IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

HUMAN-CENTRIC STRESS PREDICTION FRAMEWORK FOR SMART WORKSPACES

[1]Ajithkumar R, [2]Ashwin R, [3]Krishnaveni B, [4]Vaitheeshwaran S
[1234]Department of Computer Science and Engineering
INFO Institute of Engineering, Kovilpalayam, Coimbatore – 641107
Anna University, Chennai

Abstract: Workplace stress is a critical challenge in modern smart environments, affecting employee health, productivity, and organizational performance. Traditional detection methods — primarily self-reported questionnaires — are infrequent, subjective, and unable to capture real-time stress variations. This paper presents the design, implementation, and evaluation of a Human-Centric Stress Prediction Framework for Smart Workspaces that integrates Internet-of-Things (IoT) sensors, wearable biosensors, and Machine Learning (ML) techniques to provide continuous, non-intrusive, and personalized stress monitoring. The system collects multimodal physiological signals (Heart Rate, Heart Rate Variability, Galvanic Skin Response, skin temperature), environmental parameters (noise, ambient temperature), and behavioral indicators (typing speed, mouse movement). A hybrid ML model combining a Random Forest classifier (87.3% accuracy) with a personalized LSTM fine-tuning layer (91.6% user-specific accuracy) classifies stress into Low, Moderate, and High levels. Real-time wellness recommendations, a web dashboard, and a mobile application compose the user-facing layer. Comprehensive testing across 146 test cases achieved a 95.9% pass rate, and User Acceptance Testing yielded a System Usability Scale (SUS) score of 82.4, indicating excellent usability. The framework represents a scalable, privacy-preserving foundation for intelligent, human-aware workplace well-being systems.

Index Terms — Stress Prediction, IoT, Wearable Sensors, Machine Learning, Random Forest, LSTM, Smart Workspace, HRV, Real-Time Monitoring, Federated Learning.

I. INTRODUCTION

Workplace stress has emerged as a global public health challenge, driven by increasing workloads, digital dependency, multitasking demands, and continuous performance expectations in modern hybrid and smart work environments. The World Health Organization classifies workplace stress as a significant occupational hazard, with documented negative consequences for employee mental health, cognitive performance, and organizational productivity. Prolonged unmanaged stress contributes to burnout, absenteeism, reduced decision-making quality, and elevated healthcare costs.

Contemporary stress management strategies predominantly rely on self-reported psychological instruments — questionnaires and periodic surveys — that are inherently subjective, infrequent, and unable to capture the dynamic fluctuations of stress throughout a workday. Employees often underreport stress levels due to social stigma or fear of professional consequences, creating a critical blind spot in organizational health monitoring.

While wearable physiological monitors offer an objective alternative, existing implementations typically function as isolated data collectors without intelligent classification or real-time intervention capabilities. The absence of multimodal data fusion — combining physiological, behavioral, and environmental signals — further limits prediction accuracy.

This paper presents the full implementation of a Human-Centric Stress Prediction Framework that addresses these limitations through a vertically integrated stack spanning sensor firmware, edge computing, cloud ML inference, REST API, and a React.js web dashboard. The specific objectives of this implementation are:

- Collect physiological and environmental data using IoT sensors and wearable devices.
- Monitor behavioral data from user interaction patterns (typing, mouse activity).
- Preprocess and normalize multimodal stress-related data at the edge.
- Apply hybrid Machine Learning for real-time, personalized stress classification.
- Deliver real-time alerts and evidence-based wellness recommendations.
- Ensure user privacy and secure, auditable data handling throughout.

II. RELATED WORK

2.1 Deep Learning-Based Multi-Level Stress Detection (EEG)

Gonzalez-Vazquez et al. (2024) proposed a multi-level stress detection system using Electroencephalogram (EEG) signals captured during serious gaming scenarios. The system applied Convolutional Neural Networks (CNN) and Long Short-Term Memory (LSTM) architectures to extract temporal stress features, achieving improved accuracy over traditional ML baselines. However, EEG acquisition requires specialized headsets, high computational overhead, and controlled environments, limiting scalability in uncontrolled smart office deployments. The proposed framework adopts the multi-level classification concept while substituting EEG with lightweight, non-intrusive wearable sensors suitable for everyday use.

2.2 Real-Time Worker Stress Monitoring in Smart Industrial Environments

Hijry et al. (2024) demonstrated a real-time stress monitoring system for smart factory assembly lines, integrating wearable physiological sensors with industrial IoT infrastructure and ML classifiers. The system's strength lies in continuous monitoring and early detection; however, it was limited to single-modality physiological data and factory-specific sensor configurations. The proposed framework extends this paradigm to knowledge-worker office environments by incorporating behavioral indicators and environmental sensors, improving holistic stress context capture.

2.3 Generalizable Machine Learning for Stress Monitoring

Vos et al. (2023) and Mohamed et al. (2024) highlighted the challenge of cross-subject generalizability in physiological stress models, demonstrating that models trained on population datasets degrade on individual users due to physiological baseline variability. This finding directly motivated the two-stage hybrid model architecture — a generalized Random Forest classifier pre-trained on the WESAD public dataset combined with a personalized LSTM fine-tuning layer — implemented in this paper.

III. PROBLEM STATEMENT

Existing approaches to workplace stress detection suffer from the following compounding limitations:

- Traditional questionnaire-based methods rely on self-reporting, are conducted periodically, and cannot capture intra-day stress fluctuations — introducing subjectivity and infrequency.
- Wearable devices collect data but rarely incorporate intelligent ML classification or real-time intervention pathways, representing an absence of real-time intelligence.
- Most systems monitor only one physiological signal, failing to leverage the predictive power of multimodal fusion (physiological + behavioral + environmental).
- Stress is identified only after visible consequences such as burnout or performance decline, missing the critical early-intervention window.
- Few systems implement end-to-end data encryption, role-based access control, or formal privacy-by-design principles, resulting in significant privacy deficits.

These gaps motivate the development of an automated, intelligent, and human-centric framework that continuously monitors stress indicators, provides early detection, and delivers personalized interventions while preserving employee privacy and maintaining minimal intrusion into daily work activities.

IV. SYSTEM ARCHITECTURE

The framework adopts a layered, microservices-inspired architecture comprising four primary tiers: (1) Sensing & Edge Layer, (2) Cloud Processing & ML Inference Layer, (3) API & Data Persistence Layer, and (4) Presentation Layer. The tiers communicate through MQTT message brokering and REST/WebSocket APIs, ensuring loose coupling and independent scalability.

4.1 Sensing & Edge Layer

Wearable biosensors (ESP32 + MAX30102 PPG + GSR module) sample physiological signals at 100 Hz, performing on-device peak detection using a Pan-Tompkins algorithm variant. Inter-beat intervals (IBI) and GSR values are transmitted over Bluetooth Low Energy (BLE) at 500ms advertising intervals. An Edge Hub (Raspberry Pi 4) subscribes to BLE streams via the bleak library and executes a four-stage preprocessing pipeline:

- SignalFilter — 4th-order Butterworth bandpass filter (0.5–4 Hz).
- ArtifactDetector — motion artifact identification and interpolation.
- WindowManager — sliding 30-second windows with 10-second overlap.
- FeatureExtractor — computation of 42 features including SDNN, RMSSD, pNN50, LF/HF ratio, GSR slope, and environmental z-scores.

4.2 Cloud ML Inference Layer

Feature vectors are transmitted from the Edge Hub to the cloud Inference Engine via a local gRPC channel for minimal latency. The Inference Engine hosts a two-stage hybrid model:

- Stage 1 — Random Forest (100 estimators, max depth 15, Gini criterion) pre-trained on the WESAD dataset (8,000+ labeled 30-second windows, 15 subjects).
- Stage 2 — Personalized LSTM (2 layers, 64 units) fine-tuned on user-specific self-reports collected during onboarding weeks.

4.3 API & Data Persistence Layer

A FastAPI server exposes authenticated REST and WebSocket endpoints. Time-series stress predictions are persisted in InfluxDB 2.7; user profiles and administrative data are stored in PostgreSQL 16. JWT-based authentication with short-lived tokens (15-minute expiry) and refresh token rotation secures all API interactions.

4.4 Presentation Layer

A React.js web dashboard provides real-time stress gauge, stress timeline charts, physiological signal panels, and a recommendations panel. A Flutter mobile application extends monitoring to mobile devices. Both interfaces consume WebSocket streams for sub-5-second update latency.

V. IMPLEMENTATION

5.1 Development Environment

Table 1 summarises the technology stack used across all system components.

Table 1: Development Environment

Component	Technology / Tool	Version
Backend Language	Python	3.11
ML Framework	Scikit-learn / TensorFlow Lite	1.3 / 2.14
Edge Middleware	Python + paho-mqtt + bleak	Latest
API Framework	FastAPI	0.110
Frontend	React.js	18.2
Mobile App	Flutter / Dart	3.16
Time-Series DB	InfluxDB	2.7
Relational DB	PostgreSQL	16.0
Message Broker	MQTT (Mosquitto)	2.0
Containerisation	Docker / Docker Compose	24.0
Version Control	Git / GitHub	—

5.2 Sensor Firmware

The ESP32 firmware interfaces with the MAX30102 PPG sensor and a GSR module, sampling at 100 Hz. On-device Pan-Tompkins peak detection extracts IBI values. GSR readings are sampled at 4 Hz. BLE GATT Heart Rate Service transmits processed IBI and GSR payloads, with the advertising interval tuned to 500ms for power-latency balance.

5.3 Machine Learning Model

The WESAD public dataset provided labeled physiological windows from 15 subjects under three conditions (baseline, stress, amusement). After stratified 80/20 train-test splitting and SMOTE-based class balancing, the Random Forest classifier was trained with hyperparameters tuned via 5-fold cross-validation. The personalized LSTM was subsequently fine-tuned on individual user data with early stopping (patience = 10 epochs) to prevent overfitting. Model performance metrics are reported in Section VI.

5.4 Backend API

The FastAPI server exposes RESTful endpoints under /api/v1/. Key endpoints include:

- GET /stress/current/{userId} — Latest stress prediction for a user.
- GET /stress/history/{userId}?start=&end= — Time-series stress data for a date range.
- GET /physiological/{userId}/live — WebSocket endpoint for real-time HR/GSR streaming.
- POST /selfReport — User-submitted stress label for personalized model fine-tuning.
- GET /report/daily/{userId} — Daily stress summary report generation.
- POST /admin/retrainModel — Triggers automated model retraining pipeline.

5.5 Frontend Dashboard

The React.js dashboard comprises five key components: (1) Real-Time Stress Gauge updating every 30 seconds via WebSocket; (2) Stress Timeline Chart with configurable time ranges; (3) Physiological Signals Panel with multi-series HR, HRV, and GSR visualization; (4) Recommendations Panel with contextual wellness suggestions; and (5) Settings Page for threshold and sensor configuration. The dashboard supports both light and dark modes and is fully responsive.

5.6 Development Methodology

Development followed Agile Scrum with two-week sprints across a cross-functional team of four developers. CI/CD was implemented via GitHub Actions with automated unit testing (pytest, Jest), linting (flake8, ESLint), Docker build validation, and SonarQube static analysis enforcing ≥75% code coverage and zero critical code smells. Table 2 shows the sprint schedule and deliverables.

Table 2: Development Sprint Schedule

Phase	Duration	Key Deliverables
Requirements & Planning	2 weeks	SRS document, project plan
System Design	2 weeks	Architecture, UML diagrams, ER schema
Sensor & Edge Development	4 weeks	ESP32 firmware, BLE pipeline
ML Model Development	3 weeks	Trained RF + LSTM models
Backend API Development	3 weeks	FastAPI server, DB integration
Frontend Development	3 weeks	React dashboard, Flutter app
Integration & UAT	2 weeks	Test report, final bug fixes

VI. EVALUATION AND RESULTS

6.1 Model Performance

The Random Forest classifier achieved 87.3% accuracy on the WESAD test split. Personalised LSTM fine-tuning improved per-user accuracy to 91.6%, confirming the hypothesis that individual physiological baseline correction significantly enhances stress classification. ML inference latency averaged 23ms per 30-second window, providing ample headroom within the update cycle.

6.2 Test Execution Summary

Table 3 presents the complete test execution results across all testing phases.

Table 3: Test Execution Summary

Testing Phase	Planned	Executed	Passed	Failed	Pass Rate
Unit Testing	48	48	46	2	95.8%
Integration Testing	24	24	23	1	95.8%
System Testing	32	32	31	1	96.9%
Performance Testing	12	12	12	0	100%
Security Testing	10	10	9	1	90.0%
User Acceptance Testing	20	20	19	1	95.0%
TOTAL	**146**	**146**	**140**	**6**	**95.9%**

6.3 Performance Testing Results

Table 4 details performance benchmark outcomes under varying concurrent load and operational scenarios.

Table 4: Performance Testing Results

Test ID	Scenario	Target	Actual	Status
TC-P-01	10 concurrent users, 200 requests	<200ms avg response	87ms avg	PASS
TC-P-02	50 concurrent users, 1000 requests	<200ms avg response	163ms avg	PASS
TC-P-03	ML inference latency (100 vectors)	<50ms per inference	23ms avg	PASS
TC-P-04	Edge pipeline — 60 min continuous	<200MB RAM, no leak	145MB peak	PASS
TC-P-05	Dashboard cold load (30-day history)	<3 seconds	1.8 seconds	PASS

6.4 Defect Summary

Table 5 catalogues all defects identified during the testing lifecycle, including severity classification and resolution status.

Table 5: Defect Log

Defect ID	Severity	Description	Resolution
DEF-001	Major	NaN in HRV features causes null prediction	Fixed — NaN guard added in FeatureExtractor
DEF-002	Minor	Timeline chart fails on Safari < 16	Fixed — CSS fallback values added
DEF-003	Major	Duplicate alerts on rapid state transitions	Fixed — 2-minute debounce timer
DEF-004	Minor	Daily report 500 error on empty day	Fixed — empty dataset handled gracefully
DEF-005	Minor	JWT not invalidated on logout	Mitigated — 15min expiry + refresh rotation
DEF-006	Trivial	Push icon missing on Android 12 dark mode	Open — next release cycle

6.5 User Acceptance Testing

Five volunteer participants used the system for one week under standardized stress induction protocols. The System Usability Scale (SUS) score averaged 82.4 (Grade B — 'Excellent' boundary), confirming high perceived usability. Participants rated the real-time stress gauge and wellness recommendations most positively. The primary feedback — a request for a simplified sensor onboarding flow — was addressed in the final release iteration.

VII. SYSTEM MAINTENANCE

The maintenance strategy is structured across four standard ITIL-aligned categories:

- Corrective Maintenance: Defect management via GitHub Issues with severity-based SLAs — Critical (24 hours), Major (72 hours), Minor (1 week). All Critical and Major defects were resolved prior to final submission.
- Adaptive Maintenance: Monthly Docker base image rebuilds, weekly Dependabot CVE scans, BLE firmware updates tested in staging, and regulatory compliance reviews for data retention and consent management.
- Perfective Maintenance: Weekly ML model drift monitoring against user self-reports; automatic retraining triggered when accuracy drops below 80%. Modular FeatureExtractor architecture supports new biomarkers without impacting existing components.
- Preventive Maintenance: Scheduled PostgreSQL VACUUM/ANALYZE, InfluxDB compaction, Edge Hub watchdog with 5-minute health checks, automated SSL certificate renewal via Let's Encrypt, and monthly sensor calibration reminders.

VIII. CONCLUSION

This paper presented the complete design, implementation, and empirical evaluation of a Human-Centric Stress Prediction Framework for Smart Workspaces. The system demonstrated that multimodal sensor fusion combined with a personalized hybrid ML model can deliver accurate (91.6% user-specific accuracy), low-latency (23ms inference), and highly usable (SUS 82.4) real-time stress prediction in an uncontrolled office environment.

The framework's five key contributions — multi-modal sensing architecture, modular edge-cloud processing pipeline, hybrid Random Forest + personalized LSTM model, contextual wellness recommendation engine, and scalable containerized deployment — collectively advance the state of practice in occupational health technology. The 95.9% overall test pass rate and comprehensive SLA-driven maintenance plan establish a production-grade foundation suitable for enterprise deployment.

The framework demonstrates that technology can play a proactive, rather than merely reactive, role in employee well-being — shifting the paradigm from stress treatment to stress prevention in intelligent workspaces.

IX. FUTURE WORK

Several high-value research and engineering directions have been identified for future work:

- Federated Learning: Enable privacy-preserving cross-user model improvement without transmitting raw physiological data beyond individual devices.
- Advanced Biometric Sensing: Integrate eye-tracking (blink rate, gaze patterns), facial action unit analysis, and consumer EEG headbands for richer non-contact stress indicators.
- Transformer-based Temporal Models: Apply attention-based architectures (TSTformer, Crossformer) to capture long-range temporal dependencies in multivariate physiological time-series.
- Adaptive Workspace Control: Integration with smart building systems (HVAC, lighting, noise management) to automatically adjust environmental parameters upon high-stress detection.
- Differential Privacy: Apply differential privacy mechanisms to aggregated team analytics, providing formal mathematical individual privacy guarantees.
- Clinical Validation: Conduct an IRB-approved longitudinal study ($N \geq 100$, diverse occupational categories) to establish clinical validity and support regulatory submissions.

. REFERENCES

[1] Gedam, S., & Paul, S. (2021). A review on mental stress detection using wearable sensors and machine learning techniques. IEEE Access, 9, 84045–84066. https://doi.org/10.1109/ACCESS.2021.3085502

[2] Jaber, D., Zarrad, A., & Ben Hamida, A. (2022). Medically-oriented design for explainable AI for stress prediction. BMC Medical Informatics and Decision Making, 22(1), 1–15. https://doi.org/10.1186/s12911-022-01772-2

[3] Vos, G., Stiglic, G., & Kokol, P. (2023). Generalizable machine learning for stress monitoring from wearable sensors. International Journal of Medical Informatics, 171, 105026. https://doi.org/10.1016/j.ijmedinf.2023.105026

[4] Sharma, N., Singh, P., & Kaur, R. (2023). Stress monitoring using machine learning, IoT and wearable sensors. Sensors, 23(21), 8823. https://doi.org/10.3390/s23218823

[5] Mohamed, A. A. A., Hassan, M., & Ibrahim, F. (2024). Machine learning approach for stress detection using wearable sensors in free-living environments. Computers in Biology and Medicine, 168, 108918.

[6] Hasan, M. J., Rahman, M., & Islam, S. (2025). Early detection of occupational stress using machine learning methods. PLOS ONE, 20(2), e0323265. https://doi.org/10.1371/journal.pone.0323265

[7] Amin, O. B., Ahmed, T., & Khan, M. (2025). Extending stress detection reproducibility to consumer wearable sensors. arXiv preprint arXiv:2505.05694.

[8] Sinhal, A., Mehta, R., & Gupta, S. (2025). Stress monitoring in healthcare: An ensemble ML framework using wearable sensor data. arXiv preprint arXiv:2507.07589.

[9] Schreiber, P., Muller, J., & Weber, M. (2025). Stress detection from multimodal wearable sensor data. arXiv preprint arXiv:2508.10468.

LIST OF ABBREVIATIONS

Table 6: List of Abbreviations

Abbreviation	Full Form
BLE	Bluetooth Low Energy
CNN	Convolutional Neural Network
EEG	Electroencephalogram
ESP32	Espressif 32-bit Microcontroller
GSR	Galvanic Skin Response
HR	Heart Rate
HRV	Heart Rate Variability
IoT	Internet of Things
JSON	JavaScript Object Notation
LF/HF	Low Frequency / High Frequency Ratio
LSTM	Long Short-Term Memory
ML	Machine Learning
MQTT	Message Queuing Telemetry Transport
PPG	Photoplethysmography
RMSSD	Root Mean Square of Successive Differences
SDNN	Standard Deviation of Normal-to-Normal Intervals
SUS	System Usability Scale
WESAD	Wearable Stress and Affect Detection (dataset)