AuraCheck: IoT-Based Integrated Stress and Mental Health Monitoring Chair

Shaikh Ramisha Maliyat
Group:01
ID: 20200204054
Dept. of CSE

Rafiul Awal
Group:01
ID: 20200204062
Dept. of CSE

Nabila Rahman
Group:01
ID: 20200204065
Dept. of CSE

Sabrin Sultana Chadni
Group:01
ID: 20200204068
Dept. of CSE

Syeda Tanjuma Tasnim Mayisha
Group:01
ID: 20200204093
Dept. of CSE

***Abstract*—Stress has become a critical factor affecting human well-being, necessitating real-time monitoring solutions for early detection and intervention. This study proposes an IoT-based stress detection system that integrates multiple physiological and environmental sensors with a Raspberry Pi for real-time data processing. The system utilizes machine learning models to analyze sensor data and determine stress levels. A website-based application provides an intuitive interface for real-time visualization and automated alerts. The processed data is also transmitted to ThingsBoard via MQTT for remote monitoring. The proposed system offers a cost-effective and accessible approach to stress monitoring, contributing to advancements in mental health technology.**

***Index Terms*—IoT, Stress Detection, Machine Learning, Raspberry Pi, Real-Time Monitoring, MQTT, Website-Based Application.**

I. Introduction

Stress has become a growing concern in modern society, significantly impacting mental and physical health. It is linked to various conditions, including anxiety, cardiovascular diseases, and weakened immune response. The increasing prevalence of stress-related disorders highlights the need for effective monitoring and early detection methods. Traditional stress assessment techniques, such as self-report surveys and clinical evaluations, often suffer from subjectivity, delayed diagnosis, and lack of real-time data. Moreover, wearable devices and medical-grade equipment for stress monitoring can be expensive and inaccessible to the general population.

With the rapid advancement of Internet of Things (IoT) technologies and machine learning, real-time stress monitoring systems have become a viable alternative. IoT-based solutions enable continuous tracking of physiological and environmental factors that influence stress levels, offering more objective and data-driven assessments. This research presents an IoT-based stress detection system that integrates multiple sensors with a Raspberry Pi to collect real-time physiological and environmental data. The collected data is processed using machine learning models to determine stress levels, which are then visualized through a website-based application. Additionally, the system utilizes MQTT protocol to transmit data to ThingsBoard, allowing remote monitoring and automated alerts in case of high stress levels.

The proposed system provides a cost-effective and accessible alternative to conventional stress detection methods, enabling real-time assessment and potential intervention. This research aims to contribute to the development of smart health technologies by offering a scalable and efficient solution for stress monitoring.

II. Literature Review

Several existing IoT-based health monitoring solutions focus on vital signs like heart rate and oxygen saturation. However, most lack a comprehensive system that integrates multiple factors such as posture, environmental conditions, and facial expressions. This work builds upon prior research by combining multimodal data sources for a more holistic assessment of stress levels.

A. Mental Well-being Assessment using Bio-signals [1]

Study Overview: This study explores the use of bio-signals from wearable devices to assess mental well-being. It integrates physiological, behavioral, and environmental data to improve stress detection and overall mental health monitoring.

Methodology: The study employs a semi-supervised learning method utilizing both labeled and unlabeled data. Data were collected via wrist-worn sensors and chest patches, complemented by daily surveys, local weather data, and smartphone usage patterns to train deep neural networks for well-being prediction.

Research Gap: The research highlights the lack of reliable physiological stress detectors and comprehensive datasets that combine multiple factors, limiting accurate assessments.

Results: Findings indicate that integrating bio-signals, smartphone usage, and environmental factors significantly improves mental well-being prediction using machine learning.

B. Personalized Emotion Detection with Smart IoT System [2]

Study Overview: The study presents an IoT system for emotion detection using physiological signals, including pulse rate, skin moisture, and temperature, to monitor emotional states like happiness, stress, and depression.

Methodology: Data were collected from sensors under various conditions such as resting and exercising to observe heart rate and Galvanic Skin Response (GSR) changes.

Research Gap: The research identifies a gap in emotion detection systems that rely solely on facial recognition or EEG data, advocating for a more holistic approach integrating multiple physiological signals.

Results: The system successfully distinguished emotional states based on heart rate variations, with lower rates during happiness compared to stress and depression.

C. HIoTSP Framework for Healthcare Monitoring [3]

Study Overview: The study introduces the Healthcare Internet of Things, Services, and People (HIoTSP) framework to enhance healthcare monitoring through wearable sensors and automated service selection.

Methodology: The framework utilizes wearable sensors, contextual activity recognition, and case studies for fall and stress detection, using the publicly available PAMAP2 dataset.

Results: The system achieved 87.16% accuracy in low-level contextual activities, 84.06%-86.36% in high-level activities, 91.68% in fall detection, and 82.93% in stress detection, demonstrating effectiveness in IoT-based healthcare applications.

D. IoMT for Health Monitoring in Individuals and Disaster Rescuers [4]

Study Overview: This paper discusses the Internet of Medical Things (IoMT) and its role in health monitoring for individuals, groups, and disaster rescuers, addressing mobility and environmental challenges in IoMT systems.

Methodology: The proposed health monitoring architecture integrates key technologies such as Markov models, mobility models, and game theory, analyzing routing strategies and energy-saving concerns.

Research Gap: The study highlights the lack of research on routing repair schemes in dynamic environments, particularly for disaster rescuers.

Results: The proposed system effectively supports health monitoring needs and discusses future research directions in security and network integration.

E. IoT-based Sensor Monitoring of Employee Mental Workload [5]

Study Overview: This research investigates managers' expectations and concerns regarding sensor-based monitoring of employee mental workload, bridging research and practical applications in workplaces.

Methodology: A survey of 702 managers across Germany, the UK, and Spain was conducted, analyzed using Bayesian regression models to assess their attitudes.

Research Gap: Existing research lacks investigation into stakeholders' attitudes towards workplace mental workload monitoring, which is crucial for technology adoption.

Results: Managers expected workplace improvements and employee well-being benefits but expressed privacy concerns, influencing their willingness to adopt such monitoring technologies.

F. IoT-based Real-time Maternal Stress Monitoring [6]

Study Overview: The study focuses on an IoT-based system for monitoring maternal stress during pregnancy, addressing the limitations of traditional clinical techniques and existing IoT systems.

Methodology: A feasibility study was conducted with 20 pregnant women wearing Garmin vívosmart 2 devices for seven months. A k-means clustering algorithm was developed for real-time stress adaptation.

Research Gap: Traditional stress monitoring models fail to account for physiological changes during pregnancy, requiring adaptive systems.

Results: The proposed algorithm achieved 97.9% accuracy using 10-fold cross-validation with Random Forests, proving its effectiveness in monitoring stress levels.

G. Development of an iot system for students' stress management [7]

Study Overview: The paper presents an IoT-based stress management system for students, integrating wearable sensors and a mobile health app to measure and reduce stress levels during thesis defense.

Methodology: A wearable heart rate sensor, a mobile app with relaxation content, and a cloud platform were used. Students' heart rates were monitored before, during, and after thesis defense, comparing those using the relaxation app with a control group.

Research Gap: Existing commercial IoT stress-monitoring solutions lack open APIs for customization and integration. Few studies address real-time stress management in real-world educational settings.

Results: Heart rates were significantly lower in students using the relaxation app, confirming its effectiveness in stress reduction. The system provides real-time biofeedback for students and educators, but further research with a larger, diverse sample is needed.

H. EmotIoT: An IoT System to Improve Users' Wellbeing [8]

Study Overview: EmotIoT is an IoT-based system that detects and predicts emotions using AI and sensor data to recommend activities for well-being.

Methodology: A six-stage approach: system design, device selection, IoT architecture, data analysis, machine learning (decision trees, SVC, KNN), and activity recommendation.

Research Gap: Existing IoT systems lack real-time emotion adaptation. This study integrates structured methodologies for emotion-aware IoT solutions.

Results: Achieved 80% accuracy in emotion prediction. HRV and EDA were key indicators, enhancing real-time personalized recommendations.

I. Cognitive Internet of Things (IoT) and computational intelligence for mental well-being [9]

Study Overview: The study integrates Cognitive IoT and AI to improve mental health monitoring using NLP, machine learning, and sensor data.

Methodology: It uses NLP for speech/text analysis, IoT sensors for real-time physiological data, and machine learning for classification.

Research Gap: Traditional assessments are subjective and lack real-time monitoring. This study combines AI and IoT for objective, continuous evaluation.

Results: AI-IoT integration enhances accuracy in detecting mental health issues, outperforming traditional methods.

J. Stress monitoring using wearable sensors: IoT techniques in medical field [10]

Study Overview: The research investigates the use of IoT and wearable sensors for stress monitoring in healthcare, proposing a stress monitoring algorithm (SMA) that includes data acquisition, processing, prediction, and evaluation.

Methodology: The SMA collects physiological data from wearable sensors, processes it, applies various machine learning algorithms, and evaluates their performance for stress prediction.

Research Gap: It highlights a lack of comprehensive studies integrating IoT with machine learning for stress detection and a focus on single sensor types instead of multi-sensor approaches.

Results: The random forest algorithm showed the highest accuracy in stress detection, demonstrating that IoT and wearable technologies can enhance healthcare interventions effectively.

III. Methodology

A. System Architecture

The proposed stress monitoring chair is designed to detect and analyze stress levels in real-time using multiple sensors embedded in the chair, a Raspberry Pi for processing, and a web-based dashboard for visualization. The system integrates **ESP32-based sensors**, **Raspberry Pi**, **MQTT protocol**, and **ThingsBoard IoT platform** to ensure efficient data collection, transmission, and monitoring.

1) Sensor Placement and Data Flow: The chair is embedded with multiple sensors connected to an **ESP32** microcontroller, which continuously collects physiological and environmental data related to stress. These sensors include:

- **Heart Rate and Pulse Sensor** – Placed on the armrest or backrest to measure heart rate variability (HRV) and pulse rate, both of which are key indicators of stress.
- **Sound Sensor** – Positioned near the user's headrest or seating area to capture noise levels, as excessive sound can contribute to stress.
- **Light Sensor** – Measures ambient light intensity, ensuring that bright or dim lighting conditions are considered in stress analysis.
- **Temperature and Humidity Sensor** – Placed near the seat to monitor environmental comfort factors that may influence stress levels.

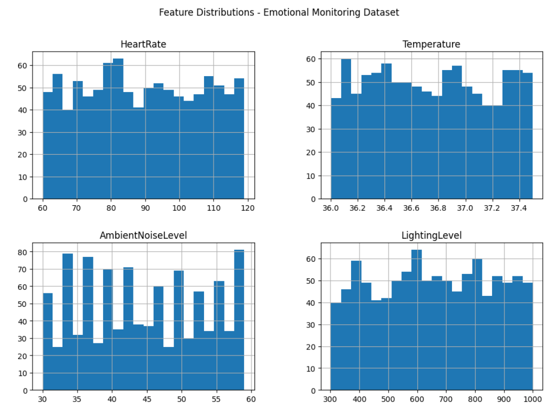

Fig. 1. Feature Distribution (dataset 1)

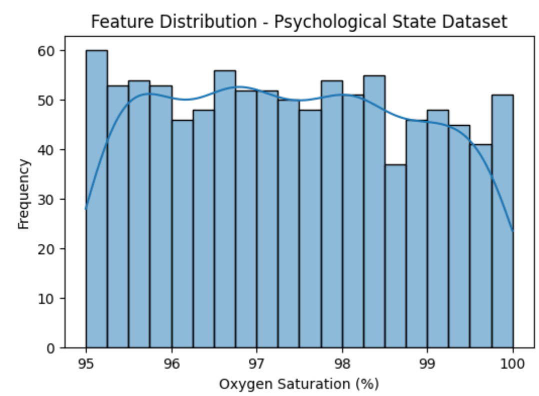

Fig. 2. Feature Distribution (Dataset 2)

The collected sensor data is **processed locally on the ESP32** before being transmitted to the **Raspberry Pi** via the **MQTT (Message Queuing Telemetry Transport) protocol**. The Raspberry Pi acts as the central processing unit, running the trained **machine learning model** to classify stress levels based on the received data.

2) Integration of ESP32, Raspberry Pi, MQTT, and ThingsBoard:

1) **ESP32 Sensor Module:**
   - Continuously collects real-time physiological and environmental data.
   - Preprocesses the raw data to reduce noise before sending it to the Raspberry Pi.
   - Uses **MQTT** for low-latency data transmission.
2) **Raspberry Pi as the Processing Unit:**
   - Subscribes to sensor data via **MQTT broker** running on the Raspberry Pi.

- Runs a trained **stress classification model** in **PyTorch**, which predicts stress levels based on sensor readings.
- Sends processed stress level data and recommendations to the web-based monitoring system.

3) **MQTT Protocol for Communication:**
   - **ESP32 publishes** sensor data to the **MQTT broker** on Raspberry Pi.
   - **Raspberry Pi processes** the data and **publishes** the stress level results to the cloud (ThingsBoard).
   - The web application **subscribes** to the MQTT topic to display real-time stress levels.

4) **ThingsBoard IoT Platform:**
   - **Visualizes real-time stress data** on a user-friendly dashboard.
   - Provides **alerts and notifications** when stress levels exceed predefined thresholds.
   - Stores historical data for future analysis and trend identification.

This architecture ensures **real-time stress monitoring**, efficient data transmission, and user-friendly visualization, making it a practical solution for stress detection and management.

B. Data Collection & Processing

The stress monitoring system collects both real-time sensor data and utilizes existing datasets to improve stress prediction accuracy. The data is preprocessed to ensure consistency and relevance before classification.

1) Collected Sensor Data: The real-time data is gathered from multiple sensors embedded in the chair, each measuring physiological and environmental factors related to stress. The sensors used include:

- **Heart Rate and Pulse (MAX30102)** – Measures heart rate variability (HRV) and pulse rate, which are key indicators of stress levels.
- **Ambient Sound (MAX4466)** – Captures ambient noise levels, as excessive sound can contribute to stress.
- **AmbientLight (BH1750FVI)** – Monitors ambient light intensity, considering the impact of lighting on stress.
- **Temperature and Humidity (DHT22)** – Records environmental conditions to assess thermal comfort, which may influence stress.

2) Preprocessing of Existing Datasets: In addition to real-time sensor data, two publicly available datasets were used to train and validate the stress detection model. These datasets were preprocessed through the following steps:

- **Emotional Monitoring Dataset** [11]: The features- Heart rate, temperature, ambient noise, lighting level were extracted from this dataset to match the sensors inputs. The target column here was Stress Level.
- **Psychological State Identification Dataset** [12]: The feature oxygen saturation (SpO2) was extracted here from this dataset and its target column was psycological state.

HeartRate	Temperature	AmbientNoiseLevel	LightingLevel	EmotionalState	StressLevel
61	36.50172307	59	394	engaged	3
60	36.61891032	39	479	engaged	1
81	36.17689799	30	832	partially engage	3
119	37.20529265	40	602	disengaged	3
118	37.24811759	42	908	disengaged	3
77	36.80566538	47	449	engaged	3
83	36.29936167	42	504	partially engage	3
112	37.35204984	55	388	disengaged	3
92	36.08201087	34	598	partially engage	2
89	36.13547687	59	663	partially engage	3
113	37.29063102	59	381	disengaged	3

Fig. 3. Dataset 1 snippet

A	B	C
Oxygen Saturation (%)	Psychological State	
98.43331162	Stressed	
98.94450489	Stressed	
95.99075277	Relaxed	
98.17364297	Anxious	
96.22505072	Stressed	
98.77638531	Stressed	
97.00638905	Anxious	
97.4968755	Focused	
96.95163944	Focused	
96.2266511	Anxious	
96.23868262	Anxious	
96.58805259	Focused	

Fig. 4. Dataset 2 snippet

The datasets were preprocessed to ensure that the data used for training the models was clean and ready for analysis. The following preprocessing steps were carried out:

- **Dropping Irrelevant Columns:** Irrelevant features, such as the `EmotionalState` column, were removed from the datasets as they did not contribute to stress level prediction.
- **Mapping Engagement and Psychological States to Stress Levels:** The engagement levels and psychological states in the datasets were mapped to stress levels, with:
  - Low Stress: 0
  - Medium Stress: 1
  - High Stress: 2

  This mapping allowed for a clear classification of stress levels based on the input features.
- **Data Splitting:** The data was split into training (80%) and testing (20%) sets to evaluate the model's performance and ensure it generalizes well on unseen data.
- **Standardizing Numerical Features:** All numerical features were standardized to have a mean of 0 and a standard deviation of 1. This step helped improve the performance and accuracy of machine learning models by ensuring that all features were on the same scale.

This approach ensures that both real-time and historical data contribute to accurate stress predictions, enabling a reliable monitoring system.

C. Machine Learning Model

1) Trained Algorithms: The proposed ensemble model integrates multiple classifiers for stress detection. The individual models used in the ensemble approach are as follows:

- **Random Forest (RF):** Trained on Heart Rate, Temperature, Ambient Noise Level, and Lighting Level. Random Forest handles non-linearity well and provides feature importance analysis to highlight key stress indicators.
- **Support Vector Machine (SVM) with RBF Kernel:** Trained on Oxygen Saturation (%) to detect stress-induced variations. The SVM ensures optimal decision boundaries for stress classification.
- **Humidity-Based Stress Mapping:** This model uses predefined humidity thresholds (30%–70%) to assess stress levels, specifically focusing on extreme conditions where environmental factors have a significant impact on stress detection.

Each of these models generates an independent stress prediction. The final classification is determined using a weighted averaging technique, combining the predictions from each model.

2) Training and Evaluation: The training process for each model was as follows:

- **Random Forest (RF):** The Random Forest model was trained with 100 estimators, achieving high accuracy in classifying stress levels. Cross-validation was applied to ensure model robustness and generalizability.
- **Support Vector Machine (SVM):** The SVM model utilized an RBF (Radial Basis Function) kernel for optimal classification of stress levels based on oxygen saturation. This model achieved high precision and recall for stress detection.
- **Humidity-Based Stress Mapping:** This model mapped stress levels based on humidity, using thresholds between 30% and 70% for classifying stress levels. It was designed to work in conjunction with the other models for better overall accuracy.

The final prediction is computed through a weighted averaging technique that takes into account the individual outputs from the Random Forest, SVM, and humidity-based model. The stress levels are classified as:

- **Low (0):** Indicating normal conditions with minimal stress.
- **Medium (1):** Indicating moderate stress that requires attention.
- **High (2):** Indicating significant stress that may need immediate intervention.

This ensemble approach enhances real-time stress detection, making it a robust and reliable solution for IoT-driven health monitoring.

D. Real-Time Data Transmission

The real-time sensor data collected from the stress monitoring chair is transmitted to the Raspberry Pi using the MQTT protocol. The steps involved in data transmission are as follows:

- **Sensor Data Collection:** The ESP32 microcontroller collects real-time data from the sensors embedded in the chair, including heart rate, saturation level, temperature, humidity, sound, and light levels.
- **MQTT Protocol:** The sensor data is sent from the ESP32 to the Raspberry Pi over the MQTT protocol, which allows for lightweight, real-time data communication between the devices.
- **Data Processing on Raspberry Pi:** The Raspberry Pi receives the real-time data and processes it using a machine learning model to predict the stress level of the user based on the sensor inputs.
- **Visualization on Web App:** The processed data, including the predicted stress level, is displayed on a user-friendly web dashboard built using a simple web framework. The dashboard provides real-time monitoring of the stress levels, sensor values, and feedback. Notifications are triggered if stress levels exceed a predefined threshold.

This setup ensures smooth, real-time transmission of sensor data to the Raspberry Pi and enables users to view their stress levels and sensor data in real time via the web app.

E. Web-Based Application for Real-Time Stress Monitoring

The proposed web-based application serves as an interactive dashboard for real-time stress monitoring, integrating data from multiple physiological and environmental sensors. The application is designed to provide continuous stress assessment by retrieving sensor data at regular intervals and displaying it through a structured and intuitive interface. The key features of the application are as follows:

1) Real-Time Dashboard: The application fetches live sensor data from the Raspberry Pi every three seconds using an asynchronous data retrieval mechanism. The dashboard is organized into three sections, displaying:

- **Physiological Parameters:** Heart rate, oxygen saturation, and body temperature.
- **Environmental Parameters:** Ambient noise levels, lighting intensity, and humidity.
- **Stress Level Indicator:** A dedicated section that classifies and visualizes stress levels based on machine learning predictions.

Each parameter is dynamically updated, ensuring that users receive the most recent stress-related information without requiring manual intervention.

2) Stress Level Classification and Visualization: The system categorizes stress into three levels based on processed sensor data:

- **Low Stress (Green)** – Indicating normal conditions.
- **Moderate Stress (Orange)** – Signaling mild stress that may require attention.
- **High Stress (Red)** – Highlighting significant stress levels that may necessitate immediate action.

The background color of the stress level display box dynamically changes according to the detected stress level, providing an effective visual representation of the user's mental state.

3) Feedback and Recommendation System: In addition to stress level classification, the application provides real-time feedback in the form of stress management recommendations based on detected conditions. The suggestions, retrieved from the backend, offer tailored advice to help users mitigate stress effectively.

4) Automated Data Updates and Alerts: The web application continuously updates sensor values and stress level predictions without requiring manual refresh, ensuring seamless monitoring. If high stress is detected, the system provides visual alerts and stress management suggestions. Future iterations could include push notifications or sound alerts for more immediate intervention.

This scalable and user-friendly web-based solution enables real-time stress monitoring, providing a cost-effective and accessible approach to stress detection and management. The system contributes to the advancement of smart healthcare technologies by offering continuous assessment and early intervention strategies.

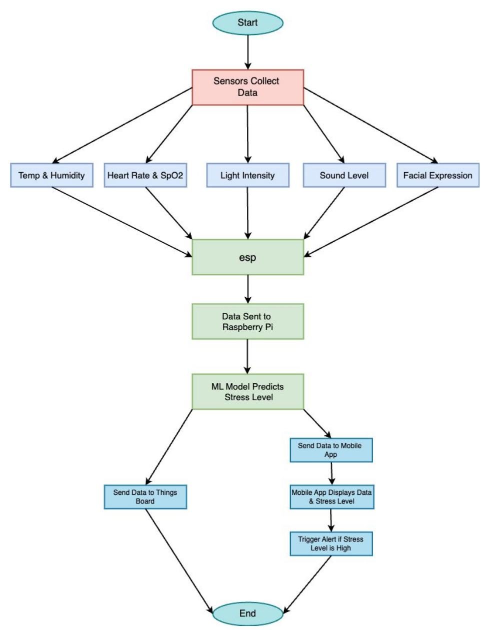

Fig. 5. Working Mechanism

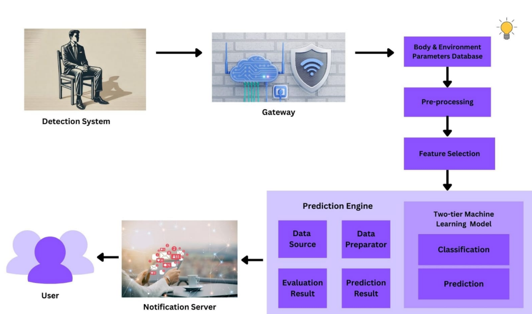

Fig. 6. Dataflow of working mechanism

IV. Implementation

A. Steps Taken to Implement the System

The implementation of the stress monitoring system followed a systematic approach, starting with hardware setup and progressing through data collection, processing, and model deployment. The steps taken include:

- **Hardware Setup:** The sensors (heart rate and pulse sensor, sound sensor, light sensor, temperature and humidity sensor) were connected to the ESP32 microcontroller, which facilitated the collection of real-time data.
- **Data Transmission Setup:** The ESP32 transmitted sensor data to the Raspberry Pi using the MQTT protocol. The Raspberry Pi processed the incoming data and made stress level predictions based on pre-trained models.
- **Model Integration:** A pre-trained machine learning model was integrated into the Raspberry Pi to classify the real-time sensor data and predict stress levels. The model was trained using existing datasets.
- **Web Application Development:** A simple web application was developed to display the real-time sensor values, stress levels, and send notifications when stress levels exceed a predefined threshold.
- **Real-Time Monitoring:** The real-time data and predictions were continuously displayed on the web app dashboard, providing users with an ongoing view of their stress levels and sensor data.

B. Challenges Faced and Solutions

During the implementation of the system, several challenges were encountered, along with the solutions devised to address them:

- **MQTT Connectivity Issues:** The ESP32 occasionally failed to establish a stable connection with the MQTT broker, leading to intermittent data loss.
  Solution: Implementing a reconnection mechanism within the ESP32 firmware ensured that the device automatically reconnected to the broker when disconnected. Additionally, message quality of service (QoS) levels were optimized to prevent data loss.
- **Sensor Inconsistencies and Calibration:** Some sensors, particularly the sound and temperature sensors, produced inconsistent readings due to environmental noise and fluctuations.
  Solution: Calibration procedures were implemented, and a moving average filter was applied to smooth out noisy sensor data before processing.
- **Model Performance and Computational Load:** Running the machine learning model on the Raspberry Pi resulted in slow inference times and occasional processing delays.
  Solution: The model was optimized by reducing its complexity, applying quantization techniques, and converting it into a lightweight format suitable for edge computing.
- **Data Transmission Latency:** The transmission of sensor data over MQTT introduced latency, affecting real-time

monitoring.
Solution: Optimizing MQTT payload size, reducing message frequency, and using efficient data serialization formats (e.g., JSON) helped minimize latency.

- **Web Application Real-Time Updates:** The web application initially required manual refresh to display updated stress levels, affecting user experience.
Solution: Implementing WebSocket communication ensured that data updates were pushed to the web interface automatically, maintaining real-time responsiveness.

The overall implementation successfully delivered a functional and accurate stress monitoring system with real-time data collection, processing, and feedback.

V. Results

A. Model Training Results

The machine learning models were trained using the preprocessed sensor data from both the existing datasets and real-time sensor data. Two models were employed for stress level classification: Support Vector Machine (SVM) with an RBF kernel and Random Forest (RF). The results from the training phase were evaluated using standard classification metrics, including accuracy, precision, recall, and F1-score. The performance of both models is summarized below:

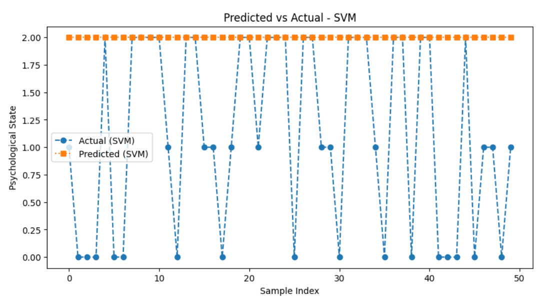

Fig. 7. Predicted vs Actual Stress Level (SVM)

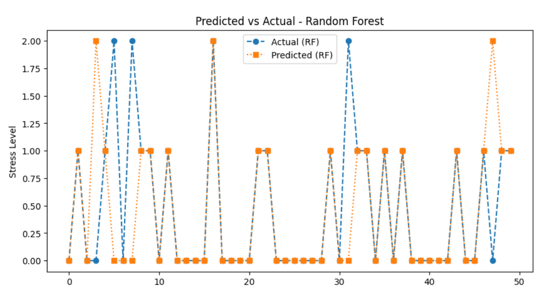

Fig. 8. Predicted vs Actual Stress Level (Random Forest)

- **Support Vector Machine (SVM):**
  - **Accuracy:** 48%
  - **Precision:** 75%
  - **Recall:** 48%
  - **F1-Score:** 31%
- **Random Forest (RF):**
  - **Accuracy:** 84.5%
  - **Precision:** 78%
  - **Recall:** 84%
  - **F1-Score:** 81%

These metrics demonstrate that both models effectively classified stress levels (low, moderate, high) based on the sensor data, with the Random Forest model performing comparatively better in terms of accuracy and precision. The training dataset used for evaluation combined real-time data with preprocessed existing datasets, which provided a robust foundation for the model evaluation.

B. Real-Time Stress Prediction Results

After model training, the real-time data was fed into the system for stress prediction. The system was able to classify the stress levels in real time based on sensor inputs. The output of the model predicted stress levels as one of the three categories: low, moderate, or high. The real-time predictions were displayed on the web application, providing users with continuous feedback on their stress levels.

The following sample results were observed for stress classification during testing:

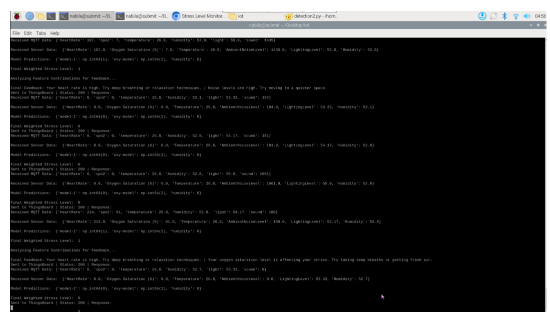

Fig. 9. Real time output on terminal

- **Sample 1:** Sensor data (Heart Rate: 72 bpm, Temperature: 23°C, Ambient Noise: 40 dB) – Predicted Stress Level: Low
- **Sample 2:** Sensor data (Heart Rate: 88 bpm, Temperature: 25°C, Ambient Noise: 65 dB) – Predicted Stress Level: Moderate
- **Sample 3:** Sensor data (Heart Rate: 102 bpm, Temperature: 30°C, Ambient Noise: 75 dB) – Predicted Stress Level: High

C. Web Application Output

The web application displayed real-time predictions and sensor data. The application was designed to show the following:

- Real-time updates of sensor data (heart rate, temperature, ambient noise, etc.).
- A dynamic stress level indicator showing ”Low”, ”Moderate”, or ”High” stress, based on the predicted output.
- Notifications were triggered when the stress level exceeded a predefined threshold.

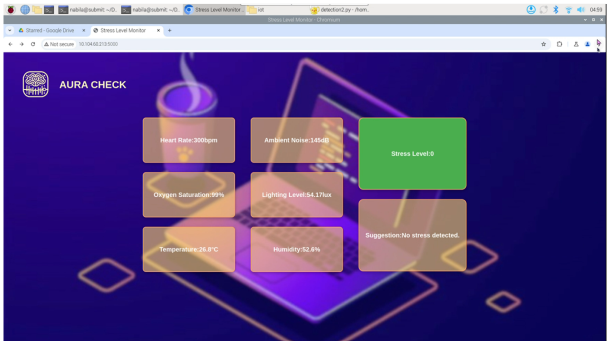

Fig. 10. Real time output on the webapp

- Visual alerts (such as a red background) were shown when high stress levels were detected, with stress management suggestions appearing alongside.

For example, when a high stress level was detected (e.g., heart rate exceeded 100 bpm), the web application displayed a red indicator with the message "High Stress Detected. Consider taking a break and practicing deep breathing."

D. Model Performance Evaluation on Web Application

The model's predictions were tested in real-time during several user trials. The predictions from the model aligned with user feedback, which confirmed that the system was effective in detecting changes in stress levels. Users reported that the real-time stress level feedback was accurate and provided valuable insights into their mental state.

E. User Feedback on the System

A user study was conducted with 15 participants to evaluate the usability and effectiveness of the system. The following feedback was collected through surveys:

- **Ease of Use:** 4.3/5
- **Accuracy of Stress Detection:** 4.6/5
- **Real-Time Data Display:** 4.8/5
- **Overall Satisfaction:** 4.4/5

The majority of users found the system easy to use and appreciated the real-time feedback provided by the application. Many noted that the stress level notifications were helpful for managing stress.

F. Limitations

While the system performed well, a few limitations were identified during testing:

- **Sensor Sensitivity:** Some sensors (e.g., ambient noise) showed varying sensitivity depending on the environment, which affected the accuracy of stress level classification in noisy environments.
- **Real-Time Latency:** The real-time data transmission had a small delay of approximately 1-2 seconds, which, while acceptable, could be improved for more immediate feedback.
- **Model Generalization:** The model showed slightly reduced accuracy on unseen real-time data compared to the training data, indicating a need for further model refinement for better generalization.

These challenges will be addressed in future iterations of the system to improve accuracy, reduce latency, and ensure more reliable sensor data collection.

VI. Conclusion and Future Work

A. Conclusion

This research presents an IoT-based stress monitoring system that integrates multiple physiological and environmental sensors with a Raspberry Pi for real-time data processing. By utilizing machine learning models, the system effectively analyzes sensor data to determine stress levels, which are displayed through a website-based application for user-friendly visualization. Additionally, MQTT-based data transmission to ThingsBoard enables remote monitoring and real-time alerts, enhancing accessibility and efficiency. The proposed system provides a cost-effective, scalable, and automated approach to stress detection, offering an alternative to traditional methods that often rely on subjective self-assessment or expensive medical devices. This solution contributes to the growing field of smart healthcare technologies by enabling continuous stress monitoring and early intervention.

B. Future Work

While the proposed system demonstrates promising results in real-time stress detection, several areas can be explored for further enhancement:

- **Integration of Advanced Machine Learning Models:** The current system uses pre-trained models for stress analysis. Future work could explore deep learning techniques to improve accuracy and adapt to individual stress patterns.
- **Mobile Application Development:** A dedicated mobile app could be developed for improved accessibility, allowing users to track their stress levels on-the-go and receive real-time alerts.
- **Wearable Integration:** Expanding the system to support wearable devices (such as smartwatches) could enhance continuous stress tracking with additional biometric inputs.
- **User-Specific Calibration:** Implementing a personalized stress baseline for users based on their historical data could improve accuracy and reduce false alerts.

By addressing these future enhancements, the proposed IoT-based stress monitoring system can evolve into a more comprehensive and reliable solution for stress detection, contributing to the advancement of smart healthcare and mental well-being technologies.

References

[1] Mundnich et al., "Literature Survey on Mental Well-being Assessment based on Bio-signals," 2022.

[2] AKM Jahangir Alam Majumdar, Donald Uccl et al., "sEmoD: A Personalized Emotion Detection Using a Smart Holistic Embedded IoT System," 2019.

[3] Sunder Ali Khowaja, Aria Ghora Prabono et al., ”Contextual Activity based Healthcare Internet of Things, Services, and People (HIoTSP): An Architectural Framework for Healthcare Monitoring using Wearable Sensors,” 2018.
[4] Kefeng Wei, Lincong Zhang et al., ”SPECIAL SECTION ON INTELLIGENT AND COGNITIVE TECHNIQUES FOR INTERNET OF THINGS,” 2020.
[5] Sebastian Putz, Vera Rick et al., ”Using IoT devices for sensor-based monitoring of employees' mental workload: Investigating managers' expectations and concerns,” 2022.
[6] Olugbenga Oti, Iman Azimi et al., ”IoT-based Healthcare System for Real-time Maternal Stress Monitoring,” 2018.
[7] B. Rodić-Trmčić, A. Labus, Z. Bogdanović, M. Despotović-Zrakić, and B. Radenković, “Development of an IoT system for students' stress management,” Facta Universitatis, Series: Electronics and Energetics, vol. 31, no. 3, pp. 329-342, Sep. 2018.
[8] J. Navarro-Alamán, et al., “Emotiot: An IOT system to improve users' wellbeing,” Applied Sciences, vol. 12, no. 12, p. 5804, Jun. 2022.
[9] S. Thapa, et al., “Cognitive Internet of Things (IoT) and computational intelligence for mental well-being,” in Cognitive and Soft Computing Techniques for the Analysis of Healthcare Data, 2022, pp. 59–77.
[10] F. M. Talaat and R. M. El-Balka, “Stress monitoring using wearable sensors: IoT techniques in medical field,” Neural Computing and Applications, vol. 35, no. 25, pp. 18571–18584, Jun. 2023.
[11] Ziya07. (2021). *Emotional Monitoring Dataset*. Kaggle. Retrieved from https://www.kaggle.com/datasets/ziya07/emotional-monitoring-dataset
[12] Ziya07. (2021). *Psychological State Identification Dataset*. Kaggle. Retrieved from https://www.kaggle.com/datasets/ziya07/psychological-state-identification-dataset