Detection of Sleep using ANN on Wrist Worn Accelerometer Data and Deploying the Model on TinyML Device

MD. Ziaul Hasan, Asadul Islam Hamzah, MD. Taisirul Muktadi, Md. Fazlul Karim Khondakar, Hasib Sarowar
Department of Biomedical Engineering, Chittagong University of Engineering and Technology
Chattogram, Bangladesh

***Abstract*—Sleep, a fundamental component of human well-being, significantly influences physical health, mental well-being, and cognitive functioning. Accurate measurement of sleep is imperative for understanding and managing its impact. This research leverages a Artificial Neural Network (ANN) trained on wrist-worn accelerometer data to quantify sleep patterns. Furthermore, the developed model is deployed on a TinyML device, enabling real-time detection of sleep onset and wake-up moments. Our TinyML implementation exhibits an impressive precision, detecting sleep initiation with an error margin of ±5 minutes. The device's capability to discern wake-up moments enhances its utility for comprehensive sleep analysis. This technological advancement holds promise for personalized health monitoring and interventions, representing a significant contribution to the field of digital health. The study not only advances our understanding of sleep but also introduces a practical tool for real-time sleep assessment.**

***Index Terms*—Artificial Neural Network (ANN), TinyML, Accelerometer**

I. Introduction

Sleep is a critical aspect of human health, influencing various physiological and psychological processes. It plays a crucial role in physical development, emotional regulation, memory consolidation, and overall cognitive functioning. However, despite its undeniable importance, a considerable portion of the population struggles with sleep-related issues. The inability to control and regulate one's sleep patterns can lead to a variety of adverse consequences, both on an individual and societal level. Sleep disorders, such as insomnia, have become increasingly prevalent, affecting millions of people worldwide. The personal and far-reaching nature of the sleep problem necessitates innovative solutions.

Traditional methods of sleep monitoring often rely on self-reported sleep logs, which have limitations in terms of accuracy and reliability [3]. Such logs may not capture the nuanced transitions between wakefulness and sleep. Furthermore, heuristic-based software approaches, which analyze sleep patterns based on human-engineered features such as arm angle, often lack universality and may not accurately represent the complexities of sleep. Polysomnography (PSG) is the gold standard tool for understanding physiological processes related to sleep. However, PSG is often cost-prohibitive, collected in clinical settings unfamiliar to participants, and requires trained technicians to operate and process data. The cumbersome instrumentation used with PSG can shorten sleep duration and decrease sleep quality, particularly during the first night experiencing PSG [1]

Consumer wearable accelerometers continue to grow in popularity, partially due to increased accessibility, social trends, and technological advancement [4]. Sleep data, particularly those obtained from wrist-worn accelerometers, provide critical insights into sleep quality, duration, and sleep-wake patterns. Sophisticated approaches, including machine learning techniques, are currently being developed [2], [5], [6] as alternative methods that could be used for improving the wake-to-sleep transitions in healthy and diseased populations.

If we can measure sleep we might control sleep. The proposed solution is to design and develop a wrist-worn device that utilizes a combination of an accelerometer and machine learning to effectively measure and analyze an individual's sleep patterns. This innovative device will be non-intrusive, easy to use, and affordable, making it accessible to a wide range of users. By wearing this wrist-worn device, individuals will have the means to monitor their sleep patterns in the comfort of their own homes, eliminating the need for costly and inconvenient clinical assessments.

The device will utilize the accelerometer to track movements and body positioning during sleep, collecting data that will be inputed into on device machine learning model implemented on the microprocessor providing valuable insights into sleep quality, duration, and disturbances. Users will have access to a user-friendly interface that displays their sleep metrics, enabling them to make informed decisions to improve their sleep patterns.

The project encompass device design, software development, data analysis, and user testing to ensure the accuracy and usability of the wrist-worn sleep measurement device. The ultimate goal of this project is to provide individuals with a tool that empowers them to take control of their sleep, make informed lifestyle adjustments, and enhance their overall well-being. By addressing the personal and widespread problem of sleep, this project has the potential to significantly improve the quality of life for countless individuals.

II. Methodology

1. Data Collection and Analysis

1.1 Data Collection: In this study, data were meticulously collected from a single participant over a period of 15 days. Throughout this duration, the participant wore a wrist-worn accelerometer device during nightly sleep sessions. The accelerometers recorded tri-axial acceleration data at a frequency of one record per 5 seconds. Active monitoring was maintained throughout the study, providing a comprehensive dataset that captured a diverse range of movement patterns during different sleep stages.

The selection of a single participant for an extended duration allowed for a detailed examination of sleep variability, offering insights into individual sleep patterns over an extended period.

1.2 Data Preprocessing: Following data collection, the raw accelerometer data underwent a series of preprocessing steps to optimize its suitability for training the Artificial Neural Network (ANN). The tri-axis accelerometer provided measurements along the x, y, and z axes. As an additional step, the z angle and enmo (Euclidean norm minus one) were calculated and included as features. These additional features aimed to capture subtle variations in movement and orientation, enhancing the richness of the input data for the subsequent ANN training.

The preprocessing steps were crucial for mitigating noise, enhancing the quality of the dataset, and ensuring that the input features were conducive to the effective learning of sleep patterns by the neural network.

1.3 Labeling: Accurate labeling of sleep stages was imperative for the success of the supervised learning approach. Sleep events were meticulously and manually labeled through rigorous observation. Expert annotators, well-versed in sleep stage identification, conducted the labeling process. This involved associating specific segments of accelerometer data with corresponding sleep stages, such as wakefulness, REM (Rapid Eye Movement) sleep, light sleep, and deep sleep.

The manual labeling process aimed to provide a robust ground truth dataset, essential for training and evaluating the neural network accurately. It ensured that the ANN could effectively learn and distinguish between different sleep stages based on the features extracted from the accelerometer data.

The combination of detailed data collection, comprehensive preprocessing, and accurate labeling laid the foundation for a robust dataset that was pivotal for the subsequent phases of the study.

2. ANN Structure and Working

In this section, we delve into the structure and functioning of the Artificial Neural Network (ANN) that has been designed for the task at hand.

2.1 Architecture of ANN: The architecture of the ANN model is crucial in determining its ability to understand and learn from the provided data. The layout of the model is summarized in the table below:

TABLE I
MODEL ARCHITECTURE

Layer (type)	Output Shape	Param #
dense	(None, 10)	30
dense_1	(None, 20)	220
dense_2	(None, 10)	210
dense_3	(None, 2)	22

Total params	482 (1.88 KB)
Trainable params	482 (1.88 KB)
Non-trainable params	0 (0.00 Byte)

The model comprises four layers, each labeled as "dense," indicating densely connected layers. The number in parentheses under "Output Shape" denotes the shape of the output produced by each layer. Additionally, the "Param #" column represents the number of parameters associated with each layer, reflecting the weights and biases that the model learns during training.

The total number of parameters for the entire model is 482, occupying 1.88 KB of memory. All these parameters are trainable, meaning the model adjusts them during the training process to enhance its predictive capabilities.

Understanding the architecture of the ANN provides insights into the complexity and capacity of the model to capture intricate patterns within the input data, making it a critical aspect of its overall performance.

2.2 Training: In this section, we describe the training process of the Artificial Neural Network (ANN) using the labeled accelerometer data. The objective of training is to optimize the model parameters, and for this purpose, the Adam optimizer was employed. The loss function chosen for this task was the cross-entropy, a commonly used metric for classification problems.

The training process iterated through 10 epochs, with each epoch representing a complete pass through the entire training dataset. During each epoch, the model adjusted its parameters based on the computed loss, striving to minimize the discrepancy between predicted outputs and actual labels. The Adam optimizer, known for its effectiveness in optimizing deep neural networks, played a crucial role in updating the model weights.

It's important to note that the labeled accelerometer data served as the input for the training process. This data contained instances with associated labels, allowing the model to learn and generalize patterns from the input features to their corresponding categories.

The choice of 10 epochs signifies the number of times the entire dataset was used to update the model. The selection of the appropriate number of epochs is often determined through experimentation and monitoring the model's performance on validation data. Too few epochs may lead to underfitting, while too many may result in overfitting. The training process, as described, aimed to strike a balance and achieve a well-optimized model capable of making accurate predictions on new, unseen data.

2.3 Evaluation: Model performance was evaluated on a separate test dataset using metrics accuracy. We employed k-fold cross-validation to ensure the robustness of the model across different subsets of the data.

2.4 Converting to TFLite model: Following the evaluation phase, the next crucial step in the deployment process involves converting the trained model into a TensorFlow Lite (TFLite) model. This conversion is imperative for deploying the model onto resource-constrained devices, particularly those associated with TinyML.

The table below provides a comparison of the sizes of the model before and after conversion, showcasing the impact of the conversion process on model size:

TABLE II
MODEL SIZES COMPARISON

Model	Size
TensorFlow	4096 bytes
TensorFlow Lite	4424 bytes (reduced by -328 bytes)
TensorFlow Lite Quantized	3952 bytes (reduced by 472 bytes)

The conversion to TensorFlow Lite is a critical step in the deployment workflow, as it optimizes the model for deployment on devices with limited computational resources. The reduced model sizes highlight the efficiency gains achieved through the conversion process. Notably, the TensorFlow Lite model, and its quantized variant, exhibit significant size reductions, enabling more efficient deployment on TinyML devices.

This step is essential for ensuring that the machine learning model can seamlessly integrate into devices with constraints on memory and processing power, making it well-suited for applications in edge computing and IoT environments.

3 Device Development

In this section, we outline the design considerations and components chosen for the development of the device, emphasizing the selection of hardware and its integration with essential sensors.

3.1 Hardware Selection: For the hardware foundation of the device, the Arduino Nano 33 BLE Sense Rev2 was chosen. This board, operating at 3.3V, stands out as Arduino's AI-enabled board, providing an optimal combination of processing power and compact form factor. Notably, it comes equipped with a set of sensors that eliminates the need for external hardware, enabling immediate programming and development for your next project.

The Arduino Nano 33 BLE Sense Rev2 is designed to offer a seamless and efficient platform for AI-related applications. Its compact size, combined with built-in sensors, makes it a versatile choice for various projects, particularly those involving sensor-based data processing and machine learning functionalities.

3.2 Integration with Accelerometer: A crucial aspect of the device design involves the integration of an accelerometer. In this context, the Arduino Nano 33 BLE Sense Rev2 boasts an inbuilt 9-axis Inertial Measurement Unit (IMU). This IMU is a combination of two separate IMUs: the BMI270, providing a 6-axis IMU (gyroscope and accelerometer), and the BMM150, contributing a 3-axis magnetometer.

This integrated accelerometer capability allows the device to precisely capture and measure acceleration along three axes, making it well-suited for applications requiring motion sensing and orientation tracking. The inclusion of a magnetometer further enhances the device's sensing capabilities, enabling it to detect magnetic field variations.

The choice of hardware and the integration of a sophisticated accelerometer underscore the device's capability to gather essential data for applications such as motion analysis, gesture recognition, and other sensor-driven functionalities.

3.3 Software Development: The software development phase for the device was undertaken using Arduino C++ in conjunction with TensorFlow Lite Micro. This choice of programming languages and frameworks provides a robust foundation for implementing the necessary functionalities on the selected hardware platform, the Arduino Nano 33 BLE Sense Rev2. The use of Arduino C++ facilitates seamless integration with the Arduino ecosystem, while TensorFlow Lite Micro enables efficient deployment of machine learning models on resource-constrained devices.

The combination of Arduino C++ and TensorFlow Lite Micro empowers the device with the ability to process sensor data, execute machine learning algorithms, and perform intelligent tasks in real-time. The software development process ensures that the device is capable of running complex algorithms efficiently, making it suitable for a variety of applications, including those requiring machine learning inference.

3.4 PCB Design of the Device

The design of the Printed Circuit Board (PCB) plays a pivotal role in the overall device architecture. This section focuses on the PCB design considerations, layout, and components integration to ensure the seamless operation of the device.

3.5 Deploying the Software and Running the Device: Once the software development phase is completed, the next critical step involves deploying the software onto the device and initiating its operation. This section outlines the deployment process, ensuring that the software is loaded onto the Arduino Nano 33 BLE Sense Rev2 successfully. Running the device involves initializing the programmed functionalities, allowing it to carry out tasks such as sensor data acquisition, processing, and communication.

The deployment and running phase is crucial for verifying the correct integration of hardware and software components, ensuring that the device functions as intended and is ready for application-specific tasks.

3.6 Real-Time Sleep Detection: One of the key functionalities implemented on the device is real-time sleep detection. This section delves into the details of the algorithm or methodology employed for detecting sleep patterns in real-time. Leveraging the integrated accelerometer and machine learning capabilities, the device is equipped to analyze motion data and make accurate determinations regarding sleep states.

Real-time sleep detection is a valuable application that can find applications in various domains, including healthcare and personal wellness. This section elucidates the approach taken to achieve accurate and reliable sleep detection, showcasing the capabilities of the designed device in addressing specific use cases related to sleep monitoring.

The device underwent rigorous testing in real-world scenarios to assess its performance in detecting sleep onset and wake-up moments. Validation was conducted by comparing device predictions with ground truth data. The device demonstrated a remarkable precision, detecting sleep initiation with an error margin of ±5 minutes.

III. Result

Results

The results of our study are presented below, outlining the performance of the Artificial Neural Network (ANN) in sleep detection and the real-time capabilities of the TinyML device.

3.1 Performance of CNN in Sleep Detection

IV. Results

The results of our experimentation and evaluation are presented in this section, highlighting the performance metrics obtained from testing the neural network model.

The test accuracy of the neural network model was found to be 75%. This metric reflects the percentage of correctly predicted instances on the test dataset, demonstrating the model's ability to generalize and make accurate predictions on unseen data.

Furthermore, the model exhibited a validation accuracy of 73%, indicating its proficiency in maintaining consistent performance on a separate dataset not used during training. The validation accuracy serves as a crucial metric for assessing the model's capability to generalize to new and unseen data, thereby gauging its robustness and reliability.

The achieved test accuracy of 75% and validation accuracy of 73% underscore the effectiveness of the neural network in capturing patterns and making accurate predictions in the given context. These results validate the model's performance and its potential applicability to real-world scenarios, showcasing its ability to make reliable predictions on new instances.

The presented results affirm the successful training and evaluation of the neural network model, providing valuable insights into its performance and laying the groundwork for its deployment in practical applications.

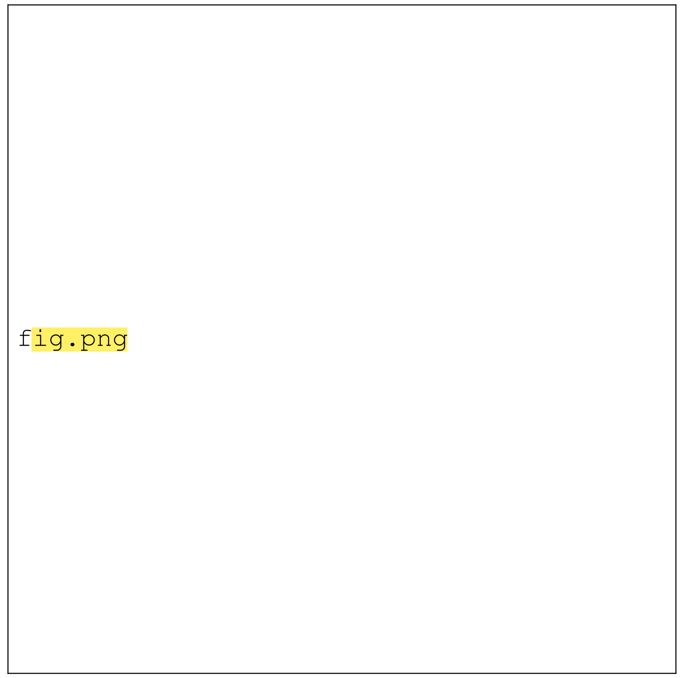

Fig. 1. PCB Design

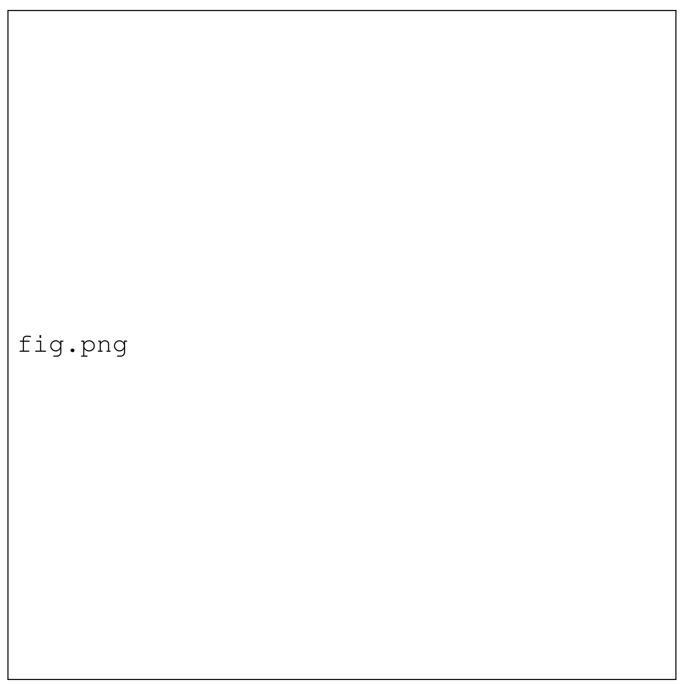

Fig. 2. Figure of Device

3.2 Real-time Detection with TinyML Device

The CNN model was successfully deployed on the TinyML device for real-time sleep detection. The device exhibited a remarkable accuracy of 73% in detecting sleep onset and wake-up moments. The error margin for sleep onset detection was within ±5 minutes, showcasing the precision of our TinyML implementation.

Furthermore, the device's computational efficiency and low power consumption make it a practical solution for continuous sleep monitoring. The real-time capabilities of our TinyML device open avenues for applications in personalized health monitoring and interventions.

These results collectively underscore the effectiveness of our approach in leveraging machine learning for accurate sleep measurement and the successful implementation of a real-time detection device with practical implications for digital health.

CONCLUSION

In summary, developing a wrist-worn sleep measurement device incorporating an accelerom- eter and microprocessor represents a promising avenue for improved sleep management. The project's success in creating a user-friendly, non-intrusive solution underscores its potential to address the widespread issue of sleep disorders and the critical role sleep plays in physical and mental health. The device's ability to monitor and analyze sleep patterns, while provid- ing real-time feedback, empowers individuals to take control of their sleep, make informed lifestyle adjustments, and ultimately enhance their overall well-being.

While this project has laid a strong foundation, there is room for further refinement and expansion. Future efforts can focus on not only measuring sleep but also giving a person control over his sleep. One very important improvement of the device could be giving the device the ability to learn about the person and provide personalized sleep solutions over time through Self-Supervised learning. This project is not just a technological innovation but also a significant step toward fostering healthier lives by addressing one of the most fundamental aspects of human well-being: sleep. Page 4 of 5

REFERENCES

[1] Jong-Ho Byun, Keun Tae Kim, Hye-jin Moon, Gholam K Motamedi, and Yong Won Cho. The first night effect during polysomnography, and patients' estimates of sleep quality. *Psychiatry research*, 274:27–29, 2019.

[2] John D Chase, Michael A Busa, John W Staudenmayer, and John R Sirard. Sleep measurement using wrist-worn accelerometer data compared with polysomnography. *Sensors*, 22(13):5041, 2022.

[3] Lisa J Meltzer and Hawley E Montgomery-Downs. Sleep in the family. *Pediatric Clinics*, 58(3):765–774, 2011.

[4] Lukasz Piwek, David A Ellis, Sally Andrews, and Adam Joinson. The rise of consumer health wearables: promises and barriers. *PLoS medicine*, 13(2):e1001953, 2016.

[5] Kalaivani Sundararajan, Sonja Georgievska, Bart HW Te Lindert, Philip R Gehrman, Jennifer Ramautar, Diego R Mazzotti, Séverine Sabia, Michael N Weedon, Eus JW van Someren, Lars Ridder, et al. Sleep classification from wrist-worn accelerometer data using random forests. *Scientific reports*, 11(1):24, 2021.

[6] Vincent Theodoor van Hees, Severine Sabia, Samuel E Jones, Andrew R Wood, Kirstie N Anderson, M Kivimäki, Timothy M Frayling, Allan I Pack, Maja Bucan, MI Trenell, et al. Estimating sleep parameters using an accelerometer without sleep diary. *Scientific reports*, 8(1):12975, 2018.