Kaunas University of Technology

Faculty of Informatics

Using Skeleton Detection to Identify Stress-Induced Changes in Human Movement Patterns

Master's Final Degree Project

Marius Klumbys

Project author

Assoc. Prof. Eglė Butkevičiūtė

Supervisor

Kaunas, 2026

Kaunas University of Technology

Faculty of Informatics

Using Skeleton Detection to Identify Stress-Induced Changes in Human Movement Patterns

Master's Final Degree Project

Artificial Intelligence in Computer Science (6211BX007)

Marius Klumbys

Project author

Assoc. Prof. Eglė Butkevičiūtė

Supervisor

Prof. Gintaras Palubeckis

Reviewer

Kaunas, 2026

Kaunas University of Technology

Faculty of Informatics

Marius Klumbys

Using Skeleton Detection to Identify Stress-Induced Changes in Human Movement Patterns

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Marius Klumbys

Klumbys Marius. Using Skeleton Detection to Identify Stress-Induced Changes in Human Movement Patterns. Master's Final Degree Project / Assoc. Prof. Eglė Butkevičiūtė; Faculty of Informatics, Kaunas University of Technology.

Study field and area (study field group): Computer Science, Informatics (B01).

Keywords: Stress detection; Body pose estimation; Facial expression analysis; Multimodal feature fusion; Supervised learning.

Kaunas, 2026. 53 pages.

Summary

This study investigates video-based stress detection using body pose and facial expression features. A unified feature extraction pipeline gathers posture and facial data from video and aggregates them into fixed windows. This research proposes and systematically evaluates a video-based stress recognition framework that fuses body pose and facial expression features and benchmarks classical machine learning and sequence models. Three models: HistGradientBoosting, SVM with RBF kernel and BiLSTM are evaluated on two datasets: SWELL-KW and StressID. On SWELL-KW dataset, the BiLSTM achieved the highest balanced accuracy (~0.60), while on StressID dataset, HistGradientBoosting reached the best balanced accuracy (~0.74). The results indicate that dataset characteristics strongly influence model performance, and while video-only features provide a meaningful stress signal, further methodological improvements are needed to enhance detection accuracy.

Klumbys Marius. Skeleto aptikimo metodų taikymas nustatant streso sukeltus žmogaus judesių pokyčius. Magistro studijų baigiamasis projektas / Assoc. Prof. Eglė Butkevičiūtė; Kauno technologijos universitetas, Informatikos fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Informatikos Mokslai, Informatika (B01).

Reikšminiai žodžiai: Streso atpažinimas; Kūno pozicijos įvertinimas; Veido išraiškų analizė; Multimodalinių požymių suliejimas; Prižiūrimas mokymasis;

Kaunas, 2026. 53 p.

Santrauka

Šiame tyrime nagrinėjamas vaizdo įrašu pagrįstas streso atpažinimas, naudojant kūno pozicijos ir veido išraiškų požymius. Vieninga požymių išskyrimo seka surenka laikysenos ir veido duomenis iš vaizdo įrašo ir agreguoja juos į fiksuotus laiko langus, tuomet streso atpažinimo sistema sujungia kūno pozicijos ir veido išraiškų požymius bei palygina klasikinius mašininio mokymosi ir sekų modelius. Trys modeliai: HistGradientBoosting, SVM Kernel RBF ir BiLSTM vertinami naudojant du duomenų rinkinius: SWELL-KW ir StressID. Naudojant SWELL-KW duomenų rinkinį, BiLSTM pasiekė didžiausią subalansuotą tikslumą (~0,60), o naudojant StressID duomenų rinkinį, geriausią subalansuotą tikslumą (~0,74) pasiekė HistGradientBoosting. Rezultatai rodo, kad duomenų rinkinio charakteristikos daro didelę įtaką modelių veikimui, ir nors vien vaizdo įrašo požymiai suteikia reikšmingą streso signalą, norint padidinti atpažinimo tikslumą, reikalingi tolesni metodologiniai patobulinimai.

Table of contents

List of figures

List of tables

List of abbreviations and terms

Abbreviations:

CNN – convolutional neural network;

MLP – multilayer perceptron;

ECG – electrocardiogram;

RESP – respiration features;

LSTM – long short-term memory;

SVM – support vector machines;

DNN – deep neural network;

EEG – electroencephalography;

GSR – galvanic skin response;

NASA-TLX – NASA task load index;

LOPO – leave-one-participant-out;

HPA – hypothalamic-pituitary-adrenal;

COCO – common objects in context;

FPS – frames per second;

FLOPS – floating point operations per second;

RBF – radial basis function;

Terms:

Recall – a metric that measures how often a machine learning model correctly identifies positive instances (true positives) from all the actual positive samples in the dataset;

ReLU activation function – outputting the input directly if it is positive, otherwise, it outputs zero;

Sigmoid function – transforms the continuous real number into a range of (0, 1), so that the input value of the next layer is within a fixed range and the weight is more stable;

Introduction

Relevance of the problem

Stress, especially psychosocial stress, is a prevalent issue with significant health implications. Stress triggers the sympathetic “fight-or-flight” response and HPA-axis, leading to hormonal surges (e.g. adrenaline, cortisol) that prepare the body for action. While short-term stress responses can be adaptive, chronic or repeated stress is linked to negative health outcomes such as anxiety, depression, cardiovascular disease, and metabolic disorders. Early identification of stress is therefore crucial in clinical settings to prevent these long-term effects. There is a growing need for non-invasive, real-time stress detection methods that can be used in everyday environments or telemedicine. In this context, human motor behavior and body language offer a promising path: stress can manifest through changes in posture, reduced movement (freezing), or fidgeting, which could serve as observable indicators. Developing a system to detect stress-induced movement changes could aid clinicians in diagnosis (e.g. identifying high stress in patients during exams or therapy) and enable continuous monitoring for preventive health care.

Aim of the project

The aim of this project is to use skeleton detection technology to identify stress-induced changes in human movement patterns. In other words - to develop a system that uses pose estimation (tracking the positions of human joints over time) to infer when a person is under stress. By analyzing skeletal movement data (e.g. joint angles, motion dynamics), the project aims to detect behavioral signatures of stress (such as reduced overall mobility or specific tension-related postures) in a measurable manner. The goal is a reliable, real-time stress detection tool that could be used in clinical practice or daily life to flag high stress levels without the need for wearable sensors or similar technologies.

Objectives of the project

1. Analyze various literature and studies to understand the concept of stress detection using artificial intelligence, including any relevant formulas and algorithms.

2. Analyze and select appropriate and customizable technologies and models that best fit the project requirements.

3. Implement skeleton and pose detection to retrieve data from input sources and structure it for use in the later components of the system.

4. Implement the stress detection system, which will determine how effectively the collected data can be used to identify signs of stress.

5. Evaluate the model in real-world scenarios by testing the system in more complex and dynamic environments to assess its performance and demonstrate the practical value of its results.

Scientific novelty

While skeleton detection has been widely used in applications such as gesture recognition, activity tracking, and human movement analysis, its use for stress-related movement analysis remains a developing research direction. This project builds on existing work in multimodal stress recognition by applying skeleton detection from standard video, together with facial landmark information, to support contactless stress analysis without requiring wearable or full-body sensor systems.

The scientific contribution of this project lies not in introducing pose-based stress detection as an entirely new modality, but in developing an integrated and practical framework for this task. Specifically, the project contributes a unified preprocessing pipeline for video-based pose and facial feature extraction, a cross-dataset feature representation suitable for stress-related movement analysis, a comparison framework for evaluating different machine learning models, and a real-time inference prototype. These elements together provide a scalable and non-invasive approach for investigating how stress may be reflected in human posture, movement, and facial cues.

1. Analysis of Video-Based Stress Detection Literature

Currently, stress is assessed via subjective surveys or biomarkers (cortisol, etc.), which can be biased or require invasive sampling. Notably, researchers envision contactless stress assessment (for example, via camera) to eventually supplement or even replace blood/saliva tests, improving accessibility and reducing costs. [1]

1.1. Human pose estimation

Human pose estimation is a computer vision technique that involves identifying and localizing specific keypoints of the human body, such as the head, shoulders, elbows, hips and knees, from images or videos [2]. This technology has a wide range of applications, including sports training, rehabilitation, human-computer interaction, and augmented reality. Recent advancements in deep learning, particularly convolutional neural networks, have significantly improved the accuracy and efficiency of the pose estimation systems, enabling both 2D and 3D pose detection. Methods such as top-down, where the model first detects the person and only then estimate the pose, and bottom-up approach, where keypoints are first detected and then grouped into individuals are commonly used. Pose estimation systems often utilize datasets like COCO and MPII for training [3][4], and frameworks like OpenPose and MediaPipe have become popular tools in this field. As pose estimation continues to evolve, it is increasingly being integrated with movement assessment and feedback systems, showing the way for automated analysis and improvement of physical performance in various domains.

Body keypoints detection has progressed beyond basic localization of body keypoints to enable advanced applications that combine movement assessment, feedback mechanisms and reasoning about human posture and motion. Studies highlight the integration of pose estimation with systems for evaluating movement quality and delivering feedback. Such systems use deep learning algorithms to detect human keypoints and assess physical movements in real-time. For instance, libraries like OpenPose and MediaPipe are employed to identify keypoints, while the movement quality is evaluated through machine learning or mathematical models, with visual or verbal feedback. These systems are very important and promising in contexts like rehabilitation and sports, where they assist users in improving their techniques by highlighting difference from optimal movements. On the other hand, there is still a huge need for research to address challenges in feedback prioritization and hardware limitations, as well as the integration of accessible technologies like webcams. [5]

Furthermore, there are some more challenges, such as occlusion, variability and context dependency. One study proposes a cascade of deep neural networks designed to refine joint localization through successive stages of prediction and correction (Fig. 1. Schematic view of the DNN-based pose regression and refinement). Unlike traditional methods that rely on part-based models or hand-crafted feature detectors, the DNN-based approach uses the full image context to regress the location of body joints, achieving state-of-the-art precision on various benchmarks.

Fig. 1. Schematic view of the DNN-based pose regression and refinement [6]

The study also emphasizes the importance of incorporating higher-resolution sub-images in coming stages of the cascade to improve accuracy in joint prediction. By eliminating the need for domain-specific feature engineering, this method demonstrates the adaptability of DNNs to various real-world applications, from healthcare monitoring to gesture-based human-computer interaction. [6]
This deep pose method achieves superior performance across multiple benchmarks. On the leeds sports dataset, it outperforms existing approaches in detecting limbs, particularly challenging ones like lower arms and legs, achieving a percentage of correct parts of 78% for upper legs compared to the next-best method at 74%. On the frames labeled in cinema dataset, the model demonstrates higher detection rates for joints like elbows and wrists, with gains of 15-20% over competing methods. The method nature allows it to handle difficult cases, such as occlusions and complex poses, better than traditional part-based models.

1.2. Stress detection

AI based stress detection has emerged as a transformative approach to understanding and managing stress through the integration of advanced machine learning techniques and multimodal data analysis. These systems use physiological and behavioral signals such as heart rate, electrodermal activity, respiration patterns, posture and facial expressions, collected via wearable sensors or video analysis [7]. Unlike traditional self-reported measures, which rely on subjective inputs, AI stress detection systems provide objective, continuous monitoring, enabling real-time identification of stress levels. Deep learning models, including convolutional neural networks and recurrent neural networks have proven particularly effective in analyzing complex, multimodal datasets. For instance, they can autonomously extract features from raw sensor data and integrate multiple modalities, such as combining physiological signals with facial emotion analysis, to improve detection accuracy [8]. AI systems not only detect stress but also classify its intensity and provide feedback. These innovations have a major impact on personalized stress management solutions and are particularly valuable in environments like healthcare and education, where timely stress detection can significantly improve performance and well-being.

The integration of diverse data, including physiological signals (heart rate variability), behavioral actions (posture and facial expressions) and computer interactions is very important in detection of stress. Using multimodal AI, systems can analyze heterogeneous datasets to capture various stress responses. For instance, the SWELL-KW dataset, used in some studies, combines features like keyboard activity, eye tracking and body posture to identify stress triggers and trends over time. The fusion of modalities significantly enhances prediction accuracy by compensating for the limitations of individual data. For example, if physiological data is noisy or unavailable, behavioral data can fill the

gap, ensuring robust stress detection across diverse contexts. This multimodal approach is particularly impactful in sedentary environments, such as workplaces, where small stress indicators might be overlooked [9]. In one study, the early fusion model achieved superior performance, with a stress detection accuracy of 96.67%, precision, recall and an F1 score of 0.95, outperforming the late fusion model, which had an accuracy of 90.45%. For NASA-TLX regression, the early fusion model demonstrated excellent precision, achieving a minimal root mean squared error of 0.036 on the test set. Body posture emerged as the most reliable indicator of stress, with an individual accuracy of 77.56%, followed by facial expressions at 74.05% and keystroke dynamics at 71.33% [9].

There are more architectures that use deep learning models like convolutional neural networks and multilayer perceptrons to analyze raw physiological data. For example, 1D convolutional neural network for chest-worn sensors and a multilayer perceptron for wrist-worn sensors. This approach eliminates the dependency on hand-crafted features, instead allowing neural networks to autonomously extract meaningful patterns from signals such as electrocardiogram and electrodermal activity [10]. The study demonstrates that deep neural networks can achieve near-perfect accuracy in binary stress detection and high accuracy in classifying multiple emotional states, such as stress. By using datasets collected through wearable sensors, the models demonstrate versatility in both chest-worn and wrist-worn applications, showcasing their adaptability to various factors. This work shows the advantages of deep learning in stress detection, particularly in scenarios requiring real-time, scalable solutions. Using this method, the CNN achieved a 99.80% accuracy for binary stress detection and 99.55% for emotion classification. The MLP also performed impressively, with accuracy rates of 99.65% and 98.38% for the respective tasks, significantly outperforming traditional machine learning models. [10]

Some studies show different approaches in different areas, some include combination of physiological data (ECG and respiration) with facial emotion analysis to capture a complete picture of stress responses. There is a study that employs a deep learning architecture with feature and decision level fusion to merge data streams, improving classification performance. By extracting facial landmarks and textures from video alongside physiological features, the model compensates for noise or missing data in one modality with complementary information from another. This dual-layered fusion approach allows for the detection of stress during simulated workplace tasks, providing insights into stress management in professional environments. [11] This study highlights the critical role of multimodal fusion in improving accuracy and ensuring reliability, particularly in applications where stress changes across individuals and situations. The proposed system uses convolutional layers and long short-term memory networks to process ECG and RESP signals. Bidirectional LSTMs are employed to analyze sequential facial features, capturing emotional changes. Feature level fusion combines features extracted from different signals into a single classifier while decision level fusion uses weighted ensemble predictions from individual models for final classification (Fig. 2. Features extracted by each branch for each signal. (a) Network branches for processing ECG, RESP and the sequential of facial features. (b) Feature-level fusion. (c) Decision-level fusion).

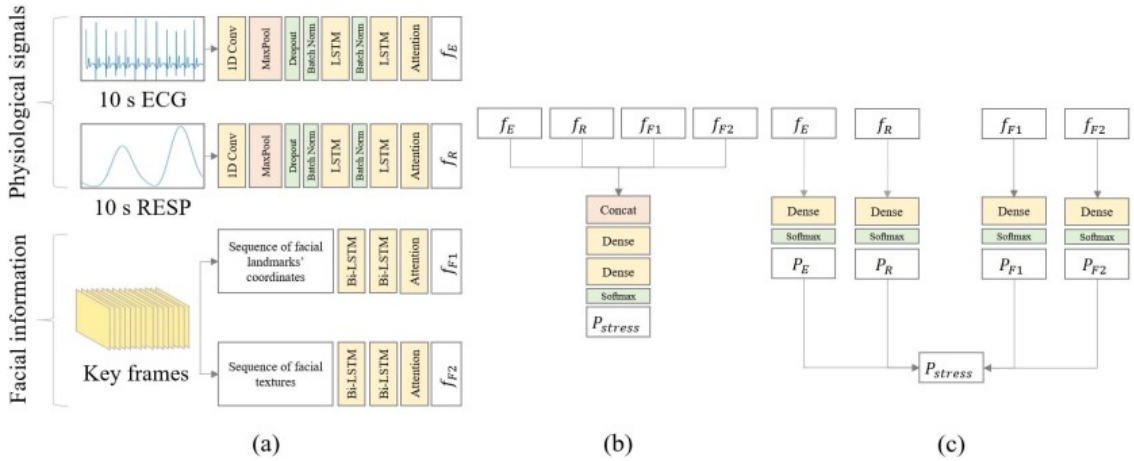

Fig. 2. Features extracted by each branch for each signal. (a) Network branches for processing ECG, RESP and the sequential of facial features. (b) Feature-level fusion. (c) Decision-level fusion [11]

There are two-level (non-stress vs. stress) and three-level (non-stress, medium stress, and high stress) classifications. Using this methodology, two-level achieved 73.3% accuracy using RESP and facial landmarks, while three-level achieved 54.4% accuracy using ECG, RESP and facial landmarks [11].

Basically, a majority part of the studies covers only researches from wearable devices as they are more precise. Wearable sensor methods primarily involve data from electroencephalography, electrocardiography, galvanic skin response and other physiological markers, while non-wearable methods include speech analysis, body posture evaluation, thermal imaging and smartphone-based activity monitoring. Researches highlight the significant role of machine learning and multimodal fusion techniques in improving stress detection accuracy and how important is feature selection, noise reduction and good classification algorithms in stress assessment. There are many challenges and limitations in current stress detection methods, such as the lack of real-world testing, dataset standardization and effective labeling techniques. [12] This shows the need for integrating contextual information (environmental conditions) and reducing device power consumption for real world applications. At the current state, wearable sensors like EEG, ECG, and GSR offer high accuracy for stress detection, with methods achieving up to 97% accuracy using advanced machine learning classifiers like SVM, random forest and deep neural networks [13]. Non-wearable techniques, such as thermal imaging and speech-based stress analysis, show promise but are often influenced by external factors like lighting and background noise.

One thing should not be forgotten, and it is a meaningful feedback for the user. Implementing stress detection systems with interactive AI-driven platforms for personalized interventions is a crucial part for the whole AI-based stress detection system as it would show its real usability for the real-world environments. This issue is addressed by some researches, they use machine learning algorithms like random forests, the system analyzes physiological parameters such as heart rate, respiration rate and snoring patterns to predict stress levels. In addition to detection, their platform incorporates an AI assistant capable of providing real-time recommendations for stress management. This dual capability enhances user engagement by offering actionable insights and support, such as guided relaxation techniques or sleep quality improvements. The addition of the interactive part bridges the gap between stress detection and stress management. The study shows the importance of combining robust machine learning models with user-specific features to create practical tools that not only detect stress

but also help users actively manage and reduce its impact [14]. Study results demonstrate the effectiveness of the random forest classifier in accurately categorizing stress levels into five categories: low/normal, medium low, medium, medium high and high. Feature importance analysis highlighted main parameters, such as heart rate variability and snoring rate, that significantly influence stress detection. The AI assistant, Gemini 1.0 Pro, improved system usability by offering advice, which includes relaxation techniques and lifestyle adjustments based on the user stress profile. The system achieved high precision, recall, and F1 scores during evaluation, showcasing its reliability. Real-time feedback and personalized recommendations were found to increase user engagement to stress management practices [14].

A lot of researchers have investigated how basic emotions are connected to stress. Turns out, our emotions can actually show how stressed we are. By analyzing things like facial expressions and tone of voice from videos or recordings, it is possible to pick up on emotional cues that point to stress. We show how we are feeling not just in our voice, but also through our facial expressions. The first step is usually to figure out what emotions are being expressed in the audio and visuals. Positive emotions like happiness, love, pride, or joy can boost our mood and help people perform better in everyday life. On the flip side, negative emotions like anger, sadness, fear, or disgust can take a toll on their health. Emotions like depression, anxiety, frustration, and anger are closely tied to stress. In particular, anger and disgust are two strong signs that someone might be feeling stressed. [15]

1.3. Literature analysis conclusions

Human pose estimation studies show how advanced deep learning models and innovative frameworks is transforming human skeletal detection into a powerful tool for movement analysis across multiple domains. One thing to keep in mind is to prepare for occlusions and precisely detect body keypoints as this is very important part, because if these functions aren’t performing accurately then the whole prediction is likely to be wrong.

Stress detection studies show lots of different approaches in detecting stress and how they are better in their own way. There are few things to have in mind from stress detection studies, one is that there are not enough researches made on this particular topic where a model directly detects stress from the movement of a human skeleton. Secondly, there are a lot of limitations regarding this domain, whether it’s software or hardware. Lastly, multimodal approach is the best way to get the most accurate results.

2. Body Pose and Facial Landmarks Feature Extraction Technologies

There are a lot of pose detection models, but in this project only one of them will be implemented (a prototype).

2.1. Requirements

These are the main requirements for the model to be suitable for a production-ready product:

- Multi-person pose tracking – the model should detect and track multiple people simultaneously (e.g. small groups of 2-5 individuals) in the camera. It should handle people entering or leaving the frame and maintain consistent tracking IDs for each person to analyze stress over time;
- Real-time processing – the system should operate in real time, processing frames with minimal lag. A target frame rate of at least 15-30 FPS is desired to ensure smooth monitoring. Real-time performance is crucial so that stress indicators (like body fidgeting or facial expressions) are captured as they happen;
- Customization – in addition to pose, the system should have the ability integrate an emotion detection module to assess facial expressions or other visual stress cues. This could involve face detection and emotion classification (e.g. recognizing expressions like fear, anger, or discomfort). The requirement is that the software can correlate body language and facial emotions to more reliably detect stress;
- Hardware requirements – from a hardware standpoint, a recommended parameters would be a system with an NVIDIA GPU (with CUDA support) to comfortably handle real-time multi-person analysis. Also, a quality webcam (1080p or better) for input is expected as part of the hardware setup;

2.2. MediaPipe Pose

Provides high-fidelity 3D body landmarks (33 keypoints including face, hands, feet) and achieves robust pose tracking [16]. Studies found MediaPipe's accuracy comparable to state-of-the-art models like MoveNet, even outperforming OpenPose in some scenarios. It is reliable under typical conditions, though extreme poses or occlusions can still pose challenges (common to most pose estimators). Designed for single-person tracking. MediaPipe's pipeline locates one person's ROI and then tracks the pose within it. Out-of-the-box it does not detect multiple people simultaneously. Developers can handle multiple people by running a person detector (e.g. YOLO) to crop each person and then applying MediaPipe per person, but this requires extra integration. Real-time performance is excellent. MediaPipe Pose uses Google's BlazePose models, achieving real-time inference even on mobile CPUs. On a typical desktop CPU it can run at 30+ FPS for one person [17]. Its lightweight design trades a bit of precision for speed, enabling low-latency tracking suitable for live video. MediaPipe provides ready-to-use solutions in Python and C++, and even JavaScript/WebAssembly for web apps. The API is straightforward (initializing a Pose object, then feeding frames), and extensive documentation and examples are available. Pre-built binaries and a PyPI package simplify installation, avoiding the need to compile from source. Runs on CPU or GPU. MediaPipe is optimized for CPUs - it can run on modest hardware (even Raspberry Pi or mobile phones) at real-time speeds [18]. GPU acceleration (via TFLite or GPU delegate) can further boost frame rates. This makes MediaPipe ideal for broad deployment without specialized hardware.

2.3. MMPose

MMPose is a toolkit supporting many models (e.g. HRNet, ViTPose, RTMPose) [19]. It can achieve very high accuracy on benchmarks - for example, the new RTMPose model reaches 75.8% AP on COCO (keypoint detection accuracy), surpassing older methods [20]. Its models are generally reliable even in complex poses, and you can choose architectures to balance accuracy vs. speed. MMPose supports both top-down and bottom-up approaches for multi-person pose estimation. This means it can first detect people then find keypoints (top-down) or directly infer multiple poses from the whole image (bottom-up), depending on the model. It handles small group scenarios well, and even crowded scenes with appropriate models. Tracking identities across frames is also supported for video-based analysis. Some MMPose models are lightweight: e.g. RTMPose can run 90+ FPS on a desktop CPU and hundreds of FPS on a GPU, demonstrating real-time capability [20]. MMPose offers a comprehensive API and configuration system, but it may have a steeper learning curve than MediaPipe. Integration typically involves installing the library and models, then using its inference APIs or even writing some glue code for custom use. Documentation is provided, and there is an active open-source community. It's excellent for researchers and developers familiar with deep learning in Python, but less straightforward for non-Python environments (no official C++ API). Memory requirements vary by model (lighter models can run in a few hundred MB of GPU memory; heavy ones need several GB). The framework is flexible to different hardware, but tuning (e.g. using half-precision, quantization) may be needed for edge devices.

2.4. OpenPose

OpenPose pioneered multi-person pose estimation and is known for robustly detecting people in complex scenes. It detects up to 135 keypoints (body, face, hands) with a single network [21]. However, its output keypoints are relatively low-resolution and less precise compared to newer models. In modern evaluations, OpenPose often shows lower accuracy (e.g. about 61.8% AP on COCO test, far below newer models). One study found OpenPose had the lowest detection performance among contemporary libraries, struggling especially with occlusions and complex poses [22]. It remains reliable in straightforward scenarios, but its older architecture (based on VGG-19 and Part Affinity Fields) can misdetect or drop keypoints under challenging conditions. OpenPose was the first real-time multi-person pose estimator and handles multiple people in an image by design. It uses a bottom-up approach (detecting all keypoints then assembling people) which inherently allows any number of people without a separate person detector. It performs well in multi-person settings and even crowded scenes, which was one of its strongest advantages when it debuted. (It also provides tracking to maintain identities frame-to-frame.) OpenPose can run in real-time with GPU acceleration. On a modern desktop GPU, it can approach ~20-30 FPS for a few people, but performance degrades with more people or high resolutions. It is considered computationally heavy - each inference is ~160 billion FLOPs. There are "light" models of OpenPose, but generally CPU-only operation is slow (often less than 1 FPS for full body model) [23]. For desktop use, an NVIDIA GPU is typically needed to meet live video frame rates. OpenPose is written in C++ but provides Python, C++, and Unity APIs. However, using it can involve compiling the library or using provided binaries. The documentation is detailed but can be overwhelming, and new users might struggle with setup. Integration into an application might require managing its output format and perhaps customizing which model components to run (e.g. body only or including hands/face). Compared to MediaPipe, OpenPose integration is less plug-and-play (no simple pip install for full functionality and you often build from source).

2.5. AlphaPose

AlphaPose is known as one of the first open-source systems to exceed 70 mAP on COCO, substantially outperforming OpenPose on that benchmark and indicating excellent joint localization [24]. AlphaPose's two-stage approach (detect person, then pose) and strong pose backbone yield reliable results even for complex poses. It is designed to handle whole-body pose (including hands & face in later versions) and is generally robust, though like all top-down methods it can drop accuracy if the person detector fails or someone is missing, it basically finds all people, then it predicts keypoints for each. This makes it capable of handling multiple individuals in a frame. It also includes an online pose tracker (PoseFlow) to maintain identities across frames, which is useful for analyzing small group interactions over time. The top-down nature means runtime scales with number of people (each person's pose is estimated separately), but it handles small groups (e.g. 5–10 people) effectively on a good system. AlphaPose has been optimized for speed: one version ran at ~20 FPS on a COCO-val dataset with ~4-5 people per image using a 2018 GPU. With modern GPUs, it can run in real time for a few people (expected 30+ FPS for 1-2 people on an average GPU, lower for many people). It also offers lighter models and settings for faster inference (trading off some accuracy). On CPU-only, real-time is unlikely unless dealing with a single person at low resolution. AlphaPose is provided as open-source code (Python, PyTorch-based). Integrating it may involve installing dependencies and using provided inference scripts or API functions. It's not as neatly packaged as MediaPipe; however, there are community forks and documentation to help with usage. Developers will need to handle the model weights and possibly the detection model configuration. The documentation includes an output format guide and some examples, but using AlphaPose in a custom app requires some coding to tie the detector and pose model together. Language support is mainly Python, so integrating into a C++ or C# app would need an extra wrapper or using it as a separate process. As a deep learning-based two-stage method, AlphaPose benefits from a CUDA-capable GPU for both the person detector (often YOLO or Faster RCNN) and the pose model. In terms of memory, the models (detector + pose network) can require a couple of GB of VRAM.

2.6. Models comparison

Basically, selected models will serve as a preprocessing tool to capture skeleton data which will be fed into a deep learning model for stress classification.

Table 1. Comparison of different pose detection models

Feature	MediaPipe Pose	MMPose	OpenPose	AlphaPose
Accuracy and Reliability	High fidelity (33 keypoints); robust tracking; slightly less precise than heavy models but very effective	State-of-art models (can exceed 75 AP); very accurate and configurable	Good but aging (~62 AP COCO); lower precision under occlusion	Very high accuracy (~73 AP COCO); reliable whole-body poses
Multi-Person Support	Single-person (multi-person requires external logic)	Yes, multi-person (top-down and bottom-up supported)	Yes, built-in multi-person (bottom-up, unlimited people)	Yes, multi-person via top-down (detect + pose)
Real-Time Performance	Excellent on CPU (mobile-ready); 30+ FPS easily	Real-time possible with optimized models (e.g. 90 FPS on CPU with	Requires strong GPU for real-time inferring; heavy computation (~160	Real-time on GPU for few people (~20-30 FPS); slower if many people

Feature	MediaPipe Pose	MMPose	OpenPose	AlphaPose
		RTMPose); heavy models need GPU	GFLOPs per image).	
Integration and Docs	Easy API in Python/C++; extensive docs and examples	Good Python API; thorough docs, but higher learning curve.	C++/Python support; setup is non-trivial (often requires building)	Python-based code; moderate integration effort (use provided scripts or adapt code).
Hardware Needs	CPU sufficient; GPU optional. Runs on modest hardware	GPU recommended for multi-person; flexible to different setups	High-performance GPU needed for 30 FPS; CPU is very slow	GPU strongly recommended; two-stage inference.
License	Apache 2.0 (free for commercial)	Apache 2.0 (business-friendly)	Non-commercial free; $25k/year for commercial use	Non-commercial free; commercial license needed via authors

Given all the information (Table 1. Comparison of different pose detection models), MediaPipe Pose is the frontrunner. It achieves a balance of sufficient accuracy, high speed, and robust tracking. In a clinical evaluation study, researchers explicitly chose MediaPipe after comparing it to OpenPose and AlphaPose, precisely because it gave the best compromise for real-world usage. They noted that while OpenPose was slightly more accurate at very strict settings, it failed to output many keypoints at those settings (low acceptance rate), whereas MediaPipe kept tracking almost all keypoints even in challenging conditions. This reliability is crucial - missing data could be problematic (e.g. if OpenPose loses track of a limb due to occlusion, the system might misinterpret it as that limb not moving, which could erroneously be seen as “freezing”). MediaPipe’s ability to handle occlusions and variations (due to learning-based holistic approach) and its one-pass pose detection (without needing external detector in most cases) simplify deployment, therefore, MediaPipe Pose will be used as the primary pose estimation engine.

2.7. Functional requirements

1. Pose detection and landmark extraction: the system shall detect human body pose keypoints from the live camera feed. For each person in view, it identifies key landmarks (e.g. head, shoulders, elbows, etc.) in either 2D or 3D coordinates. This forms the basis for analyzing body posture and movement patterns related to stress (such as restless movements or defensive body language).

2. Pose tracking over time: the software shall track individual poses across frames, maintaining a unique ID for each person. This means once a person is detected, their pose in subsequent frames is associated with the same person. Tracking enables the system to observe changes over time (e.g. a person’s pose becoming more slouched gradually) rather than treating each frame in isolation. It also prevents confusion when people move or swap positions.

3. Multi-person support: the system shall handle multiple people simultaneously in the camera frame. It will detect and track poses for each person present (up to a defined maximum, based on small group settings, such as 5 people). For each person, their pose and stress indicators are analyzed independently, and the system can potentially compare or aggregate results (for example, identifying overall group stress levels versus individual).

4. Emotion recognition (facial analysis): the system shall perform facial emotion recognition for each detected person. This involves detecting faces (likely using face bounding boxes) and classifying expressions or other facial features that correlate with stress. Emotions such as anxiety, frustration, or confusion could be detected via facial landmarks (e.g. furrowed brows, frowns). This functional requirement is critical to capture the “emotional” aspect of stress, complementing the physical pose cues.

5. Stress level inference: using the pose and emotion data, the software shall infer a stress level or indicator for each person. This could be a categorical output (e.g. “high stress” vs “low stress”) or a continuous score. The inference might be done through a predefined heuristic or an AI model trained to recognize stress from body language and facial cues. This function encapsulates the core AI decision making of the system.

2.8. Non-functional requirements

1. Performance: the system must perform efficiently. This includes meeting real-time processing demands (as noted, e.g. 20 FPS or higher with minimal latency). Latency from camera capture to output display should be low (ideally under 100ms). The system should also manage CPU/GPU load such that it does not freeze or stutter during operation - utilizing multi-threading or GPU offloading to maintain smooth performance. If multiple people are present, performance should degrade gracefully (e.g. slight FPS drop for each additional person, but not a complete breakdown).

2. Accuracy and precision: the pose detection and emotion recognition components should be accurate in their predictions. For pose, this means a high percentage of keypoints detected correctly (with low error) under various conditions (more than 80%). For emotion, it means the expression classification should match the person’s actual affect with a high true-positive rate.

3. Reliability and robustness: the system must be reliable during continuous use. It should not crash or hang even if running for long sessions (more than 1 hour). Robustness also means handling varying lighting conditions and environments - for example, it should work in both well-lit and moderately low-light rooms. If lighting changes or if a person partially leaves the frame and comes back, the system should adapt and continue tracking. Additionally, it should handle cases like occlusion (people obstructing each other) gracefully, perhaps with a slight drop in accuracy but quick recovery when occlusion ends.

4. Security and privacy: the system should ensure that any data (video frames, analysis results) is handled securely, especially if stored or transmitted. Since stress data can be sensitive, if there’s any logging, it should be protected. Also, the application should be protected from unauthorized access - for example, if it has a web interface or network features, it must have proper authentication.

2.9. Architecture

The class diagram summarizes the system's end-to-end architecture (Fig. 3. System class diagram). At the data layer, DatasetLoader reads posture and facial datasets, and FeatureAligner combines modalities into a session-aligned dataset. Feature extraction is split into BodyFeatureExtractor and FaceFeatureExtractor, which feeds data into a WindowAggregator that computes the window-level summaries used by the models. Preprocessing is handled by Imputer (missing-value handling) and Normalizer (participant-level or global normalization), while SequenceBuilder optionally constructs fixed-length temporal sequences for sequence models.

For training and evaluation, ModelTrainer fits the chosen model and writes it into a ModelBundle that stores the trained estimator, feature ordering, normalization stats, and metadata. Evaluator computes performance metrics and produces diagnostic plots. At inference time, WebcamCapture supplies live frames to the feature extractors, the InferenceEngine maintains rolling windows, builds feature vectors, applies imputation and normalization, and produces predictions using the stored ModelBundle. This structure keeps the data pipeline consistent across training and inference while allowing different model types to plug into the same preprocessing and feature logic.

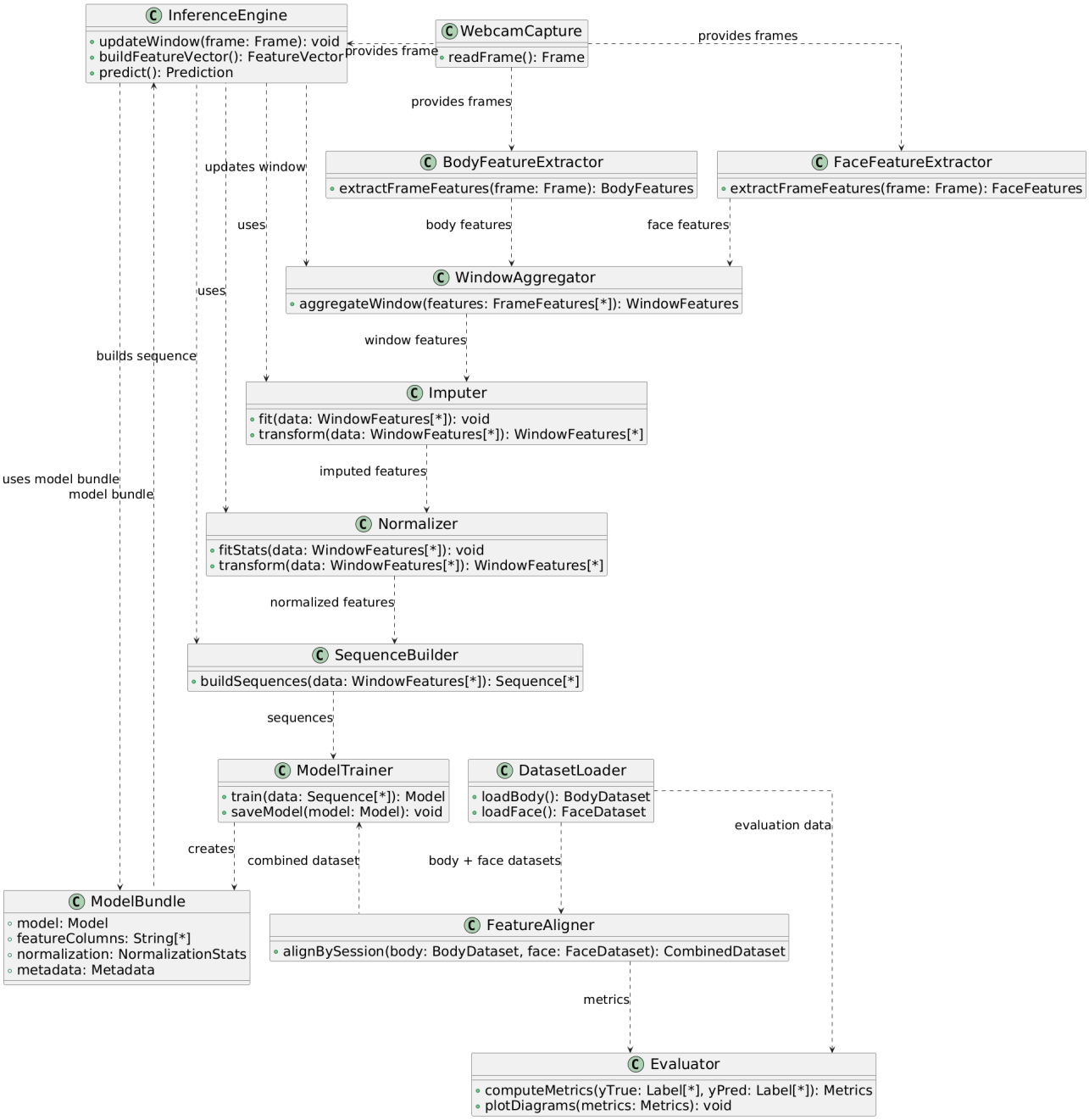

Fig. 3. System class diagram

The sequence diagrams illustrates the real-time flow of the live stress monitoring system from raw webcam input to final stress prediction and visualization.

Training: as seen in training sequence diagram (Fig. 4. System training sequence diagram), user triggers dataset loading. The DatasetLoader reads body and face data, and the FeatureAligner aligns sessions by participant and condition. The aligned data is aggregated into minute windows, passed through Imputer and Normalizer, and optionally converted into sequences by SequenceBuilder. The ModelTrainer fits the model, stores it in a ModelBundle, and hands evaluation results to Evaluator, which reports metrics and plots back to the user.

Inference: as inference sequence diagram (Fig. 5. System inference sequence diagram) shows, the user starts the webcam through WebcamCapture. Frames flow to BodyFeatureExtractor and FaceFeatureExtractor, then to WindowAggregator for rolling feature aggregation. The aggregated features are imputed, normalized, and optionally turned into sequences before the InferenceEngine queries the ModelBundle and outputs stress probability to the researcher. This loop repeats continuously during live use.

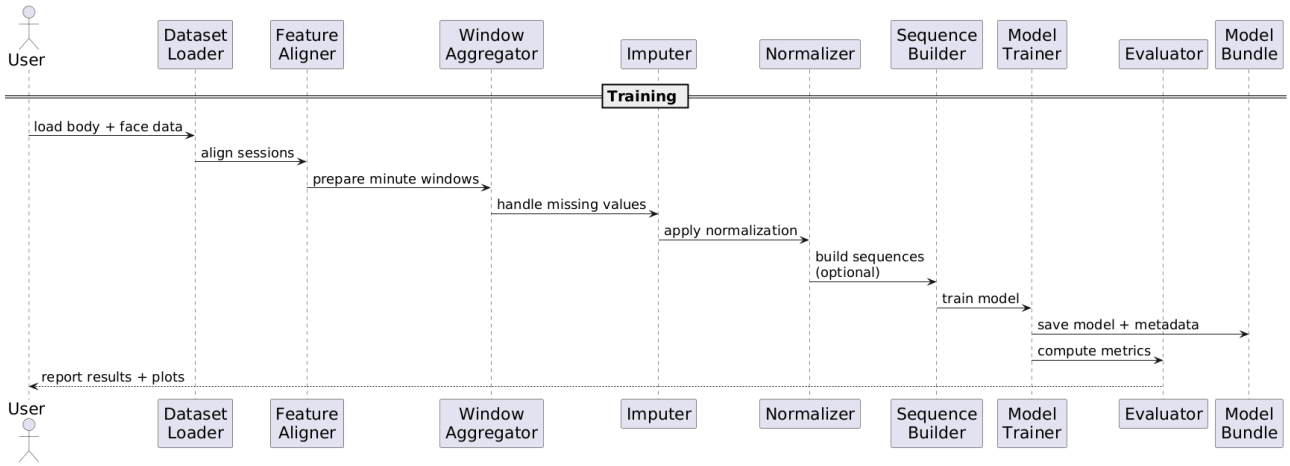

Fig. 4. System training sequence diagram

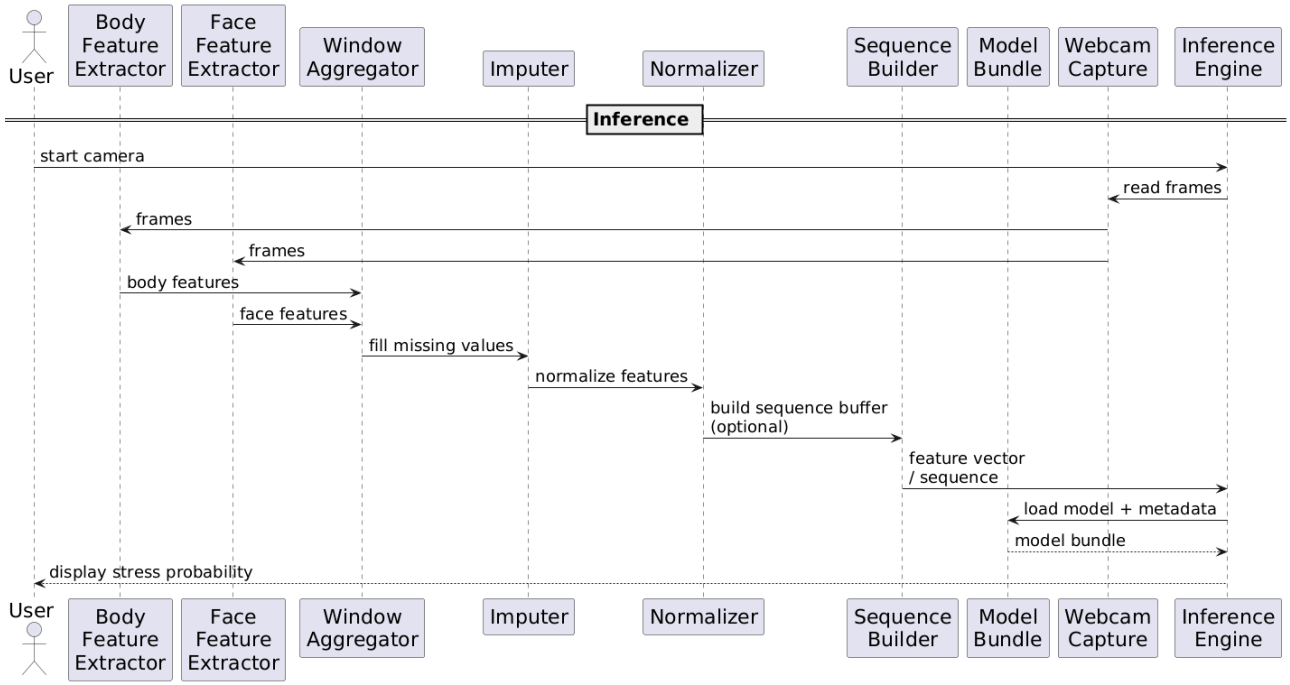

Fig. 5. System inference sequence diagram

The use case diagram shows the system's high-level interactions for both training and inference (Fig. 6. System use-case diagram). A Researcher performs the offline workflow: loading datasets, aligning modalities, preprocessing features, training the model, evaluating results, and saving the model bundle. An Operator runs the live workflow: starting inference and viewing stress probability outputs.

Both actors can configure parameters (e.g. dataset mode, window length, model settings). The includes show core dependencies: dataset loading leads to alignment and preprocessing, model training includes evaluation and saving, live inference includes display of stress probability.

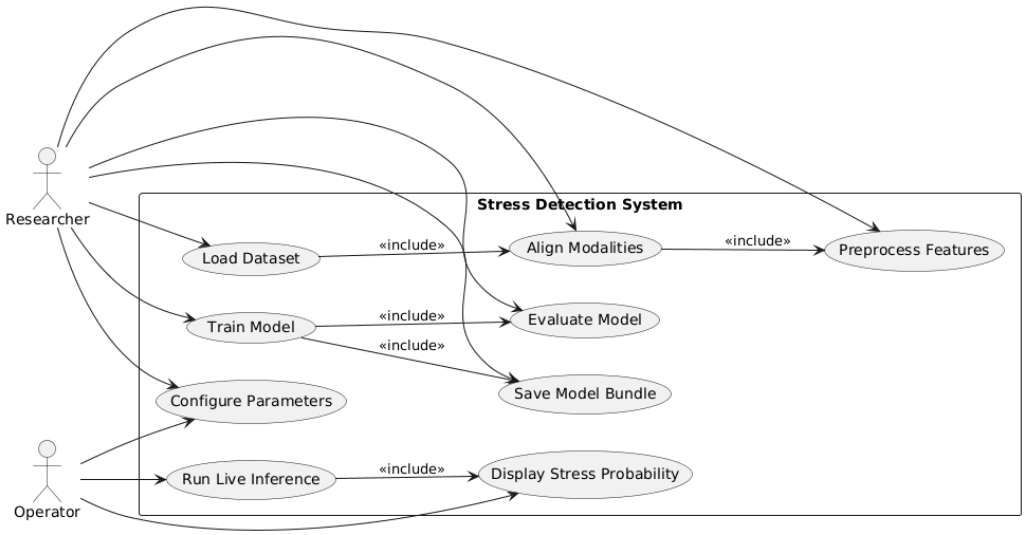

Fig. 6. System use-case diagram

The state diagram shows an abstract description of system's behavior (Fig. 7. System state diagram). From the initial Idle state, the training flow begins by loading the datasets, aligning body and face sessions, preprocessing (imputation and normalization), and optionally building sequences. The system then enters the Training state (model fitting), transitions to Evaluating for metrics and plots, saves the model bundle (ModelSaved), and returns to Idle ready for inference.

The inference flow also starts from Idle, moving to CameraReady when the webcam is started. The system cycles through Capturing frames, Windowing (rolling aggregation), feature vector construction, and Predicting to produce stress probability outputs, which are displayed in Displaying. It loops continuously between Displaying and Capturing until the webcam is stopped, at which point it returns to Idle.

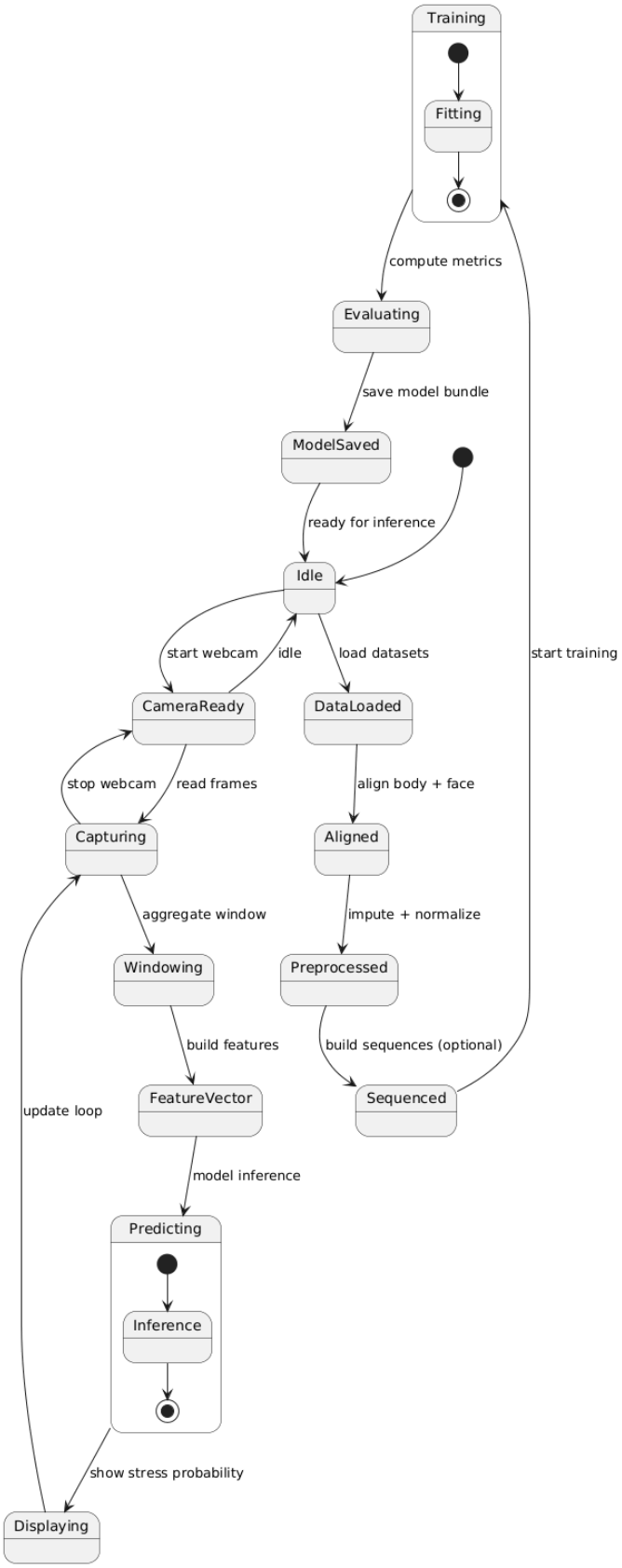

Fig. 7. System state diagram

3. Multimodal Stress Recognition Framework

3.1. Training data

This research uses and compares two datasets - SWELL-KW and StressID dataset, the latter required an official letter from university to be allowed to use it.

3.1.1. SWELL-KW dataset

Dataset description

SWELL-KW dataset is a rich multimodal dataset designed to study stress, workload and user behavior during typical work tasks. It was collected from 25 participants performing realistic office activities under three conditions - neutral, time pressure and interruptions. The dataset includes synchronized recordings of computer interaction, physiological signals, facial expressions and body posture. [25]

Dataset preparation

This dataset was prepared by loading the body-posture CSV and facial-expression TXT files, removing the extra header row in the body file, and converting all feature columns to numeric while keeping participant identifier (PP) and timestamp only as identifiers. This was done to ensure the inputs are clean, numeric, and consistent, because the raw files mix text headers with numeric values, therefore models cannot train on mixed types.

For the combined body and face dataset, the two sources had different row counts per participant and condition, so alignment was performed by mapping rows across time and then fusing the features into one minute-level record. This was required to avoid dropping data or mismatching minutes. The condition label C was then mapped to the target label (binary stress vs non-stress, or 3-class) so the model objective is explicit. Missing values were filled with medians to prevent NaNs during training, and features were normalized per participant using training-only statistics, with a global fallback for the held-out subject. This normalization reduces subject-specific scale differences and improves leave-one-participant-out generalization without leaking test information.

3.1.2. StressID dataset

Dataset description

StressID dataset is also a multimodal dataset designed to study stress identification. It contains RGB facial video, audio and physiological signals (ECG, EDA, Respiration). Different stress-inducing stimuli are used: emotional video-clips, cognitive tasks and public speaking. The total dataset consists of recordings from 65 participants that performed 11 tasks. The experimental set-up ensures synchronized, high-quality, and low noise data. [26]

Dataset preparation

The second dataset was prepared by using only the video modality and the label/meta CSV files, since audio and physiological signals were excluded for a body and face only model. Each video was processed frame-by-frame with the body and face helper scripts to extract posture and facial

expression features (Fig. 8. Example frames from StressID dataset). Frames were sampled at a fixed FPS and aggregated into fixed-length windows (30-second windows with a stride), producing per-window averages and standard deviations. This windowing step was required to turn variable-length videos into uniform feature rows suitable for models.

Fig. 8. Example frames from StressID dataset [26]

Labels were loaded from the provided StressID label files, cleaned, and matched to videos by subject/task/video identifiers. Unlabeled videos (e.g. baselines) were excluded by default to avoid training on unknown targets. Feature columns were converted to numeric values, missing values were imputed (median) to prevent NaN failures, and per-subject normalization was applied during training to reduce inter-participant scale differences without leaking test-subject information. The final output of preparation was a single feature CSV with metadata columns (subject/task/video/window) plus body and face features and label columns, ready for leave-on-participant-out training.

3.2. Stress detection models

HistGradientBoosting (HGB), Support Vector Machines with an RBF kernel (SVM-RBF), and Bidirectional Long Short-Term Memory (BiLSTM) models were used to implement the stress detection pipeline. HGB is strong on tabular engineered features and captures non-linear interactions robustly, SVM-RBF provides a margin-based non-linear classifier that performs well on smaller, high-dimensional datasets, and BiLSTM is suited for temporal modeling, capturing short-term dynamics in posture and facial features that static models can miss.

3.2.1. HistGradientBoosting

HistGradientBoosting is currently one of the most powerful algorithms available in Scikit-Learn for structured (tabular) data. It is an ensemble method based on decision trees, but with a critical architectural optimization inspired by LightGBM (Fig. 9. HistGradientBoosting architecture).

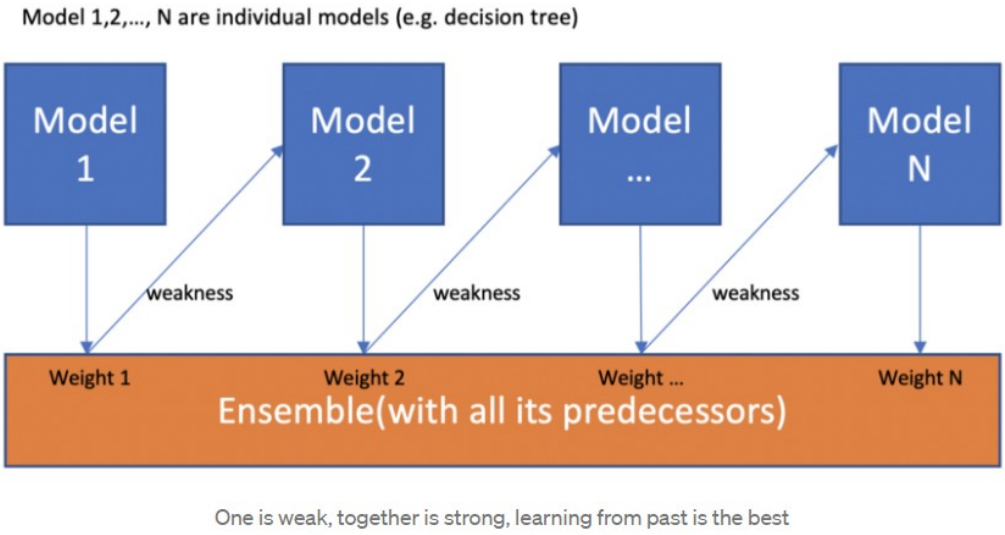

Fig. 9. HistGradientBoosting architecture [27]

This model builds a series of "weak learners" (decision trees) sequentially, where each new tree attempts to correct the errors of the previous ones. However, its primary innovation lies in how it handles feature splitting. Instead of sorting every individual data point to find the optimal cut for a decision tree node (process that is computationally expensive), the algorithm discretizes continuous input features into integer-valued "bins" (typically up to 255). By constructing histograms of these bins, the model can locate optimal split points significantly faster, reducing the complexity from O(N×logN) to O(N), where N is the number of samples. [28]

For stress detection on both SWELL-KW and StressID, HistGradientBoosting is well-suited to the quality and reliability of video-based sensors because it natively handles missing values. In SWELL-KW (Kinect posture + FaceReader) and in StressID (body/face from video), dropouts are common when a subject turns away, covers the face, or landmarks fail, producing NaNs. Unlike SVMs or standard neural networks, which often require explicit imputation or can fail on missing data, HistGradientBoosting learns how to route missing values during training, effectively treating the absence of a signal as a potentially informative feature rather than an error.

3.2.2. SVM Kernel RBF

Support Vector Machine (SVM) is a powerful supervised learning algorithm that approaches classification as a geometric optimization problem rather than a logical rule-based one (Fig. 10. SVM architecture). Its primary goal is to find the optimal "hyperplane" - a decision boundary in N-dimensional space that distinctly separates the data points of one class from another.

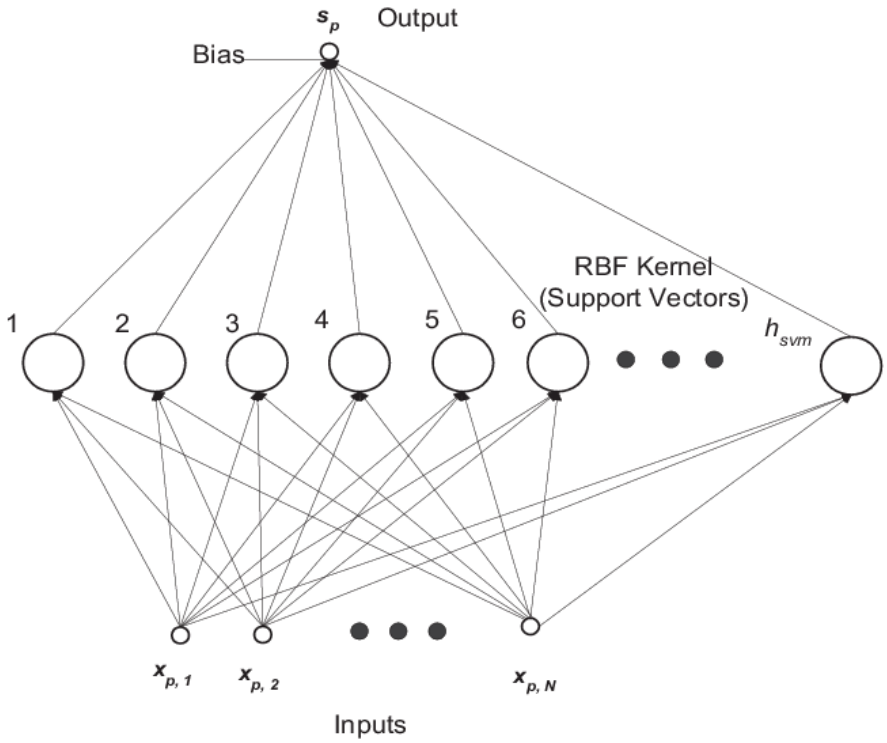

Fig. 10. SVM architecture [29]

The model's algorithm attempts to maximize the margin (distance between the hyperplane and the nearest data points from either class) to ensure the model is as generalizable as possible. Critically, for complex physiological data that cannot be separated by a simple straight line, SVM utilizes a technique called the "Kernel Trick" (commonly the Radial Basis Function or RBF). This mathematical transformation projects the input data into a higher-dimensional space where the relationship between stress features becomes linearly separable, allowing the model to capture complex, non-linear patterns without heavy computational costs. [29]

For both SWELL-KW and StressID, an SVM with an RBF kernel is a strong fit because the feature sets are wide and non-linear, combining many posture measurements with facial expression signals. These high-dimensional vectors can be challenging for simpler models, but SVMs remain effective in such spaces by maximizing the margin and using the RBF kernel to model complex boundaries. This makes it well suited to capture subtle combinations of cues, such as eyebrow tension, gaze direction, and shoulder posture, that together indicate stress even when no single feature is decisive.

3.2.3. BiLSTM

Bidirectional Long Short-Term Memory (BiLSTM) is a deep learning architecture that fundamentally changes how a model understands time. While a standard LSTM reads data linearly from past to future, a BiLSTM processes the input sequence in two directions simultaneously (Fig. 11. BiLSTM architecture).

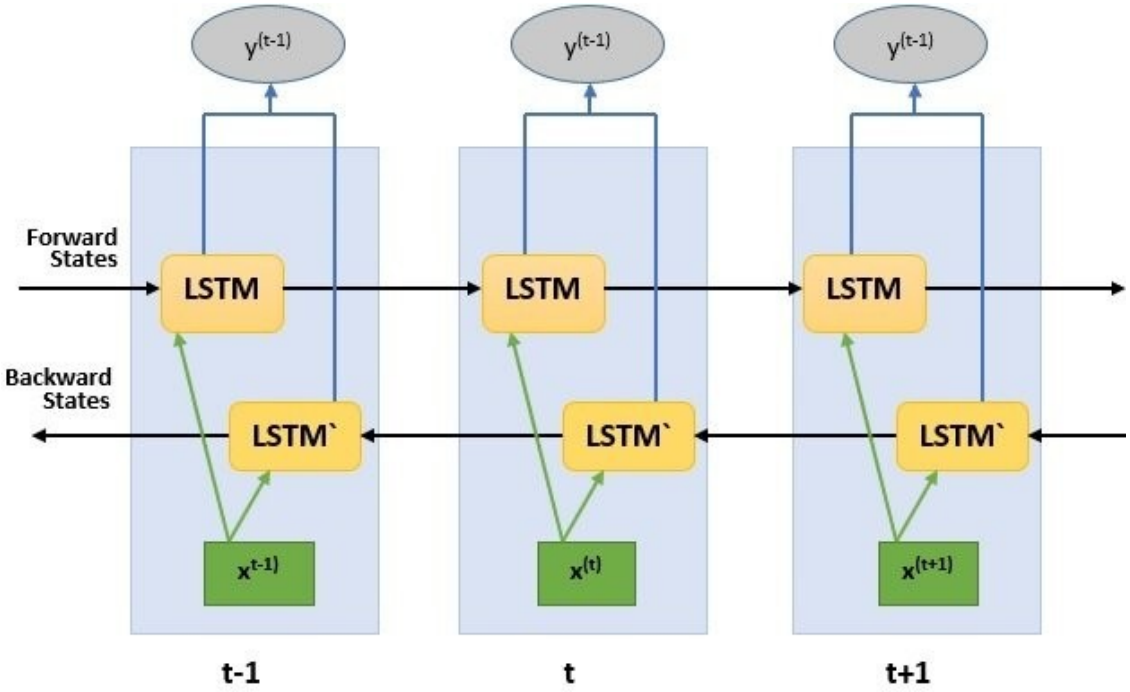

Fig. 11. BiLSTM architecture [30]

The model maintains two separate hidden layers: one that reads the sequence forward (from time t=0 to t=N) and another that reads it backward (from t=N to t=0). At every single time step, the model combines the insights from both the past and the future before making a prediction. This effectively gives the model hindsight, it doesn't just know what led up to a specific moment, but also how that moment resolved. This dual-context approach allows it to capture complex, long term dependencies in time-series data that a unidirectional model might miss. [30]

For both SWELL-KW and StressID, the BiLSTM is included to model stress as a temporal pattern rather than a single instant. Stress behaviors often unfold over several windows, so a sudden movement or facial change is only meaningful in context. A bidirectional LSTM can incorporate both preceding and subsequent windows, allowing it to distinguish a brief, harmless motion (e.g. a stretch followed by relaxation) from a sustained stress response (e.g. continued fidgeting or rigid posture). This temporal view complements the static models and makes it better suited for capturing stress dynamics across time.

3.3. Feature extraction

Video frames were processed with custom feature extraction scripts based on MediaPipe for retrieving body posture and face landmarks data.

3.3.1. Pose estimation

SWELL-KW provides 3D Kinect joint coordinates, while StressID relies on videos, therefore this extractor standardizes both sources into a consistent 3D-style representation by deriving 3D pose from the available inputs (true 3D for Kinect, estimated 3D from video-based landmarks) (Fig. 12. 3D visualization and 2D keypoints of a person). This alignment ensures that downstream models operate on comparable posture features across datasets.

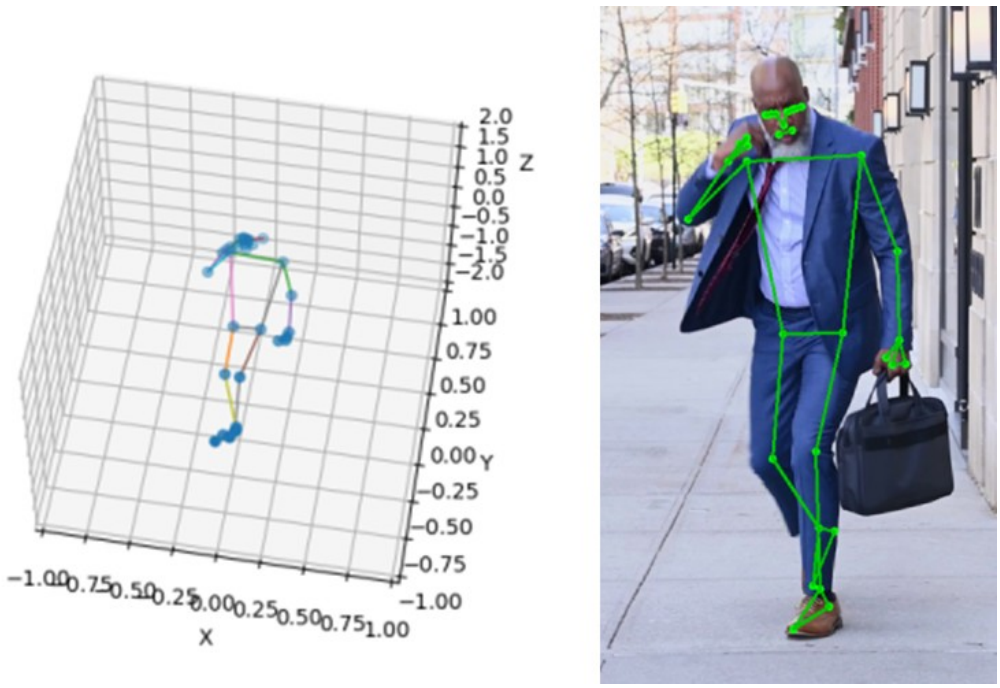

Fig. 12. 3D visualization and 2D keypoints of a person

Table 2. Example of pose estimation values

avgDepth(avg)	2588.577
avgDepth(stdv)	161.244
rightShoulderAngle(avg)	124.751
rightShoulderAngle(stdv)	12.924
…	...

This is a real-time pipeline that uses a webcam to run MediaPipe Pose per frame and then produces Kinect-style posture features so the same representation can be used for both SWELL-KW and StressID (Table 2. Example of pose estimation values), which came to a total of around 94 body posture feature columns per dataset. It follows four stages: first, frames are captured and optionally downscaled for speed, secondly, MediaPipe Pose is applied to obtain landmarks, then post-processing smooths jitter and when necessary, applies a shoulder-width scale heuristic to approximate meter-level coordinates, and lastly, feature synthesis reconstructs pseudo-Kinect joints (e.g. HipCenter/Spine/ShoulderCenter midpoints, head proxy, hand points) and computes segment-to-segment and segment-to-plane/axis angles using SWELL-KW naming conventions, including a best-effort depth estimate from shoulder pixel width. A rolling window of per-frame features is maintained to output windowed averages and standard deviations aligned with the SWELL-KW avg/stdv columns, while optionally rendering the filtered skeleton for visual inspection.

3.3.2. Face landmarks estimation

Face landmarks detection didn’t need big changes which was the case with pose estimation, this implementation was only adjusted to SWELL-KW dataset features.

Fig. 13. Face landmarks detection

The face landmark extractor uses MediaPipe Face Mesh (Fig. 13. Face landmarks detection) to track face in real time and derives a SWELL-style feature row with lightweight geometric heuristics, and the same module is reused for both SWELL-KW and StressID to keep the facial feature space consistent. It computes head orientation, mouth openness, left/right eye closure from eye-aspect ratio, eyebrow raise/lower scores from brow-to-lid distances, and gaze direction. Overall, about 41 face expression feature columns have been extracted from each dataset.

3.3.3. Summary

HistGradientBoosting, SVM Kernel RBF and BiLSTM represent three different approaches to supervised stress detection. HGB is well suited for tabular body pose and facial expression features because it can model non-linear relationships, handle feature interactions and remain robust to noisy or partially missing inputs. SVM-RBF is useful for high dimensional feature spaces because it learns a non-linear margin-based decision boundary, allowing subtle combinations of posture and facial cues to separate stress from non-stress. BiLSTM is designed for sequential data and can model how behavior changes over time, which is important because stress is often reflected through sustained or evolving patterns rather than a single isolated frame or window. All of them were trained on roughly 135 combined body and face numeric feature columns for SWELL-KW dataset and 137 extracted body and face feature columns for StressID dataset.

4. Evaluation of Stress Detection Models and Results

4.1. Training results on SWELL-KW dataset

4.1.1. HistGradientBoosting

This implementation builds a stress-detection model using a HistGradientBoostingClassifier trained on SWELL-KW minute-level features. The system fuses body-posture (Kinect-style) and facial-expression modalities. Training is evaluated with leave-one-participant-out (LOPO) validation to measure cross-subject generalization, and inference uses webcam input with MediaPipe to compute the same SWELL-KW-style features in real time.

Because the body and face timestamps do not align, the fusion is performed by session-level alignment per participant and condition. With each session, body rows are used as the reference timeline, face rows are mapped to body rows by normalized index (first ↔ first, last ↔ last). This preserves session structure without relying on mismatched timestamps, yielding a combined row for each body minute.

Tested hyperparameters:

- Model 1: learning_rate=0.05, max_depth=3, max_leaf_nodes=31, min_samples_leaf=50, l2_regularization=1.0. This combination introduces structural constraints. Restricting depth to 3 and increasing the minimum samples per leaf forces the model to learn broader, more generalized rules. L2 regularization penalizes extreme parameter weights.
- Model 2: learning_rate=0.03, max_depth=2, max_leaf_nodes=15, min_samples_leaf=100, l2_regularization=1.0. This is a heavily constrained, highly conservative model. Tree depth is limited to 2 (essentially functioning as decision stumps), leaf nodes are halved, and leaves require a massive 100 samples to form. The lower learning rate (0.03) ensures the model updates slowly and smoothly.
- Model 3: max_depth=None, max_leaf_nodes=31, min_samples_leaf=20, l2_regularization=0.0. By leaving depth unconstrained and using no L2 regularization, this model has the highest capacity to learn complex, non-linear patterns. However, it is the most prone to overfitting.

Table 3. HGB training results on SWELL-KW dataset

Metric	Model 1	Model 2	Model 3
Accuracy	~ 61%	**~ 63%**	~ 61%
Balanced accuracy	~ 55%	**~ 56%**	**~ 56%**
Precision	~ 64%	~ 64%	**~ 65%**
Recall	~ 86%	**~ 92%**	~ 82%
F1	~ 72%	**~ 74%**	~ 71%

In evaluating the HistGradientBoosting models via Leave-One-Person-Out (LOPO) cross-validation on the SWELL-KW dataset, a clear trend emerged regarding model complexity. From the results (Table 3. HGB training results on SWELL-KW dataset) it is seen that the unconstrained baseline

model (Model 3) exhibited signs of overfitting to subject-specific physiological traits, yielding the lowest F1-score (0.7071). Introducing strict regularization, specifically by limiting max_depth to 2, increasing min_samples_leaf to 100, applying L2 regularization, and reducing the learning rate to 0.03 substantially improved generalizability to unseen subjects. This strongly constrained configuration achieved the highest overall performance (Accuracy: 0.6276, F1-Score: 0.7410). Notably, this robust generalization manifested primarily through a significant increase in Recall (0.9249), suggesting that simpler, highly regularized models (Model 2) are better equipped to identify universal physiological markers of stress across different individuals with a minor trade-off in precision (Fig. 14. Best HistGradientBoosting training results on SWELL-KW dataset).

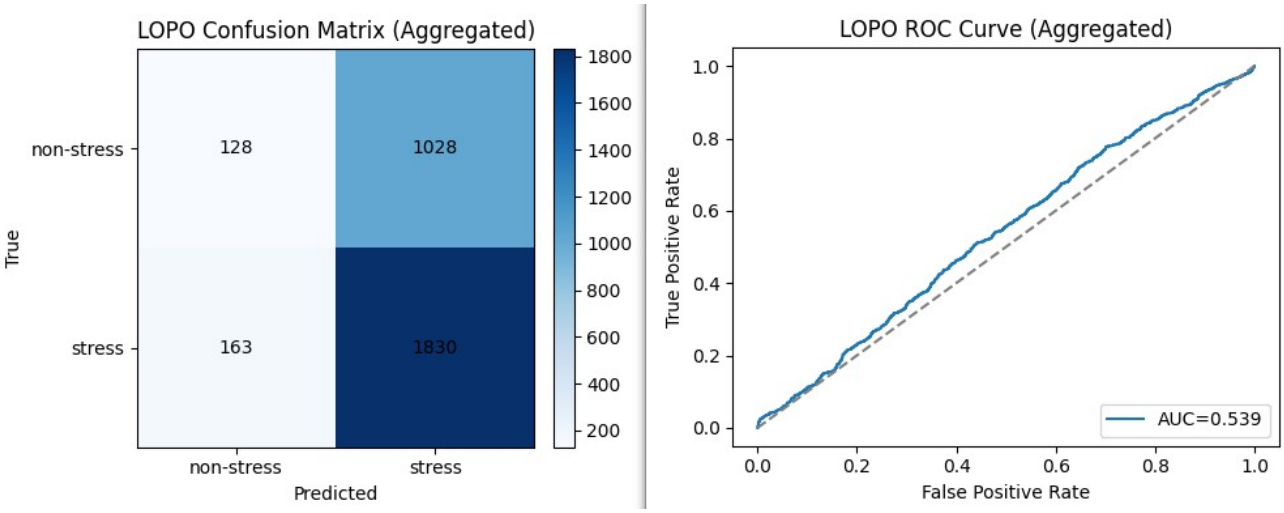

Fig. 14. Best HistGradientBoosting training results on SWELL-KW dataset

4.1.2. SVM Kernel RBF

This implementation uses a Support Vector Machine with an RBF kernel to classify stress vs non-stress using SWELL-KW minute-level features. The pipeline mirrors the HGB setup: it supports body, face, or combined modalities, applies participant-level normalization, evaluates with leave-one-participant-out validation and saves a model bundle for inference.

For the SVM experiments, the same body-face fusion strategy as the HGB setup was retained. Since the body and face streams are not time-aligned, fusion is done at the session level per participant and condition: body rows define the reference timeline, and face rows are mapped by normalized index to create a combined minute-level record for each body row.

Tested hyperparameters:

- Model 1: C=3, gamma=0.01. The high C heavily penalizes training errors, trying to fit the subjects in the training set perfectly.
- Model 2: C=0.3, gamma=0.05. The low C applies strong regularization, telling the model to prioritize a broad margin over getting every training point right.
- Model 3: C=1.0, gamma=scale (automatically adjusts based on data variance). This is a balanced, moderate approach.

Table 4. SVM Kernel RBF training results on SWELL-KW dataset

Metric	Model 1	Model 2	Model 3
Accuracy	~ 59%	**~ 63%**	~ 60%
Balanced accuracy	~ 55%	**~ 56%**	~ 55%
Precision	**~ 65%**	~ 63%	~ 64%
Recall	~ 78%	**~ 100%**	~ 82%
F1	~ 69%	**~ 76%**	~ 70%

Analysis of the RBF-SVM models on the SWELL-KW dataset under LOPO cross-validation revealed critical insights into the risk of decision boundary collapse (Table 4. SVM Kernel RBF training results on SWELL-KW dataset). While the strongly regularized model (Model 2) showed the highest nominal F1-score (0.7597), this was entirely driven by an anomalous, perfect Recall (~100%). This indicates that the low C penalty caused the model to over-generalize, effectively defaulting to predicting the positive class to maximize the margin, acting essentially as a majority-class classifier rather than learning meaningful physiological distinctions. Conversely, the model with weak regularization (Model 1) suffered from overfitting to the training subjects, yielding the lowest Accuracy (0.5880). Ultimately, the baseline configuration (Model 3) demonstrated the most robust and authentic generalization, avoiding the pitfalls of both overfitting to individual subject nuances and collapsing under excessive regularization (Fig. 15. Best SVM Kernel RBF training results on SWELL-KW dataset).

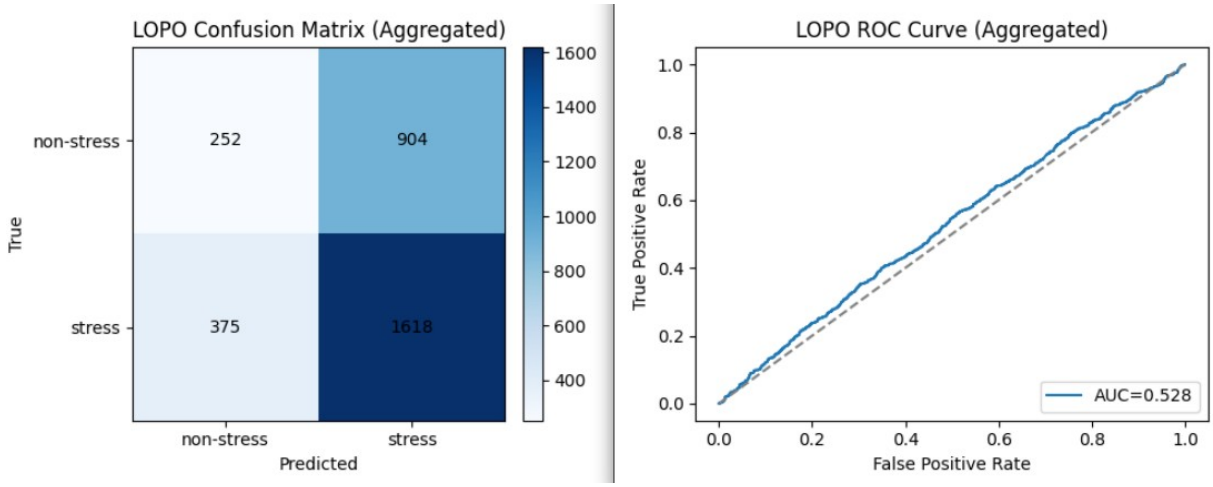

Fig. 15. Best SVM Kernel RBF training results on SWELL-KW dataset

4.1.3. BiLSTM

The BiLSTM model extends the stress-detection pipeline by modeling temporal dynamics across multiple minutes of multimodal features. Unlike the HGB and SVM models, which treat each minute independently, the BiLSTM consumes fixed-length sequences of consecutive minutes, allowing it to capture short-term temporal patterns in posture and facial behavior. It uses two stacked bidirectional

LSTM layers with dropout applied after each block, followed by a sigmoid output for binary stress. Training uses the Adam optimizer with binary cross-entropy loss and early stopping to prevent overfitting. This implementation uses the same body-face fusion strategy that is applied in the HGB/SVM pipelines.

Tested hyperparameters:

- Model 1: seq-len=2, lstm-units=32, dropout=0.5. This model looks at an extremely short time window (just 2 steps). To prevent overfitting on such limited context, it uses a small hidden state (32 units) and aggressively drops out 50% of its connections during training.
- Model 2: seq-len=3, lstm-units=64, dropout=0.3. This model slightly increases the time window (3 steps) and doubles the learning capacity (64 units), while reducing the dropout penalty.
- Model 3: seq-len=10, lstm-units=64, dropout=0.3. This model keeps the higher capacity of Model 2 but drastically expands its temporal view, looking at 10 consecutive time steps before making a prediction.

Table 5. BiLSTM training results on SWELL-KW dataset

Metric	**Model 1**	**Model 2**	**Model 3**
Accuracy	~ 61%	~ 57%	**~ 62%**
Balanced accuracy	~ 56%	~ 55%	**~ 60%**
Precision	~ 63%	~ 64%	**~ 67%**
Recall	**~ 92%**	74%	~ 69%
F1	~ 73%	~ 66%	**~ 77%**

In evaluation of the BiLSTM architectures using LOPO cross-validation on the SWELL-KW dataset, the length of the temporal sequence (seq-len) emerged as the most critical hyperparameter. Models restricted to short temporal windows (Model 1 and Model 2) struggled to generalize, with the heavily regularized configuration exhibiting a strong bias toward the positive class, resulting in an artificially inflated Recall (0.9194) but poor Balanced Accuracy (0.5551) (Table 5. BiLSTM training results on SWELL-KW dataset). However, expanding the sequence length to 10 while maintaining moderate capacity (Model 3) yielded the most robust and balanced model. This configuration achieved the highest overall performance (F1-score: ~0.77, Balanced Accuracy: ~0.60), demonstrating that sufficient temporal context is essential for recurrent networks to capture the genuine, unfolding physiological markers of stress rather than overfitting to transient noise (Fig. 16. BiLSTM training results on SWELL-KW dataset).

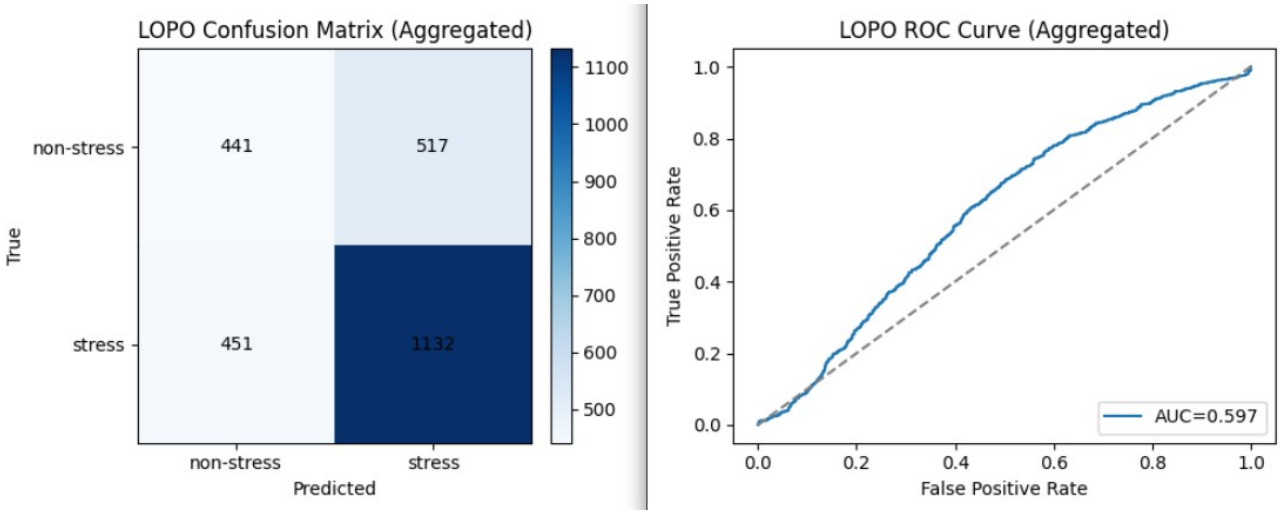

Fig. 16. BiLSTM training results on SWELL-KW dataset

4.1.4. Issues encountered and their solutions

During models training and their inference testing on SWELL-KW dataset, various problems had appeared, their descriptions and solutions are described as follows:

1. Timestamp mismatch between body and face datasets: no exact timestamp overlap between modalities, so direct merges yielded zero matches. To tackle this problem, session-level alignment by normalized index per participant/condition was implemented, preserving relative time order without relying on timestamps.
2. Low initial performance with C3-only stress mapping: treating only C3 as “stress” produced near-chance LOPO results. Re-mapping stress to C2+C3, which is aligned with SWELL-KW protocol improved performance substantially.
3. Participant variability hurting generalization: large scale differences across participants led to unstable LOPO performance. Adding participant-level z-score normalization with global fallback improved stability and F1.
4. Flat probability output at inference: the SVM often produced near-constant probabilities during live inference despite changing input features. Debugging showed a nearly constant decision function, indicating the RBF decision surface was flat in the observed input region. This is a model sensitivity issue rather than an inference bug and suggests that further tuning or additional feature scaling adjustments would be required for more responsive real-time output.
5. Inference delayed excessively: the initial inference design added one sequence step only at the prediction interval, which delayed the first prediction by several minutes. This was resolved by decoupling sequence updates from prediction updates.

4.1.5. Models comparison

Overall, HGB and SVM RBF are nearly tied on SWELL-KW and both show high recall with modest balanced accuracy, indicating a tendency to favor the stress class, while BiLSTM achieves the best balanced accuracy and F1, but at the cost of lower recall. In practical terms, HGB/SVM are stronger when maximizing stress detection is the priority, whereas BiLSTM is more balanced between stress and non-stress predictions.

Table 6. Best models training results comparison on SWELL-KW dataset

Metric	HistGradientBoosting	SVM Kernel RBF	BiLSTM
Accuracy	**~ 63%**	~ 60%	~ 62%
Balanced Accuracy	~ 56%	~ 55%	**~ 60%**
Precision	~ 64%	~ 64%	**~ 67%**
Recall	**~ 92%**	~ 82%	~ 69%
F1	~ 74%	~ 70%	**~ 77%**

BiLSTM is the most robust model out of the three. Generally, it predicts both states in the most stable way, while the other two are more towards to a single state (Table 6. Best models training results comparison on SWELL-KW dataset). These, on SWELL-KW dataset trained models likely achieved only low-modest performance because the dataset is relatively small and highly variable across participants, while LOPO evaluation is strict and punishes subject-specific cues. The minute-level labels may not align perfectly with short-term posture or facial changes, introducing label noise. In the combined setting, body-face alignment is approximate rather than time-synchronized, which can blur multimodal signals. Kinect and FaceReader data also contain dropouts and measurement noise, and stress vs non-stress classes are not perfectly balanced, which can skew decision boundaries.

4.2. Training results on StressID dataset

4.2.1. HistGradientBoosting

This implementation is the same HGB model, but now trained on StressID video features extracted from body posture and facial expressions. It uses a HistGradientBoostingClassifier and is evaluated with leave-one-subject-out validation to measure generalization to unseen participants. The pipeline mirrors the SWELL-KW setup but adapts to StressID labels and video-based inputs.

StressID provides task-level labels for each subject and task, along with video recordings. A custom extractor processes each video, samples frames at 3 FPS and computes pose and face based features using MediaPipe, which are then aggregated into fixed windows (default 30s with 15s stride) to match the clip-level labels while producing multiple samples per video.

Tested hyperparameters:

- Model 1: max_depth=3, min_samples_leaf=50, l2=1.0, learning_rate=0.05. A structurally constrained model.
- Model 2: max_depth=2, min_samples_leaf=100, l2=1.0, learning rate=0.03. Highly constrained, acting as an ensemble of very generalized, weak learners.
- Model 3: nnconstrained depth, standard learning rate (0.1), no L2 penalty. High capacity, high risk of overfitting.

Table 7. HGB training results on StressID dataset

Metric	Model 1	Model 2	Model 3
Accuracy	~ 71%	**~ 72%**	~ 69%
Balanced accuracy	~ 73%	**~ 74%**	~ 71%

Metric	Model 1	Model 2	Model 3
Precision	~ 67%	**~ 68%**	~ 65%
Recall	~ 52%	**~ 53%**	~ 52%
F1	~ 52%	**~ 54%**	~ 51%

As seen in training results (Table 7. HGB training results on StressID dataset), applying the HistGradientBoosting configurations to the StressID dataset under LOPO cross-validation highlighted distinct challenges inherent to the dataset. Consistent with prior experiments, the heavily regularized model (Model 2) yielded the best overall generalization, outperforming the unconstrained baseline across all metrics (Balanced Accuracy: 0.7412, F1-Score: 0.5384) (Fig. 17. Best HistGradientBoosting training results on StressID dataset). However, the StressID results are characterized by extreme inter-subject variability, evidenced by massive standard deviations in Precision (±0.3091) and Recall (±0.3242). Furthermore, a divergence between high Balanced Accuracy and low F1-scores suggests that the models adopted a conservative prediction strategy, successfully identifying baseline physiological states but struggling to consistently detect stress events across different individuals.

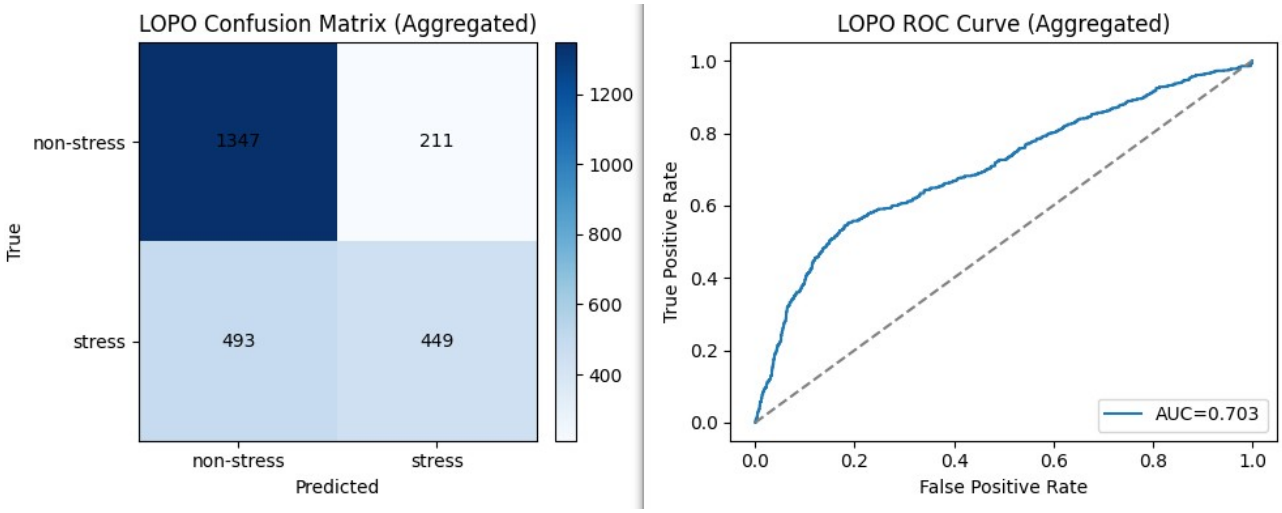

Fig. 17. Best HistGradientBoosting training results on StressID dataset

4.2.2. SVM Kernel RBF

This implementation uses RBF-kernel Support Vector Machine on StressID dataset. For the SVM experiments, the same StressID preprocessing pipeline used for HGB was reused (3 FPS, MediaPipe body and face extraction with SWELL-KW style features and fixed-window aggregation), so the SVM model operates on the identical windowed feature table without additional data preparation.

Tested hyperparameters:

- Model 1: C=0.3, gamma=0.05. This model prioritizes a wide, generalized margin over correctly classifying every training point.
- Model 2: C=3, gamma=0.01. This model penalizes misclassifications heavily, trying to strictly separate the classes in the training data.

– Model 3: C=1.0, gamma=scale. The balanced approach, letting the heuristic adapt to the data's variance.

Table 8. SVM Kernel RBF training results on StressID dataset

Metric	Model 1	Model 2	Model 3
Accuracy	~ 62%	~ 66%	**~ 70%**
Balanced accuracy	~ 55%	~ 67%	**~ 72%**
Precision	0%	~ 57%	**~ 63%**
Recall	0%	~ 51%	**~ 52%**
F1	0%	~ 49%	**~ 52%**

Evaluation of the RBF-SVM models on the StressID dataset exposed a profound sensitivity to margin hyperparameter tuning, resulting in catastrophic model collapse under strong regularization (Table 8. SVM Kernel RBF training results on StressID dataset). Specifically, the configuration utilizing a soft margin (Model 1) completely failed to generalize across LOPO folds, yielding a Precision, Recall, and F1-score of 0.0000. This indicates that the SVM, unable to identify a cohesive cross-subject stress boundary amid the dataset's high physiological variance, collapsed into a majority-class predictor (baseline state). In contrast, the baseline configuration (Model 3) provided the necessary balance between margin hardness and boundary flexibility, achieving the highest overall performance (Balanced Accuracy: ~0.72%, F1-Score: ~0.52%) (Fig. 18. Best SVM Kernel RBF training results on StressID dataset). These results starkly highlight that hyperparameter configurations which cause over-prediction of stress in one dataset (e.g. SWELL-KW) can cause a complete inversion to under-prediction in another with differing variance profiles.

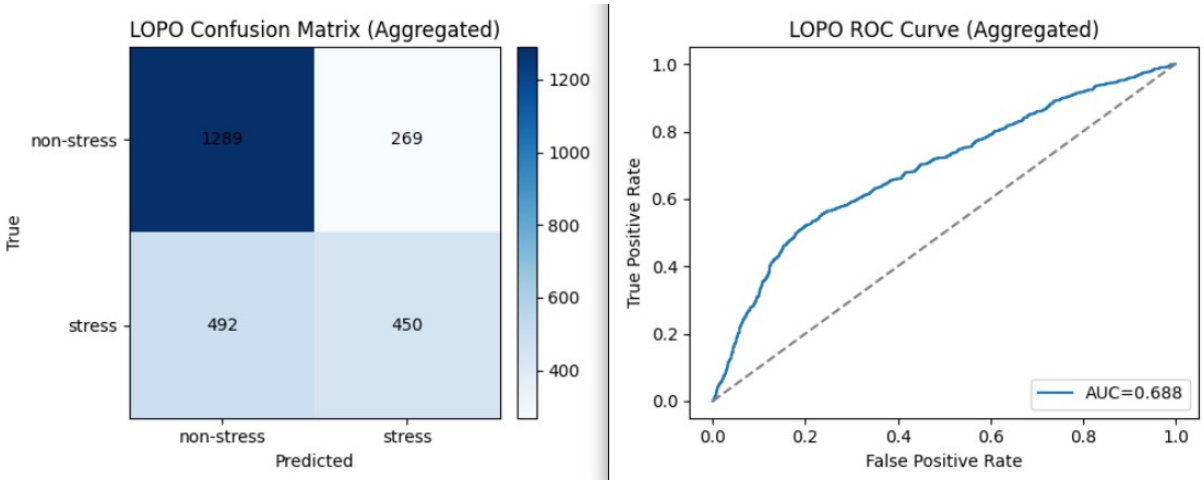

Fig. 18. Best SVM Kernel RBF training results on StressID dataset

4.2.3. BiLSTM

The implementation trains the BiLSTM model on synchronized body pose and facial feature windows extracted from StressID dataset. Like previously, features are first aggregated at the window level from sampled frames, then transformed into temporal sequences with a configurable length and stride. Each sequence is built within a (subject, task, video) segment to preserve temporal continuity and

avoid mixing subjects or tasks. The model architecture is made of a two-layer BiLSTM followed by a dense classifier (sigmoid). Optimization uses Adam with early stopping on validation loss.

Tested hyperparameters:

- Model 1: seq-len=2, lstm-units=32, dropout=0.5. This is a highly constrained network.
- Model 2: seq-len=3, lstm-units=64, dropout=0.3. This slightly extends the temporal window while maintaining the higher capacity.
- Model 3: seq-len=2, lstm-units=64, dropout=0.3. This model looks at the same tiny time window but has double the learning capacity and less dropout.

Table 9. BiLSTM training results on StressID dataset

Metric	**Model 1**	**Model 2**	**Model 3**
Accuracy	~ 70%	**~ 75%**	~ 66%
Balanced accuracy	~ 69%	**~ 70%**	~ 67%
Precision	**~ 63%**	~ 6%	~ 53%
Recall	~ 43%	~ 2%	**~ 48%**
F1	**~ 44%**	~ 3%	~ 43%

The evaluation of BiLSTM architectures on the StressID dataset further underscored the extreme difficulty of cross-subject temporal modeling in highly variant physiological data (Table 9. BiLSTM training results on StressID dataset). Strikingly, the model configuration with a slightly extended temporal context and higher capacity (Model 2) suffered from catastrophic decision boundary collapse. While yielding a deceptively high Accuracy (0.7457), it produced near-zero Precision (0.0629) and Recall (0.0165), indicating a complete failure to detect the minority stress class and a default to majority-class prediction. Conversely, the most heavily constrained model (Model 1) managed to avoid this collapse, achieving the highest F1-score (0.4377) and Precision (0.6250) (Fig. 19. Best BiLSTM training results on StressID dataset). However, the objectively low overall F1-scores across all BiLSTM configurations on StressID, especially when contrasted with the success of longer-sequence BiLSTMs on the SWELL-KW dataset suggests that universal, cross-subject temporal signatures of stress are either absent or obscured by individualized physiological noise within this specific dataset.

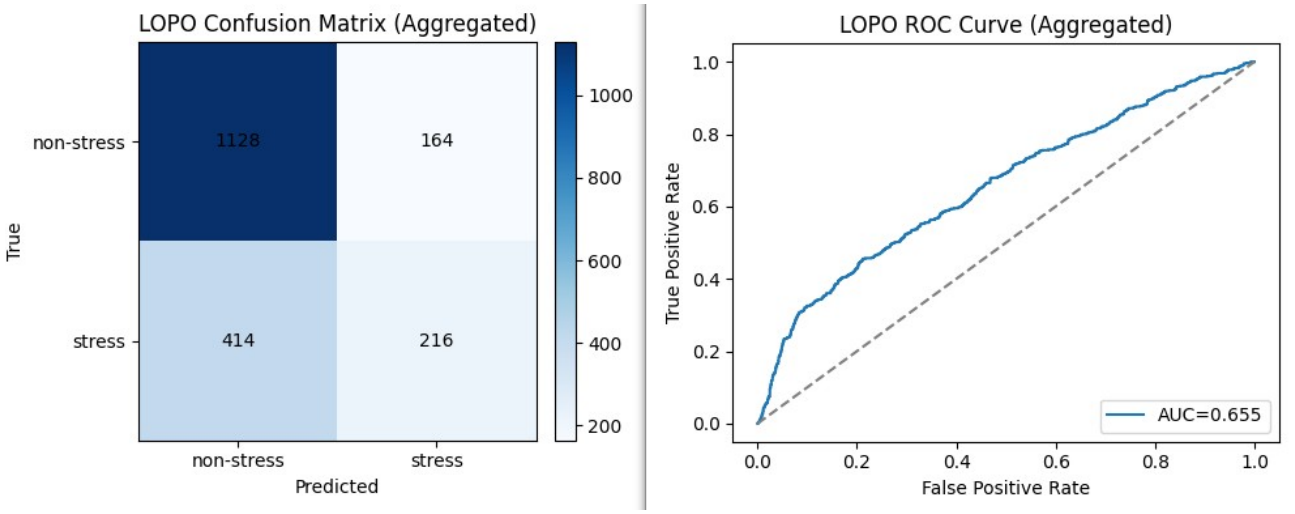

Fig. 19. Best BiLSTM training results on StressID dataset

4.2.4. Issues encountered and their solutions

During models training and their inference testing on StressID dataset, various problems had appeared, their descriptions and solutions are described as follows:

1. Mismatch between available videos and labels: some subjects and tasks appear in labels but not in the video folder, and baseline videos exist without labels. This is solved by adjusting the extractor, it joins labels by subject_task and skips unlabeled videos unless explicitly allowed. Baseline is excluded by default.
2. Need for scalable extraction: processing hundreds of videos sequentially is slow. To tackle this, parallel CPU extraction was introduced to speed up feature generation.
3. LOPO folds with zero positives: some test folds contained only non-stress sequences, causing precision/recall to collapse to zero. Reduced sequence length and stride so shorter videos still generate positive sequences.
4. Overfitting at longer sequence lengths: validation loss increased while training accuracy kept rising. To improve this, the implementation was updated to reduce epochs and rely on early stopping to capture the best epoch.

4.2.5. Models comparison

HGB and SVM RBF deliver the best overall balance between classes with similar recall and F1, while the BiLSTM underperforms across all metrics, suggesting that the sequence model is not yet capturing additional temporal signal beyond what the windowed features already provide. The small gap between HGB and SVM indicates that the choice between them can be guided by practical considerations (training speed, interpretability), whereas BiLSTM would need further tuning to be competitive.

Table 10. Best models training results comparison on StressID dataset

Metric	HistGradientBoosting	SVM Kernel RBF	BiLSTM
Accuracy	**~ 72%**	~ 70%	~ 70%
Balanced Accuracy	**~ 74%**	~ 72%	~ 69%
Precision	**~ 68%**	~ 63%	~ 63%
Recall	**~ 53%**	~ 52%	~ 43%
F1	**~ 54%**	~ 52%	~ 44%

As seen from the training results (Table 10. Best models training results comparison on StressID dataset), HistGradienBoosting is the most robust model out of the three. It’s almost on the same level as SVM, but is very slightly better looking from metrics perspective. StressID models show moderate performance but higher than SWELL-KW likely because the extracted video features are more consistent across subjects and the dataset provides more uniform recordings per task, yielding a cleaner signal when aggregated into 30-second windows. The larger number of windowed samples per video also helps classical models (HGB/SVM) to improve generalization. At the same time,

results remain moderate because labels are still task-level, stress behaviors are subtle and vary widely by person, and that the LOPO protocol is demanding.

Overall, the results these models achieved are pretty good and comparable to StressID study and their video-only results (Fig. 20. StressID study results).

	2-class		3-class	
Baseline	F1-score	Accuracy	F1-score	Accuracy
Physiological only	0.66 ± 0.05	0.58 ± 0.04	0.50 ± 0.05	0.48 ± 0.06
Video only	0.67 ± 0.03	0.62 ± 0.04	0.58 ± 0.05	0.56 ± 0.05
Audio only	0.67 ± 0.04	0.62 ± 0.04	0.56 ± 0.06	0.54 ± 0.06
Feature fusion + SVM	0.64 ± 0.09	0.56 ± 0.05	0.55 ± 0.06	0.51 ± 0.05
Feature fusion + MLP	0.66 ± 0.04	0.61 ± 0.03	0.51 ± 0.07	0.51 ± 0.07
Feature fusion + DBN	0.58 ± 0.06	0.52 ± 0.05	0.30 ± 0.09	0.32 ± 0.04
SVM + Sum rule fusion	$\mathbf{0.72 \pm 0.05}$	0.64 ± 0.05	0.62 ± 0.05	$\mathbf{0.58 \pm 0.07}$
SVM + Product rule fusion	0.71 ± 0.05	0.63 ± 0.05	0.61 ± 0.05	0.56 ± 0.07
SVM + Average rule fusion	$\mathbf{0.72 \pm 0.05}$	$\mathbf{0.65 \pm 0.05}$	$\mathbf{0.63 \pm 0.05}$	$\mathbf{0.58 \pm 0.07}$
SVM + Maximum rule fusion	$\mathbf{0.72 \pm 0.05}$	0.64 ± 0.05	0.61 ± 0.06	0.57 ± 0.07

Fig. 20. StressID study results

Only the 2-class should be compared, because even though StressID dataset has 3 stress classes in their dataset and research, only 2 were used in this study and if comparing the same dataset results it is seen that on our study the accuracy is even better, but the F1 score is worse.

4.3. Testing inference

Before testing, best performing models has to be selected, and these are SWELL-KW BiLSTM and StressID HistGradientBoosting (Table 11. Best performing models).

Table 11. Best performing models

Metric	**SWELL-KW BiLSTM**	**StressID HistGradientBoosting**
Accuracy	~ 62%	**~ 72%**
Balanced Accuracy	~ 60%	**~ 74%**
Precision	~ 67%	**~ 68%**
Recall	**~ 69%**	~ 53%
F1	**~ 77%**	~ 54%

Testing was done using inference script that loads the model and displays stress probability. For demonstration purposes, inference time was lowered to 3 seconds (default for SWELL-KW trained modes is 60 seconds and for StressID trained models its 30 seconds), this means that the prediction is done every 3 seconds. Three stages were captured: moderate stress, where moderate body and face tension were simulated, relaxed position and heavily tensed body and face position (Fig. 21. Different body and face positions. a) Very relaxed body and face position. b) Moderate body and face tension. c) Moderately relaxed body and face tension. d) Heavily tense body and face).

Fig. 21. Different body and face positions. a) Very relaxed body and face position. b) Moderate body and face tension. c) Moderately relaxed body and face tension. d) Heavily tense body and face

The results are pretty good and trustworthy, the relaxed state got the lowest probability (0.115), while the moderate position got higher probability (0.643) and the heavily tense position got the highest (0.896). The main and most optimistic point of this test was that the model accurately distinguished the heavily tensed position from relaxed position, even though they are visually similar.

Next experiment was done by filming and inferring stress probability at work for two days. The results for Day 1 and Day 2 are presented in Fig. 22. Day 1 results comparison and Fig. 23. Day 2 results comparison respectively. To capture the relationship between physiological and behavioral data, this experiment utilized smartwatch (Samsung Galaxy Watch 5) metrics representing integrated stress scores derived from heart rate variability (HRV) and movement, sampled at 30-minute intervals. This baseline was compared against stress inference model probability scores every 30 seconds. To ensure compatibility between these two distinct data streams, the high-frequency video results were processed using a 10-minute rolling average, while linear interpolation was applied to bridge any data gaps resulting from intermittent facial occlusions or the subject moving out of the camera frame.

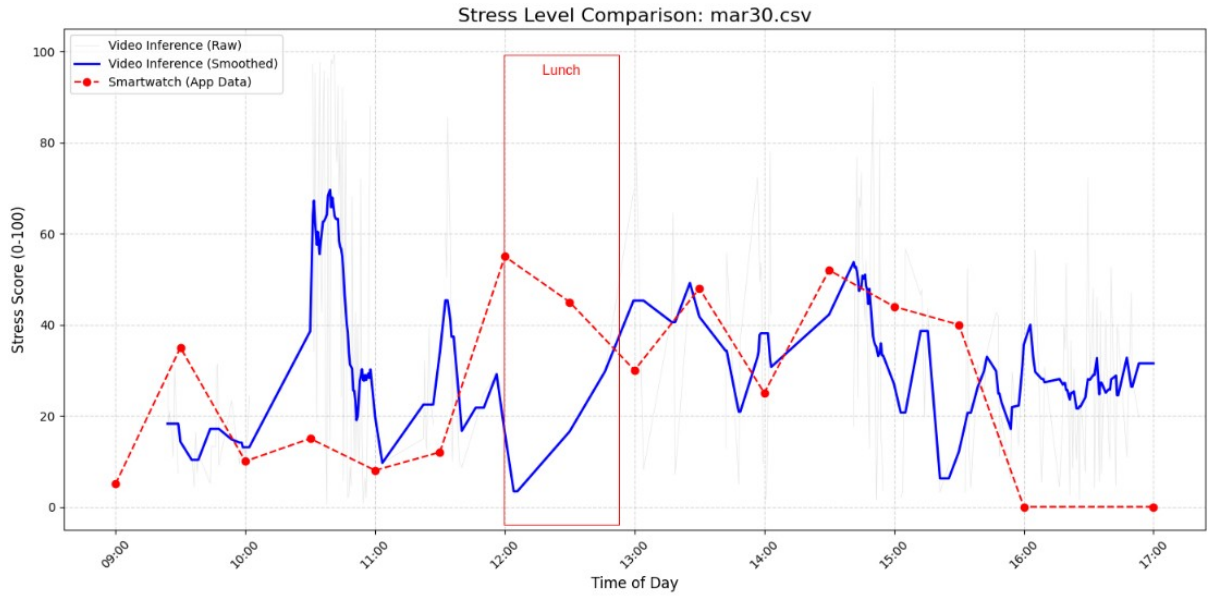

Fig. 22. Day 1 results comparison

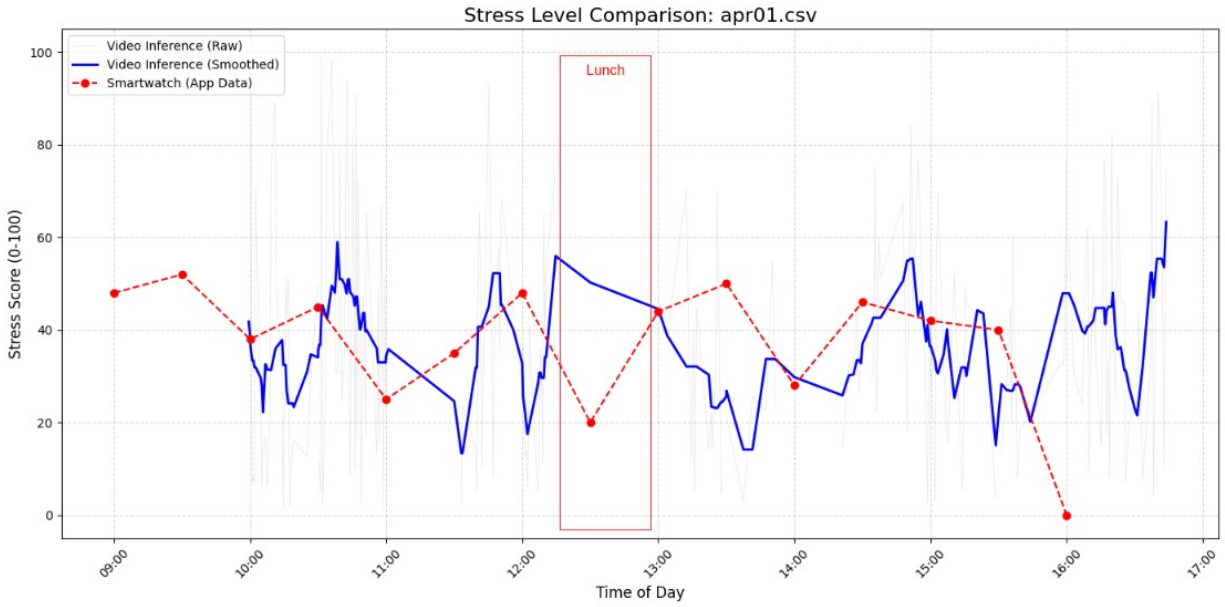

Fig. 23. Day 2 results comparison

Across the observation periods (two days), the smoothed video inference data demonstrated a reasonable directional alignment with the smartwatch baseline. While the wearable device provided a consistent long-term monitoring state, the video inference model’s higher sensitivity allowed for identifications of "micro-stressors" and rapid fluctuations that the smartwatch’s lower sampling rate naturally smoothed over. However, analysis of the raw video data revealed irregular tracking gaps correlating with periods of high physical activity or non-frontal head orientation, these dropouts highlight the essential environmental constraints of video-based sensing when compared to the continuous, skin-contact reliability of traditional wearables.

4.4. Discussion

The proposed novelty of this study is not a new model architecture, but a unified video-based stress detection pipeline that applies the same body pose and facial feature representation across two different datasets, SWELL-KW and StressID. Instead of relying on physiological or audio signals, the system uses only visual information and converts both datasets into a comparable SWELL-style feature space. This makes it possible to evaluate classical tabular models and a temporal deep model under the same preprocessing, validation, and inference framework.

The work included building body pose and facial feature extraction modules, aligning body and face modalities, preparing SWELL-KW and StressID into trainable feature tables, applying participant-level normalization and missing-value handling, and training three supervised models: HGB, SVM-RBF and BiLSTM. The models were evaluated with leave-one-participant-out validation to measure how well they generalize to unseen subjects. The combined body and face feature set worked better than relying on a single modality and participant/subject normalization helped reduce person-specific variation. Classical models, especially HGB and SVM-RBF, were stable on windowed tabular features, while BiLSTM was useful for testing whether temporal modeling adds value. StressID produced stronger balanced accuracy than SWELL-KW, likely because the extracted video windows gave more consistent samples.

Compared with studies that use physiological sensors, this approach is less intrusive because it only requires video, and in contrast with studies focused on a single dataset or one model type, this work compares multiple model families across two datasets using the same evaluation strategy. The limitation is that visual-only stress detection is generally harder and more sensitive to lighting, occlusion, pose visibility and label quality, so results may be lower than sensor-based systems but are more practical for non-contact monitoring.

Conclusions

1. An extensive literature review was conducted to examine how stress affects observable human behavior, especially posture, movement and facial expression. Even though there is not much literature about similar systems, the reviewed studies show that stress can influence body rigidity, shoulder position, gaze behavior, facial tension, and movement patterns. This supports the main assumption of the work: stress can be estimated not only through biochemical markers, wearable sensors, or self-reports, but also through non-contact visual cues. Therefore, the proposed camera-based system is relevant because it explores a less intrusive alternative for stress detection.
2. Several pose estimation technologies were analyzed to select a suitable method for real-time body tracking. MediaPipe Pose was chosen because it provides a practical balance between speed, accessibility and accuracy for real-time inference. Other solutions such as OpenPose, AlphaPose and MMPose can offer advantages such as higher precision or stronger multi-person tracking, but they are heavier and less convenient for a lightweight prototype. As a result, MediaPipe Pose was considered the most appropriate option for a webcam-based stress detection system.
3. Stress-related datasets suitable for visual stress detection were found to be limited, especially when body posture and facial features are required. SWELL-KW and StressID were selected because they contain multimodal stress related recordings and usable labels for supervised learning. SWELL-KW provided structured body and facial features, while StressID required feature extraction from video recordings. This dataset selection allowed the system to be tested on two different sources and helped evaluate whether the same visual feature pipeline could be applied beyond a single dataset.
4. Three supervised models were implemented and trained on both datasets: HistGradientBoosting, SVM with an RBF kernel and BiLSTM. These models were selected to compare classical tabular learning, kernel based non-linear classification, and temporal sequence modeling. The results showed that the best model depended on the dataset: BiLSTM performed best on SWELL-KW according to balanced accuracy, while the best StressID result was achieved by the HGB model in the final experiments. This indicates that no single model is universally optimal and that dataset structure, label quality and temporal consistency strongly affects model performance.
5. A real-time inference pipeline was created to connect feature extraction, preprocessing, model loading, and live stress prediction from webcam input. The system was tested on different body and facial tension states, including relaxed, moderately tense and heavily tense behavior. The pipeline was able to react to visible changes in posture and facial expression, showing that the trained models can be used outside offline dataset evaluation. This demonstrates the practical feasibility of turning the trained classifiers into an interactive stress estimation prototype.
6. Overall, the results suggest that video-based stress inference is a viable non-intrusive direction for stress detection, especially when body and facial cues are combined. Wearable and physiological sensors still provide stronger continuous baselines and more direct biological signals, but video-based inference offers advantages in comfort, accessibility and detection of visible high intensity moments. The findings show that camera-based stress recognition can complement traditional methods, although further work is needed to improve robustness across lighting, occlusion, subject variability and real-world conditions.

AI tools usage

In this study, AI tools were used only as supportive aids for improving grammar, writing style and generating ideas for data visualization. All study design decisions, algorithm development, data processing pipelines, analyses and interpretations were created by the author. Any AI-assisted suggestions were carefully reviewed, verified and adapted where appropriate.

List of references

1. Richer, R.; Koch, V.; Abel, L.; et al. Machine learning-based detection of acute psychosocial stress from body posture and movements [online]. 2024. Available from: https://doi.org/10.1038/s41598-024-59043-1
2. Cao, Z.; Simon, T.; Wei, S.-E.; Sheikh, Y. Realtime multi-person 2D pose estimation using part affinity fields [online]. 2017. Available from: https://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf
3. COCO Consortium. MS COCO keypoint detection task [online]. [no date]. Available from: https://cocodataset.org
4. Max Planck Institute for Informatics. MPII human pose dataset [online]. [no date]. Available from: https://www.mpi-inf.mpg.de/departments/computer-vision-and-machine-learning/software-and-datasets/mpii-human-pose-dataset
5. Tharatipyakul, A.; Srikaewsiew, T.; Pongnumkul, S. Deep learning-based human body pose estimation in providing feedback for physical movement: A review [online]. 2023. Available from: https://www.sciencedirect.com/science/article/pii/S2405844024126205
6. Toshev, A.; Szegedy, C. DeepPose: Human pose estimation via deep neural networks [online]. 2013. Available from: https://arxiv.org/pdf/1312.4659
7. Zhang, J.; Yin, H.; Zhang, J.; Yang, G.; Qin, J.; He, L. Real-time mental stress detection using multimodality expressions with a deep learning framework [online]. 2022. Available from: https://www.frontiersin.org/journals/neuroscience/articles/10.3389/fnins.2022.947168
8. Xiang, J. Z.; Wang, Q. Y.; Fang, Z. B.; Esquivel, J. A.; Su, Z. X. A multi-modal deep learning approach for stress detection using physiological signals: Integrating time and frequency domain features [online]. 2025. Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC11997569
9. Walambe, R.; Nayak, P.; Bhardwaj, A.; Kotecha, K. Employing multimodal machine learning for stress detection [online]. 2023. Available from: https://arxiv.org/pdf/2306.09385
10. Li, R.; Liu, Z. Stress detection using deep neural networks [online]. 2020. Available from: https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-020-01299-4
11. Seo, W.; Kim, N.; Park, C.; Park, S.-M. Deep learning approach for detecting work-related stress using multimodal signals [online]. 2022. Available from: https://www.researchgate.net/publication/363048616_Deep_Learning_Approach_for_Detecting_Work-Related_Stress_Using_Multimodal_Signals
12. Lazarou, E.; Exarchos, T. P. Predicting stress levels using physiological data: Real-time stress prediction models utilizing wearable devices. AIMS Neuroscience [online]. 2024. Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC11230864
13. Hemakom, A.; Atiwiwat, D.; Israsena, P. ECG and EEG based detection and multilevel classification of stress using machine learning for specified genders: A preliminary study. PLOS ONE [online]. 2023. Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC10473514/
14. Bhole, Y. Stress detection using AI and machine learning [online]. 2024. Available from: https://www.ijert.org/research/stress-detection-using-ai-and-machine-learning-IJERTV13IS080017.pdf

15. Gupta, M. V.; Vaikole, S.; Oza, A. D.; Patel, A.; Burduhos-Nergis, D. P.; Burduhos-Nergis, D. D. Audio-visual stress classification using cascaded RNN-LSTM networks [online]. 2022. Available from: https://pmc.ncbi.nlm.nih.gov/articles/PMC9598122/
16. Google AI Edge. Pose landmarker guide for MediaPipe Pose [online]. 2025. Available from: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker
17. Bazarevsky, V.; Grishchenko, I.; Raveendran, K.; Zhu, T.; Zhang, F.; Grundmann, M. BlazePose: On-device real-time body pose tracking [online]. 2024. Available from: https://ar5iv.labs.arxiv.org/html/2006.10204
18. Alves, M. G.; Chen, G.-L.; Kang, X.; Song, G.-H. Reduced CPU workload for human pose detection with the aid of a low-resolution infrared array sensor on embedded systems [online]. 2023. Available from: https://www.mdpi.com/1424-8220/23/23/9403
19. OpenMMLab. MMPose [online]. [no date]. Available from: https://github.com/open-mmlab/mmpose
20. Jiang, T.; Lu, P.; Zhang, L.; Ma, N.; Han, R.; Lyu, C.; Li, Y.; Chen, K. RTMPose: Real-time multi-person pose estimation based on MMPose [online]. 2023. Available from: https://arxiv.org/pdf/2303.07399
21. CMU Perceptual Computing Lab. OpenPose [online]. [no date]. Available from: https://github.com/CMU-Perceptual-Computing-Lab/openpose
22. Yamazaki, M.; Mori, E. Rethinking deconvolution for 2D human pose estimation [online]. 2021. Available from: https://arxiv.org/pdf/2111.04226
23. CMU Perceptual Computing Lab. Maximizing OpenPose speed [online]. [no date]. Available from: https://cmu-perceptual-computing-lab.github.io/openpose/web/html/doc/md_doc_06_maximizing_openpose_speed.html
24. Fang, H.; Li, J.; Tang, H.; Xu, C.; Zhu, H.; Xiu, Y.; Li, Y.; Lu, C. AlphaPose: Whole-body regional multi-person pose estimation and tracking in real time [online]. 2022. Available from: https://arxiv.org/pdf/2211.03375
25. Koldijk, S.; Sappelli, M.; Verberne, S.; Neerincx, M.; Kraaij, W. The SWELL knowledge work dataset for stress and user modeling research [online]. 2014. Available from: https://cs.ru.nl/~skoldijk/Papers/ICMI%202014%20paper_final_cr.pdf
26. Chaptoukaev, H.; Strizhkova, V.; Panariello, M.; Dalpaos, B.; Reka, A.; Manera, V.; Thummler, S.; Ismailova, E.; W., N.; Bremond, F.; Todisco, M.; Zuluaga, M. A.; Ferrari, L. M. StressID: A multimodal dataset for stress identification [online]. 2023. Available from: https://openreview.net/attachment?id=qWsQi9DGJb&name=supplementary_material
27. Al-A'araji, N. H.; Almamory, S.; Shakarchi, A. Classification and clustering based ensemble techniques for intrusion detection systems: A survey [online]. 2021. Available from: https://www.researchgate.net/publication/349907931_Classification_and_Clustering_Based_Ensemble_Techniques_for_Intrusion_Detection_Systems_A_Survey
28. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.-Y. LightGBM: A highly efficient gradient boosting decision tree [online]. 2017. Available from: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf
29. Narasimha, P. L.; Malalur, S. S.; Manry, M. Small models of large machines [online]. 2008. Available from: https://www.researchgate.net/publication/221438405_Small_Models_of_Large_Machines

30. Elsayed, N.; Zaghloul, Z. S.; Azumah, S. W.; Li, C. Intrusion detection system in smart home network using bidirectional LSTM and convolutional neural networks hybrid model [online]. 2021. Available from: https://www.researchgate.net/publication/351869045_Intrusion_Detection_System_in_Smart_Home_Network_Using_Bidirectional_LSTM_and_Convolutional_Neural_Networks_Hybrid_Model