Smart Driver Drowsiness Detection Using LSTM Technology Based on Heart Rate Monitoring

Jon Endri[1], Inas Afifah[2], Suroso Suroso[3], Rumiasi Rumiasi[4], Carlos RS[5], Ade Silvia Handayani[6*]

[1-6] Politeknik Negeri Sriwijaya, Palembang 30139, Indonesia

*ade_silvia@polsri.ac.id

Abstract. This study presents the development of a smart driver drowsiness detection system using Long Short-Term Memory (LSTM) technology based on heart rate monitoring. Traffic accidents are frequently caused by factors such as drowsiness, sleep deprivation, alcohol consumption, or drug use. Among these, microsleep—a brief and involuntary lapse into sleep lasting a few seconds—poses significant risks, often resulting in fatal consequences while driving. The objective of this research is to implement the LSTM algorithm as an early warning system for microsleep in drivers and to develop a drowsiness detection device equipped with a pulse heart rate sensor. The LSTM algorithm, known for its ability to model long-term dependencies, has proven highly effective in time series prediction. It is utilized here for real-time analysis of drivers' heart rate data.

Keywords: driver drowsiness detection, long short-term memory, heart rate monitoring, microsleep

1 Introduction

Traffic accidents are a pressing global issue, contributing significantly to non-communicable health problems and socioeconomic losses. The World Health Organization (WHO) estimates that road traffic injuries are among the leading causes of death worldwide, particularly in low- and middle-income countries [1]. These accidents often lead to severe consequences, including physical disabilities, financial burdens, and fatalities, negatively impacting public health and welfare. In Indonesia, data from the National Traffic Police Corps reported 94,617 traffic accidents in 2022, with a substantial portion attributed to driver negligence and impaired concentration. Key contributing factors include mobile phone use, substance use, fatigue, and drowsiness, with microsleep being particularly perilous

Microsleep, characterized by brief, involuntary episodes of sleep lasting a few seconds, is often triggered by monotonous tasks such as driving or screen exposure. During microsleep episodes, individuals may lose sensory awareness, exhibit excessive blinking, and be unable to maintain alertness, making them oblivious to the dangers posed [2][3]. Research indicates that up to 32% of drivers have experienced accidents related

N. L. Husni et al. (eds.), *Proceedings of the 8th FIRST 2024 International Conference on Global Innovations (FIRST-ESCSI 2024)*, Advances in Engineering Research 261,
https://doi.org/10.2991/978-94-6463-678-9_31

to drowsiness, underscoring the critical need for advanced detection systems [4]. Physiological markers, particularly heart rate fluctuations, have been identified as reliable indicators for detecting drowsiness and the onset of microsleep, as heart rate tends to decrease during these states [5].

Emerging technologies in artificial intelligence, particularly Long Short-Term Memory (LSTM) networks, have revolutionized the ability to analyze sequential data and predict time-dependent patterns. LSTM, a form of recurrent neural network, is particularly well-suited for detecting drowsiness by analyzing real-time physiological data, such as heart rate, due to its capacity to model long-term dependencies [6][7]. However, existing drowsiness detection systems often face limitations. These include bulky designs, lack of real-time prediction, dependency on multiple parameters such as behavioral and physiological data, and the absence of an integrated early warning mechanism [8].

This research addresses these gaps by proposing a novel driver drowsiness detection system based on LSTM technology and heart rate monitoring. By leveraging a compact and wearable pulse heart rate sensor, the system continuously monitors the driver's physiological state. Using an LSTM-based model, it predicts potential drowsiness episodes and provides early warnings through vibration, on-screen notifications, and real-time alerts. The proposed system aims to enhance road safety by offering a practical, efficient, and innovative solution to mitigate the risks associated with driver drowsiness and microsleep.

2. Methodology

This study involves several steps, such as conducting a literature review, setting up the hardware, developing an algorithm model using Long Short-Term Memory (LSTM), implementing the LSTM model on the hardware, testing, and collecting data. To construct the hardware, several components are required, including a pulse heart rate sensor, NodeMCU, OLED display, and a buzzer. Each component is crucial for drowsiness detection: the heart rate sensor monitors physiological indications, the NodeMCU processes data and issues alerts, the OLED presents real-time information, and the buzzer delivers audible warnings. The effective amalgamation of these elements will provide precise identification and mitigation of drowsiness in real-time situations. The effective amalgamation of these elements will provide precise identification and mitigation of drowsiness in real-time situations, hence enhancing safety and diminishing the likelihood of road accidents.

Long Short-Term Memory (LSTM) algorithm model can be constructed with the dataset comprising heartbeats recorded by the pulse sensor that was collected first. A pre-processing procedure will be conducted on the multiple available datasets. At this step, data cleaning will be conducted to address duplicate and corrupted data, and data augmentation will also be implemented to enhance the dataset through rotation trials. The dataset will be partitioned into two segments: 67% for training and 33% for testing. The training data comprises the dataset utilized for training the LSTM model, whereas the testing dataset functions as the dataset employed for system evaluation post-training, ensuring that it remains unaltered and free from any processing. The LSTM architecture will be developed at this step, comprising multiple layers to generate BPM prediction outputs. The constructed LSTM model is subsequently trained with the training data. This phase entails iterative learning, wherein the model identifies patterns within the data and adjusts weights to minimize prediction errors. Subsequent to training, the model undergoes evaluation using testing data to verify its performance on unfamiliar data. Model assessment is performed to assess the model's performance using the metrics MSE, MAE, MAPE, and RMSE. Following the assessment, it will be ascertained if the model is optimal; if affirmative, the superior model will be chosen, and if negative, it will revert to the training phase. The optimal model is designated as the final model, which is subsequently utilized for real-time heart rate prediction and monitoring.

The software development process commences with the initialization of the Flask application and the blueprint designated for managing requests. The LSTM model and scaler are subsequently loaded to facilitate predictions. The application includes a feature to transmit messages over Telegram if necessary. A route designated as /predict is established to accept POST requests that provide heart rate data in JSON format. Upon receiving the request, the heart rate data is extracted from the JSON, subsequently scaled, and reshaped to conform to the input specifications of the LSTM model. This model is subsequently employed to forecast sleepiness state utilizing the analyzed heart rate data. Subsequent to acquiring the prediction findings, the data is rescaled for enhanced interpretation. The device subsequently assesses the driver's state to ascertain if it is drowsy. Should the predictive analysis suggest that the driver is fatigued, the program will transmit a warning message using Telegram. Otherwise, the program will solely provide the prediction outcome. All this information is subsequently encapsulated in a JSON response, which is transmitted back to the driver before the process concludes.

The instrument operates by utilizing heart rate data collected through a heart rate monitor. The Long Short-Term Memory method will predict the subsequent heart rate data. If the subsequent heart rate is low or below 65 bpm, it will automatically activate the buzzer, dispatch a telegram message, and exhibit the BPM number on the OLED panel, signifying that the driver is likely experiencing drowsiness. During the drowsiness prediction phase, a data processing procedure is executed utilizing the driver's heart rate data as input. The data will be analyzed utilizing the Long Short-Term Memory (LSTM) model to forecast driver tiredness.

The comprehensive system testing procedure is vital in comprehending the tool's workflow. The comprehensive system testing comprises two components, specifically: Evaluating the gadget by juxtaposing the BPM sensor with the BPM oximeter to assess the accuracy of the BPM sensor in detecting heartbeats. The formula for calculating the percentage accuracy of the pulse heart rate sensor measurement involves dividing the heart rate value obtained from the BPM sensor by the heart rate value from the BPM oximeter, followed by multiplying the result by 100 to derive the percentage accuracy, as expressed in the following equation.

$$level\ of\ accuracy\ (\%) = \frac{sensorpulse}{manual} \times 100 \quad (1)$$

Evaluating the accuracy of the drowsiness detection gadget in predicting drowsiness using heart rate data analysis. Evaluating the optimal performance of the Long Short-Term Memory method by ascertaining the subsequent values,

a. Root Mean Square Error (RMSE),
A low or near-zero Root Mean Square Error (RMSE) number signifies superior prediction outcomes.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2} \quad (2)$$

b. Mean Absolute Percentage Error (MAPE),
A decreased Mean Absolute Percentage Error (MAPE) indicates enhanced prediction accuracy, and MAPE can be classified into four distinct levels.

$$MAPE = 100 \times \left(\frac{1}{m}\right)\sum_{i=1}^{m}\left|\frac{Y_i - X_i}{Y_i}\right| \quad (3)$$

- Score <10% = Highly Accurate
- Score between 10% and 19% = Good
- Score between 20% and 49% = Normal
- > 50% = Inaccurate

c. Mean Absolute Error (MAE),
The Mean Absolute Error (MAE) is determined by averaging the absolute error values, as the calculation ensures that the difference between the actual and anticipated values remains positive.

$$MAE = \left(\frac{1}{m}\right)\sum_{i=1}^{m}|X_i - Y_i| \quad (4)$$

d. Mean Squared Error (MSE).
A low or near-zero Mean Squared Error (MSE) score signifies that the predicted outcomes correspond closely with the actual data, making them suitable for future predictive analyses.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2 \quad (5)$$

3. Result and Discussion

3.1. Hardware Setup Result

The comprehensive outcomes of the hardware design are illustrated in Figure 1. The hardware configuration incorporates the NodeMCU ESP8266 housed within a box alongside additional component. An OLED display is positioned at the top, accompanied by an on/off button and a battery charging connection adjacent to it. A pulse heart rate sensor is put at the bottom, and a buzzer is located at the rear.

Fig. 1. Hardware Setup Result

3.2. Long Short-Term Memory (LSTM) Algorithm Model Results

Prior to initiating the data training procedure, the crucial initial step is to assemble a dataset comprising two primary variables: time (in minutes) and heart rate (in BPM, or beats per minute). This data was acquired from several responders utilizing a Pulse Heart Rate Sensor. This sensor monitors heart rate in real-time and is linked to NodeMCU, a microcontroller proficient in wireless data transmission. The NodeMCU transmits data to the Raspberry Pi, a single-board computer utilized for data recording and storage. The Raspberry Pi operates as a receiving server that aggregates data for subsequent analysis.

The device's heart rate detection system generates 522 entries in TXT format, documenting measurement results. The data is subsequently transformed into CSV format for additional processing. The Heart Rate column is extracted and converted into a NumPy array, while the dataset is normalized utilizing the MinMaxScaler from sklearn. This normalization ensures that all parameters, including time and BPM, contribute proportionally to analyses or prediction models, particularly in machine learning algorithms sensitive to feature scale. Table 1, illustrates the outcomes of this normalization.

Table 1. Normalized Data Result

No	Normalized Heart Bit (bpm)
1	0.39583325
2	0.5
3	0.45833325
4	0.5416666
5	0.4375
6	0.666666
7	0.47916663
8	0.52083325
9	0.375
10	0.5625
11	0.6041666
12	0.4375
13	0.45833325
14	0.41666663
15	0.58333325
16	0.1875
17	0.5
18	0.45833325

The Long Short-Term Memory (LSTM) model architecture was constructed with TensorFlow and Keras to forecast heart rate based on historical data. The model is a sequential architecture with a single LSTM layer with four units and an output Dense layer. The model is optimized with the 'mean-squared-error' loss function and the 'ADAM' optimization algorithm to minimize prediction errors during training. The model undergoes training for 100 epochs, utilizing a batch size of 1 and a validation split of 0.2. The model is subsequently included into a device engineered to monitor and analyze heartbeats in real-time for the detection of drowsiness in drivers. The evaluation measures the model's capacity to generalize and apply knowledge acquired from training data to predict outcomes on novel data. The model's efficacy is assessed by Mean Square Error (MSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Root Mean Squared Error (RMSE). The Mean Squared Error (MSE) for the training dataset is 77.80, whereas for the testing dataset it is 69.47. The MAE is 6.42, the MAPE is 8.87%, and the RMSE is 8.82, demonstrating constant efficacy in minimizing substantial mistakes.

3.3. Software Design Result

Figure 3, illustrates a Flask application handling POST requests to the `/api/predict` endpoint, with a status code of 200 signifying successful processing. The server's message "Message sent successfully!" verifies API functioning, enabling it to process heart rate data requests in alignment with the application's goals.

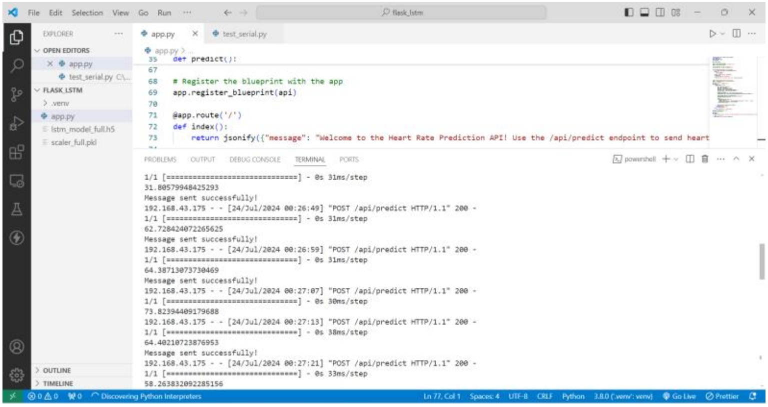

Fig. 3. Log Result of API Flask testing

3.4. Test Result

The system's efficacy in accurately detecting heartbeats is assessed by evaluating the accuracy of the BPM sensor and the precision of the Long Short-Term Memory (LSTM) model in its predictions. A comparison was conducted using a BPM oximeter to verify the sensor's precision. The findings indicated a minor discrepancy between the BPM recorded by the sensor and the oximeter, suggesting that the sensor typically yields precise readings.

Fig. 4. OLED display in a drowsy condition

The most significant discrepancy was observed with a BPM sensor reading of 62.72 and an oximeter reading of 67, achieving an accuracy level of 93%. The Long Short-Term Memory (LSTM) model was evaluated to forecast heart rate using the acquired data. The system evaluated sleepiness conditions and modified the buzzer status accordingly. The OLED display presented heart rate data in BPM when the device operated under standard settings. Upon the LSTM model's prediction of a drowsy state, the screen exhibited the message "BPM NOW," enabling the driver to undertake requisite measures.

3.5. Data Analysis

The research evaluates the BPM sensor and BPM oximeter for heart rate assessment, revealing consistent and precise outcomes. Long Short-Term Memory is employed to identify drowsiness through the analysis of a driver's pulse rate. A sequential model exhibiting consistent MAE, MAPE, and RMSE values is developed, demonstrating stable performance. The model has been trained and saved, and the Flask API is utilized to accept data from NodeMcu. The technology verifies the anticipated BPM value and transmits alerts using Telegram. The buzzer activates for 5 seconds, generating a vibration, while the OLED screen exhibits the current BPM and the driver's status.

4. Summary

The results of the tests and analysis conducted lead to several key conclusions. The training loss was 0.0348, while the testing loss was 0.0314%, indicating that the prediction model performs well with minimal errors in heart rate prediction. Additionally, the BPM sensor was tested successfully and functioned accurately, providing reliable heart rate measurements. The integration of the BPM sensor with the Telegram system and buzzer also proved effective, enabling timely notifications regarding the driver's condition. This system ensures efficient monitoring and alerts, contributing to improved safety and driver health management.

References

[1] T. Toroyan, M. M. Peden, and K. Iaych, "WHO launches second global status report on road safety," *Inj. Prev.*, vol. 19, no. 2, p. 150, 2013, doi: 10.1136/injuryprev-2013-040775.

[2] Q. Abbas, M. E. A. Ibrahim, S. Khan, and A. R. Baig, "Hypo-Driver : A Multiview Driver Fatigue and Distraction Level Detection System," 2022, doi: 10.32604/cmc.2022.022553.

[3] Z. Wang and M. Chen, "Study on Driver Cross-Subject Emotion Recognition Based on Raw Multi-Channels EEG Data," 2023.

[4] A. Madikhonov, "" EXPLORATION OF REAL-TIME EYE DROWSINESS DETECTION THROUGH DIVERSE COMPUTER VISION MODELS : AN INVESTIGATIVE STUDY "," vol. 2, no. 4, pp. 111–118, 2024.

[5] N. Investigation, "Sleep Stage Classification Using EEG Signal Analysis :," pp. 50–70, 2016, doi: 10.3390/e18090272.

[6] F. Faraji, F. Lotfi, J. Khorramdel, A. Najafi, and A. Ghaffari, "Drowsiness Detection Based On Driver Temporal Behavior Using a New Developed Dataset," pp. 1–7.

[7] A. Kumar and S. Chakravarthy, “Energy-Efficient Deep Neural Networks for EEG Signal Noise Reduction in Next-Generation Green Wireless Networks and Industrial IoT Application,” 2023.

[8] J. Jeong, B. Yu, and D. Lee, “Classification of Drowsiness Levels Based on a Deep Spatio-Temporal Convolutional Bidirectional LSTM Network Using Electroencephalography Signals,” pp. 26–28, 2019, doi: 10.3390/brainsci9120348.