FMDB *Transactions on Sustainable Computing Systems*

Hybrid Feature Selection Model with BH-XGBoost for Predicting the Drive Sleepiness in Naturalistic Road Driving Condition

C. N. Sangeetha[1,*], K. Vinoth Kumar[2], M. Arunadevi Thirumalraj[3], S. Venkatasubramanian[4], Jouma Ali Al-Mohamad[5]

[1,2]Department of Electrical and Electronics Engineering, New Horizon College of Engineering, Bengaluru, Karnataka, India.
[3]Department of Computer Science and Engineering, Karunya Institute of Technology and Science, Coimbatore, Tamil Nadu, India.
[3]Department of Computer Science and Business Management, Saranathan College of Engineering, Trichy, Tamil Nadu, India.
[4]Department of Computer Science and Business Systems, Saranathan College of Engineering, Trichy, Tamil Nadu, India.
[5]Department of Computer and Mobile Communications Engineering, Al-Shahbaa Private University, Aleppo, Aleppo Governorate, Syria.
sangeethacn.nhce@newhorizonindia.edu[1], dr.vinothkumar@newhorizonindia.edu[2], aruna.devi96@gmail.com[3], veeyes@saranathan.ac.in[4], jalmohamad@su.edu.sy[5]

Abstract: In 20% of fatal car accidents, drivers are tired. One of the most promising and commonly available technologies is vehicle driver fatigue monitoring. This item will determine if video-based analysis and machine learning can detect tiredness. Method development and evaluation were based on the experiences of 13 Swedish freeway drivers. Each driver participated in two 90-minute sessions, one during daylight (low sleepiness) and one at midnight (high sleepiness). The KSS sleepiness detection model outputs an alert or weary (KSS 7-9) or a score of 1-9. For driver tiredness prediction, we recommend BH-XGBoost, which uses Bayesian hyperparameter optimisation. Feature selection improves the model's classification accuracy. HBC enhances the BRO algorithm. For participant connection, the BRO method uses the chicken swarm optimisation (CSO) algorithm. An elite player update method that combines random exploration and directed updating within a confined zone may enhance growth. Real-time drowsiness detection can help control fatigue and prevent driving accidents by identifying severe indicators of weariness. With 100 iterations, the suggested model achieved 81% accuracy and 91% precision, and 79% accuracy and 83% precision without a feature selection model.

Keywords: Karolinska Sleepiness Scale; Driver Fatigue; Chicken Swarm Optimization; Battle Royale Optimization; Bayesian Hyperparameter Optimization; Elite Random Guidance; Gaussian Time Domain Network.

Received on: 13/11/2024, **Revised on:** 07/02/2025, **Accepted on:** 19/03/2025, **Published on:** 07/09/2025

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSCS

DOI: https://doi.org/10.69888/FTSCS.2025.000476

Cite as: C. N. Sangeetha, K. V. Kumar, M. A. Thirumalraj, S. Venkatasubramanian, and J. A. Al-Mohamad, "Hybrid Feature Selection Model with BH-XGBoost for Predicting the Drive Sleepiness in Naturalistic Road Driving Condition," *FMDB Transactions on Sustainable Computing Systems*, vol. 3, no. 3, pp. 176-191, 2025.

[*]Corresponding author.

1. Introduction

More people lose their lives, and more than 50 million are incapacitated worldwide every year due to car accidents [1]. Approximately 90% of vehicle events are caused by driver mistakes, such as impaired driving, accidental errors, and dangerous driving behaviours [2]. These errors are closely linked to crash risk. It is well-known that human factors, such as inattentive driving, sleepiness, alcohol, anger, and speeding, are major causes of fatal accidents [7]; [8]; [9]. According to data gathered in the United States, impaired driving accounts for around 35% of all traffic-related fatalities [6]. In 2016, there were 10,497 people killed due to drunk driving [10]. As a result of impairment, the driver is unable to exert sufficient control over the vehicle's steering and braking actuators [11]. The driver's steering, acceleration, and braking orders, as well as the command sequencing during manoeuvring, are reflected in the vehicle's sensor data, making it crucial for assessing vehicle motion [12]. Vehicle motion retort is an indicator of the driver's sufficient or inadequate orders and may thus be used to determine the driver's responsiveness to potential risks [3].

Driver conduct has a major impact on the efficiency and security of our transportation network [4]; [5]. Fortunately, as automation improves, vehicles may one day be equipped to take over driving duties entirely or, at the very least, execute a safe stopping manoeuvre if their drivers become too tired to do so. This technology, however, is in its infancy and still some ways off [13]; [16]. Therefore, both fully and conditionally autonomous cars require driver fatigue monitoring systems, the former to persuade drowsy motorists to pull over for a sleep and the latter to determine when the driver is too exhausted to resume control [14]; [15]. Most existing fatigue detection systems rely on either (i) data collected from the vehicle itself, such as lane keeping presentation, (ii) data collected from the driver themselves, as yawning and eye actions, or (iii) data collected from the driver's physiological sensors, such as electrooculography (EOG) [17]; [18]. However, the landscape for fatigue detection systems is shifting with the advent of advanced driver assistance systems and automated driving. Physiological sensor-based systems are sometimes intrusive and/or susceptible to motion artefacts, rendering vehicle-based data useless as indicators of tiredness until proper longitudinal and lateral posture has been established by the vehicle [19].

Video-based systems have the highest promise for accuracy, are the most user-friendly, and are the most practical method for detecting driver weariness [20]. Camera positioning, the number of cameras, and algorithms for detecting fatigue may need to be modified for future autonomous cars [21]. This is because the driver may be out of the camera's view or performing duties that obscure the camera's view of the motorist's face. Some of the proposed methods have the drawback of only recognising yawning and eyelid closure as signs of exhaustion. The high proportion of time spent with the eyelids closed (i.e., PERCLOS) and yawning is thus considered an indicator of exhaustion [22]. As we've seen, though, this isn't always the case, so these approaches only solve part of the problem [23]. Although very accurate eyelid closure detection is crucial, it is merely the beginning of the classification process for true tiredness [24]. Another drawback is that many studies rely on data from driving simulators or driving-based video games, and occasionally with drivers who are asked to appear drowsy by voluntarily extending their breaks, among other methods [25]. The extent to which such methods apply to the actual world and genuine weariness is unknown [40]. The main contribution of the research work is mentioned as follows:

- In this research, we describe a video-based driver drowsiness detection system that is suitable for real-time applications. We test it on a real-world dataset collected from drowsy drivers in a motorway driving scenario that mimics real-world conditions [29].
- To facilitate the feature selection process, this work proposes a novel hybrid algorithm (HBC). Participants in the BRO procedure are first stratified into elite and regular players using the CSO algorithm's level mechanism.
- Then, a rule is developed for top players that combines random exploration with strategic updates. Finally, refine the procedures for updating regular players, who now must follow the elite players' index. With the novel population-level method, individuals search the solution space in two distinct ways, boosting the BRO algorithm's global exploration accuracy.
- The algorithm's hyperparameters are also quite important. The same technique, when applied to new data with adjusted hyperparameters, can further enhance prediction accuracy.
- This study proposes a drowsiness prediction approach called BH-XGBoost, which optimises XGBoost performance by configuring hyperparameters during model training.

2. Related Works

Driver fatigue was investigated by Akrout and Mahdi [24], who proposed innovative approaches to the autonomous regulation of driver sleepiness. These remedies must rely solely on visual indicator analysis and be completely non-intrusive. These metrics are derived from the location and time data embedded in a video feed. Our primary focus is on using facial recognition to identify drivers exhibiting inappropriate behaviour. In fact, we've suggested a fusion system that accounts for yawn and somnolence detection, as well as 3D head posture estimation. The results of evaluating the three databases on the fusion system demonstrate the effectiveness of our proposed method. In this study, Li et al. [25] present a fatigue detection system that utilises

a neural network trained to recognise facial expressions and incorporates additional facial information. To effectively separate the contextual information available in sequentially ordered picture frames, we employ a neural network with a GRU layer. Additionally, we developed a multi-task (MTCNN) model for extracting detailed facial characteristics. Once the locations of the facial feature points are known, we may determine the angle formed by the face's top and lower eyelids, lips, and brows.

By comparing the subject's facial characteristics to a standard 3D face model, we can also determine the subject's head attitude at three different angles. Finally, we train a judgment network using data on the evolution of six characteristics across 20 successive images. Upon completion of the training phase, the judgment network model will make an instantaneous determination of the driver's tiredness level based on the six feature curves derived from the most recently entered 20 frames. The workstation platform can achieve 55 FPS of real-time detection using this fatigue-detection approach. The technique enables real-time detection at 26 FPS on the AGX embedded platform. This technique for identifying tiredness can achieve an accuracy of 97.47%. Gao et al. [26] present a new time- and space-frequency-based multi-dimensional feature fusion GTNet, for tiredness detection. It is composed of two distinct networks: the CSFNet (CSF Network) and the GTNet (Gaussian Time Domain Network). The experimental evidence demonstrates that the suggested approach can reliably differentiate between wakefulness and drowsiness. When associated with approaches, the accuracy rates on both the custom and SEED-VIG datasets are greater.

Additionally, using the brain topology map, we assess the role of each brain area in recognising fatigue. Using the heat map, we also investigate the fluctuating patterns in each frequency band and the importance of people in awake and sleepy states. Our findings have the potential to inspire new directions in the study of brain fatigue and play a decisive part in advancing the discipline. Using an Fp1 EEG channel, Shahbakhti et al. [27] introduce a novel approach for detecting driver weariness. To begin extracting blink-related information, the moving standard deviation technique is used to detect eye blink intervals (EBIs). Second, the EBIs are removed from the EEG signal by the discrete wavelet transform [47]. Third, different linear characteristics are retrieved from the filtered EEG signal's sub-bands. At last, a classifier is provided with the most salient features identified by the investigation to distinguish between drowsy and awake driving. In this study, we examine two distinct types of databases. The first is used to fine-tune the proposed approach for detecting and filtering eye blinks, measuring nonlinear EEG activity, and selecting candidate features [48].

The second serves only to verify that the adjusted settings are stable. Critical Outcomes: The AdaBoost classifier's findings across both databases indicate that the proposed technique for driver tiredness identification is reliable, with sensitivities of 90.2% and 87.7%, specificities of 88.4% and 86.8%, and accuracies of 88.4% and 86.8%, respectively. Since single-channel prefrontal EEG headbands are already on the market, the suggested approach can be utilised to identify driver fatigue in real-world settings [46]. For the first time, Shang et al. [28] describe a non-intrusive and effective approach for detecting driver weariness and emotional state. Driver fatigue indicators, including driver eye closure time (PERCLOS dataset, the proposed fatigue detection algorithm achieves an accuracy of 73.32%, while the emotion recognition network achieves an accuracy of 73.32%. Future studies in the field of assisted safe driving can benefit from the composite score generated via time series fusion, as it can more precisely capture the driver's condition across various contexts [45]. The MATCN-GT model in Croft and England [31] comprises a multi-scale attentional temporal convolutional neural network block (MATCN) and a graph Convolutional-Transformer block (GT block). The MATCN block analyses the characteristics of EEG signals between separate electrodes, and the GT block retrieves features directly from the raw EEG signal without prior knowledge [44].

Additionally, we develop a multi-scale attention module to prevent the loss of crucial data related to electrode correlations. To better capture the relationships between long-distance electrodes, we augment the graph convolutional network with a Transformer module. Experiments on the public dataset SEED-VIG demonstrated that the MATCN-GT model outperformed state-of-the-art algorithms, achieving an accuracy of 93.67%. The GT block has also increased accuracy by 3.25% compared to the standard graph convolutional neural network. The MATCN block outperforms conventional feature extraction techniques in terms of accuracy across various test subjects [42]. When it comes to detecting driver weariness, Zhou et al. [30] present a functional brain network-based approach that aims to integrate features and algorithms for maximum efficiency. To collect EEG signal data from several people engaged in a prolonged monotonous cognitive activity, we first set up a simulated driving experiment. Second, a functional brain network is built by computing the Pearson correlation coefficient for each pair of EEG signal channels [43]. Then, eight different machine learning algorithms [SVM], [LR], decision tree [DT], k-nearest neighbour [KNN], latent dirichlet allocation [LDA], and gradient boosting machine [GBM]) They are used as classifiers for fatigue detection.

The best set of features and algorithms is finally determined. According to the findings, the logistic regression technique performs best when using a feature set consisting of node degree, local efficiency, and typical path length [39]. This set obtains a classification accuracy of 92.92%. The models above focused on either ML or DL for driver sleepiness detection, without considering feature selection. Irrelevant features lead to poor classification accuracy; for example, the technique in Bakker et al. [32] achieved only 92%. Hence, it is crucial for optimal feature selection, which is achieved through a hybrid optimisation

model in the research work. Moreover, the XGBoost optimisation is not carried out using any existing techniques; this research focuses on introducing Bayesian optimisation [37].

3. Materials and Methods

Both the driver's face and the forward-facing driving situation are analysed in the first step to extract relevant characteristics. From the facial video, these characteristics indicate eye-opening and closing data. From the forward-facing camera, they indicate lane locations, exits, and changes. Both videos represent gaze direction, head stance, and facial expression. Due to the small size of the available sleepiness-specific dataset, this initial phase relies heavily on pre-trained models. Second-stage processing takes the retrieved features and predicts their association with drowsiness. Using data collected over the last 5 minutes, the drowsiness classification model may either make a binary judgment (awake or asleep) or provide a continuous, regression-like prediction. Python and C++ are used in the first and second stages, respectively. These two stages are further broken down into several modular subcomponents (detailed below) that utilise multiple deep learning frameworks and both multithreaded CPU and GPU computations [33]. With this setup, a single multicore i7 CPU laptop with an NVIDIA GTX 1080 can achieve around 25 frames per second. Through careful algorithm/software tweaking and by leveraging the multicore CPU to run the various components in parallel, we achieved the reasonably high throughput necessary for a real-time application [36].

3.1. Sleepiness Database

Twenty individuals (10 men and 10 women) were recorded for algorithm development and validation [34]. They averaged 23.1 years behind the wheel, drove 18172 10240 kilometres annually, and scored 6.9 on the Epworth Drowsiness Scale, indicating a mild to typical tendency toward excessive daytime drowsiness. Drivers had to be between 30 and 60 years old, in generally excellent health according to their own assessments, of a healthy weight (BMI 30), and neither shift workers nor professional drivers were allowed to participate. Two drivers were removed from the research while data were being collected for failing to follow the rules. Later, five more motorists were omitted due to missing information or poor data quality. The sample consisted of thirteen different motorists. The Swedish government has approved (N2007/5326/TR) to conduct roadside trials involving sleep-deprived drivers. The local Linköping ethics committee (dnr 2010/153-3) approved the study. All drivers signed informed consent forms. Over the three nights preceding the experiment day, participants were instructed to maintain a sleep diary and aim for at least seven hours of sleep each night. They were instructed to rise at 7:00 sharp on the morning of the trial. In addition to not drinking alcohol for three days, participants were asked to stay off nicotine and caffeine for the duration of the trial, which began three hours before the first driving session and ended at the end of the day.

Each driver stayed awake from the wee hours of the morning through two separate driving sessions, once during the day and once at night. Two people may participate in the experiment on each experimental day, as the experiment was designed to accommodate this. The first participant began driving at 15:30 and 00:15, and the second participant began driving at 17:45 (alert condition) and 02:45 (sleep-deprived condition), respectively. Thus, the awake condition was conducted during periods of strong circadian drive and low homeostatic sleep pressure, while the sleep-deprived condition was conducted during periods of low circadian drive and high homeostatic sleep pressure.

The driving lesson, which lasted around 90 minutes, took place on the E4, a public highway, just outside of Linköping, Sweden. With 9,000 to 14,000 cars daily, this route has a posted speed limit of 110 km/h. Before heading back toward Linköping, the participants spent the first half of the midday voyage with the sun in their eyes at an altitude of 20-30 degrees (which may have caused the kind of realistic eye squinting discussed above). The Volvo XC70, equipped with an automatic transmission, was used in the study. The Smart Eye Pro system, comprising three cameras, infrared cameras, and infrared illumination (Smart Eye AB, Gothenburg, Sweden), was used to capture the facial videos. The infrared camera system can capture sharp images in the dark.

As a bonus, it reduces the distinction between day and night photographs, making them seem almost indistinguishable. Not only were speeds and directions recorded, but also films were shot from the front and back of the car. In our analysis, we included only the forward-facing and facial footage captured by the central Smart Eye camera, positioned just above the steering wheel. A designated safety driver was always present in case any of the other drivers became too drowsy to continue or showed signs of being unable to drive safely. During data collection, the safety driver had no contact with the driver. The vehicle was equipped with a safety driver who could take over the controls in an emergency. This occurred on five separate occasions when the drivers fell asleep at the wheel. Subjective sleepiness was measured using the KSS, with individuals providing verbal ratings of their drowsiness. Extremely alert, very tired but making a significant effort to stay awake (9) are the nine anchored levels used by KSS. Every 5 minutes during the driving sessions, the participants reported their degree of tiredness. The second-stage model is trained using these KSS scores as benchmarks [35].

3.2. Sleepiness Detection Stage 1: Feature Selection

The most important information gleaned from forward-facing cams concerned lane deviation and lane changes. First, the approach by Zou et al. [34] for lane recognition and intra-lane location. It was pretrained using the TuSimple open lanes dataset and summarises information about lane changes and lane deviations [38]. In this step, lane modifications that cause deviations are filtered out (because they would muddy the results of the lane deviation analysis). The final outputs of the first- and second-stage models are concatenated, yielding a fixed-size feature vector with 574 elements (550 + 24). These features are optimally selected using hybrid optimisation, as explained in the following section.

3.2.1. Battle Royale Optimisation Algorithm

In 2017, Zou et al. [34] suggested the BRO procedure. It resembles the popular battle royale video game, in which players must explore their surroundings and utilise objects to stay alive (Figure 1).

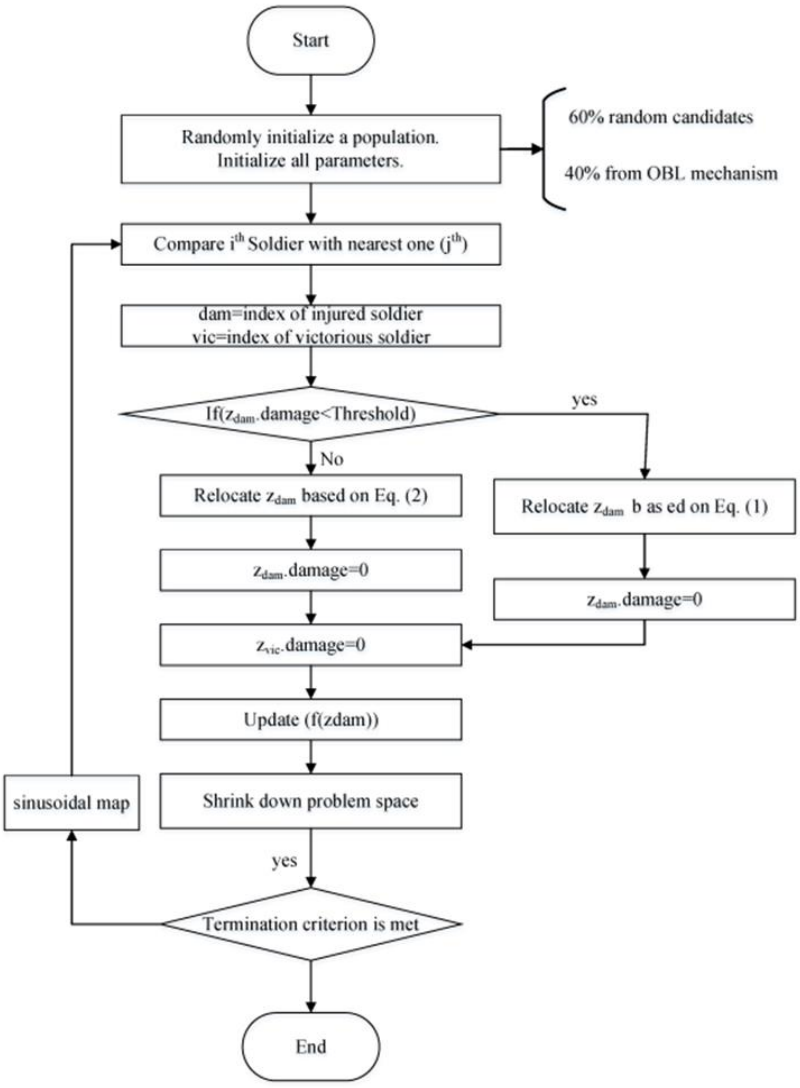

Figure 1: Developed battle royal algorithm

The goal of the game is to stay alive while avoiding enemy fire in a safe zone that keeps getting smaller. Any player who is not in a safe zone will take damage according on their health value, and if they go too far from cover, they could be killed. The BRO approach begins with a group of n people who are randomly placed in a space with D dimensions. Information on the ith player's position and damage in the tth iteration of the population is denoted by x_it (i [1, n]) and x_(i, dam)t, respectively. As the process repeats, the nearest two players in terms of Euclidean distance will engage in combat. The player who is in a better position and stays there without taking damage is the winner. The player who is in a worse position is thought to be on the

losing side because both players are attacking each other. As seen in equation (1), the winner gets an update on their damage. At the same time, the location of the loser changes based on the current position, as shown in equation (2), so that they can protect themselves and attack from different angles.

$$x_{i,dam}^{t+1} = x_{i,dam}^{t} + 1 \quad (1)$$

$$x_i^{t+1} = x_i^t + r(x_{best} - x_i^t) \quad (2)$$

$$x_i^{t+1} = r(ub_d - lb_d) + lb_d \quad (3)$$

$$lb_d = x_{best} - SD(\overline{x_d}) \quad (4)$$

$$ub_d = x_{best} - SD(\overline{x_d}) \quad (5)$$

$SD(\overline{x_d})$ is the best option right now; it's better than the population's standard deviation. If lb_d and ub_d are too large relative to the solution space's minimum and maximum, respectively, return them to their initial values. As shown in Equation (6), the area update threshold, D, is dynamically adjusted over time.

$$\Delta = \Delta + \text{round}\left(\frac{\Delta}{2}\right) \quad (6)$$

The initial value Δ is $\log_{10}(T_{max})$, where T_{max} is the maximum sum of iterations.

3.2.2. Chicken Swarm Optimisation Procedure

The CSO algorithm is a metaheuristic procedure published in 2014 that draws inspiration from the foraging behaviour of wild chickens [41]. The algorithm's computational performance is significantly enhanced by introducing a level mechanism and establishing hierarchical relationships among individuals in the population. The CSO algorithm classifies the population into roosters, hens, or chicks. Additionally, the rooster establishes a random order for the population's relocation; the hens follow the rooster, and the chicks follow their mother hen. As the process iterates, the level mechanism is updated, individuals learn from one another, and engage in healthy competition to get the best possible outcome. The solution space has dimension D, and there are N chickens in it. The position of the ith chicken at time t is denoted as X_it (i[1, N]). Equations (7) and (8) depict the position update of the rooster subgroup, whose number is N_r. Roosters have the best position in the population.

$$X_i^{t+1} = X_i^t * (1 + \text{randn}(0, \sigma^2)) \quad (7)$$

$$\sigma^2 = \begin{cases} 1 & f_i \le f_k \\ \exp\left(\frac{(f_k - f_i)}{|f_i + \varepsilon|}\right) & \text{otherwise} \end{cases} \quad (8)$$

Where $\text{randn}(0, \sigma^2)$ is a random Gaussian with a deviation of σ^2, ε is the least continuous in the computer, k is a random sum ($k \in [1, Nr], k \neq i$), f represents the fitness value of a separate. There are N_h hens, and they will all try to outperform one another by updating their position relative to a rooster. Equations (9) -(11) depict the position update.

$$X_i^{t+1} = X_i^t + s_1 * r(X_{r1}^t - X_i^t) + s_2 * r(X_{r2}^t - X_i^t) \quad (9)$$

$$s_1 = \exp\left(\frac{f_i - f_{r1}}{abs(f_i) + \varepsilon}\right) \quad (10)$$

$$s_1 = \exp(f_{r2} - f_i) \quad (11)$$

Where X_{r1}^t characterises the site of the rooster shadowed by the ith hen, Xrt2 is an arbitrarily designated position for either a rooster or a hen. $X_{r1}^t - X_{r2}^t$. The number of chicks is N_c, the chick moves with the site, which is updated as shown in equation (12).

$$X_i^{t+1} = X_i^t + FL * (X_m^t - X_i^t) \quad (12)$$

Where X_m^t is position of the mother hen, FL($FL \in [0, 2]$) is the subsequent coefficient.

3.2.3. Hybrid Improved Algorithm (HBC)

This work presents an HBC algorithm, a variant of the BRO method, to improve the computational performance of the BRO procedure. This section introduces the HBC algorithm, describes its fundamental principles, and outlines the steps involved in implementing it. According to equation (13), the BRO algorithm classifies players into two categories, "Elite players" and "ordinary players," and the HBC algorithm results from integrating the CSO algorithm. By designating certain individuals in the population as elite players and the remaining individuals as ordinary players, the population's level mechanism is formed. Each high-tier player's update target is paired at random with a low-tier player, establishing the level connection. Every Gn iteration, the level mechanism and level relationship are reset.

$$x_n = [x_a, x_b] \tag{13}$$

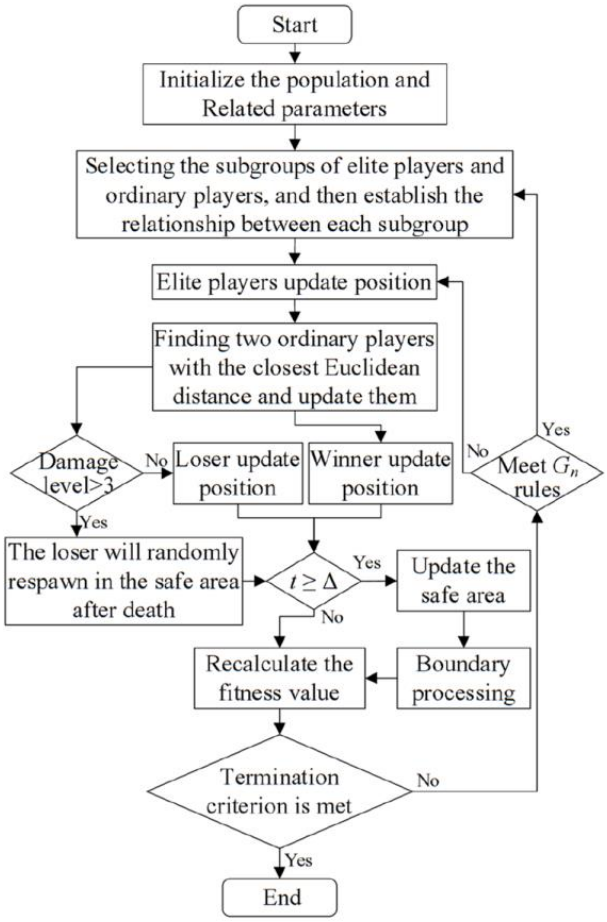

Figure 2: Flowchart of the HBC model

Where x_a is the subgroup elite player, x_b is the subgroup an ordinary player. The HBC algorithm benefits from the superior positional data carried by the top player subgroup., $X_{i,a}^{t}(i \in [1, n_a])$ stands for the coordinate data it represents. The algorithm's solution may be more precise if it were given unrestricted access to its immediate surroundings. This is why we recommend the elite player update rule, which alternates between exploratory randomness and direction updates. The CSO algorithm's use of the rooster random walk's update mechanism facilitates better directional exploration by top players. Equation (14) depicts the updating rule.

$$X_{i,a}^{t+1} = \begin{cases} x_{i,a}^{t} * \left(1 + \text{randn}(0, \sigma^2)\right) & p \geq 0.5 \\ x_{i,a}^{t} + \sigma^2 * r\left(x_{best} - x_{i,a}^{t}\right) & p < 0.5 \end{cases} \tag{14}$$

Where $x_{best}$is the position, $\text{randn}(0, \sigma^2)$ is a random sum in equation (8). $p \in [0, 1]$. Standard players make up the bulk of the population, and their positions are represented by the vector x_(i,b)t (i[1,n_b]). Improving the algorithm's global exploration ability and convergence time may depend on its capacity to efficiently and diversely explore the solution space. Therefore, the

conventional player update rule is enhanced in the HBC algorithm, with the following enhancements based on the player algorithm. To begin, the Euclidean distance is used to determine the neighbouring player of the regular player. The fitness levels of the two average players should then be determined. The player in the stronger position triumphs (xtj, b), and nobody is hurt. If a player is in an unfavourable position (xti, b), they take an extra one damage. Lastly, to fully utilise the tremendous position information in the population, the winner is updated with the global ideal position as the target, as shown in equation (15), and the loser is updated with the target, as described in equation (16).

$$x_{j,b}^{t} = x_{j,b}^{t} + r\left(x_{best} - x_{j,b}^{t}\right) \quad (15)$$

$$x_{i,b}^{t+1} = x_{i,b}^{t} + r\left(x_{k,a}^{t} - x_{i,b}^{t}\right) \quad (16)$$

Where $x_{k,a}^{t}$ is an elite player arbitrarily designated, x_{best} is the global ideal position. Figure 2 presents the flow chart of the proposed hybrid model.

3.2.4. Implementation of HBC Algorithm

The following are the precise procedures for implementing the HBC algorithm, in accordance with its premise described in the preceding section:

Algorithm 1: HBC's Model Description
Determine the sample size, or n. The percentage of top-tier players (na) and the percentage of regular players (nb). The maximum number of iterations possible (T_max). The population is initialised at random, and its fitness is determined based on the regional update threshold () and other relevant characteristics. The population is split into elite and regular players based on their fitness values. Additionally, the connection between subsequent groups is strengthened. Adjust the stats of the top players using equations (8 and 14), and those of the rest of the team using equations (2, 3) and (15). Calculate new fitness values for each person and update the algorithm's secure region using equations (4) through (6). Keep track of the highest fitness index and the fittest person. Determine whether the termination condition is met by either the fitness value or the maximum number of iterations. The algorithm will complete if this condition holds. Then, return to step 2.

.

3.3. Classification using Machine Learning

Based on optimisations and enhancements to the gradient progressive regression tree technique, XGBoost (eXtreme Gradient Boosting) is used. Due to its superior learning impact and fast training model speed, it quickly gained popularity after its introduction. The technique efficiently utilises the CPU's multi-threading capabilities, enabling it to create a boosting tree and perform parallel operations. The results of this method in Kaggle's Higgs sub-signal recognition competition have been really promising. Its core idea is to sort features in advance, which improves categorisation trees. The projected value in XGBoost is derived from a combined model of k decision trees:

$$\hat{y}_i = \sum_{t=1}^{k} f_t(x_i) \quad (17)$$

$\hat{y}_i$ is the predicted value of x_i. ft represents a decision tree, and $f_t(x_i)$ is the foretold value of f_t given x_i. The loss function L can then be expressed in terms of the forecast value. $\hat{y}_i$ and true values y_i:

$$L = \sum_{i=1}^{n} l(y_i, \hat{y}_i) \quad (18)$$

Overfitting can be prevented and the model's generalisation capacity enhanced by including a regularisation term. In Eq. 19, we rewrite the loss function, and in Eq. 20, we define the regular term (f_t), where T is the total number of leaf nodes.

$$Obj = \sum_{i=1}^{n} l(\hat{y}_i, y_i) + \sum_{t=1}^{k} \Omega(f_t) \quad (19)$$

$$\Omega(f_t) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2 \quad (20)$$

When adding a decision tree to an XGBoost model, the model learns a novel function and its coefficients to match the residuals from the previous stage of the prediction process. The ith sample x_i's prediction is given by Eq. 21, and the step-t model's objective function is given by Eq. 22.

$$\hat{y}_i^t = \hat{y}_i^{t-1} + f_t(x_i) \quad (21)$$

$$Obj^{(t)} = \sum_{i=1}^{n} L\left(y_i, \hat{y}^{t-1} + f_t(x_i)\right) + \Omega(f_t) \quad (22)$$

Use Taylor growth to enlarge the above equation:

$$Obj^{(t)} \approx \sum_{i=1}^{n} \left[L(y_i, \hat{y}^{t-1}) + g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)\right] + \Omega(f_t) \quad (23)$$

Among them, g_i represents the first copy of $L(y_i, \hat{y}^{t-1})$ to $\hat{y}^{t-1}$, and hi represents the second copy of $(y_i, \hat{y}^{t-1})$to $\hat{y}^{t-1}$. Furthermore, identical function values on the same leaf node might be merged. Therefore, we can conclude the following:

$$w_j = \frac{G_j}{H_j + \lambda} \quad (24)$$

Therefore, relieving w_j into the objective function is basic as:

$$Obj^{(t)} = -\frac{1}{2} \sum_{j=1}^{T} \frac{G_j^2}{H_J + \lambda} + \gamma T + C \quad (25)$$

As mentioned above, the equation simplifies the value. As shown, XGBoost supports a tunable goal function, restricts calculations to first- and second-order derivatives, and yields a reduced equation.

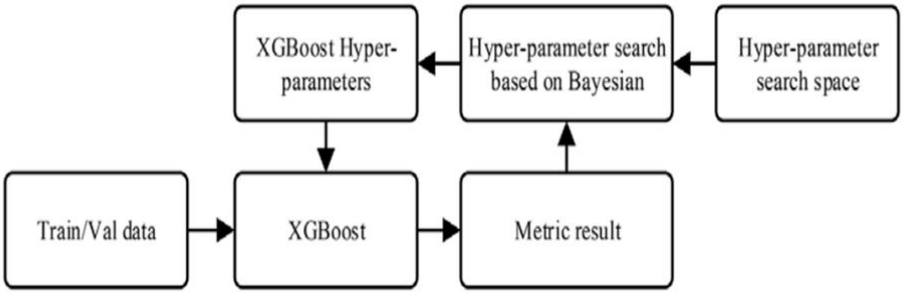

Figure 3: Working flowchart of the proposed classifier

Choosing appropriate XGBoost parameters is crucial. The effectiveness of the XGBoost procedure model is controlled by the parameters max_depth and learning_rate [47]. Figure 3 presents the working flow of the proposed model.

3.3.1. Hyper-Parameter Principle

In machine learning, hyperparameters are the parameters that define a model's architecture. The number of classes in a clustering algorithm, the sum of topics in a topic perfect, etc., are all examples of hyperparameters. The algorithms employed in machine learning have found many applications. To tailor the machine learning model to various issues, its hyperparameters must be adjusted. Bayesian hyperparameter optimisation is chosen for this article. Network learning seeks to discover the mapping y = f(x), where y is the output and x is the size of the mapping, determined by the vector. Bayesian methods focus on establishing a probability model of the goal function by adjusting the model's hyperparameters. By applying the acquisition function, we can efficiently search for and ultimately choose the best possible set of hyperparameters. Let's consider the GBRT hyperparameter to be an optimisation target in an n-dimensional space. The hyper-parameter that yields the smallest value of the loss function, f (), is located in the set AXd, as indicated in the following equation.

$$\theta^* = \arg\min_{\theta \in A} f(\theta) \quad (26)$$

Since the structure of UNKNOWN is presumed to be masked by observational noise, no prior understanding of its components is required.

$$y(\theta) = f(\theta) + \varepsilon, \text{and } \varepsilon \sim N(0, \sigma^2_{noise}) \quad (27)$$

There are two primary choices within the Bayesian framework. The optimisation process begins by selecting a hypothesis, p(f|D), that reflects beliefs about the function f. Second, the acquisition function for deciding the next test point is set by the posterior model. Based on the data sample D, the Bayesian framework constructs a model of the objective function using the hypothesis function p(f|D). The model chooses between optimisation and development based on the present p(f|D) model. Surrogate functions, such as the Tree Parzen Estimator (TPE) and the Random Process, are the primary distinguishing features of the Bayesian optimisation model.

3.3.2. BH-XGBoost Technique

In this research, we employ Bayesian theory to fine-tune the XGBoost's hyperparameters for forecasting wind power over the next few days. The XGBoost search space for each hyperparameter must be defined before training begins. The initial values of the search space hyperparameters, referred to as the initial hyperparameters, are randomly selected for the first iteration.

4. Results and Discussion

The accuracy of the final classifier was measured using the test dataset. It's essential to note that the training dataset was used to select the winner, and the test dataset was used solely to evaluate it. This was done to ensure that no information would be lost during classifier selection. The applicability of the resulting classifier to drivers who were hidden from view was also estimated using a leave-one-out evaluation. The method involved using all but one of the individuals in the study as training data. All participants underwent this process again, and the average accuracy and specificity were calculated. The effectiveness of the suggested model is tested using the following equations.

$$\text{Recall} = \frac{tp}{tp+fn} \quad (28)$$

$$\text{Precision} = \frac{tn}{tn+fp} \quad (29)$$

$$\text{Accuracy} = \frac{tp+tn}{tp+fp+tn+fn} \quad (30)$$

$$F - \text{Measure} = \frac{tp}{tp+1/2(fp+fn)} \quad (31)$$

Where TP refers to the properly classified process, FP to the incorrectly classified one, FN to the incorrectly normal one, and TN to the correctly abnormal one, the existing models, such as ADB, DT, RF, KNN, SVM, and LDA, are taken from Bakker et al. [32] and implemented with our dataset, then results are averaged in Table 1.

Table 1: Analysis of the proposed model with the existing technique

Models	Precision	Recall	F-Measure	AUC	Accuracy
ADB	0.812	0.860	0.835	0.867	0.84
DT	0.791	0.791	0.791	0.877	0.79
RF	0.918	0.916	0.915	0.943	0.81
KNN	0.791	0.791	0.791	0.791	0.88
SVM	0.909	0.909	0.909	0.927	0.91
LDA	0.93	0.93	0.93	0.93	0.93
XGBoost	0.96	0.966	0.966	0.986	0.97
BH-XGBoost	0.974	0.974	0.974	0.989	0.98

In this analysis, we evaluated different methods; the first, ADB, achieved precision, recall, F-Measure, AUC, and accuracy of 0.812, 0.860, 0.835, 0.867, and 0.84, respectively.

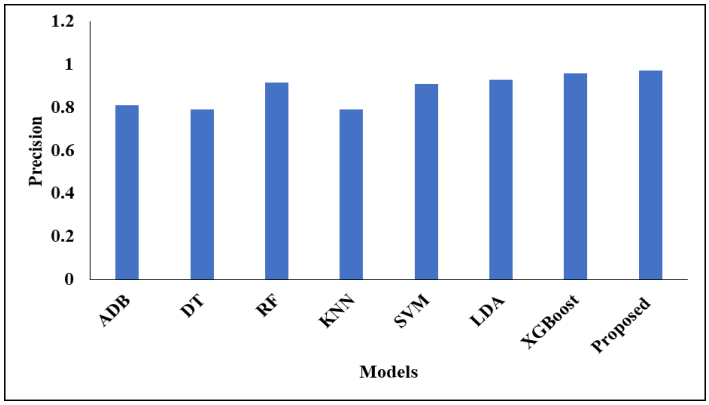

Figure 4: Analysis of precision

Another method achieved precision, recall, and F-Measure values of 0.791, an AUC of 0.877, and a final accuracy of 0.79. Additionally, the RF achieved precision, recall, F-measure, and AUC values of 0.918, 0.916, 0.915, and 0.943, respectively, along with an accuracy value of 0.81. After that, the KNN achieved precision, recall, F-Measure, AUC, and accuracy values of 0.791, 0.791, 0.791, 0.791, and 0.88, respectively.

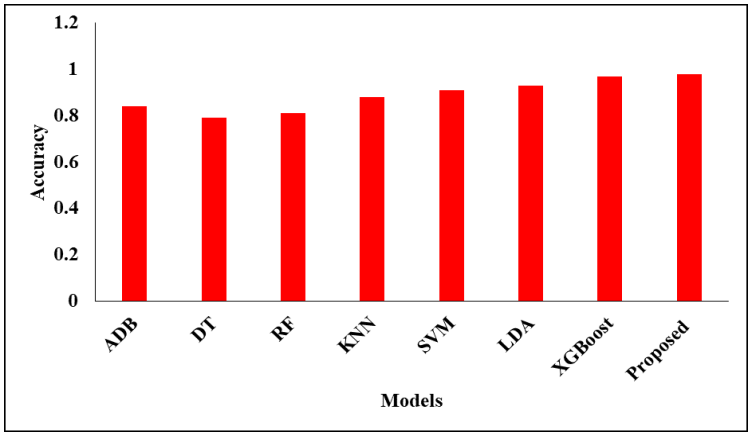

Figure 5: Accuracy analysis

Then, SVM achieved precision, recall, F-measure, AUC, and accuracy values of 0.909, 0.909, 0.909, 0.927, and 0.91, respectively. Additionally, another LDA achieved precision, recall, and F-Measure values of 0.93, as well as an accuracy value of 0.93. However, another model, XGBoost, achieved a precision of 0.96, a recall of 0.966, an F1 score of 0.966, an AUC of 0.986, and an accuracy of 0.97.

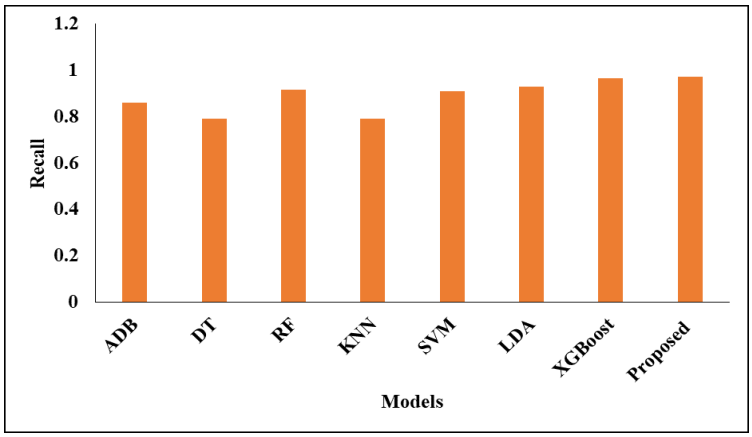

Figure 6: Comparison of recall

The last model, BH-XGBoost, achieved precision, recall, and F-Measure values of 0.974, 0.974, and 0.974, respectively, along with an AUC of 0.989 and an accuracy of 0.98.

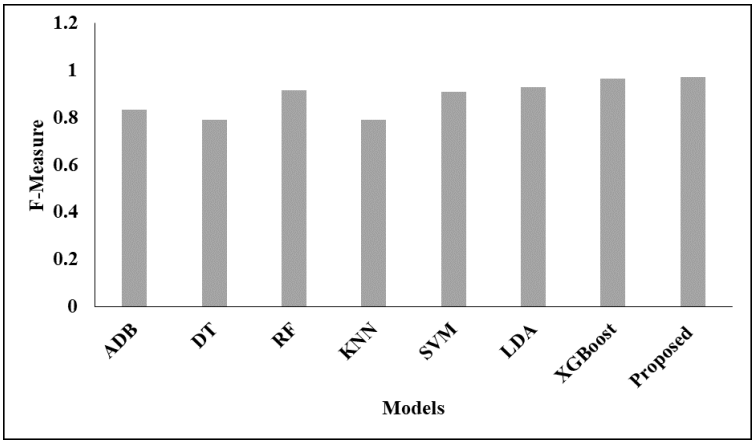

Figure 7: F-measure analysis

In this comparison, the BH-XGBoost model achieved better results than the other models.

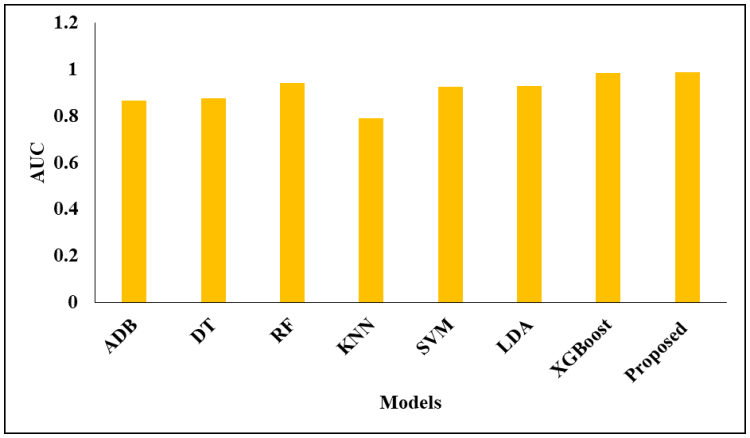

Figure 8: Validation on AUC

Figures 4 to 8 present graphical analyses of the various models. Table 2 below represents the analysis of feature selection using the Proposed Model. In this analysis, we absorbed different sums from 5 to 200 iterations. In the proposed model, the 5th iteration achieved a precision of 0.812, while the one without HBC achieved 0.786. In the projected model, the recall value of 0.860 and the value without HBC attained a precision value of 0.857. In the proposed model, the f-measure value of 0.835, as well as the one without HBC, achieved a precision value of 0.823. In the proposed model, the accuracy was 0.84, and the value without HBC achieved a precision of 0.82. Another iteration is the 50th, where the proposed model reached a precision of 0.791, and without HBC, it also achieved a precision of 0.775.

Table 2: Analysis of feature selection with the proposed model

No. of Iterations	Precision		Recall		F-Measure		Accuracy	
	Proposed	Without HBC	Proposed	Without HBC	Proposed	Without HBC	Proposed	Without HBC
5	0.812	0.786	0.860	0.857	0.835	0.823	0.84	0.82
50	0.791	0.775	0.791	0.760	0.791	0.776	0.79	0.78
100	0.918	0.774	0.916	0.838	0.915	0.804	0.81	0.80
150	0.909	0.909	0.909	0.775	0.909	0.776	0.88	0.66
200	0.974	0.720	0.974	0.735	0.974	0.727	0.98	0.73

In the proposed model, the recall value of 0.791, as well as the value without HBC, achieved a precision value of 0.760. In the projected model, the f-measure value of 0.791, as well as the one without HBC, achieved a precision value of 0.776. In the proposed model, the accuracy was 0.79, and in the model without HBC, it was 0.78. The 100th-iteration proposed model achieved a precision of 0.918, while the model without HBC achieved 0.774. In the proposed model, the recall value of 0.916 and the value without HBC achieved a precision value of 0.838. In the proposed model, the f-measure value of 0.915, as well as the one without HBC, achieved a precision value of 0.804.

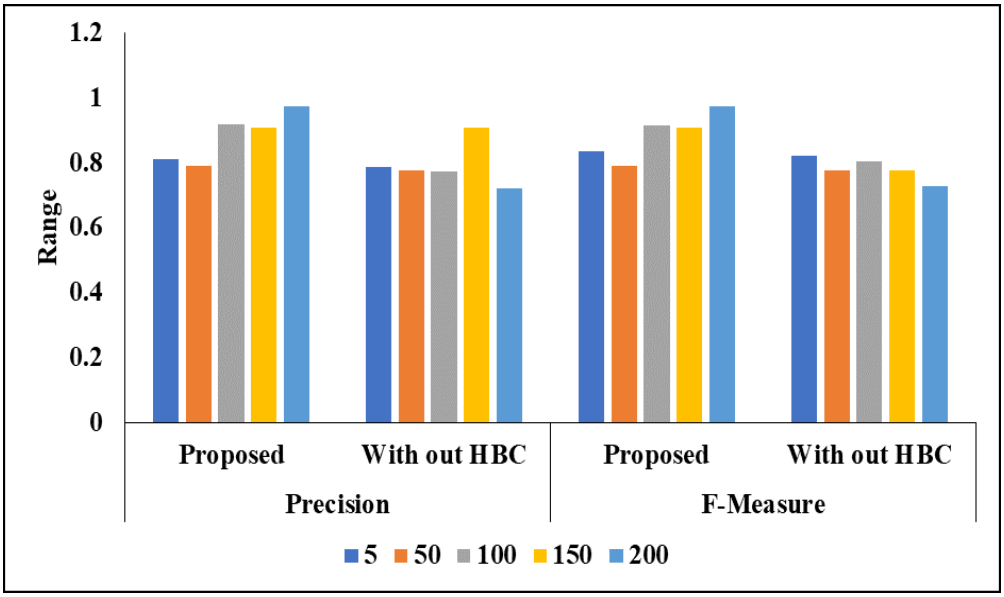

Figure 9: Analysis of the proposed model without HBC

In the proposed model, the accuracy was 0.81, and in the model without HBC, it was 0.80. After 150 iterations, the projected model reached a precision of 0. In the proposed model, ll was 0.909, and the one without HBC achieved a precision of 0.776. In the proposed model, the accuracy is 8, whereas in the model without HBC, it is. In the proposed model, the 200th iteration reached a precision of 0.974, and without HBC, it also achieved a precision of 0.920. In the proposed model, the recall value of 0.974 and the value without HBC attained a precision value of 0.735. In the proposed model, the f-measure was 0.974, whereas the one without HBC achieved a precision of 0.73. In the proposed model, the accuracy was 0.98, and in the model without HBC, it was 0.73. Figures 9 and 10 present a graphical comparison of different metrics.

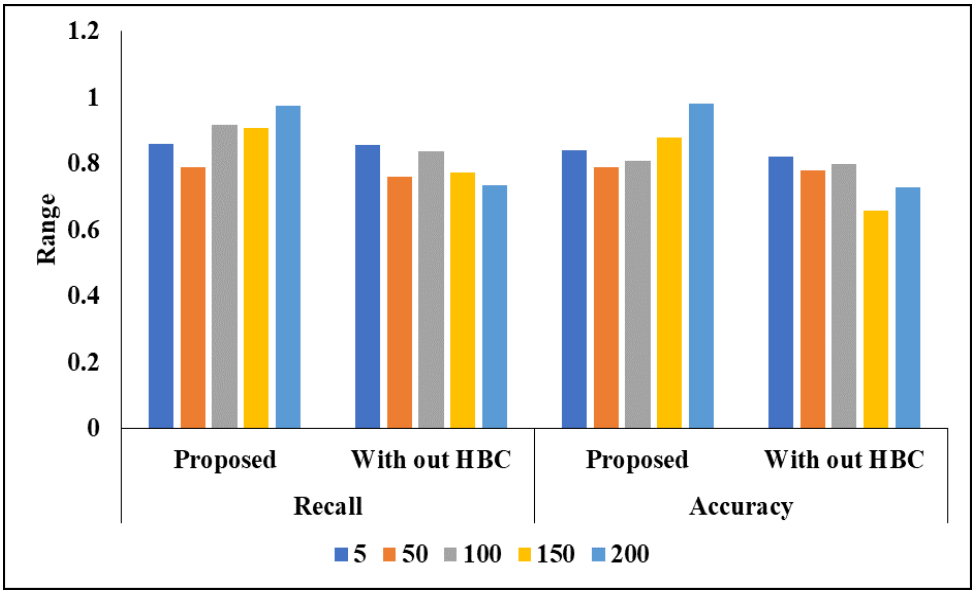

Figure 10: Analysis of the proposed classifier without feature selection

5. Conclusion

It is now possible to use machine learning with a tailored feature selection approach. An XGBoost classifier-based system for detecting drowsy driving has been built and thoroughly evaluated using data collected from actual roads and traffic conditions. In this study, we present the HBC algorithm, a hybrid enhanced BRO method, to choose the best characteristics. The CSO level method is included in the BRO population division, which separates the best players from the rest of the population. The connection between the subgroups has been established for further observation. The elite player updating approach incorporates both random exploration and directed update. An enhanced update approach for regular players involves the loser updating to the ideal position and the winner updating to the top player. The findings emphasise the need for individualised methods to detect drowsiness. The best-fit regression can be generated using a Bayesian optimisation hyperparameter in the proposed technique. The assessments were conducted using a high-validity dataset gathered from tired drivers in naturalistic driving conditions; therefore, the findings are encouraging. The classifier's sensitivity suffered greatly when tested on data from an individual it had not been trained on. As a future study of the research work, it is necessary to collect data from a larger sample of drivers (with a wide diversity of facial features/geometries, in alert and drowsy states, and under task-related tiredness conditions).

Acknowledgement: The authors acknowledge the valuable support and cooperation of New Horizon College of Engineering, Karunya Institute of Technology and Science, Saranathan College of Engineering, and Al-Shahbaa Private University.

Data Availability Statement: Data are taken from Reference [24] and the data are publicly available.

Funding Statement: This research and the preparation of the manuscript were carried out without any external financial assistance or funding support.

Conflicts of Interest Statement: The authors declare that there are no conflicts of interest related to this work. The study is an original contribution by the authors, and all relevant citations and references have been properly acknowledged.

Ethics and Consent Statement: This study was conducted in accordance with established ethical standards. Informed consent was obtained from all participants before their involvement in the research.

References

1. I. Sar, A. Routray, and B. Mahanty, "A Review on Existing Technologies for the Identification and Measurement of Abnormal Driving," *International Journal of Intelligent Transportation Systems Research*, vol. 21, no. 1, pp. 159-177, 2023.
2. G. Sikander and S. Anwar, "Driver fatigue detection systems: A review," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 6, pp. 2339–2352, 2018.
3. N. Muthukumaran, N. R. G. Prasath, and R. Kabilan, "Driver Sleepiness Detection Using Deep Learning Convolution Neural Network Classifier," *in Proc. 2019 Third Int. Conf. on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)*, Palladam, India, 2019.
4. Q. Abbas and A. Alsheddy, "Driver fatigue detection systems using multi-sensors, smartphone, and cloud-based computing platforms: a comparative analysis," *Sensors*, vol. 21, no. 1, p. 56, 2021.
5. Z. Zhao, N. Zhou, L. Zhang, H. Yan, Y. Xu, and Z. Zhang, "Driver fatigue detection based on convolutional neural networks using EM-CNN," *Computational Intelligence and Neuroscience*, vol. 2020, no. 1, p. 7251280, 2020.
6. I. Jahan, K. M. A. Uddin, S. A. Murad, M. S. U. Miah, T. Z. Khan, M. Masud, S. Aljahdali, and A. K. Bairagi, "4D: A real-time driver drowsiness detector using deep learning," *Electronics*, vol. 12, no. 1, p. 235, 2023.
7. B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection system based on multi-task ConNN," *IEEE Access*, vol. 8, no. 1, pp. 12491–12498, 2020.
8. Y. Ma, B. Chen, R. Li, C. Wang, J. Wang, Q. She, Z. Luo, and Y. Zhang, "Driving fatigue detection from EEG using a modified PCANet method," *Computational Intelligence and Neuroscience*, vol. 2019, no. 1, p. 4721863, 2019.
9. Z. Wang, B. Yu, X. Pan, and C. Qin, "Research on fatigue detection algorithm of vehicle driver based on computer vision," *in Proc. 3rd Int. Conf. on Intelligent Computing and Human-Computer Interaction (ICHCI)*, Guangzhou, China, 2023.
10. B. K. Savaş and Y. Becerikli, "Real time driver fatigue detection based on SVM algorithm," *in Proc. 2018 6th Int. Conf. on Control Engineering and Information Technology (CEIT)*, Istanbul, Turkey, 2018.
11. Z. Liu, Y. Peng, and W. Hu, "Driver fatigue detection based on deeply-learned facial expression representation," *Journal of Visual Communication and Image Representation*, vol. 71, no. 8, p. 102723, 2020.

12. Y. Zhang, H. Guo, Y. Zhou, C. Xu, and Y. Liao, “Recognising drivers’ mental fatigue based on EEG multi-dimensional feature selection and fusion,” *Biomedical Signal Processing and Control*, vol. 79, no. 1, p. 104237, 2023.
13. C. Zhao, Z. Gao, Q. Wang, K. Xiao, Z. Mo, and M. J. Deen, “FedSup: A communication-efficient federated learning fatigue driving behaviors supervision approach,” *Future Generation Computer Systems*, vol. 138, no. 1, pp. 52–60, 2023.
14. F. Luo, B. Chen, X. Ran, W. Ouyang, and L. Shang, “PEO-PDMS-based triboelectric nanogenerators as self-powered sensors for driver status monitoring,” *Chemical Engineering Journal*, vol. 451, no. 1, p. 138961, 2023.
15. W. Liu, J. Qian, Z. Yao, X. Jiao, and J. Pan, “Convolutional two-stream network using multi-facial feature fusion for driver fatigue detection,” *Future Internet*, vol. 11, no. 5, p. 115, 2019.
16. R. Li, R. Gao, and P. N. Suganthan, “A decomposition-based hybrid ensemble CNN framework for driver fatigue recognition,” *Information Sciences*, vol. 624, no. 5, pp. 833–848, 2023.
17. I. A. Fouad, “A robust and efficient EEG-based drowsiness detection system using different machine learning algorithms,” *Ain Shams Engineering Journal*, vol. 14, no. 3, p. 101895, 2023.
18. P. K. Singh, A. Gupta, M. Upadhyay, A. Jain, M. Khari, and P. S. Lamba, “Multimodal Driver Drowsiness Detection from Video Frames,” *Journal of Mobile Multimedia*, vol. 19, no. 2, pp. 567–586, 2023.
19. X. Chen and Y. Gao, “Multi-view classification via twin projection vector machine with application to EEG-based driving fatigue detection,” *in Proc. Int. Conf. on Data Mining and Big Data (DMBD)*, Beijing, China, 2022.
20. K. Gupta, S. Choubey, N. Yogeesh, P. William, T. N. Vasanthakumari, and C. P. Kale, “Implementation of Motorist Weariness Detection System using a Conventional Object Recognition Technique,” *in Proc. 2023 Int. Conf. on Intelligent Data Communication Technologies and Internet of Things (IDCIoT)*, Bengaluru, India, 2023.
21. Y. Ed-Doughmi, N. Idrissi, and Y. Hbali, “Real-time system for driver fatigue detection based on a recurrent neuronal network,” *Journal of Imaging*, vol. 6, no. 3, p. 8, 2020.
22. J. S. Bajaj, N. Kumar, R. K. Kaushal, H. L. Gururaj, F. Flammini, and R. Natarajan, “System and method for driver drowsiness detection using behavioral and sensor-based physiological measures,” *Sensors*, vol. 23, no. 3, p. 1292, 2023.
23. L. Pan, C. Yan, Y. Zheng, Q. Fu, Y. Zhang, Z. Lu, Z. Zhao, and J. Tian, “Fatigue detection method for UAV remote pilot based on multi feature fusion,” *Electronic Research Archive*, vol. 31, no. 1, pp. 442–466, 2023.
24. B. Akrout and W. Mahdi, “A novel approach for driver fatigue detection based on visual characteristics analysis,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 14, no. 1, pp. 527–552, 2023.
25. D. Li, X. Zhang, X. Liu, Z. Ma, and B. Zhang, “Driver fatigue detection based on comprehensive facial features and gated recurrent unit,” *Journal of Real-Time Image Processing*, vol. 20, no. 2, p. 19, 2023.
26. D. Gao, P. Li, M. Wang, Y. Liang, S. Liu, and J. Zhou, “CSF-GTNet: A novel multi-dimensional feature fusion network based on Convnext-GeLU-BiLSTM for EEG-signals-enabled fatigue driving detection,” *IEEE Journal of Biomedical and Health Informatics*, vol. 28, no. 5, pp. 2558-2568, 2023.
27. M. Shahbakhti, M. Beiramvand, E. Nasiri, S. M. Far, W. Chen, J. Sole-Casals, M. Wierzchon, A. Broniec-Wojcik, P. Augustyniak, and V. Marozas, “Fusion of EEG and Eye Blink Analysis for Detection of Driver Fatigue,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 31, no. 4, pp. 2037-2046, 2023.
28. Y. Shang, M. Yang, J. Cui, L. Cui, Z. Huang, and X. Li, “Driver Emotion and Fatigue State Detection Based on Time Series Fusion,” *Electronics*, vol. 12, no. 1, p. 26, 2023.
29. H. Jia, Z. Xiao, and P. Ji, “End-to-end fatigue driving EEG signal detection model based on improved temporal-graph convolution network,” *Computers in Biology and Medicine*, vol. 152, no. 1, p. 106431, 2023.
30. Y. Zhou, C. Zeng, and Z. Mu, “Optimal feature-algorithm combination research for EEG fatigue driving detection based on functional brain network,” *IET Biometrics*, vol. 12, no. 2, pp. 65-76, 2023.
31. D. Croft and M. England, “Computing with CodeRunner at Coventry University: Automated summative assessment of Python and C++ code,” *in Proc. 4th Conf. on Computing Education Practice*, New York, United States of America, 2020.
32. B. Bakker, B. Zablocki, A. Baker, V. Riethmeister, B. Marx, and G. Lyer, “A multi-stage, multi-feature machine learning approach to detect driver sleepiness in naturalistic road driving conditions,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 4791–4800, 2022.
33. T. Åkerstedt, A. Anund, J. Axelsson, and G. Kecklund, “Subjective sleepiness is a sensitive indicator of insufficient sleep and impaired waking function,” *Journal of Sleep Research*, vol. 23, no. 3, pp. 240–252, 2014.
34. Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, “Robust lane detection from continuous driving scenes using deep neural networks,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 41–54, 2019.
35. TuSimple, “TuSimple Lane Detection Challenge for CVPR2017,” *Online Dataset Report*, 2017. Available: https://github.com/TuSimple/tusimple-benchmark [Accessed by 12/09/2024].
36. T. R. Farshi, “Battle royale optimization algorithm,” *Neural Computing and Applications*, vol. 33, no. 4, pp. 1139–1157, 2021.

37. S. Akan and T. Akan, “Battle royale optimizer with a new movement strategy,” in Handbook of Nature-Inspired Optimization Algorithms: The State of the Art, Volume I: Solving Single Objective Bound-Constrained Real-Parameter Numerical Optimization Problems, *Springer International Publishing*, Cham, Switzerland, 2022.
38. K. K. Azar, A. Kakouee, M. Mollajafari, A. Majdi, N. Ghadimi, and M. Ghadamyari, “Developed design of battle royale optimizer for the optimum identification of solid oxide fuel cell,” *Sustainability*, vol. 14, no. 16, p. 9882, 2022.
39. X. Meng, Y. Liu, X. Gao, and H. Zhang, “A new bio-inspired algorithm: chicken swarm optimization,” *in Proc. Int. Conf. on Swarm Intelligence (ICSI)*, Hefei, China, 2014.
40. S. Deb, X. Z. Gao, K. Tammi, K. Kalita, and P. Mahanta, “Recent studies on chicken swarm optimization algorithm: a review (2014–2018),” *Artificial Intelligence Review*, vol. 53, no. 5, pp. 1737–1765, 2020.
41. W. Osamy, A. A. El-Sawy, and A. Salim, “CSOCA: Chicken swarm optimization based clustering algorithm for wireless sensor networks,” *IEEE Access*, vol. 8, no. 3, pp. 60676–60883, 2020.
42. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” *in Proc. 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, San Francisco, California, United States of America, 2016.
43. Y. Qiu, J. Zhou, M. Khandelwal, H. Yang, P. Yang, and C. Li, “Performance evaluation of hybrid WOA-XGBoost, GWO-XGBoost and BO-XGBoost models to predict blast-induced ground vibration,” *Engineering with Computers*, vol. 38, no. 4, pp. 4145–4162, 2021.
44. S. S. Dhaliwal, A. A. Nahid, and R. Abbas, “Effective intrusion detection system using XGBoost,” *Information*, vol. 9, no. 7, p. 149, 2018.
45. S. Putatunda and K. Rama, “A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of XGBoost,” *in Proc. 2018 Int. Conf. on Signal Processing and Machine Learning*, Shanghai, China, 2018.
46. Y. Wang and X. S. Ni, “An XGBoost risk model via feature selection and Bayesian hyper-parameter optimization,” *arXiv preprint, arXiv:1901.08433*, 2019. Available: https://arxiv.org/abs/1901.08433 [Accessed by 09/09/2024].
47. S. Janizadeh, M. Vafakhah, Z. Kapelan, and N. M. Dinan, “Hybrid XGBoost model with various Bayesian hyperparameter optimization algorithms for flood hazard susceptibility modeling,” *Geocarto International*, vol. 37, no. 25, pp. 8273-8292, 2021.
48. M. Zivkovic, L. Jovanovic, M. Ivanovic, N. Bacanin, I. Strumberger, and P. M. Joseph, “XGBoost hyperparameters tuning by fitness-dependent optimizer for network intrusion detection,” *in Communication and Intelligent Systems: Proceedings of ICCIS 2021*, Greater Noida, India, 2022.