RedziSENS Corp.

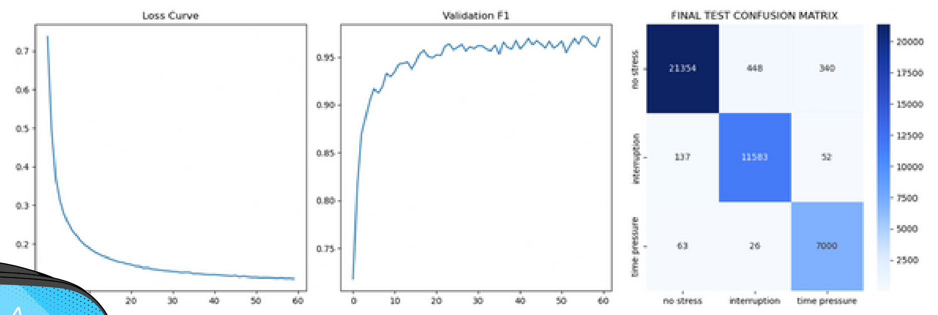
STRESS-LEVEL MANAGEMENT VIA SMARTWATCH MONITORING

Project Summary and Objectives

- Primary Objective - development of a high-fidelity, real-time stress detection system designed for local execution on wearable devices.
- Focused on privacy and low latency.
- Utilizes a hybrid CNN-GRU architecture
- Achieved a high F1 score of 0.99.
- Powered by Firebase for live monitoring and data visualization.

Model Development and Evaluation

- Conducted “tournament-style” comparison of 8 deep learning architectures.
- Evaluated several candidates against baseline models, including LSTM, ResNet, pure CNN and pure GRU.
- Pure GRU exhibited overconfidence in reliability diagrams, while pure CNN lacked the necessary temporal memory to accurately track stress.
- Hybrid CNN-GRU selected as optimal architecture
- Outcome - achieved a high performance and mobile-efficient model



The Machine Learning Pipeline

- Utilized raw sensor data from the SWELL dataset to establish stress monitoring.
- Implemented a comprehensive cleaning and normalization workflow to ensure data consistency and quality.
- Performed feature selection by streamlining input space from 16 to 5 critical features, minimizing computational load.
- Conducted comparative analysis between baseline models and more advanced architectures.
- Finalized the model by exporting to TorchScript.

Engineering and Implementation

- Developed the wearable application in Native Kotlin to overcome Tizen OS limitations and access raw sensor streams.
- Engineered a specific BPM cap (40-180) to filter out motion artifacts and protect the model from sensor noise.
- Integrated Firebase Real-Time Database to facilitate instant, zero-latency dashboard updates without page refreshes.

Ethical Consideration and Commercialization

- Enforced "Privacy by Design" by processing all data locally, ensuring sensitive biometrics never leave the device.
- Implemented ethical safeguards to prevent "anxiety loops" and distinguish between distress and healthy exercise.
- Expanded commercial reach to corporate burnout monitoring and users without wearables via a web survey.

Reflections and Future Work

- Successfully pivoted from cross-platform frameworks to native Kotlin development to ensure deep hardware integration and system reliability.
- Implemented custom BPM capping logic (40-180) for outliers in sensor data.

- Combine the dashboard and web survey into a unified user interface.
- Refine the data processing pipeline to minimize or eliminate the initial 30-sample buffering delay.

Conclusion

- Successfully transitioned a data science experiment to functional prototype
- Employed a dual-pathway approach to maximize platform accessibility and context.
- Demonstrated that complex AI can be optimized for hardware without sacrificing accuracy.