Fatigue Detector System

using Bayesian Networks

Kamil Bernacik, Artur Stefańczyk, Mikołaj Wielgos

Introduction

This scholarly report delineates a comprehensive analysis of the Fatigue Detector System, encompassing an integral data simulation and visualization module, with methodological foundations derived from empirical research on professional driver fatigue detection in working environment[1]. The investigative approach leverages two distinct Bayesian network architectures: a dedicated data generation simulation module and a rigorously independent fatigue detection network.

Fatigue evaluator module for simulator

Based on the Polish research findings[2], the study illuminates the nuanced environmental determinants of driver fatigue, delineating critical contextual variations between urban high-traffic corridors and monotonous extra-urban routes, with empirical evidence suggesting heightened driver somnolence in congested urban environments. Consequently, we operationalized a comprehensive set of environmental states for simulation, encompassing meteorological conditions, traffic density, and road typology, while integrating temporal factors such as circadian rhythms and duration of continuous wakefulness. The developed flat Bayesian network strategically interconnects these parametric nodes to a terminal "fatigue loss" evaluator, with conditional probability distributions meticulously calibrated through iterative system development. Leveraging this probabilistic framework, we quantify driver attentional degradation across a complex state space of 180 potential configurations, conceptualizing driver focus as a gradated metric initialized at 100 points, with complete cognitive exhaustion mandating immediate restorative intervention upon reaching zero.

Environment simulator module

In this probabilistic simulation framework, environmental parameters exhibit stochastic variability at each discrete time interval, representing potential microvariations in meteorological conditions, road typology, and traffic dynamics. Critically, given a 5-minute temporal resolution and a predefined 12-hour diurnal cycle, systematic day-night state transitions are algorithmically implemented at precisely 144-tick increments, ensuring systematic progression of circadian environmental parameters.

kbernacik@student.agh.edu.pl | astefanczyk@student.agh.edu.pl | wielgos@student.agh.edu.pl

The possible weather states are as follows:

- clear
- rain
- foggy
- snowy
- sunny

Furthermore traffic condition consists of one of these three: low, medium, high. Lastly, road type is oscilating between highway, city and rural.

Driver simulator module

The fatigue evaluator module translates calculated focus points into a driver simulator module, which generates biometric proxy indicators representing physiological stress and attentional degradation. Based on empirical research establishing baseline and drowsy state ranges, we map focus point levels to specific biometric metrics: heart rate, heart rate variability (HRV), electrodermal activity (EDA), percentage of eyelid closure (PERCLOS), blink duration, and blink frequency. As focus points decrease, corresponding biometric indicators dynamically shift, with metrics like PERCLOS demonstrating significant deviation from baseline values[3], thereby simulating progressive cognitive and physiological fatigue manifestations.

Biometric fatigue detector module

The fatigue detection module processes simulated biometric signals through a sophisticated Bayesian network comprising 4,096 states, employing empirically derived probabilistic inference mechanisms. The alarm threshold is calibrated at 60%, established through experimental validation and grounded in foundational research literature. The network's conditional probability distributions are strategically weighted, enabling differential impact of individual biometric metrics on the comprehensive fatigue score, thus providing a nuanced and adaptive fatigue assessment mechanism.

```
        self.WEIGHTS = {
             'HeartRate': 1,
             'HRV': 1,
             'EDA': 1,
             'PERCLOS': 4,
             'BlinkDuration': 4,
             'BlinkRate': 2
         }
```

Code snippet 1. Default weights setup

Example CPD adjustment for PERCLOS metric (as a part of Biometric Fatigue detector module's Bayesian Network).

```
        cpd_perclos = TabularCPD(
            'PERCLOS', 4, [
                [0.05],  # Normal/Rested (0.10-0.20)
                [0.1],  # Slightly Fatigued (0.20-0.30)
                [0.25],  # Fatigued (0.30-0.40)
                [0.6]   # Very Fatigued (>0.40)
            ])
```

Code snippet 2. CPD values for PERCLOS metric

Scenarios

We've prepared a different scenarios to run simulation with. The scenarios were developed to be used in EnvironmentSimulator and also DriverSimulator modules. The modular scenario design enables flexible combinations between environment and driver parameters, facilitating comprehensive fatigue detection system testing across diverse operational conditions.

Environment selections:

- normal
- bad
- worst

Driver selections

- normal
- low heart rate
- fast blinker
- eyes wide open
- sweaty palms
- drowsy

Evaluations

We tested our model based on the previous listed scenarios with the evaluation function:

```
accuracy = valid_alarms / (total_alarms + missed_alarms)
```

Code snippet 3. Accuracy metric calculation formula

Where *valid_alarms* is a total of valid alarms triggered throughout simulation run, *total_alarms* - total summed up amount of alarms, *missed_alarms* - alarms missed - alarms that should have been triggered but were not.

	normal	bad	worst
normal	62.48%	68.97%	69.08%
low heart rate	47.71%	56.01%	59.62%
fast blinker	45.91%	52.84%	59.69%
eyes wide open	8.14%	5.274%	0.95%
sweaty palms	57.92%	61.09%	53.86%
drowsy	32.47%	34.40%	35.13%

Table 1. Evaluation table for each scenario combination

Visualization module

The visualization module, developed utilizing the Pygame library, presents a comprehensive graphical user interface that meticulously displays system-wide information, offering intuitive guidelines and real-time results to enhance model interpretability and adaptability.

Picture 1. Graphical User Interface developed for the system

Results and conclusion

While our baseline implementation demonstrates efficacy for standard scenarios, professional deployment would necessitate personalized weight calibration. The current model's performance limitations are particularly evident in the "eyes wide open" scenario, where default weightings—overly reliant on PERCLOS and blink metrics—yield suboptimal results due to insufficient discriminative input from these specific physiological indicators.

Further improvements

Potential enhancements include implementing a parametric optimization algorithm to determine optimal universal model weights and Conditional Probability Distributions (CPDs), as well as developing a more flexible Graphical User Interface that supports dynamic scenario selection and custom scenario configuration, currently constrained by hardcoded implementations.

Inference time and use of resources

The Bayesian network's inference time proves computationally negligible, with the majority of system resources predominantly allocated to rendering the visualization module. Graphical rendering and interface interactions consume substantially more computational power, representing the most resource-intensive component of the fatigue detection system's architecture.

Technical requirements

To successfully run the system, user is required to download the source code as well as the following libraries/packages:

- python 3.11
- numpy 2.2
- pgmpy 0.1
- pygame 2.6

References

[1] https://onlinelibrary.wiley.com/doi/10.1155/2020/4219562
[2] https://link.springer.com/article/10.1007/s00484-018-1643-y
[3] https://academic.oup.com/sleepadvances/article/4/1/zpad006/7000589