Combining Multiple tinyML Models for Multimodal Context-Aware Stress Recognition on Constrained Microcontrollers

Michael Gibbs*, Kieran Woodward* and Eiman Kanjo*†,
*Smart Sensing Lab, Department of Computer Science, Nottingham Trent University, Nottingham, UK
†Department of Computing, Imperial College London, London, UK

Abstract—As stress continues to be a major health concern, there is growing interest in developing effective stress management systems that can detect and mitigate stress. Deep Neural Networks (DNNs) have shown their effectiveness in accurately classifying stress, but most existing solutions rely on the cloud or large obtrusive devices for inference. The emergence of tinyML provides an opportunity to bridge this gap and enable ubiquitous intelligent systems. In this paper, we propose a context-aware stress detection approach that uses a microcontroller to continuously infer physical activity to mitigate motion artifacts when inferring stress from heart rate and electrodermal activity. We deploy two DNNs onto a single resource-constrained microcontroller for real-world stress recognition, with the resultant stress and activity recognition models achieving 88% and 98% accuracy respectively. Our proposed context-aware approach improves the accuracy and privacy of stress detection systems while eliminating the need to store or transmit sensitive health data.

Introduction

In recent years, stress has become a major health concern affecting individuals across different age groups and professions. Stress can lead to a wide range of physical and mental health issues, such as anxiety, depression, and cardiovascular disease [8]. As a result, there is a growing interest in developing effective stress management systems that can detect and mitigate stress in real-world environments, with many adopting artificially intelligent methods.

Advances in deep learning are resulting in ever expanding capabilities and applications ranging from voice assistants to autonomous driving. However, the successful deployment of a Deep Neural Network (DNN) relies upon two stages: training and inference. Much of the literature focuses on the development and training of DNNs but not the real-world inference. This is often due to the fact that DNNs are difficult to employ, as AI-based solutions commonly require a large amount of computational power. This is often achieved using cloud-based solutions or systems that are impractical for deployment.

The edge computing paradigm allows small devices to be placed close to the end-user at the point of data collection, such as sensors, etc., to quickly and locally process data. For example, microcontrollers enable embedded systems in a wide range of real-world applications such as fitness trackers, smart home devices, industrial sensors, and medical devices. The emergence of edge computing and DNNs paves the way for ubiquitous intelligent systems. In recent years, the use of edge computing devices has increased significantly; this is due to the many advantages that edge computing offers over traditional centralised architectures. Some of the main advantages of edge computing include the following:

- Lower latency
- Increased privacy
- Improved scalability

DNNs have shown their effectiveness in accurately classifying stress, but most existing solutions rely on the cloud for inference. Edge computing has the potential to aid real-world stress recognition, as

Digital Object Identifier 10.1109/XXX.0000.0000000

bridging these capabilities to embedded devices can reduce the amount of data stored and transmitted, hence increasing the privacy of sensitive health data. Photoplethysmography (PPG) sensors enable the non-invasive monitoring of Heart Rate (HR). Automatic analysis of PPG has rendered it valuable in clinical and non-clinical settings. However, tracking heart rate with PPG is difficult due to motion artifacts, which are major causes of signal degradation, as they obscure the location of the heart rate peak in the spectra. Therefore, context recognition in the form of detecting physical activity is essential to improve the performance of stress detection. By integrating exercise recognition, stress detection systems can become more reliable by mitigating any motion artifacts from the signal.

Human Activity Recognition (HAR) is the task of inferring actions carried out by a person and is an increasingly popular research topic with the potential to improve healthcare, which can allow for more accurate stress detection. However, to implement such systems on a large scale, efficient and low-power intelligent algorithms that can run on low-cost, resource-constrained microcontrollers should be developed.

AI on the Edge research is still in its infancy and the trade-off between classification accuracy and embedded device constraints must be assessed. Most edge computing research frequently uses larger devices, such as Raspberry Pis or smartphones; however, many real-world applications, such as wearable devices for stress recognition, require smaller electronics as they are to be worn on-body. These tiny microcontrollers offer an ideal opportunity for real-world, on-device inference. However, they do present additional challenges, as they are resource- and power-constrained. In this paper, we present the deployment of a novel multimodal multi-*model* approach applied to context-aware stress detection, where stress inference only occurs if no physical activity is inferred from a separate classification model. All of which is performed on-device using an ultra-low power, low-cost and resource-constrained microcontroller. This tinyML approach applied to both models will demonstrate the potential of using multiple models while also leveraging several input modalities. The contributions of this work are as follows:

- The novel combination of two models to provide real-time contextual activity recognition information to improve the accuracy of stress inference.
- The deployment of the context-aware multi-model approach on an ultra-low power, resource-constrained microcontroller for real-world on-device inference.

The remainder of the paper is organised as follows; Section 2 outlines the background, Section 3 discusses the methodology, Section 4 outlines the results, Section 5 offers a discussion, and Section 6 concludes.

Background

Many existing DNNs have high computational cost and memory requirements and, therefore, rely on centralised servers. However, it is more efficient to process the data on-device close to the data source. This section highlights previous research in HAR as well as stress detection, while also relating such technologies back to edge computing.

Human Activity Recognition

Extensive research has shown the proficiency of sensor-based configurations at classifying HAR. Wearable sensors attached to users are commonly used for data collection, these sensors include accelerometers, gyroscopes and magnetometers and are often embedded within smartphones and smartwatches.

Sensor-based HAR is a time-series classification problem. Deep Learning (DL) enables high-level feature representations to be automatically learned from raw data. Both convolutional neural networks (CNN) and long short-term memory (LSTM) networks have shown promise for HAR. CNNs have been widely used to learn spectral patterns of sensor signals, whereas LSTMs have been used to capture temporal dependencies.

HAR is a common application area when developing machine learning on low-power devices, as a CNN-based feature learning approach for HAR has been developed [12]. This approach used IMU data from 20 healthy subjects to classify walking, walking upstairs, walking downstairs, sedentary, and sleeping, achieving 96.4% accuracy. A considerable computational speedup was achieved using the proposed approach compared to SVM and MLP. The model was deployed onto a smartphone for real-world use. A 1D CNN-based method for human activity recognition has also been developed [7]. Activity data was collected using a smartphone's IMU, including walking, running, and standing still. Acceleration data were converted to vector magnitude data and used as input to the 1D CNN achieving an accuracy of 92.71%.

Approaches have been developed to improve classification on constrained microcontrollers. Optimisation techniques, such as pruning and quantisation have enabled model size to reduced by 10 times without severely impacting model accuracy [4]. However, the techniques developed have had little testing on-device to evaluate real-world performance. Despite numerous approaches claiming to develop new approaches for edge computing, few have deployed and tested their models on-device to explore the real-world impact. However, Ghibellini et al. [3] have shown the possibility of classifying human activities on the edge using an ultra-low power microcontroller using quantisation. A 1D CNN was used to develop a classification model to infer three activities (running, falling, and normal state). The model was deployed on an Arduino BLE 33 Sense and achieved 97% accuracy with quantisation reducing the model size by 53%. Similarly, a CNN has been used to classify five specific movements in bed (agitation, idle, in bed, out of bed, and bed movement) for the elderly [9]. The model was deployed to an nRF52 development board achieving 88.96% accuracy. However, the data used in this study were collected from only one person, creating bias.

While edge computing includes powerful devices such as smartphones, the use of ultra-low power microcontrollers show greater potential due to the ability to add external sensors such as for physiological monitoring. Furthermore, although HAR research has made extensive use of smartphones for on-device processing, it is not feasible to use a smartphone for the processing of external sensors, such as physiological sensors required for stress detection.

Stress recognition

Non-invasive physiological sensors such as Electro-Dermal Activity (EDA) and Heart Rate (HR) present a significant opportunity to assess stress in the real-world due to their direct correlation with the sympathetic nervous system. EDA and HR have been shown to detect poor mental wellbeing, such as stress, using a 1D CNN with 92.3% accuracy, outperforming comparative models such as LSTM [11]. Similarly, EDA and Heart Rate Variability (HRV) were used in a wearable device to measure stress during driving [5]. The wearable device took measurements over a 5-minute period to detect stress levels with an accuracy of 97.4% and found that HRV and EDA are highly valuable. EDA and HR signals have also been used to infer stressed and relaxed states using K Nearest Neighbour and Fisher discriminant analysis, achieving 95% accuracy stating that 5-10 second intervals are suitable for real-time stress detection [1].

The ability to use non-invasive sensors to measure HR and EDA allows small devices to accurately determine stress levels in real-time and should be further utilised to detect stress. However, physiological signals do not account for the context in which devices are used, as context can play a significant role in perceived stress levels meaning additional sensors are required. In particular, physical activity can significantly alter human physiology including HR and EDA [10], showing consideration is needed to not infer stress when exercising to avoid false positives. Therefore, we propose an edge computing approach to infer stress only when no physical activity is detected.

The ability to classify sensor data using small microcontrollers offers many opportunities for real-world inference due to their small footprint. However, edge computing still faces many challenges such as the constrained nature of the devices and the high memory requirement of AI models. It is a challenging proposition to overcome many of these challenges; however, advances in software techniques such as quantisation and the use of context aim to improve the classification of models on limited hardware. Overall, while there has been much research on effective DL models for stress classification, they have rarely been deployed on microcontrollers, and there has been no consideration of how human activity context could improve the performance of stress modelling.

Methodology

This work outlines our tiny Machine Learning (tinyML) multimodal multi-*model* approach combining an activity recognition classifier with stress detection. The multi-model context-aware system consists of the following key components:

- Inertial Measurement Unit (IMU) sensors - The built-in IMU of the Arduino Nano 33 Sense is used to collect 3-axis accelerometer data for human activity recognition.
- HR and EDA sensors - These external sensors connect to the microcontroller to provide physiological signals for stress detection.
- HAR model - A CNN classifies each incoming accelerometer sample as either resting or active.
- Stress detection model - A separate CNN acts on physiological data to classify stress, but only once resting state is inferred.
- Microcontroller - The Arduino Nano 33 Sense provides computing power, sensor connectivity and executes both models.

We assume the wearable sensors offer clean unaffected signals and consistent placement across participants. For modelling, we assume the CNN architectures generalise well to new data based on validation results. The model architectures are constrained to relatively simple 1D CNNs given the hardware limita-

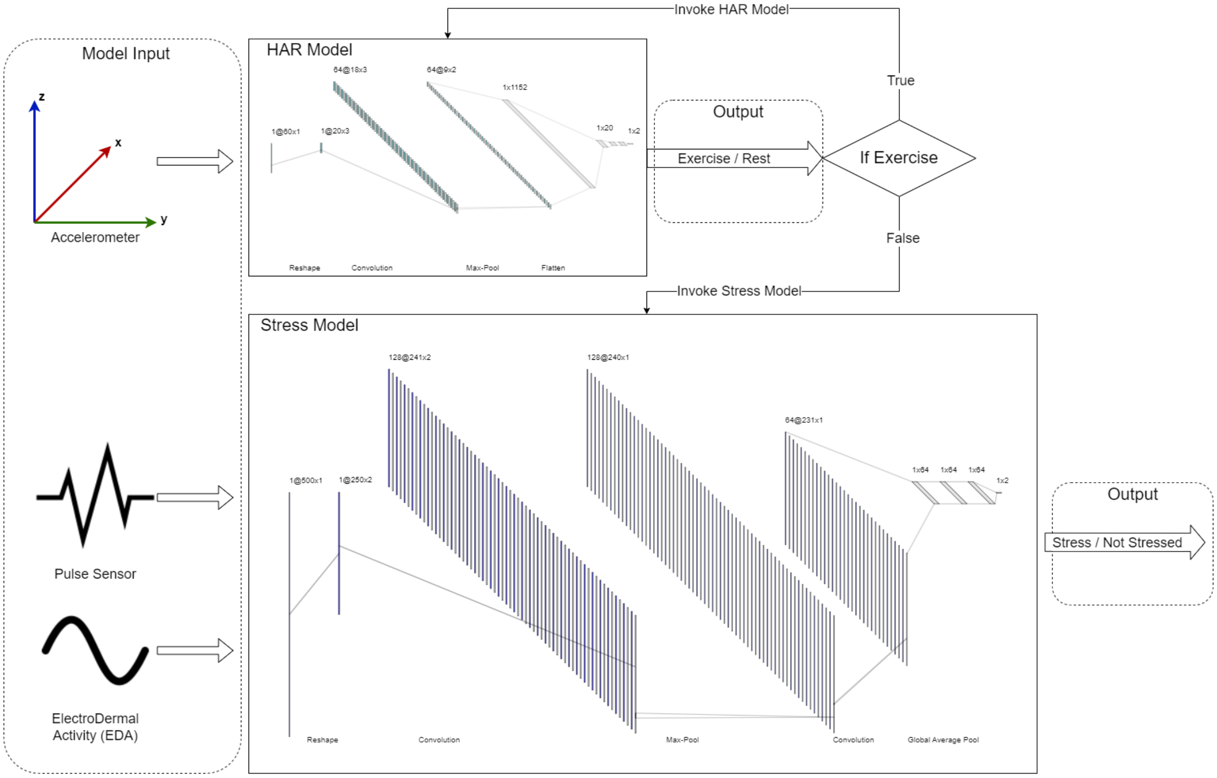


FIGURE 1. The proposed system containing the HAR model (top) and stress model (bottom), along with the internal logic.

tions. Only a single context factor of physical activity was used to trigger stress classification rather than exploring multiple contextual variables.

Evaluation is performed by simultaneously collecting all sensor data streams but withholding the physiological data from the stress model until inactive periods are detected, mirroring real-world operation. The same datasets are used for both training and evaluation.

Datasets

In order to complete the multi-model context-aware approach, a dataset was required for both the HAR and stress recognition to train the classifiers.

HAR dataset We used the WISDM: Wireless Sensor Data Mining dataset [6]. The dataset contains 3-axis accelerometer data sampled at 20Hz for walking, jogging, climbing upstairs, climbing downstairs, sitting, and standing. Data was collected from 29 participants using an Android phone in their front trouser leg pocket.

Preprocessing the data was required as the data was recorded using an Android phone, and the timestamp was recorded using the phone's up-time. Some rows of data had a timestamp, but no attached data. These rows were removed during the cleaning process.

Due to the requirement of only needing to infer when the user is performing physical activity or resting, we condensed the classes into exercise and rest. Walking, jogging, climbing upstairs and climbing downstairs were combined into the exercise class, while standing and sitting were combined to form the rest class.

Stress dataset A lab-based stressor experiment has been conducted in which participants' stress response was stimulated using the Montreal stress test [2]. This experiment induced stress in 20 healthy participants aged 18-50 between June and September 2019 as approved by Nottingham Trent University Human Ethics Board, application number 600. Participants wore hand-held non-invasive sensors on their fingers. The sensors recorded HR Beats Per Minute (BPM), raw HR amplitude, HRV, and EDA, each sampled at 30Hz to collect physiological data while experiencing relaxed and stressed states of mental wellbeing.

To allow the effects of stress and mental arithmetic to be investigated separately, the experiment had three test conditions; rest, control, and experimental. Each participant was initially briefed before completing a 3-minute rest period where participants looked at a static computer screen where no tasks were displayed. This was followed by 3 minutes of the control condition, where a series of mental arithmetic questions were displayed which participants answered, followed by another 3-minute rest period. The participants then completed the stressor experiment where the difficulty of the questions increased and the task time limit was adjusted to be 10% less than the average time taken to answer questions during the training, taking it just beyond the individual's mental capacity. The time pressure along with a progress bar showing their progress compared with an artificially inflated average were both designed to induce stress during the 10-minute experiment. A similar number of samples were collected from both relaxed and stressed data, helping to reduce bias in the classification model.

Micocontroller

The Arduino Nano 33 Sense is a small form factor, ultra-low power, and cost-effective microcontroller. It is based on the ARM Cortex-M4 32-bit processor, which is an energy-efficient microcontroller that is well-suited for a wide range of applications. Although the Cortex-M4 processor is capable of performing advanced calculations and processing, it has limited processing power.

In addition to its processing power limitations, the Arduino Nano 33 Sense also has limited memory with only 256KB of SRAM, which can make it difficult to store larger samples and complex models. This means that methods such as model compression and quantisation are required to reduce the size of their models and make them more suitable for deployment. We have therefore utilised TensorFlow Lite for Microcontrollers to develop small models capable of fitting on the microcontroller. While there are other microcontrollers that are more powerful, the Arduino Nano 33 Sense is an extremely low-cost microcontroller enabling much wider deployment and also has extensive support for TensorFlow Lite for Microcontrollers.

Sensors

A non-invasive sensor-based approach presents the most significant opportunity to assess stress, as sensors can be easily connected to microcontrollers and used inconspicuously in the real-world.

HR sensors are commonly used within wearable computing systems as they can be embedded within a wide range of devices due to their small footprint and provide insights into the autonomous nervous system. Therefore, the same PPG sensor used in the Montreal stressor data collection experiment was connected to the microcontroller to measure HR.

Similarly, the same EDA sensor from the data collection experiment has been connected to the microcontoller. EDA is often used to train affective models to classify stress as it is directly related to the sympathetic nervous system, which controls rapid involuntary responses to dangerous or stressful situations.

Finally, motion data measures movement through accelerometers, gyroscopes, and magnetometers which can be used to measure physical activity. The Arduino Nano 33 Sense has an LSM9DS1 IMU which includes a 3D digital linear acceleration sensor. This in-built accelerometer has been utilised for the real-world HAR inference.

Classification Features

Feature extraction is typically a crucial process when developing classification models. However, the memory limitations of the Arduino 33 Sense limit the ability to extract complex features.

Due to the constrained nature of the microcontroller, the only feature used from the PPG sensor was the raw HR amplitude along with the raw EDA values. Additional features such as BPM and HRV were examined, but when the model was trained using these features, it was too large to fit on the microcontroller in addition to the HAR model due to the limited 1MB of flash memory.

Similarly, for the activity recognition model, the raw 3-axis accelerometer values were used from the Arduino Nano 33 Sense on-board IMU sensor.

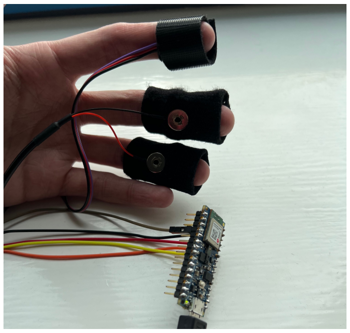


FIGURE 2. Person wearing the HR and EDA sensors connected to an Arduino Nano 33 Sense.

Network Architecture

DNNs provide higher accuracy compared to classic machine learning approaches and their sample-by-sample streaming capabilities better match real-time edge processing. Previous research shows the benefits of using CNNs for both HAR and stress detection. While LSTM networks are frequently used to classify time-series data, previous work shows CNNs can outperform LSTMs for time-series classification such as stress recognition [11]. Furthermore, our testing showed the resultant model size remains smaller when using 1D CNN, making them ideal for tinyML applications.

HAR Model A one-dimensional Convolutional Neural Network (1D CNN) was selected for the HAR model due to its high performance and smaller model sizes compared to LSTM. A CNN is composed of convolutional layers that employ a filter to slide over the one-dimensional time series data. The CNN architecture consists of a single convolutional layer, a dropout layer with a rate of 0.2 to prevent overfitting followed by three dense layers, and a softmax output layer. The high accuracy and automated feature learning of CNNs makes it better suited for generalised mobility detection, despite its higher complexity.

Stress Model Similarly, a 1D CNN was selected for the stress detection model. The original input data is partitioned into segments of fixed length. These data are segmented over an overlapping sliding window with an experimentally chosen window size of 256 samples and a step size of 24, after testing various window sizes ranging from 16 to 2056. The network architecture consists of two 1-dimensional convolutional layers, followed by max-pooling operations. Batch normalisation layers are incorporated, as well as a dropout layer with a rate of 0.2 to avoid overfitting, prior to the softmax activation function.

To reduce model size, the stress model was tested using only one convolutional layer. However, this resulted in a significant accuracy sacrifice of approximately 27%. Thus, two convolutional layers must be in the network architecture to achieve significant accuracy.

Model Quantisation

The Arduino Nano 33 Sense is a powerful and versatile microcontroller that is capable of performing a variety of machine learning tasks on-device. However, due to its limited memory, it is often necessary to use techniques such as model quantisation to reduce the size of machine learning models and make them more suitable for deployment.

Two commonly used policies for post-training quantisation (PTQ) are *int8* and *float16* quantisation, which are both methods to reduce the precision of floating-point numbers used in machine learning models. In a standard machine learning model, floating-point numbers are typically represented using 32 bits, which can make the model large and memory intensive. By using *float16* quantisation, the number of bits used to represent each floating-point number is reduced to 16 bits. For the *int8* method, the number of bits is reduced from 32 to 8, thus a maximum model reduction by a factor of 4. However, this means representing the model parameters as full 8-bit integers, truncating them at the decimal place.

It was necessary to ensure that both models could fit onto the target board, namely the Arduino Nano 33 Sense. To do so, PTQ of varying policy (*int8*, *float16* and *float32* / unquantised) were tested. By introducing a smaller space for the data to occupy, the information incurs a rounding error from the original value. Typically, as a consequence of this, the accuracy of a model is reduced. An acceptable accuracy sacrifice varies depending on the application.

Results

An iterative process was adopted during development, which meant setting some performance targets. The results should be more significant than a guess, as this demonstrates that the model performs beyond randomness. Therefore, the accuracy should be more significant than one standard deviation in a normal distribution. As both models are binary classifiers, this puts our target accuracy at 68%.

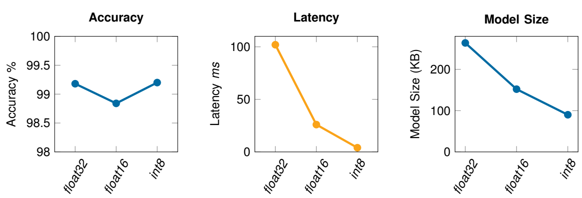
The stress model was initially tested using *float16* quantisation, but this meant using 92% of the available storage. Therefore, to ensure that both models could fit on the microcontroller, they each required quantisation, with the HAR model requiring *float16* quantisation and the stress model *int8* quantisation. In this case, the accuracy of the HAR model was only reduced by 0.34% while the latency was reduced from 102 to 70 ms. Similarly, *int8* quantisation had no impact on the accuracy of the stress model, but decreased latency and reduced memory usage. Therefore, a minor accuracy sacrifice was deemed appropriate to demonstrate two models on a single microcontroller while still performing at a high level of significance.

Upon model deployment, the stress and HAR models achieved an accuracy of 88% and 98% respectively, both using hold-out validation with a 20% test split. A

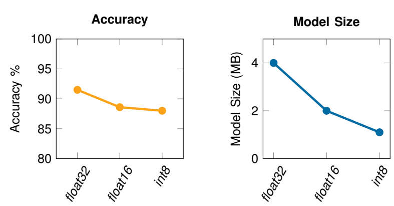
Stress Model			
Class	Precision	Recall	F1
Non-Stress	0.86	0.95	0.90
Stress	0.94	0.83	0.88
Latency		Accuracy	0.88
3642*ms*		Model Size (.h file)	1.1MB

HAR Model			
Class	Precision	Recall	F1
Rest	0.97	0.92	0.94
Exercise	0.99	1.00	0.99
Latency		Accuracy	0.98
26*ms*		Model Size (.h file)	152KB

TABLE 1. Performance metrics for the stress (top) and physical activity (bottom) classification models.



(a) HAR Model metrics



(b) Stress Recognition Model metrics

FIGURE 3. Graph of performance metrics over varying quantisation policies of the HAR model (a) and the stress model (b). Latency was unavailable for *float32* and *float16* quantisation for the stress model due to its large model size.

further breakdown of the performance metrics for each model can be found in Table 1.

The resulting models were able to detect both physical activity and stress with high precision. This enabled the final context-aware approach to be deployed on the Arduino Nano 33 Sense, where physical activity is continuously inferred. Thence, once no physical activity is detected, stress is inferred.

Robust evaluation of the multi-model context-aware system requires validating both the individual model accuracies as well as the overall workflow. Hold-out validation allows reporting performance on unseen data in an unbiased manner. However, evaluating the models separately does not fully validate the system's real-world operation. To address this, we also performed end-to-end validation with live sensor streams. The activity recognition model classifies incoming motion data and only triggers stress classification during periods of inactivity. This end-to-end approach ensures models are evaluated with live data capturing noise and artifacts. The classifiers were run concurrently as designed rather than separately to validate the activity context modelling and switching logic.

The end-to-end validation of the multi-model system with live sensor data streams confirmed the expected performance based on the hold-out testing of individual models. The activity recognition model operated with high accuracy on motion data, properly classifying periods of activity versus inactivity. During these classified inactive states, the stress detection model was selectively triggered as intended and output predictions within the expected accuracy range reported during hold-out testing. The overall workflow of selective stress classification based on activity context was successfully validated. No problems were encountered with concurrency, data flows, or model switching logic. Latency remained low and suitable for real-time operation. This end-to-end validation on live data from multiple sensors confirmed that the integrated system performs as designed based on the individual model metrics.

Discussion

In this paper we have examined the feasibility of running a context-aware HAR model that triggers a stress detection model all performed independently on an Arduino Nano 33 Sense. Our results confirm the potential to run two classification models on a resource-constrained, low-cost microcontroller. Each of the models achieved high accuracy (0.98 and 0.88 for activity recognition and stress detection respectively), demonstrating the potential for real-world inference. This demonstrates similar results for HAR [9], however, our results demonstrate improved performance for rest compared to [9] that achieved 67.3% for the respective idle class, demonstrating the benefits of binary classification. Furthermore, many studies do not report on key metrics such as latency [3] or model size [9] which are key considerations when deploying on microcontrollers. Most research at the edge for HAR relies on smartphones where model sizes can be larger such as 16MB for an activity recognition CNN [12], which is not comparable to the deployment on resource-constrained microcontrollers.

Context plays an important role in stress detection, as the activity in which an individual is engaged can significantly impact physiology. In particular, physical activity can drastically alter both HR and EDA, which are common biomarkers for stress detection. Therefore, our work combining a HAR classification model with stress detection on a microcontroller provides a novel approach for context-aware stress detection that can help improve the reliability of real-world stress inference.

However, limitations were encountered; in particular, the limited memory of the Arduino Nano 33 resulted in the use of simpler models. Initially, a CNN with 4 convolutional layers was used for stress detection that achieved 95.6% accuracy, a 7.6% increase over the 2 layer CNN; however, this used 97% of the microcontroller's available dynamic memory when using *int8* quantisation. Therefore, the number of convolutional layers was reduced to two layers, reducing the model size by 67%. To further reduce the size of the stress model, the model was trained using only one convolutional layer; however, this reduced the accuracy to 61.8%, a 26.7% reduction in accuracy. The model was tested using *float16* quantisation with 2 convolutional layers but this used 92% of the microcontroller's available storage. Therefore, 2 convolutional layers were used to train the model and *int8* quantisation was used to ensure a suitable model size. We also explored the use of LSTM networks, but the models performed worse and larger than comparative CNNs. Additionally, only raw HR and EDA data were used to classify stress. Additional features, such as BPM and HRV, made the model too large to deploy to the microcontroller with the additional HAR model, demonstrating the trade-offs required for real-world deployment.

Figure 3 illustrates the latency and model size of the HAR model reduce using *float16* and *int8* quantisation, while accuracy remains uniform. This demonstrates the benefits of quantisation when deploying onto ultra-low power hardware. However, the HAR model was limited to a minimum of *float16* quantisation due to the inconsistent nature of the WISDM dataset. The expected IMU (LSM9DS1) input data would range between $[0, 1]$, however training data were rarely between these values. Most training samples had a high range and interquartile range, implying a need for complex scaling for reliable integer results. Therefore, if *int8* quantisation was used, this could incur a high preprocessing cost in the final implementation.

These limitations demonstrate the challenging nature of running two classification models on a single microcontroller. While it is possible, careful consideration must be taken to reduce model size while retaining performance. In the future, new microcontrollers with increased memory may simplify deployment with increasingly complex models.

Overall, this tinyML context-aware classification approach has the potential to improve numerous domains beyond stress detection. The capability to run one classification model, which can provide context to a second more complex model running on-device, is a novel approach that can help increase model performance and reduce power consumption.

Conclusion

In this work, we have demonstrated how a classification model can be used to provide context to a second model on the edge using an ultra-low power and low-cost microcontroller. Here, we have examined how activity recognition can be used to help reduce motion artifacts in real-world stress detection, but this framework has many potential real-world applications.

In the future, our goal is to further deploy this approach and test this context-aware approach in other domains. The use of new microcontrollers with more memory and alternative quantisation methods will also be explored to enable on-device inference from more complex models.

REFERENCES

[1] Alberto De Santos Sierra et al. "Two stress detection schemes based on physiological signals for real-time applications". In: *Proceedings - 2010 6th International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP 2010*. 2010. ISBN: 9780769542225.

[2] Katarina Dedovic et al. "The Montreal Imaging Stress Task : using functional imaging to investigate the ..." In: *Psychiatry & Neuroscience* 30.5 (2005), pp. 319–325. ISSN: 1180-4882.

[3] Alessandro Ghibellini, Luciano Bononi, and Marco Di Felice. "Intelligence at the IoT Edge: Activity Recognition with Low-Power Microcontrollers and Convolutional Neural Networks". In: *Proceedings - IEEE Consumer Communications and Networking Conference, CCNC* (2022), pp. 707–710. ISSN: 23319860.

[4] Shubham Gupta et al. "A TinyML Approach to Human Activity Recognition". In: *Journal of Physics: Conference Series* 2273.1 (2022). ISSN: 17426596.

[5] J.A. Healey and R.W. Picard. "Detecting Stress During Real-World Driving Tasks Using Physiological Sensors". In: *IEEE Transactions on Intelligent Transportation Systems* 6.2 (June 2005), pp. 156–166. ISSN: 1524-9050.

[6] Jennifer R Kwapisz, Gary M Weiss, and Samuel A Moore. “Activity Recognition using Cell Phone Accelerometers”. In: *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data (at KDD-10), Washington DC*. 2010. ISBN: 9781450302241.
[7] Song Mi Lee, Sang Min Yoon, and Heeryon Cho. “Human activity recognition from accelerometer data using Convolutional Neural Network”. In: *2017 IEEE International Conference on Big Data and Smart Computing, BigComp 2017* (Mar. 2017), pp. 131–134.
[8] Neil Schneiderman, Gail Ironson, and Scott D. Siegel. “Stress and Health: Psychological, Behavioral, and Biological Determinants”. In: *Annual Review of Clinical Psychology* 1 (Nov. 2004), pp. 607–628.
[9] Kristof Tjonck et al. “Real-Time Activity Tracking using TinyML to Support Elderly Care”. In: *2021 30th International Scientific Conference Electronics, ET 2021 - Proceedings* (Sept. 2021).
[10] Solveig Vieluf et al. “Exercise-induced changes of multimodal interactions within the autonomic nervous network”. In: *Frontiers in Physiology* 10.MAR (Mar. 2019), p. 240. ISSN: 1664042X.
[11] Kieran Woodward et al. “Towards Personalised Mental Wellbeing Recognition On-Device using Transfer Learning "in the Wild"”. In: *IEEE International Smart Cities Conference 2021*. 2021.
[12] Tahmina Zebin et al. “Design and Implementation of a Convolutional Neural Network on an Edge Computing Smartphone for Human Activity Recognition”. In: *IEEE Access* 7 (2019), pp. 133509–133520. ISSN: 21693536.

Michael Gibbs is a PhD student in the Smart Sensing Lab, Nottingham Trent University, UK. His research interests include tinyML, edge computing and model optimisation. M. Gibbs and K. Woodward contributed equally to this work. Contact him at michael.gibbs@ntu.ac.uk

Kieran Woodward is a Research Associate in the Smart Sensing Lab, Nottingham Trent University, UK. His current research interests include affective computing, tinyML, and pervasive mobile gaming. Kieran received a PhD in Pervasive Computing and Data Science from Nottingham Trent University. Contact him at kieran.woodward@ntu.ac.uk

Eiman Kanjo is a Professor of Pervasive Sensing and the head of the Smart Sensing lab at Nottingham Trent University. She is also the Provost’s Visiting Professor at Imperial College London. She conducts research in Mobile Sensing, wearable, tinyML and Edge Computing. She is a recent winner of the Top 50 Women in Engineering by the Women in Engineering Society. She is the NTU-Turing Network Academic lead funded by Alan Turing Institute. She received her PhD in computer Science from Abertay University of Dundee, UK. Contact her at eiman.kanjo@ntu.ac.uk