (number of layers).
- **Self-attention transformer:** `s_heads` and `s_layers`.

The Multi-Layer Perceptron (MLP) in each layer of the Transformer has the two fully connected layers that are commonly referred to as the feedforward network (FFN). The hidden layer size of the MLP is an important hyperparameter as it closely relates to the model capacity and compute efficiency.

In standard Husformer architectures, the FFN applies two linear transformations

[21]:

$$\text{fc}_1 : (\text{embed\_dim} \rightarrow \alpha \times \text{embed\_dim}) \tag{3.1}$$

$$\text{fc}_2 : (\alpha \times \text{embed\_dim} \rightarrow \text{embed\_dim}) \tag{3.2}$$

where α is the expansion factor which refers to the dimensions of the hidden layer. The typical alpha in standard Transformer settings is $\alpha = 4$, which means the embedding dimension will be expanded by a factor of 4.

To allow for FFN size to be flexible for optimization, we use α as a tuned hyperparameter:

$$\text{Hidden Size} = \alpha \times \text{embed\_dim} \tag{3.3}$$

By changing α, we can ramp up the expressiveness and efficiency of the FFN. Table 3.2 illustrates the hidden sizes for different values of α when embed_dim = 30.

α	**FFN dimension size**
1	30
2	60
3	90
4	120

Table 3.2: FFN dimension sizes for different values of α when embed_dim = 30.

Optimizing α requires a trade-off between cost and model capacity. Bigger α values increase the total number of parameters and compute per forward pass, which may improve representational power, but may lead to higher useness cost of latency and memory use. Conversely, smaller values, may reduce computations, but may lose the ability to capture complex representations. For our experiments, we evaluate efficiency of different α values with and without pruning.

Although it may seem reasonable to treat the model dimension (d_m) and the FFN size independently, both are inseparably tied together in the standard architecture of a Transformer. The d_m serves a shared purpose in transforming information (to attention dimensions) for the FFN, while also being flexible in tensor compatibility. The model dimension d_m establishes the sizes of the token embeddings and the output space of the attention mechanism. All attention heads project into this space, so the

d_m is a central design concern connecting all of a Transformer's core layers. The FFN, while occurring after the multi-head attention, generally operates on vectors of dimension d_m. Its typical configuration is depicted in equations (3), (4), and (5). For example, if $d_m = 64$, the FFN projects internally to the size of 256, then projects back to 64, so that the input and output shapes remain consistent. Thus, varying d_m also varies the dimensional interface for the FFN, and arbitrary changes to the size of the FFN will require additional projection layers to compensate for incompatible shapes. Clearly, these additional projection layers entail additional parameters and could introduce artifacts such as biases, that could make ablation studies or capacity comparisons less interpretable. Therefore, in practice, keeping the d_m and FFN size together makes scores or evaluations on model variants more plausible.

Conversely, parameters such as the number of attention heads (c_{heads}, s_{heads}) and the number of layers (c_{layers}, s_{layers}) in cross-modal and self-attention branches are considered more independent than dependent parameters. This independence arises in modern frameworks of transformers which can be perceived as modular. Although similar attention branches (i.e., modality-specific components or hierarchical components) may be running in either sequence or parallel, they each retain independent configurations during their operation. Each branch will take its own input stream and will often define residual connections and project layers to keep their individual interfaces compatible. This independence means that if we change the number of layers or heads in one modular branch, any assumptions made on the internal shape in another branch are unaffected. The independence of the components in transformers as we are iteratively fill out their parameters provides opportunities for research on architectural depth and parallelism, in cross-modal or multi-stream formats, with fewer constraints on using structures differently. The number of layers and FFN size would require co-tuning because they are so coupled, but modular parameters like number of heads or layers are independent proposals that can provide future research momentum.

This systematic investigation of hyperparameter space is an initial step in finding a configuration that strikes a good balance with model accuracy and computing time.

The configuration that has been chosen is ideal for multimodal learning tasks that have limited computational resources.

To better understand how the components all work, an **ablation study** is designed to measure the effect of the L, H, d_m and FFN modules individually and together. The ablation study systematically evaluate all possible pairwise combinations (e.g. $\{L+H\}$, $\{H+d_m\}$, $\{d_m+FFN\}$), triplet combinations (e.g. $\{L+H+FFN\}$), and when all four modules are used together. This way we can assess the additive and interaction effect of different components by looking at them hierarchically. It shows the impact of each of the modules in isolation and all their combinations. Such approach informs architecture refinement, and is helpful in making decisions about how to reduce, or increase the model complexity for effective deployment of the full multimodal model.

3.3 Experimental Setup

3.3.1 Experimental Settings

Hardware Platform: All experiments were conducted on an 8GB VRAM NVIDIA GeForce RTX 2070 GPU-enabled laptop. The laptop is supported with Microsoft Windows 10 (version 10.0.22621.3958) and is equipped with CUDA 12.7 and NVIDIA-SMI driver version 566.36. The GPU consumes power from 3W (idleness) to 41W (peak load). The system includes an Intel-based processor and 16GB of RAM, which is sufficient to conduct the computations for training and testing deep learning models.
Software Frameworks: We run our software on Python 3.9 and run deep learning models using GitHub codes [18] based on the PyTorch framework. CUDA is used to enable GPU acceleration for parallel processing.

3.3.2 Datasets & Quantitative Metrics

The study makes use of two datasets: Wearable Stress and Affect Detection (WESAD) [16] and Cognitive Load (CogLoad) [5].

Table 3.3: Specifications of Experimental Platform

Laptop Platform	
GPU	2304-core NVIDIA GeForce RTX 2070, 8GB VRAM, CUDA 12.7, NVIDIA-SMI 566.36
CPU	Intel-based processor
Memory	16GB RAM
Operating System	Microsoft Windows 10 (version 10.0.22621.3958)
Power	3W (idleness) to 41W (peak load)
Software Frameworks	
Programming Language	Python 3.9
Deep Learning Framework	PyTorch (GitHub-based implementations)
GPU Acceleration	CUDA 12.7

WESAD dataset

WESAD is a popular open-source multimodal stress and emotion recognition dataset within the fields of physiological and affective computing. The dataset comprises multimodal physiological sensor data recorded from 15 subjects, with each subject experiencing a controlled experiment for inducing various affective states, such as neutral, stress, amusement, and baseline states. Data were collected with two wearable sensors: RespiBAN, a chest-worn sensor that delivers galvanic skin response (GSR), respiration (RESP), electrocardiogram (ECG), electromyogram (EMG); and Empatica E4, a wrist-worn sensor that delivers GSR(wrist) and blood volume pulse (BVP), which are illustrated in figure 3-2.

To facilitate model input, these signals were segmented into fixed-size windows (700 samples for chest signals and 64 or 4 samples for wrist signals), with each segment reshaped to structured input formats suitable for model training. Only segments with consistent labels across the entire window were retained, and segments with ambiguous labeling (i.e., multiple labels within a window) were discarded. Furthermore, to focus on meaningful emotional states, segments corresponding to neutral labels were filtered out.

Additionally, to evaluate the generalization performance of our model, we adopted

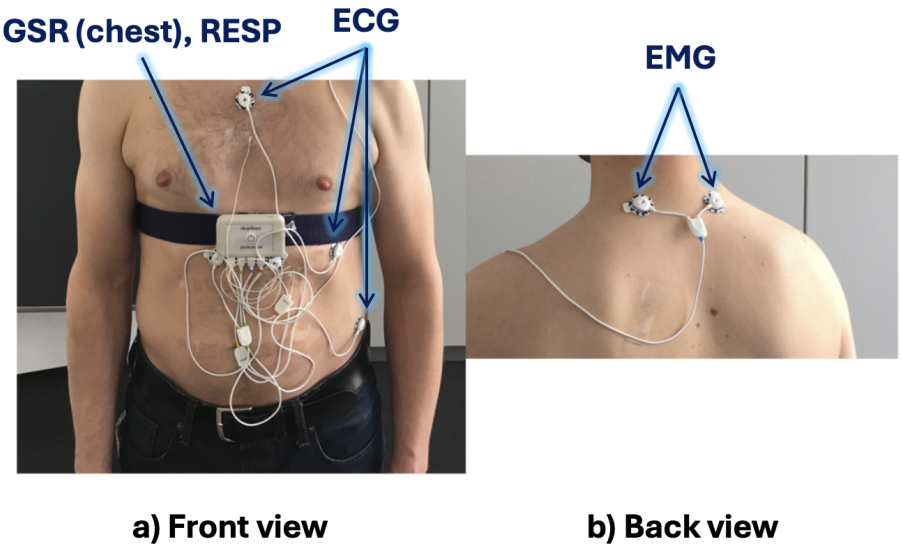

Figure 3-2: Placement of GSR (Chest), RESP, ECG, and EMG sensors. For completeness, note that GSR (Wrist) and BVP are attached to the non-dominant hand, although not illustrated (Deployed from [16] and modified).

a **10-fold cross-validation (CV)** strategy applied to the WESAD dataset as in the prior work. This approach enables robust model validation by systematically rotating training, validation, and testing splits across the data.

Given N samples, the data is partitioned into 10 approximately equal-sized subsets or *folds*. For each fold $i \in \{0, 1, \ldots, 9\}$, the following procedure is executed:

- **Validation Set:** The i-th fold, corresponding to indices from $i \cdot \frac{N}{10}$ to $(i+1) \cdot \frac{N}{10}$, is designated as the validation set
- **Test Set:** The subsequent $(i+1)$-th fold is used as the test set. For the last fold $(i = 9)$, the test set wraps around and includes the first $\frac{N}{10}$ samples, preserving fold size consistency.
- **Training Set:** All remaining samples not included in the current validation or test sets are assigned to the training set.

This results in non-overlapping training, validation, and test sets for each of the 10 folds, with each sample participating in all three roles (training, validation, testing) across the full cross-validation cycle.

In the `WESAD()` function, this procedure is implemented using slicing operations on the shuffled index array. For each fold, the respective sample indices for training, validation, and testing are passed to the `pkl_make()` function. This function constructs and serializes the fold-specific datasets into separate `.pkl` files.

Each `.pkl` file contains input data from six physiological modalities, along with corresponding labels and sample identifiers. This setup enables consistent evaluation of the model's performance across multiple independent folds, with the final performance metric averaged over all folds to reduce the risk of overfitting or sample-specific bias.

CogLoad dataset

The CogLoad [5] is an open-source, publicly available dataset that is ready-to-use in `.pkl` file format. This dataset contains physiological signals (EEG, GSR, BVP), collected from 23 participants, using a Microsoft Band. Each participant completed a rest session and six dual-task tasks, all of which sets cognitive load. The first task of the dual-task was randomly selected from standard psycho-physiological tasks from the work of Eija et al. [6]. The second task (the stimuli) consisted of participants clicking on visual stimuli that appeared on screen while they completed the first task. After completing the task, the participants reported their perceived cognitive workload using the TLX scale from the NASA-TLX questionnaire. Baseline (resting state) data was also collected for reference.

Quantitative Metrics

Our evaluation metrics from classification and computation perspectives are average multi-class accuracy (Acc), average multi-class F1 score (F1), cross-entropy loss (CE), mean absolute error (MAE), training duration (in hours), memory used (in MB), and total trainable parameters.

Let n = the number of classes. **Average multi-class accuracy** is defined by:

$$Acc = \frac{1}{n}\sum_{i=1}^{n} accuracy_i \tag{3.4}$$

where $accuracy_i$ is the binary accuracy for class i, defined by:

$$accuracy_i = \frac{TP_i + TN_i}{length_i} \tag{3.5}$$

forcing we note that TP_i and TN_i refer to the true positive and true negative predictions for class i respectively and $length_i$ refers to the number of samples in that class. **Macro average F1 score** is computed as:

$$F1 = \frac{1}{n}\sum_{i=1}^{n} f1_i \tag{3.6}$$

where $f1_i$ is the F1 score for class i, meaning the harmonic mean average between precision and recall:

$$f1_i = \frac{2 \cdot precision_i \cdot recall_i}{precision_i + recall_i} \tag{3.7}$$

$$precision_i = \frac{TP_i}{TP_i + FP_i}, \quad recall_i = \frac{TP_i}{TP_i + FN_i} \tag{3.8}$$

FP_i and FN_i, are the number of false positive and false negative predictions for class i, respectively.

In terms of loss-based evaluation, we additionally include **cross-entropy loss**, defined over all N samples and n classes as:

$$CE = -\frac{1}{N}\sum_{j=1}^{N}\sum_{i=1}^{n} y_{ji}\log(\hat{y}_{ji}) \tag{3.9}$$

where y_{ji} is the ground truth label and $\hat{y}_{ji}$ is the predicted probability for class i on sample j.

We also report **mean absolute error (MAE)**, a general-purpose metric indicat-

ing average prediction error magnitude:

$$MAE = \frac{1}{N}\sum_{j=1}^{N}|y_j - \hat{y}_j| \tag{3.10}$$

To supplement classification metrics we also indicate computational cost. We give total **training time** as:

$$T_{\text{train}} = \sum_{e=1}^{E} t_e \tag{3.11}$$

where t_e indicates the elapsed time of for epoch e, both total number of training epochs is E.

Memory efficiency is measured via peak **memory consumption** at training time:

$$M_{\text{peak}} = \max_{t\in[0,T_{\text{train}}]} memory(t) \tag{3.12}$$

Additionally, **trainable parameter count** is introduced to account for model complexity:

$$N_{\text{params}} = \sum_{l=1}^{L} P_l \tag{3.13}$$

where P_l refers to the number of parameters in layer l, and L is the total number of layers.

Husformer Model Modification: The original code produces three separate model files representing which task scenarios involved 3, 4, and 5 modalities. The modified version successfully incorporates all 6 modalities of the WESAD dataset. The Husformer showes the ability to manage modes efficiently, and feasibility was proofed by the new framework that accommodates multiple types of modal data.

Label Mapping Correction: The primary improvement of the original execution relates to the improved alignment of label mapping to that of the WESAD dataset documentation. The prior label mappings included incorrect labels that may have reduced the classification performance of the model in previous execution cycles. The improved model ensures accurate labeling alignment which provides more reliable classifications reducing or mitigating potential misclassifications via increased accu-

racy. The fixed label mapping is as follows:

Table 3.4: Label Mapping for WESAD Dataset

Label	**Condition**
-1 or 0	Stress
2	Amusement
3	Baseline

This adjustment guarantees that the dataset has consistency, which improves the reliability of the model training process. In order to evaluate the effectiveness of the model, we carried out a comprehensive evaluation using several performance metrics and provide a robust understanding of the classification capabilities. The results were illustrated in Figure 3-3, showing key trends in accuracy, F1-score, loss, and mean absolute error (MAE) across various classification settings.

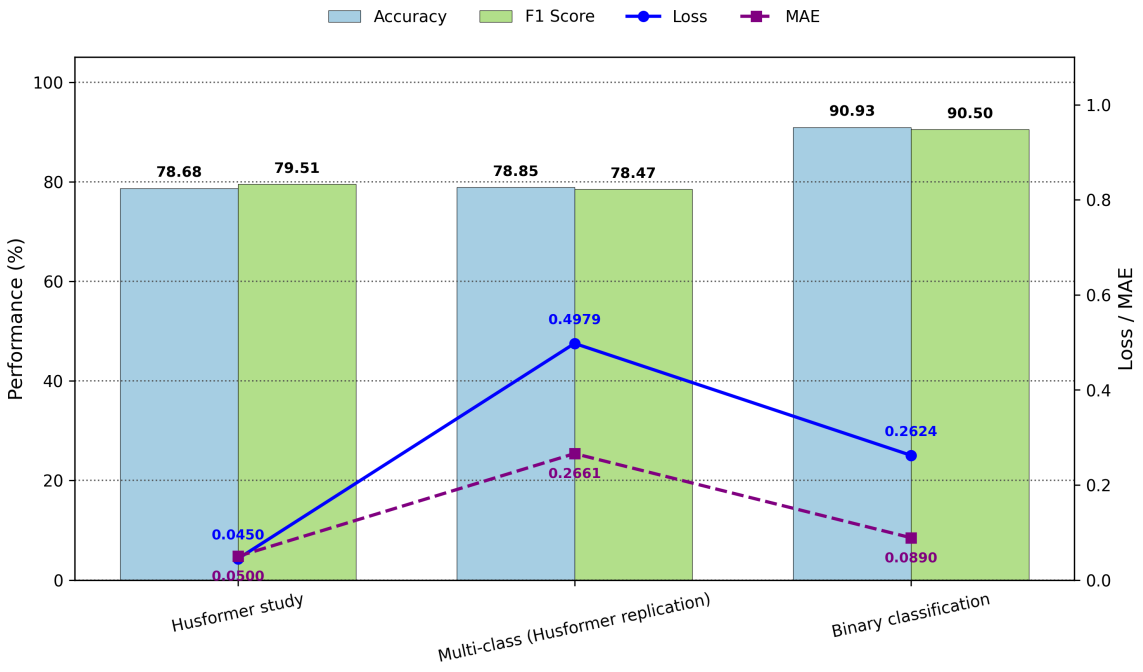

Figure 3-3: Performance Metrics of Experimental Results

The comparative performance from the Figure 3-3 of the Multi-class model closely matches with the outcome of our reference paper. The model produced accuracy

(78.85%) and F1-score (78.47%) results matching that of the paper (accuracy of 78.68% and F1-score of 79.51%), indicating that the implementation is successfully replicating the approach of the original study.

The binary classification model produced the best results with 90.93% accuracy and a F1-score of 90.50%. However, the class distribution is not balanced, because our motivation is not to allow False Positives (FPs), in other words, emphasis on stress recognition. In regard to the binary classification model, the results indicate that recasting the classification problem is simply better for predictive accuracy. The binary model also had the lowest loss (0.2624) and MAE (0.0890).

Chapter 4

Results and Analysis

4.1 Experimental Results

4.1.1 Strategic Ordering of Parameter Tuning

In order to progressively evaluate the relative importance of each hyperparameter in the Efficient-Husformer architecture, we conduct **parameter-isolation approach**. This way we measure the performance difference for a hyperparameter by varying its value while holding all other parameters at their default settings [20]. This strategy allows for the direct assessment of individual hyperparameter contributions to model accuracy, computational efficiency, and robustness when applied to multimodal stress detection in the WESAD and CogLoad dataset.

The default settings give us a reliable baseline which allows us to assume the performance change is solely initiated by the hyperparameter taken in consideration. Each hyperparameter is tested through a valid range of values, which were discussed in optimization space section.

By testing each hyperparameter separately, we indicate which parts of the model shift performance the most. For example, a significant performance improvement seen by varying $FFN\ size$, while all other parameters remain unchanged, would imply that feedforward capacity is an important tuning target. A different perspective, if performance is stable, across several values of s_layers this might imply that

Table 4.1: Layers priority configuration results at the default $c\_heads = 3$, $s\_heads = 3$, $d_m = 30$, $FFN = 120$

No	Cross-Modal Layers	Self-Attention Layers	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	1	1	0.2661	0.1266	0.8951	0.8955	0.45	653.68	81,910
2	2	2	0.4868	0.2320	0.8240	0.8165	0.75	1157.89	160,240
3	3	3	0.4682	0.2232	0.8295	0.8125	1.09	1652.86	238,570
4	4	4	0.5166	0.2549	0.8053	0.8015	1.36	2155.12	316,900
5	**5**	**5**	**0.5862**	**0.2715**	**0.7766**	**0.7787**	**1.74**	**2652.86**	**395,230**

Table 4.2: Heads priority configuration results at the default $c\_layers = 5$, $s\_layers = 5$, $d_m = 30$, $FFN = 120$

No	Cross-Modal Heads	Self-Attention Heads	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	1	1	0.4629	0.2434	0.8196	0.8116	1.42	1583.21	395,230
2	2	2	0.5552	0.2830	0.7820	0.7796	1.51	2091.66	395,230
3	**3**	**3**	**0.5862**	**0.2715**	**0.7766**	**0.7787**	**1.74**	**2652.86**	**395,230**

the depth of the self-attention stack has minimal contribution with the boundaries evaluated.

This prioritization approach, allows a staged-optimization workflow. Parameters with higher impact targetted first in the finetuning phase, and low impact parameters could either be fixed or tuned at lesser resolution to preserve compute. The knowledge gained, could also be used to instruct the ablation study which considers interactive behaviour of high-impact parameters.

4.1.2 Quantitative Assessment of Key Factors

Results on WESAD dataset

Based on all of the experiments conducted in this study, we propose an evidence-based order for modifying the transformer architecture: the order is to change the attention layers (L) and heads (H), the model dimension (d_m), and finally the FFN hidden size (FFN). This order is driven by empirical performance and cost.

We looked at configurations with different numbers of **cross-modal and self-attention layers** while holding all other architectural factors at their defaults ($c\_heads = 3$, $s\_heads = 3$, $d\_m = 30$, $FFN = 120$). The results in table 4.1 demonstrate that with each added layer the performance decreased rather than improved. The configuration with 1 cross-modal layer and 1 self-attention layer achieved the best accuracy of 0.8951, the lowest MAE of 0.1266, and lowest training time of 0.45 hours, while requiring the least memory (653.68 MB) and parameters (81,910). As the layers in-

creased, both the MAE and training time were noticeably worse, with the accuracy of 0.7766 and highest MAE of 0.2715 at the default settings of 5 and 5 layers.

This suggests that deeper architecture of Husformer does not provide performance gains for this task and may, in fact, lead to overfitting or inefficient learning. Therefore, the best value for layers is **1 cross-modal and 1 self-attention layer**, offering superior performance and resource efficiency.

Similar to the number of layers, the results of this study suggest that beyond one cross-modal and one self-attention head, there is no performance gain based on the empirical results in Table 4.2. The performance metrics for the configuration of 1 cross-modal head, and 1 self-attention head (Accuracy=0.8196; MAE=0.2434) were highest; and with the additional advantages evaluation in the study, least time to train (1.42 hours) and least memory use (1583.21 MB) of every evaluated configuration. In contrast, adding more heads (2 or 3) leads to worsening performance metrics, such as increased MAE and reduced F1-scores, despite identical parameter counts (395,230).

The pattern shown indicates that adding more attention heads may cause over-parameterizing without a corresponding gain in learning. Thus, the best case configuration is 1 head for each attention type at a the best ratio of model accuracy to computational expense.

Table 4.3 shows the impact of changing only **model dimension size**. All configurations were fixed to $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, and $FFN = 120$, effectively isolating that d_m as the sole variable.

The model with $d_m = 18$ gave the best overall performance on most evaluation metrics. It had the lowest loss (0.4896), the lowest MAE (0.2597), and the highest accuracy and F1-score (both 0.7932). In contrast, increasing d_m to default 30 value resulted in a slight performance degradation (loss = 0.5111, MAE = 0.2658, accuracy = 0.7767, F1 = 0.7787), despite a significantly larger model size and resource demand (2652.86 MB memory vs. 2289.62 MB; 395,230 parameters vs. 146,290).

Although an increase of d_m is generally associated with richer feature space and increased potential to model complexity, which is important in the analysis of physiological signals with limited data, there is also an increased computational burden and

Table 4.3: Model dimension size priority configuration results at the default $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, $FFN = 120$

No	d_m	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	9	0.6780	0.3604	0.7373	0.7152	1.52	1993.34	39,154
2	18	0.4896	0.2597	0.7932	0.7932	1.61	2289.62	146,290
3	**30**	**0.5111**	**0.2658**	**0.7767**	**0.7787**	**1.74**	**2652.86**	**395,230**

Table 4.4: FFN dimension size priority configuration results at the default $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, $d_m = 30$

No	FFN	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	30	0.4404	0.2419	0.8110	0.8113	1.67	2461.20	203,080
2	60	0.5046	0.2434	0.7979	0.7970	1.69	2525.71	267,130
3	90	0.5278	0.2612	0.7877	0.7858	1.74	2598.04	331,180
4	**120**	**0.5111**	**0.2658**	**0.7767**	**0.7787**	**1.74**	**2652.86**	**395,230**

risk of overfitting. The WESAD dataset is multi-modal but still has a limited subject and sample size. With that in mind, full responsibilities for a completely expanded parameter space with $d_m = 30$ to leverage, seems unreasonable. Practically, it is also worth mentioning that the model that used $d_m = 18$ was trained more efficiently with a small training time increase (+0.09 hours) from $d_m = 9$, while also providing substantially better performance. Since the memory footprint and parameter count still remained moderate, $d_m = 18$ is more scalable and deployment-friendly than default $d_m = 30$.

Table 4.4 presents the results of systematically varying the **feedforward network (FFN) size** parameter with all other hyperparameters held constant at default $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, and $d\_m = 30$. The size of the FFN layer is important in defining the model capacity to transform and modify feature representations throughout a feedforward operation following the attention operation, helping to determine the learning dynamics and generalization of the model.

Among the values presented in the table, the smallest FFN configuration of 30 showed the best results of all the configurations available in the features selected by increasing FFN dimension. In terms of loss (0.4404), MAE (0.2419), accuracy (0.8110), and F1 score (0.8113), value of 30 denoted superior generalization and efficiency, rather than underfitting in reference to all larger FFN dimension values indicating that in this transformer architecture and focusing on the characteristics of

Table 4.5: Ablation Study Results

Ablation Pair	L	H	D	FFN	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
Default	**5**	**3**	**30**	**120**	**0.5111**	**0.2658**	**0.7767**	**0.7787**	**1.74**	**2652.86**	**395,230**
L + H	1	1	30	120	0.4236	0.2135	0.8312	0.8235	0.39	408.92	81,910
L + d_m	1	3	18	120	0.2999	0.1486	0.8841	0.8815	0.42	575.20	30,874
L + FFN	1	3	30	30	0.3155	0.1407	0.8815	0.8819	0.43	634.03	43,480
H + d_m	5	1	18	120	0.4767	0.2567	0.7917	0.7898	1.31	1178.01	146,290
H + FFN	5	1	30	30	0.5020	0.2715	0.7957	0.7869	1.37	1368.12	203,080
d_m + FFN	5	3	18	30	0.3546	0.1649	0.8632	0.8644	1.59	2166.27	76,360
L + H + d_m	1	1	18	120	0.3855	0.2096	0.8446	0.8460	0.35	304.45	30,874
L + H + FFN	1	1	30	30	0.4021	0.2114	0.8299	0.8313	0.38	368.24	43,480
L + d_m + FFN	1	3	18	30	0.3859	0.2108	0.8479	0.8423	0.42	562.12	16,888
H + d_m + FFN	5	1	18	30	0.6260	0.3090	0.7737	0.7566	1.30	1058.63	76,360
L + H + d_m + FFN	1	1	18	30	0.4365	0.2289	0.8334	0.8235	0.37	287.53	16,888

the dataset, the smaller FFN dimension prevented unnecessary redundancy.

Importantly, as FFN dimension value increased, accuracy saw an overall downward trend. Across the 60, 90, and 120 feedforward values, the loss increased from 0.5046 (60) to 0.5111 (120), and accuracy declined from 0.7979 (60), to 0.7767 (120). There was an increasing number of parameters and additional memory usage when larger FFN values were set (from 203,080 parameters to 395,230 parameters); but either the additional parameters did not make a substantive contribution or the memory requirement inhibited deeper learning.

This behavior may be a result of overparameterization, along with the limited sample size of the WESAD dataset. Additionally, increasing the FFN size expands the model's internal layers at a greater rate compared to the attention layers potentially breaking the balance between learning representations and refining attention. Further, $FFN = 30$ has lower memory usage (2461.20 MB) and a lower training time (1.67 hours), which makes it the most performant and the lowest resource consumption, which makes $FFN = 30$ useful for lightweight deployment scenarios.

4.1.3 Ablation Study

As a means of better understanding the independent and joint effects of the primary architectural dimensions on model performance, we conducted a comprehensive ablation study. In this ablation study, we adapted combinations of four architectural parameters: total number of encoder layers (L), number of attention heads (H), dimensionality of the hidden representation (d_m), and FFN dimension (FFN). The purpose of this study was to provide a measure of the predictive accuracy, efficiency,

and model measure of complexity of each design choice. Table 4.5 shows the results summaries across 11 ablation configurations. By fixing the effects of single and paired combinations, we will derive important information about the effects of each property on the model's accuracy and scalability.

The **best performing configuration** is $\{L+d_m\}$, with one layer, three attention heads, model dimension of 18, FFN dimension of 120. The outputs were characterized with the lowest loss (0.2999) having MAE of 0.1486, the best accuracy (0.8841) and F1 -score (0.8815). Notably, this configuration occupied a lightweight architecture (only 30k parameters and 575MB memory). This suggests that depth and dimensionality are crucial synergistic factors in learning effective representations, especially in resource-constrained environments.

Regardless of the various combinations involving FFN size (FFN), results consistently demonstrate that a smaller FFN should be generally considered or even preferred. For example, the configuration $\{L+FFN\}$ and $\{d_m+FFN\}$ with $FFN = 30$ both achieved strong results (accuracy 0.8815 and 0.8632, respectively) and tended to be superior to configurations with larger FFN sizes (e.g. $\{H + FFN\}$ with 203k parameters yielded only 0.7957 accuracy). This again aligns with past experience that MLP layers are subject to diminishing returns as they become over-parameterized and possibly overfitted.

Attention head (H) seems more beneficial when added to the right level of dimensionality (e.g. to $\{L + d_m\}$, $\{L + d_m + FFN\}$), but does not hold much merit in isolation or shallow combinations. Since $\{L + H\}$ (both at size equal to 1) still performed relatively well (accuracy 0.8312), it indicates that the model derives utility from at least a minimal attention mechanism, but possibly not extensive multi-head attention.

Results on CogLoad dataset

As shown in Table 4.6, increasing the number of both cross-modal and self-attention layers from 1 to 5 leads to a noticeable increase in training time, memory usage, and model parameters. The best performance in terms of lowest loss (0.0413) and MAE

Table 4.6: Layers priority configuration results on CogLoad dataset at the default $c\_heads = 3$, $s\_heads = 3$, $d_m = 30$, $FFN = 120$

No	Cross-Modal Layers	Self-Attention Layers	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	1	1	0.0413	0.0217	0.9643	0.9639	0.31	420.51	57,398
2	2	2	0.0645	0.0353	0.9457	0.9509	0.56	780.12	113,348
3	3	3	0.0633	0.0292	0.9460	0.9410	0.79	1132.78	169,298
4	4	4	0.0684	0.0365	0.9436	0.9382	1.06	1485.34	225,248
5	**5**	**5**	**0.0695**	**0.0375**	**0.9427**	**0.9378**	**1.41**	**1837.90**	**281,198**

Table 4.7: Heads priority configuration results on CogLoad dataset at the default $c\_layers = 5$, $s\_layers = 5$, $d_m = 30$, $FFN = 120$

No	Cross-Modal Heads	Self-Attention Heads	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	1	1	0.0532	0.0279	0.9542	0.9527	1.39	1432.47	281,198
2	2	2	0.0668	0.0381	0.9360	0.9348	1.37	1635.81	281,198
3	**3**	**3**	**0.0695**	**0.0375**	**0.9427**	**0.9378**	**1.41**	**1837.90**	**281,198**

Table 4.8: Model dimension size priority configuration results on CogLoad dataset at the default $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, $FFN = 120$

No	d_m	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	9	0.0863	0.0526	0.9183	0.9114	1.38	782.33	103,706
2	18	0.0595	0.0361	0.9351	0.9288	1.46	1315.42	189,698
3	**30**	**0.0695**	**0.0375**	**0.9427**	**0.9378**	**1.41**	**1837.90**	**281,198**

(0.0217) was achieved with only 1 layer. The accuracy and F1 score also peaked at 96.43% and 96.39%, respectively, in this configuration.

Table 4.7 presents the effect of varying the number of cross-modal and self-attention heads while keeping the number of layers fixed at 5. The optimal configuration was achieved with 1 head per modality, yielding a loss of 0.0532 and an MAE of 0.0279, outperforming the default setting of 3 heads.

As shown in Table 4.8, increasing the model dimension d_m from 9 to 30 improved the accuracy from 91.83% to 94.27% and F1 score from 91.14% to 93.78%. However, the best MAE (0.0361) and lowest loss (0.0595) were obtained at $d_m = 18$, suggesting a trade-off between model capacity and generalization performance. The highest dimensionality ($d_m = 30$) led to the largest model size and memory usage, but did not consistently yield the best performance.

Table 4.9 investigates the effect of varying the FFN dimension size. The most accurate model (95.53%) with the lowest MAE (0.0287) was achieved with FFN = 30, a significantly smaller size compared to the default 120.

Table 4.9: FFN dimension size priority configuration results on CogLoad dataset at the default $c\_layers = 5$, $c\_heads = 3$, $s\_layers = 5$, $s\_heads = 3$, $d_m = 30$

No	FFN	Loss	MAE	Accuracy	F1	Train Time (hours)	Memory Used (MB)	Num Params
1	30	0.0512	0.0287	0.9553	0.9521	1.35	1457.32	187,698
2	60	0.0568	0.0325	0.9496	0.9467	1.37	1713.54	235,448
3	90	0.0723	0.0363	0.9438	0.9332	1.45	1594.77	143,948
4	**120**	**0.0695**	**0.0375**	**0.9427**	**0.9378**	**1.41**	**1837.90**	**281,198**

4.2 Summary of Results and Analysis

We evaluated the performance of our multimodal transformer-based architecture across two distinct datasets: WESAD and CogLoad. Each experimental setup explored the sensitivity of the model to changes in the number of cross-modal and self-attention layers, attention heads, model dimension size d_m, and feed-forward network size (FFN). All experiments were conducted under consistent default settings unless specified otherwise. The evaluation metrics guided our evidence-based recommendations.

First, we determined that shallower architectures with fewer attention layers significantly outperform deeper variants. The configuration with a single cross-modal and self-attention layer achieved the highest accuracy, lowest MAE, and minimal resource usage. Adding more layers led to degraded performance and increased computational cost, indicating overfitting or inefficiency in deeper models.

Second, varying the number of attention heads showed that a minimal configuration of one cross-modal and one self-attention head yields the best trade-off between performance and efficiency (lowest memory and training time). Increasing the number of heads did not improve results and introduced unnecessary overhead, suggesting that the model benefits from simplified attention mechanisms.

Third, we evaluated different model dimensions d_m while keeping other parameters fixed. A reduced model dimension of 18 offered superior performance compared to the default 30, while also lowering parameter count and memory footprint. This confirms that a moderately sized feature space balances learning capacity and generalization in low-sample scenarios like WESAD.

Fourth, the FFN dimension size was found to influence generalization. A smaller

FFN size of 30 consistently outperformed larger configurations, achieving the highest accuracy and lowest loss. Larger FFNs increased parameter counts without improving model performance, likely due to overparameterization and overfitting risks.

The ablation study further revealed that combinations involving depth (L) and model dimension (d_m) delivered the most effective configurations. The best setup $\{L + d_m\}$ used only one encoder layer and a model dimension of 18, resulting in the lowest loss and the highest accuracy, emphasizing scalability and practicality. Similarly, the combination of depth and FFN size $\{L + FFN\}$ also delivered nearly comparable results, reinforcing the importance of depth in the model's performance.

In contrast, the CogLoad dataset demonstrated high classification performance even at higher model complexities. The optimal configuration in terms of accuracy (0.9643) and F1 score (0.9639) was again achieved with the shallowest setup of one cross-modal and one self-attention layer. Increasing the number of layers slightly reduced the model's performance, though the impact was less pronounced compared to WESAD. For instance, with five layers each, the loss rose to 0.0695, MAE to 0.0375, and accuracy and F1 score marginally declined to 0.9427 and 0.9378, respectively. This suggests the CogLoad dataset may inherently be less complex or more separable, allowing simpler models to perform nearly optimally.

The trend regarding attention heads was consistent across both datasets: using a single attention head led to either the best or comparable results with significantly reduced resource usage. Increasing the number of heads did not yield further gains and instead introduced unnecessary computational overhead. In terms of model dimensionality, CogLoad again favored more compact models. The optimal configurations used relatively fewer parameters, and the gains from larger d_m or FFN values were negligible. The lowest loss and error were consistently observed in settings with fewer parameters, echoing the trend observed on WESAD.

From the collective analysis, we derive a prioritized order for hyperparameter tuning based on empirical impact:

Chapter 5

Discussion and Future Work

The findings from this study show that the Efficient-HusFormer architecture notably advances the balance of accuracy, efficiency, and resource cost tradeoffs for multi-class stress classification from physiological data. In particular, the optimization strategy with decoupling of the cross-modal transformer and self-attention transformer, was shown to be effective as it allows for mode-fusion abilities and classification functionalities to be tuned independently, enabling more interpretability and resource efficiency. While this is a advantage, one limitation of the current approach is that, even if the self-attention and cross-modal attention mechanisms are decoupled in terms of structural design, they are still using the same number of layers and number of attention heads. This design decision may hinder specialization for parameter tuning for both attention types to be personalized as needed, which should be considered moving forward, so each can be improved and explored to serve their distinct purpose.

We explored how the model dimension can be considered a tunable hyperparameter that allows for greater flexibility in resource constraints. The results suggest that there is a tradeoff that can be controlled through d_m. The model could be compressed to operate under limited resources (e.g., edge devices) when d_m was varied while still achieving acceptably reasonable performance.

To explore how to reduce computational cost while minimizing the effect on accuracy for multiphase stress classification, we applied hyperparameter optimization (HPO) to the model. The results suggest that considerable efficiencies can be achieved

when tuned systematically. This feasibility demonstrates that HPO can be utilized to adaptively modify the HusFormer architecture to allow for a range of deployment contexts with varying constraints on performance. Most notably, the layers (L) and heads (H) in the attention mechanisms had the largest impact on model performance, which aligns with the expected theorical relationship between richer attention mechanisms providing the capability to better represent interdependencies in multi-modal time-series data.

A thorough ablation study showed which parts of the architecture were the most important to results. The number of layers and d_m size had large positive influences on performance. These results further emphasize the need for systematic architectural exploration when designing effective multimodal fusion designs. Although extensive ablation experiments were conducted on the WESAD dataset, a detailed ablation study for the CogLoad dataset remains pending. Future work will focus on systematically analyzing the impact of each architectural component on the CogLoad task to better understand its robustness, generalization capacity, and sensitivity to hyperparameter variations.

Although Efficient-HusFormer has shown promising improvements to performance and efficiency, there are still areas for future work. One important avenue is improving cross-subject generalization to help accommodate inter-individual variability in physiological responses to stress (due to biological, psychological, and contextual factors), that were only shared between subjects in current evaluation. Future work could utilize domain adaptation, or personalized calibration to improve the model's ability to accommodate these variations.

A second area for improvement involves expanding the type and variety of available modalities in the model. In the current study, six physiological modalities from the WESAD dataset and three modalities from the CogLoad dataset were tested, and expanding the sensor streams to include other types of behavioral or environmental data could broaden the feature space and yield more complex predictive possibilities. Additionally, there is opportunity to expand the dataset either by utilizing existing datasets, or through the collection of additional physiological recordings from more

diverse subjects, which would strengthen model robustness, and increase utility.

As the dataset grows in diversity, it will become increasingly important to adopt evaluation strategies that rigorously assess generalization across individuals. Future work could explore implementing Leave-One-Subject-Out (LOSO) cross-validation. In LOSO, the model would be trained on all subjects except one, and tested on the excluded subject, repeating this process for each subject. This approach would offer an even stricter evaluation of subject independence and better reflect real-world generalization capabilities.

These directions provide important opportunities to build upon the successes of Efficient-HusFormer, and to make further advances in the field of affective computing, and health monitoring with physiological signals.

Chapter 6

Conclusion

In this work, we proposed Efficient-HusFormer, a novel transformer-based architecture for multi-modal physiological stress detection, designed with efficiency, flexibility, and interpretability in mind. We tested multi-modal physiological stress detection and whether optimizing hyper-parameters helped using the WESAD and CogLoad dataset, an established multimodal dataset with several physiological signals such as ECG, EMG, respiration, GSR and BVP.

We developed our architecture based on the original HusFormer backbone with some important changes to its architecture - notably, separating the cross-modal and self-attention components. With that in place, we assigned independent configurations of layers and attention heads to the inter-modal (cross-modal attention) and intra-modal (self-attention) attention so we could see if separating these components corresponds to any separation in multi-modal fusion task success.

As part of getting deep insights into performance, we calculated a deep hyper-parameter optimization (HPO) that focused on a few key architectural parameters, namely famous parameters like the number of layers and heads, FFN dimension size, and in particular we treated the model dimension as a tunable hyperparameter which can yield performance differences dependent on available resources. Along with the HPO, we ran a large ablation study to identify which architectural components impacted best classification accuracy and computational costs.

The findings from our experimentation using the WESAD and CogLoad dataset

demonstrated that by decoupling each of the attention components, we were able to exhibit additional flexibility and interpretability, while maintaining the classification accuracy and possibly improving it at the same time. We found that adjusting the model dimension can be a determinant in computational cost, especially when there are restrictions with resources; however, there will not be appreciable predictive power loss.

The original Husformer reported a mean classification accuracy of 78.68% $\pm$ 2.05 and an F1-score of 79.51% $\pm$ 2.28. In contrast, the best configured model of Efficient-Husformer represented the multimodal case of L and d_m, executed with a single transformer layer, 3 encoded attention heads, a model dimension of 18 and a feed-forward network (FFN) dimension of 120. This resulted in the lowest loss (0.2999), the mean absolute error (MAE) of 0.1486, maximum accuracy (0.8841) and F1-score (0.8815). Ultimately, this was done through a lightweight architecture with only 30K parameters, that only consumed 575 MB of memory, which also presents an additional potential for deployment to edge and resource restricted networks. Similarly, the L and FFN configuration, with 1 transformer layer, 3 attention heads, a model dimension of 30, and FFN dimension of 30, yielded a loss of 0.3155, MAE of 0.1407, accuracy of 0.8815, and an F1-score of 0.8819. This model trained in 0.43 hours, used 634.03 MB of memory, and consisted of 43,480 parameters.

These values correspond to absolute improvements of 9.73 percentage points in accuracy and 8.64 percentage points in F1-score over the baseline means. In relative terms, this constitutes an improvement of approximately 12.37% in accuracy and 10.87% in F1-score, demonstrating the effectiveness of the optimized architecture in enhancing classification quality.

In terms of efficiency, Efficient-HusFormer has a drastic reduction in the number of parameters compared to a baseline of transformer models. Efficient-HusFormer enables on-device applications without sacrificing performance. Furthermore, the model is still interpretable due to the attention mechanisms being decoupled, allowing the ability to better specify modality-specific contributions to the ultimate prediction. However, an important limitation is that while we have decoupled the architecture

mechanism for cross-modal and self-attention, we made sure to use similar layer and head counts for both cross-modal and self-attention. In future studies, this could be further differentiated for specialization.

In summary, our work have shown that careful architectural HPO can dramatically improve transformer-based models for detecting physiological stress. Efficient-HusFormer does not only improve in accuracy, but moreover, it establishes a new standard in terms of computational efficiency, laying the groundwork for the practical application in next-generation health monitoring systems.

Bibliography

[1] Asma Abu-Samah, Dalilah Ghaffa, Nor Fadzilah Abdullah, Noorfazila Kamal, Rosdiadee Nordin, Jennifer C. Dela Cruz, Glenn V. Magwili, and Reginald Juan Mercado. Deployment of tinyml-based stress classification using computational constrained health wearable. *Electronics*, 14(4), 2025.

[2] Behnam Behinaein, Anubhav Bhatti, Dirk Rodenburg, Paul Hungler, and Ali Etemad. A transformer architecture for stress detection from ecg. In *Proceedings of the 2021 ACM International Symposium on Wearable Computers*, ISWC '21, page 132–134, New York, NY, USA, 2021. Association for Computing Machinery.

[3] Pramod Bobade and Vani M. Stress detection with machine learning and deep learning using multimodal physiological data. *2020 Second International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 51–57, 2020.

[4] Krishna Teja Chitty-Venkata, Murali Emani, Venkatram Vishwanath, and Arun Somani. Neural architecture search for transformers: A survey. *IEEE Access*, PP:1–1, 10 2022.

[5] Martin Gjoreski, Tine Kolenik, Timotej Knez, Mitja Luštrek, Matjaž Gams, Hristijan Gjoreski, and Veljko Pejović. Datasets for cognitive load inference using wearable sensors and psychological traits. *Applied Sciences*, 10(11), 2020.

[6] Eija Haapalainen, SeungJun Kim, Jodi F. Forlizzi, and Anind K. Dey. Psycho-physiological measures for assessing cognitive load. In *Proceedings of the 12th ACM International Conference on Ubiquitous Computing*, UbiComp '10, page 301–310, New York, NY, USA, 2010. Association for Computing Machinery.

[7] Seyed Amir Hossein Aqajari, Emad Kasaeyan Naeini, Milad Asgari Mehrabadi, Sina Labbaf, Nikil Dutt, and Amir M. Rahmani. pyeda: An open-source python toolkit for pre-processing and feature extraction of electrodermal activity. *Procedia Computer Science*, 184:99–106, 2021. The 12th International Conference on Ambient Systems, Networks and Technologies (ANT) / The 4th International Conference on Emerging Data and Industry 4.0 (EDI40) / Affiliated Workshops.

[8] Peng Jin, Bo Zhu, Li Yuan, and Shuicheng YAN. Moh: Multi-head attention as mixture-of-head attention, 2025.

[9] Wonse Jo, Ruiqi Wang, Go-Eum Cha, Su Sun, Revanth Krishna Senthilkumaran, Daniel Foti, and Byung-Cheol Min. MOCAS: A Multimodal Dataset for Objective Cognitive Workload Assessment on Simultaneous Tasks . *IEEE Transactions on Affective Computing*, 16(01):116–132, January 2025.

[10] Gil Keren and Björn Schuller. Convolutional rnn: An enhanced model for extracting features from sequential data. *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 3412–3419, 2016.

[11] Brown Klinton and Abram Gracias. Evaluating the performance of cnn, rnn, and transformer models for real-time activity recognition. *Healthcare*, 12 2024.

[12] Sander Koelstra, Christian Muhl, Mohammad Soleymani, Jong-Seok Lee, Ashkan Yazdani, Touradj Ebrahimi, Thierry Pun, Anton Nijholt, and Ioannis Patras. Deap: A database for emotion analysis;using physiological signals. *IEEE Trans. Affect. Comput.*, 3(1):18–31, January 2012.

[13] Lucas Lange, Borislav Degenkolb, and Erhard Rahm. Privacy-preserving stress detection using smartwatch health data. In *GI-Jahrestagung*, 2023.

[14] Karthikeyan Palanisamy. Multiple physiological signal-based human stress identification using non-linear classifiers. *Elektronika ir Elektrotechnika*, 19:80–85, 01 2013.

[15] Bishwajit Roy, Lokesh Malviya, Radhikesh Kumar, Sandip Mal, Amrendra Kumar, Tanmay Bhowmik, and Jong Wan Hu. Hybrid deep learning approach for stress detection using decomposed eeg signals. *Diagnostics*, 13(11), 2023.

[16] Philip Schmidt, Attila Reiss, Robert Duerichen, Claus Marberger, and Kristof Van Laerhoven. Introducing wesad, a multimodal dataset for wearable stress and affect detection. In *Proceedings of the 20th ACM international conference on multimodal interaction*, pages 400–408, 2018.

[17] Fatma Shaheen, Brijesh Verma, and Md. Asafuddoula. Impact of automatic feature extraction in deep learning architecture. In *2016 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, pages 1–8, 2016.

[18] SMARTlab-Purdue. Smartlab-purdue/husformer: This repository contains the source code for our paper: "husformer: A multi-modal transformer for multimodal human state recognition". https://arxiv.org/abs/2209.15182.

[19] Yuzhu Su, Likun Ge, and Gaoxia Wei. Random forest model predicts stress level in a sample of 18,403 college students. In *Proceedings of the 2024 4th International Conference on Artificial Intelligence, Big Data and Algorithms*, CAIBDA '24, page 588–593, New York, NY, USA, 2024. Association for Computing Machinery.

[20] Daphne Theodorakopoulos, Frederic Stahl, and Marius Lindauer. *Hyperparameter Importance Analysis for Multi-Objective AutoML*. IOS Press, October 2024.

[21] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[22] Juan Vazquez-Rodriguez, Grégoire Lefebvre, Julien Cumin, and James L. Crowley. Transformer-based self-supervised learning for emotion recognition, 2022.

[23] Tuong Thuy Vu, Van Thong Huynh, and Soo hyung Kim. Multi-scale transformer-based network for emotion recognition from multi physiological signals. In *Asian Conference on Pattern Recognition*, 2023.

[24] Ruiqi Wang, Wonse Jo, Dezhong Zhao, Weizheng Wang, Arjun Gupte, Baijian Yang, Guohua Chen, and Byung-Cheol Min. Husformer: A multimodal transformer for multimodal human state recognition. *IEEE Transactions on Cognitive and Developmental Systems*, 16(4):1374–1390, 2024.

[25] Ziqing Yang and Houwei Cao. Decompose time and frequency dependencies: Multivariate time series physiological signal emotion recognition, 2024.

[26] Yiqun Yao, Michalis Papakostas, Mihai Burzo, Mohamed Abouelenien, and Rada Mihalcea. MUSER: MUltimodal stress detection using emotion recognition as an auxiliary task. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2714–2725, Online, June 2021. Association for Computational Linguistics.

[27] Sayyedjavad Ziaratnia, Tipporn Laohakangvalvit, Midori Sugaya, and Peeraya Sripian. Multimodal deep learning for remote stress estimation using cct-lstm. In *2024 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 8321–8329, 2024.