Expert Systems With Applications 192 (2022) 116389

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

An efficient automatic accident detection system using inertial measurement through machine learning techniques for powered two wheelers

A. Jackulin Mahariba [a], Annie Uthra R. [a], Golda Brunet Rajan [b,*]

[a] Department of Computational Intelligence, School of Computing, SRM Institute of Science and Technology, Kattankulathur, Tamil Nadu, India
[b] Department of CSE, Government College of Engineering, Salem, Tamil Nadu, India

ARTICLE INFO

Keywords:
Accident detection
Accident severity
Fall identification
Two-wheeler accident

ABSTRACT

The two-wheeler accidents in most populated and developing countries have become vulnerable and six accidents happen every hour on average. This paper proposes an efficient automatic accident detection system that attempts to detect the occurrences of the accidents in powered two-wheelers (PTW) automatically using vehicle-dependent parameters and the physiological parameters of the rider in real-time. The proposed system builds an accident detection system in PTW using three steps namely, critical event detection system, accident detection system, and severity assessment system. The critical event detection system reads the accelerometer sensor values from the On-Board Diagnostic (OBD) unit mounted on the PTW and classifies the state of the vehicle as normal, fall-like, and fall through the enhanced decision tree algorithm. The enhanced decision tree algorithm uses a tanh function to calculate entropy values. The rules are extracted to fix the threshold by pruning the decision tree to identify the fall of the vehicle and the rider. Due to the unstable nature of PTW and the rider, a novel Adaptive Sequence Window algorithm (ASW) is proposed to substantiate and validate the occurrence of accidents based on the sequence of states identified. Once the accident is detected, the Decision Support System (DSS) running on the OBD mounted on the PTW decides the severity of the accident by combining the three parameters namely fall of the vehicle, fall of the rider, and pulse rate of the rider using the first-order predicate logic rules. The enhanced decision tree algorithm outperforms the other classifiers such as naïve Bayes, artificial neural network, and recurrent neural network with an accuracy of 99.8%. The OBD unit mounted on the PTW and the rider's helmet is used to detect the occurrence of accidents automatically along with its severity with less time. The ASW algorithm enables the system to detect the fall of the vehicle and rider within five minutes and prevents false positives. Further, the information can be communicated based on the severity of the accident to the emergency medical service for quick response.

1. Introduction

Mopeds, scooters, and motorcycles are collectively called the term Powered Two Wheelers (PTW) which comes out in different forms and designs throughout the world. The considerable and affordable cost of PTW, not only plays a dominant role in commuting and tourism but also results in a tremendous increase in the number of PTW users. There is almost double the sales of PTW in 2010 when compared to 2020 sales worldwide. Among all the road users PTW users are most vulnerable to fatal accidents because of the unstable nature of its design. The rising pattern of severe road traffic accidents in developing countries has created a significant need to understand the key factors involved in such accidents and the major injuries and fatalities.

In the absence of public transportation with a high density of population, two-wheelers are becoming more popular, and the proportion of two-wheeler accidents is increasing. The most vulnerable population group of fatal accidents on PTW in countries with more working population group has the age range from 30 to 59 years. Peak working hours and extreme weather conditions are favorable conditions for accidents. The accidents caused by the fault in the motorcycle parts are very negligible, they accounted for 1.8% when compared to the other factors. The fault of the driver is the main risk factor which accounts for 78% of

Abbreviations: PTW, Powered two Wheeler; OBD, On-Board Diagnostic Unit.

* Corresponding author.
E-mail address: goldabrunet@gcesalem.edu.in (G.B. Rajan).

https://doi.org/10.1016/j.eswa.2021.116389
Received 5 April 2021; Received in revised form 1 November 2021; Accepted 7 December 2021
Available online 18 December 2021

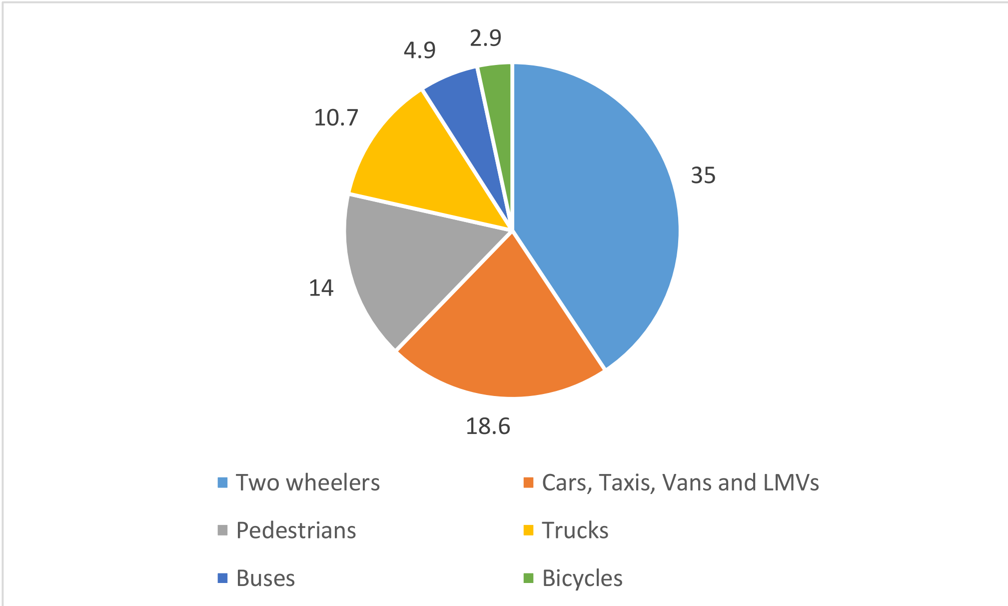

Fig. 1. Distribution of fatality rate among road users in accidents.

causes of road accidents. The inexperienced drivers are high-risk road users. The growth of new drivers in the driving population is accompanied by inadequate training and testing facilities, and thus increases the high risk of road accidents. Improved road conditions, vehicle standards, enforcement of strict traffic rules and regulations, speed control may reduce the fatalities in road accidents.

According to a report released in October by the Ministry of Road Transport and Highways of India (centre, 2019), more than a third (35%) of those who died in road accidents in 2019 were two-wheeler riders, which is depicted in Fig. 1.

The increased fatality rate in motorcycle accidents is due to the highly unstable nature of motorcycles. The delay in accident notification and emergency medical care is the prevalent reason behind two-wheeler accident fatalities. Hence, a high-accuracy, low-cost, light-weight, and low-maintenance automatic accident detection system for PTW is required. It motivates the researchers to design an automatic accident detection system for two-wheelers. Accident detection system for two-wheelers has the following pitfalls to be addressed to make the system more reliable, robust, and accurate. First is that existing accident detection systems use the accelerometer, gyroscope values, and report the fall by comparing the norm of the measured value with the threshold. But the proof of concept behind fixing the threshold value is through vehicle dynamics only like sudden deceleration and change in lean angle. The vehicle dynamics vary according to the nature of each accident. Secondly, close and continuous monitoring of the vehicle dynamics after the fall detection is necessary. Existing accident detection systems detects the occurrence of accident through fixed waiting time. But the time duration required to do close monitoring varies from case to case.

If the occurrence of an accident is detected automatically and reported to the emergency management team, then there will be a huge drop in the death rate in PTW accidents due to the reduction in response time. To address this concern the advancement in technologies and communication, the Intelligent Transportation System (ITS) evolves in the modern era.

There is a need to design a system to identify the occurrence of accidents and emergency notification due to the escalating prevalence of road accidents and victim fatalities, particularly in two-wheeler accidents. But most of the existing designs support only four-wheelers with the built-in On-Board unit for accident detection by embedding sensors and linking it to the car's Engine Control Unit (ECU). As a result, the vehicle's cost is also increased. Such an accident detection system is not feasible for a two-wheeler due to design constraints that heavy equipment or devices cannot be mounted on the PTW's body. Hence, a high-accuracy, low-cost, light-weight, faster, and low-maintenance automatic accident detection system for PTW is required. The proposed system for accident detection meets these requirements and serves the purpose efficiently. The proposed system uses a real-time unsupervised dataset to build an automatic accident detection technique. The dataset is labeled through pre-processing step, the k-means clustering technique. The best classification algorithm is identified after experimenting with various classifiers on the dataset to classify the state of the vehicle and rider. A novel Adaptive Sequence Window algorithm is proposed to ensure the occurrence of the fallen state. Finally, the severity of the accident is detected using predicate logic.

The novelty of the proposed work is introduced in the classification technique which identifies the state of the vehicle and rider. The decision tree classifier is modified with the tanh function for entropy calculation to improve the performance of the classifier and convergence time. Accident detection in two-wheelers is uncommon due to the uncertain and highly unstable nature of PTW design which is well addressed in this paper. A novel accident detection algorithm is devised using an adaptive sequence windowing technique that could discriminate the real fall state from the actions like skidding, tipping, simple hit, and sliding resulting in avoiding false-positive cases. Adaptive window size is derived based on the number of transitions across the states during the journey.

The rest of the paper is presented as follows: Section 2 provides the insights observed from various research findings in accident detection and fall detection based on inertial measurement values using different machine learning algorithms. Also, it summarizes the different indices followed to measure the crash/accident severity. Section 3 explains the proposed automatic accident detection system which is achieved through intelligent critical event detection, adaptive sequence window algorithm, and accident severity categorization. Section 4 provides the results and discussion. Finally, section 5 provides the conclusion of the proposed work.

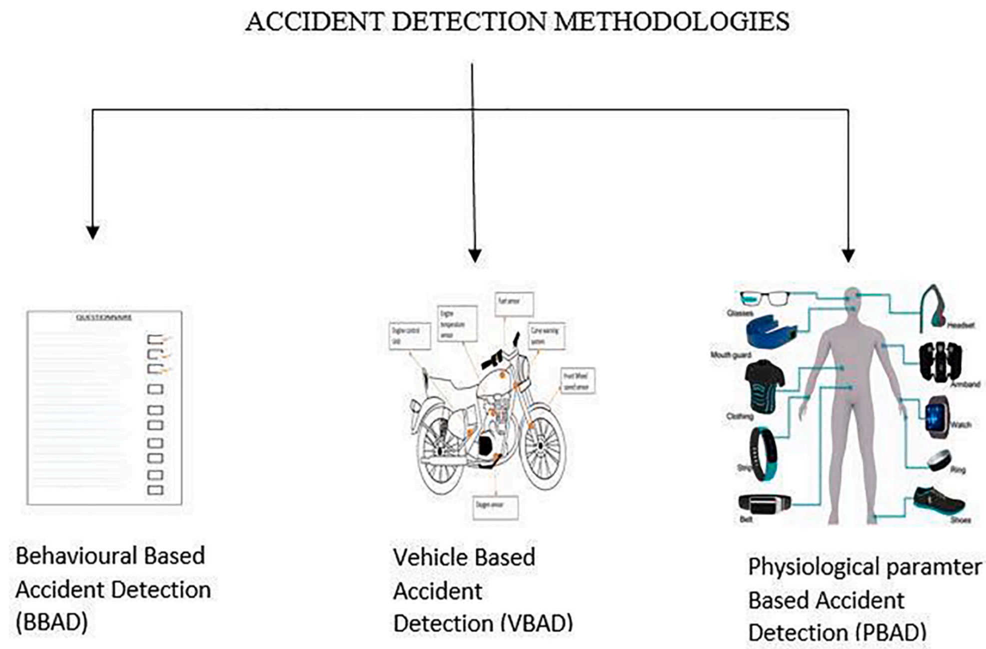

Fig. 2. Types of accident detection methodologies.

2. Literature survey

A descriptive epidemiological study was conducted on past accident records to understand the issues and challenges behind the occurrence of road accidents. An automatic accident detection system is defined as a subdomain under the autonomous vehicle design, where the human intervention in all aspects of driving and managing on the road is merely less or it is controlled completely by the special OBD fitted to the engine of the vehicle. Such an autonomous driving mechanism is not feasible in the case of PTW, hence special investigation and detailed study are required to detect the accident in PTW. A road traffic accident can be classified based on the type of vehicles and the reason involved in it such as pedestrians, pedal cyclists, motorcyclists, drivers of cars, commercial and passenger vehicles, animal-vehicle collisions, mass casualty incidents, and Act of God.

2.1. Automatic accident detection system

An automatic accident detection system specially designed for PTW is required to reduce the number of death cases in fatal accidents. A complete study on the PTW user behavior is required to build an accident detection system. Also, the model should be designed after careful analysis of all the reasons behind road accidents, especially PTW crashes and accidents. In general, accident detection can be carried through three main parametric ways such as behavioral-based accident detection (BBAD), vehicular parameter-based accident detection (VBAD), and physiological parameter-based accident detection (PBAD) as shown in Fig. 2.

The behavioral-based accident detection system involves post-accident analysis. Different types of the questionnaire were framed to understand the importance of different riding behaviors and the role of human factors in fatal road accidents and the results are analyzed to exhibit driving violation and accident-free counterparts in dangerous situations (Elliott, Armitage, & Baughan, 2007; Stephens, et al., 2017; Sumer, Ozkan, & Lajunen, 2006). These studies will show a strong causal and effect relationship between the parameters involved in road accidents. The main drawback of these kinds of the study was the type of participants involved in the survey, the quality of the questions framed, and the sampling method used during the analysis of the survey results.

To have a better understanding of the reasons behind road crashes in terms of environmental and vehicle-dependent parameters, an Accimap was generated through Rasmussen's risk management framework (Sharon & Natassia, 2015). It provides a guide for the vehicle manufacturing industry for the development of better systems that can overcome the common risk-causing factors in the vehicle. There are many accident detection techniques identified through research literature. They are accident detection systems using smartphones, GPS and GSM communication technologies, simple mobile applications, vehicular ad hoc networks, and so on.

In most cases of accidents, the primary cause is over speeding. High-speed driving increases the risk factor and severity of the accident. The kinetic energy on the moving vehicle is transformed into a destructive force during the accident. The sudden unexpected deceleration is used to identify the occurrence of an accident. The deceleration value is calculated by using the initial velocity of the vehicle. If the estimated deceleration value is less than the predicted value, it is detected as an accident. Then the location in the GPMRC message will be communicated to the emergency center through GSM communication. An OBD unit is designed (Amin, Jali, & Reaz, 2012) to serve the purpose of detecting accidents and communicating them to the rescue service.

Yuki Kobayashi and Takumi Makabe of Honda R&D CO., Ltd of Japan released a white paper for crash detection of PTW airbag system. The airbag is triggered to be inflated when deceleration occurs due to steered collision (Kobayashi & Takumi, 2013). The threshold value is described for crash detection using longitudinal displacement of the front suspension of PTW. This system can be applied to all types of PTW. Baramy et. al (Baramy, Singh, Jadhav, Javir, & Tarleka, 2016) proposed a methodology to automatically detect the accident using sensors fitted on the body of the vehicle. A shock sensor is mounted on all sides of the vehicle to detect the impact force experienced on it. The output of these sensors is connected to an OR gate, such that the hit or crash on any side of the vehicle can also be considered. This phenomenon is very similar to the triggering of an airbag sensor in four-wheelers. An ultrasonic sensor is fitted on the roof of a four-wheeler. An ultrasonic sensor reads the distance between the vehicle with the other vehicles in the front and rear sides. The accident occurs, when the other vehicles are closer than the minimum threshold distance (Khalil, Javid, & Nasir, 2017). But this methodology may give a high false rate because in the real-time driving scenario the vehicles may intervene or overtake at different angles of position concerning the vehicle. It can't be applicable for two-wheelers due to the lack of space to mount the sensor on PTW. Hence a special design is required for PTW.

The drunken drive is also a major cause of the accident, alcohol sensor (Priyanka, Darshini, Shavi, & Begum, 2018) is used to detect the content of alcoholic odor in the breath of the rider. The engine will remain OFF until this odor is below the permitted threshold which is set to near zero. An experimental study is made by Andy et al (Cheng, Ng, & Lee, 2013) to understand the correlation between the violation of driving rules and visual attention of drivers on an accident-involved motorcycle ride. This study proves that the rider who violates traffic rules had a high hazard perception and thus leads to the involvement of accident. Ali et.al (Ali & Eid, 2015) has proposed an automated system for accident detection for four-wheelers, it monitors the parameters such as accelerometer, gyroscope, impact force, and speed. The readings are coded as low, medium–high and their combination is analyzed using fuzzy logic to find the collision index. If the resultant collision index is high then it is reported as an accident. An alert system for Intelligent Transportation System (ITS) was proposed by (Fernandes, Alam, Gomes, Ferreira, & Oliveira, 2016) using eCall service to European Union to provide rapid assistance to the victims.

A wearable device with an accelerometer is designed for remote health monitoring systems (Gibson, Amira, Ramzan, Casaseca-de-la-Higuera, & Pervez, 2016; Amelie, 2018) in which a multiple classifiers system with comparator function is used for fall detection. This system uses algorithms such as ANN, KNN, RBF-based ANN, PPCA based decision tree, and LDA. The fall is classified as a strong and soft fall as per the severity of the injury, also the fall direction is identified using the accelerometer wavelet signals. Denis Atalar et al classified PTW road crashes into five categories such as overtaking, loss of control, rear-end collision, turning, and others using latent cluster analysis (Atalar & Thomas, 2019). This cluster analysis works well with a large number of categorical features in the dataset when compared to other machine learning algorithms.

A simple low complexity threshold-based fall detection algorithm was proposed (Boubezoul, Espié, Larnaudie, & Bouaziz, 2013) for PTW. This algorithm uses a three-axis accelerometer and gyroscope sensor readings as input to identify the fall or ejection of a rider which in turn triggers the inflation of the safety jacket worn by the rider to prevent fatal injury during an accident. An automatic smart accident detection system was proposed using Fuzzy logic (Ali & Eid, 2015). The VBAD system describes the modeling of vehicle-dependent parameters such as accelerometer, gyroscope, GPS, tachometer, and shock-sensor to identify the occurrence of an accident. Due to the socio-economic cost factor of PTW, the implementation of such OBD in PTW is not feasible. The manufacturing design of PTW involves most cost-cutting operations due to its usage in terms of mobility with low power.

The PBAD includes the measurement of pulse rate, ECG, and drowsiness of the rider. The wearable sensors are used to design the PBAD system (Vivien Melcher et al, 2015). The main drawback of this system is maintenance and its robustness. Wearing such a safety system and handling it safely is an additional overhead for PTW users. Many variables can lead to the likelihood of a rider accident, particular rider features and rider execution have also been identified to be major components.

The related works show that many studies have ignored the fall-like condition in the fall detection system using accelerometer values. On analyzing the related work carried out in the field of accident detection can be broadly categorized into two types.

1. Accident detection system using inertial systems in OBD mounted on PTW
2. Accident detection through machine learning algorithms with earlier crash data

2.2. Crash severity

A simple accident detection technique either using an inertial method or learning algorithm is not reliable for any Decision support system (DSS) to initiate the emergency management for victims through rescue service to prevent the loss of life. The crash severity plays a significant role in designing an efficient DSS for PTW accidents. Salvatore Cafiso et al analyzed the ISTAT database, which contains accident records of PTW in Italy. From the dataset only 8 parameters are considered, they are road type, time of the day, road geometry, age of driver, partner collision, type of the vehicle (PTW), circumstances, and collision type (Cafiso, Cava, & Pappalardo, 2012). Linear regression, mixed logit model, neural network are used to assess the crash severity (Li & Bai, 2008; Moore, Schneider, Savolainen, & Farzaneh, 2011; Jeong, Jang, Bowman, & Masoud, 2018; Tanga, Lianga, Hana, Lib, & Huanga, 2019) in a road accident. Real-time accident detection systems like wreckwatch (White, 2011), AADS, Belted Safety jacket (Grassi, Barbani, Baldanzini, Barbieri, & Pierini, 2018), and e-Box (Gelmini, Strada, Tanelli, Saravesi, & Tommasi, 2019; SaveDrives, 2016) monitor the physiological parameters such as pulse rate and ECG to determine the crash severity.

The main research gap identified in existing accident detection systems are listed below,

1. In existing accident detection systems, especially in systems that detect accidents using an accelerometer, a threshold value is set to detect the occurrence of an accident. Fixing a hard threshold value using only accelerometer values and categorizing events as fall or non-fall will result in high false positives. The fall-like case is ignored while detecting the fall using a threshold. Hence, depending on the threshold value set for fall detection, a simple lean over left, right turn, and curve can also be classified as a fall.
2. In existing accident detection techniques, personalization of the system concerning the rider concerning their riding style and physiological vitals such as pulse rate is not addressed.

The importance of designing an automatic accident detection system will help PTW road users from losing their life in any type of crash/ road accident. The proposed design of the accident detection system for PTW includes both VBAD and PBAD. Once the occurrence of an accident is detected, it can be easily communicated through many telecommunication services available in ITS, which helps in the speedy recovery of the road users affected by the accidents. Hence the objective of this paper is to provide an efficient automatic accident detection system for PTW using VBAD and PBAD techniques. The parameters are measured and modeled using machine learning algorithms to identify the

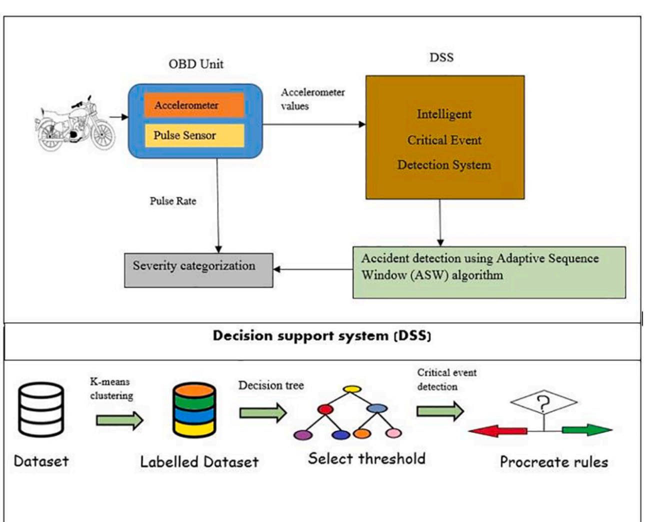

Fig. 3. Automatic accident detection system.

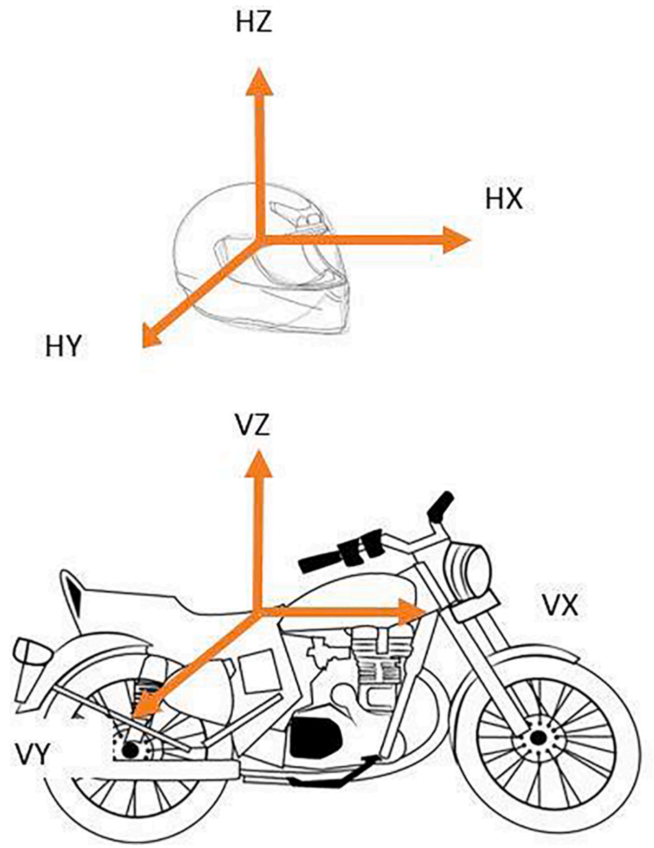

Fig. 4. X, Y, Z axes of accelerometers mounted on the vehicle and the helmet.

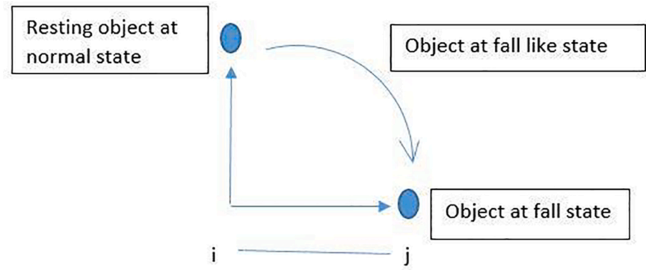

Fig. 5. Illustration of falling object mechanics with the three states.

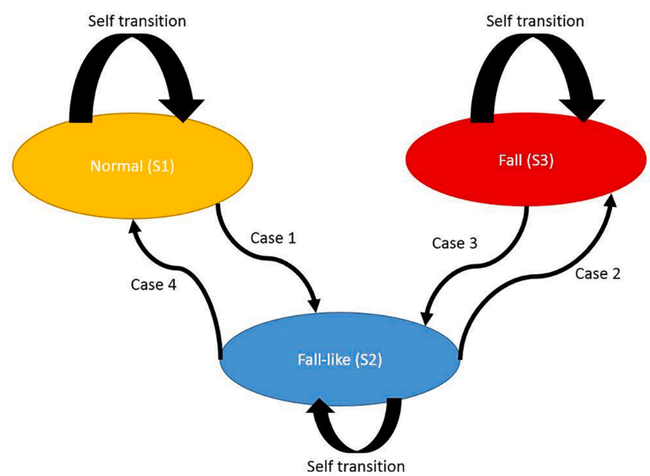

Fig. 6. Sequence of states.

occurrence of falls and their severity class to detect an accident.

In the proposed work, a decision support system is designed using predicate logic rules, which accepts the pulse rate of the rider along with the states of the vehicle and rider as input. It categorizes the severity of accidents as low and high after the occurrence of an accident. Models of high computational cost and storage cannot be imparted for accident detection in PTW due to the cost and design constraints of the vehicle. Hence there is a need for an exclusive cost-effective and efficient automatic accident detection system for PTW.

3. Automatic accident detection system

The proposed accident detection system in PTW is carried out in three steps namely, critical event detection system, accident detection system, and severity assessment system. The integration of these phases is shown in Fig. 3. The proposed system operates on the data collected from only two sensors viz, the accelerometer and the pulse sensor so that the system is economically feasible for all riders in practice. The purpose of a critical event detection system is to identify the fall of the vehicle or the rider through the accelerometer readings. The accident detection system determines whether the fall identified by the critical event detection system is indeed an accident using an adaptive sequence window. In the proposed work, the critical event, namely, a fall, happens and persists until the buffer window is filled with a fall state is referred to as an accident. Once the accident detection system confirms an occurrence of an accident, the severity of the accident is assessed for further action.

3.1. Critical event detection system

The state of a two-wheeler riding at a particular instance of time can be classified into three categories namely Normal (NR), Fall-Like (FL), and Fall. To assess the state of the ride, the On-Board Diagnostic (OBD) unit deploys two accelerometers – one on the vehicle (V) and the other on the rider's helmet (H). Each accelerometer reports X, Y, and Z values which represent the orientation of the accelerometer. The orientation of the accelerometers can be interpreted as the orientation of the vehicle and the rider as they are fitted on top of them. Fig. 4 shows the accelerometer axes on the vehicle and the rider concerning the normal state. As the accelerometers are mounted on the vehicle and helmet separately, their tri-axial values are represented as VX, VY, VZ, HX, HY, and HZ respectively.

The normal state (S1), represents the position in which the vehicle and the rider are in an upright position. Fall-like state (S2), defines the position in which either the vehicle or rider's orientation is tilted less than or equal to $\pm 45^{\circ}$ with respect to the normal state on the Z-axis, which acts in the opposite direction of gravitational force (G). Fall state (S3), characterizes the position in which either the vehicle or rider's orientation is changed more than $\pm 90^{\circ}$ on Z-axis with respect to the normal state. Typically in a two-wheeler ride, the initial state is S1, the state transition from normal to fall follow a strict sequencing order such as, normal to fall-like and fall-like to fall which is proved using the following theorem.

Theorem: The state transition of the vehicle and rider follows a strict sequencing order normal – fall-like – Fall when the rider meets an accident while driving.

Proof. Since the rider's orientation is opposite the gravitational force G, the initial state of a ride is called normal, which is stable and steady-state. Newton's law of motion states that any external force acting on an entity causes it to move or be displaced. When a rider is involved in a collision, an external force acts on him, which may be G causing the rider to tilt the vehicle more than $45^{\circ}$from normal or G-force. As a result, the rider's state shifts from normal to fall, where the rider makes contact with the ground. The rider is pushed towards the ground by a G force,

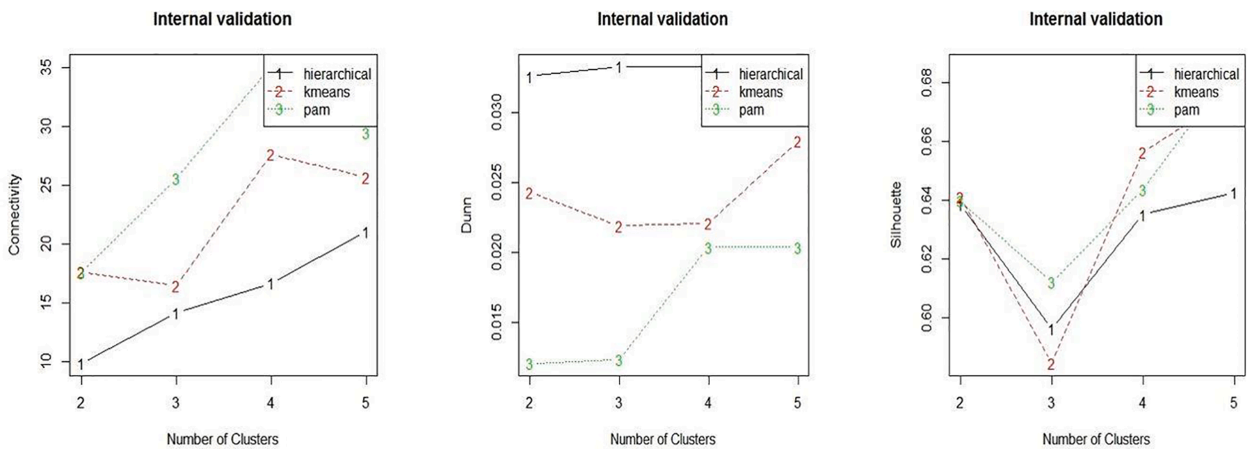

Fig. 7. Cluster internal validation results of hierarchical, K means, and PAM.

which acts as momentum control and produces displacement in the direction of the earth's surface, resulting in a fall-like state. When the rider encounters an accident while driving, the state change of the vehicle and rider follows a strict sequencing order of normal to fall-like to fall.

Fig. 4 illustrates the state transition of vehicle/rider at time *'i'* when an external force is applied on the vehicle/rider at a normal state. Vehicle/rider reaches fall state at time *'j'* with displacement.

As seen in Fig. 5, the vehicle/rider moves in a fall-like state from i + 1 to j-1. As a result, every falling entity (vehicle) follows a strict sequence of normal, fall-like, and fall state conditions. Fig. 6 represents the sequential order of three states. The onset of a fall is known as a critical event.

The accelerometer constantly senses the X, Y, and Z directions of the vehicle and rider at a sampling rate of 45 Hz. By specifying a threshold, the critical event is defined based on the observed values from the accelerometer mounted on the PTW. Machine learning algorithms are used to assess the threshold. To make it consistent throughout the paper, the tri-axial accelerometer values are called AX, AY, and AZ.

3.1.1. Data pre-processing

The proposed system uses the dataset released by Abderrahamane boubezoul *et. al.* in 2019, where the observations are made on the realistic falls in different scenarios in the controlled environment. The continuous sensor values recorded on each ride are combined to form a dataset with all three states namely, normal, fall-like, and fall. The tri-axial accelerometer values from the dataset are used to determine the state of the vehicle. In an unlabeled raw dataset, state identification is hard. As a result, machine learning methods are used to cluster the data points initially. The data points with similar characteristics are expected to club together to form a cluster. The number of clusters is pre-determined from the domain knowledge and a cluster centroid is randomly chosen for each other. The data points are associated with any one of the clusters based on similar measures. Distance and density are the two main measures available to estimate the centroid in clustering techniques. In density-based clustering, the data distribution is used to identify the cluster head, whereas, in distance-based clustering, distance measures are used to estimate object similarity. The accelerometer measurements in the dataset under experimentation have a skewed structure, inclining closer toward normal riding values. Because the critical event occurs sporadically during a trip and lasts only for few seconds. Thus, the use of density-based clustering is ruled out in our scenario. Therefore distance-based clustering techniques such as Hierarchical clustering, K-means, and Partition Around Medoids (PAM) are used to cluster the accelerometer values in the dataset with tri-axial accelerometer values.

For all the three clustering techniques, Euclidean distance is utilized as the common measure. The clustering techniques and their algorithm are described in detail.

Hierarchical clustering:

The data points ($d_1, d_2, \ldots, d_N$) are grouped by building a hierarchy of clusters(C). This technique does not require the number of clusters to be known in advance. The bottom-up approach of hierarchical clustering considers each data point as a singleton cluster initially, then aggregates the nearest cluster and pairs up until all clusters are merged to be in a single cluster.

The hierarchical clustering algorithm is presented below,

```
for i = 1 to N: // size of the dataset is N
    for j = 1 to i:
        dis_mat[i][j] = eucdist(di, dj) // distance calculated for lower triangular matrix,
    since the distance matrix is symmetric.
each data point is assigned as a singleton cluster (Ci)
repeat
    merge the two cluster having minimum distance
    update dis_mat[i][j]
until only a single cluster remains
```

K-means clustering:

Each data point (X) is categorized by calculating the distance between it and the centroid, and then assigning it in the cluster(c) with the centroid (C) nearest to it.

The K-means clustering algorithm is presented below,

Repeat
For each × in data
Dist = []; C = []; // initialize the centroid and distance measured
For i to n:
Dist[i] = eucdist(X, centroids[i]) // distance calculated from each centroid
$C[i] = \arg\min_{i=1}^{n} eucdist(c_i, X)$ *// nearest centroid to the data point is extracted*
Assign X to the nearest centroid from C[i]
For each i in C[i]:
$c_i = \frac{1}{S_i}\sum_{X_i \in S_i} X_i$ *// compute the new centroid; S_i set of all points assigned to cluster i*
until *no change in* c_i

PAM clustering: This clustering technique intends to reduce the average dissimilarity between data points(X) to their centroid. PAM clustering represents centroid as medoids(M), the center point of the cluster. The dissimilarity(E) is measured using the following equation,

$$E = |M_i - X_i|$$

The cost involved in finding K-medoids is given as follows,

$$Cost = \sum_{M_i}\sum_{X_i \in M_i}|X_i - M_i|$$

The PAM clustering algorithm is presented below,

Set k out of n data points as medoids (M[1…k])
Assign each data point to the nearest M[i] // find eucdist(X_i, M_i)

(*continued on next page*)

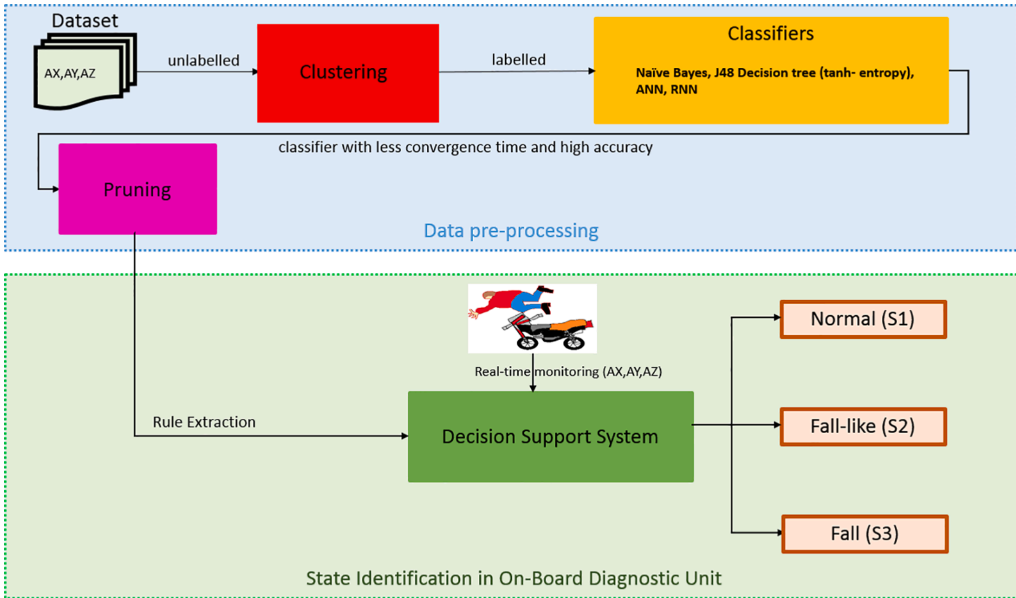

Fig. 8. Framework to classify states using machine learning algorithms.

(*continued*)

Compute Cost
while(Cost decreases)
For each M[i], for each X[i] $\notin$ M[k]
Swap X[i] and M[i], assign each X[i] to the nearest M[i] and compute Cost
If ($Cost_i > Cost_{i-1}$)
Undo Swap X[i] and M[i]

Cluster validation metrics aid in the selection of the most appropriate clustering algorithm. The criteria utilized to select the best clustering technique are connectivity, Dunn index, and silhouette width. The better clustering occurs when the connectivity is low, with a maximum on the Dunn index and silhouette width. In all the three measures comparatively, the K-means clustering technique yields good results, also the number of clusters is to be fixed as three 3 (Normal, Fall-like, and Fall) as aforementioned. The performance of the clustering algorithm is compared with cluster internal validation as shown in Fig. 7. On observing the cluster internal validation result, K-means clustering with K = 3 (Normal, Fall-like, and Fall) is chosen and applied on the dataset with the features AX, AY, and AZ to label the data set.

3.1.2. State identification through classification

A framework is developed to characterize the states of the accelerometer values in the dataset as depicted in Fig. 8.

After performing the clustering, the dataset is labeled with three states namely fall, fall-like and normal. The classification algorithms such as Naïve Bayes, Decision tree, Artificial Neural Network (ANN), and Recurrent Neural Network (RNN) are used on the labeled dataset to extract the rules and threshold values. The proposed system introduced a decision tree algorithm with tanh entropy calculation and designed ANN and RNN architectures to find out the best suitable model to detect the critical event. Based on the results obtained the proposed system used a rule-based algorithm is used to detect the critical event.

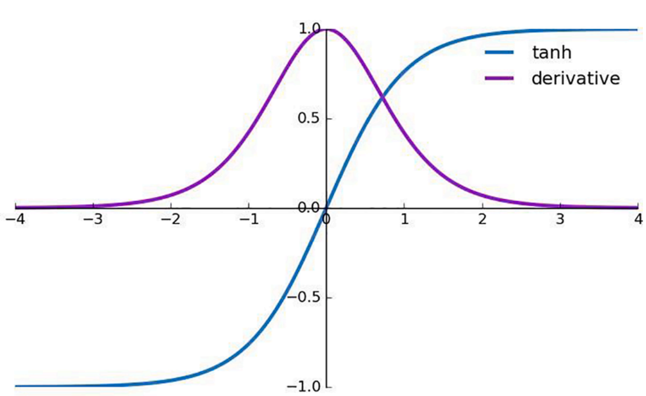

Fig. 9. Characteristic curve of tanh function and its derivative.

3.1.2.1. ID3 Decision Tree Algorithm. The decision tree algorithm is a data classification algorithm that uses a series of predefined Boolean questions to categorize data. The training patterns are used to create these predefined problems. Depending on the answer the data provides to the query, each question splits the input data into two categories. This procedure is repeated for different questions on each of the separated data which leads to the generation of a tree. Each question on a node thus tries to separate data belonging to different classes and the amount of inter-class data on a node denotes the purity of that node which is measured by the entropy of the node. Lower the entropy higher the purity of the node. In the ID3 decision tree algorithm, the entropy is calculated for the entire dataset using equation (1)

$$\rho_D = -\sum p_i log p_i \tag{1}$$

The entropy of each feature is calculated to measure the disorder accounted for each feature 'f' using equation (2) such that 'f' splits the dataset into 'n' number of subsets {D_1, D_2,…., D_n}.

$$\rho_f = -\sum_{i=1}^{n} \frac{D_i}{D} \rho_{Di} \tag{2}$$

Information gain is defined as the difference between the entropy measures of the root node and the feature.

$$I.G = \rho_D - \rho_f \tag{3}$$

The query at the root node is an attribute with low entropy but high information gain. It is also subdivided into subsets based on its entropy and information gain in comparison to other attributes. When the feature is continuously valued, the decision tree often takes more convergence time because the brute force on continuous values to come up with the correct split value is a time-consuming process. In this research, the continuous-valued accelerometer readings have to be

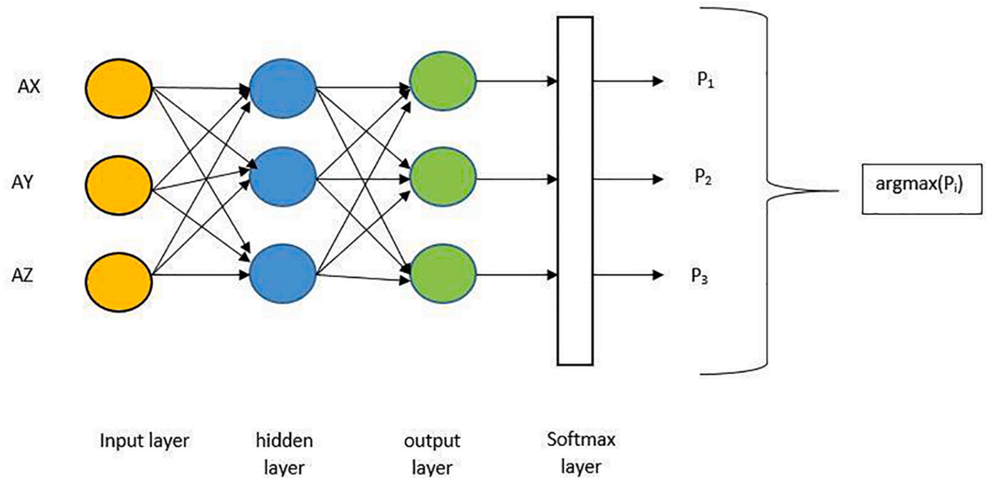

Fig. 10. ANN Architecture diagram.

classified into normal, fall-like, or fall using a decision tree. As the number of unique values for accelerometer readings is too high, the decision tree is tall with a height of at least n, where 'n' is the number of unique accelerometer readings.

Calculation of Entropy using tanh function

To improve the performance of the decision tree classifier and convergence time, the tanh function is introduced in the entropy calculation which performs well, when compared to simple entropy calculation with information gain in the ID3 algorithm. The range of the tanh function is from (-1 to 1). The advantage of using tanh is that the negative inputs will be mapped strongly negative and the zero inputs will be mapped near zero in the tanh graph and the function is differentiable. Also, the tanh function is monotonic while its derivative is not monotonic so that it converges easier with low computation cost and results in an accurate classifier. The monotonicity of the tanh function is shown in Fig. 9.

The entropy calculation of the ID3 decision tree algorithm is modified using the tanh function to calculate the new Entropy (ρ_2). In the modified decision tree algorithm, the entropy is calculated for the entire dataset using equation (4)

$$\delta_D = tanh\left(\sum p_i log p_i\right) \tag{4}$$

Entropy of each feature is calculated using equation (5)

$$\delta_f = tanh\left(\sum_{i=1}^{n} \frac{D_i}{D}\delta_{D_i}\right) \tag{5}$$

And the information gain is given as,

$$I.G = |\delta_D - \delta_f| \tag{6}$$

3.1.2.2. ANN Classifier. ANN classifier identifies the similarity in the samples of known classes and learns the weights between layers of

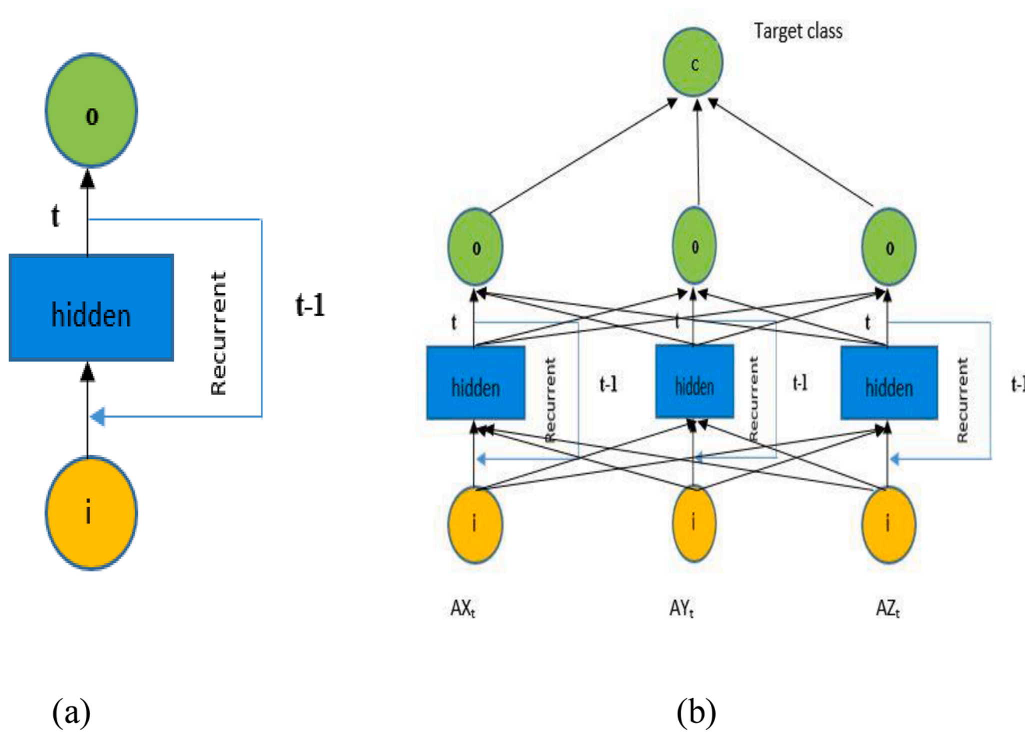

Fig. 11. (a) General structure of RNN (b) RNN Architecture diagram to classify the states.

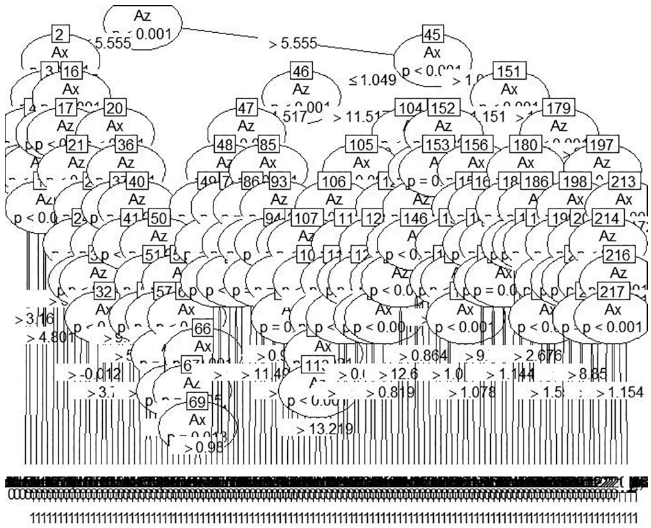

Fig. 12. The tree generated using Decision tree algorithm for classification of states using accelerometer values.

neurons to separate the samples into different classes. The proposed system constructed a simple neural network with one input layer, one hidden layer, and an output layer as shown in Fig. 10. Each layer has three neurons. The input values AX, AY, and AZ are fed through the three neurons in the input layer. The learning rate used in this classifier is 0.1 and the tanh function is used as the activation function. The output class is obtained in the form of one hot encoding from the three neurons in the output layer.

In ANN classifier the output vector is computed using the following equations,

$$O_t = \tanh(Wh_t + b_t) \tag{7}$$

The backpropagation algorithm is used to update the weights at each iteration. The gradient value is calculated using the following equation,

$$\frac{\partial L}{\partial W} = \frac{\partial L_t}{\partial O_t} \sum_{k=0}^{t} \left(\prod_{i=k+1}^{t} \frac{\partial h_i}{\partial h_{i-1}} \right) \frac{\partial h_k}{\partial W} \tag{8}$$

Where W represents Weights, b represents bias, h represents the output of the hidden states and L refers to the loss function. The mean squared error function is used as the loss function to learn the parameters during backpropagation. The tanh value is calculated using the following equation,

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9}$$

3.1.2.3. RNN classifier. RNN classifier is similar to the ANN except for the recurrence in the hidden layer. The recurrent connection enables the network to influence the current prediction (t) based on the learning from the previous (t-1) iteration and current (t) input, as shown in Fig. 11. a. The main advantage of using RNN is keeping the previous state in memory and predicting the current state. A simple RNN architecture is built for the classification of states using accelerometer values as shown in Fig. 11. b. The input and output layer is represented as 'i' and 'o' respectively. AX_t, AY_t, and AZ_t are the accelerometer values observed at time 't', the hidden layer accepts the current input (AX_t, AY_t, and AZ_t) and the recurrence learning from the previous observation (t-1) to classify the sample obtained at time 't'.

As compared to other classifiers, the decision tree algorithm with entropy measured using the tanh function provides better results. So, the results of the modified decision tree algorithm are used to derive rules and set the threshold in state identification.

3.1.3. Rule extraction with threshold

As the tree generated many paths of varying heights as depicted in Fig. 12, without pruning the decision tree cannot be used for rule extraction which also increases the complexity of the system. The pruned decision tree generated using the J48 decision tree algorithm is used to extract the rules for critical event detection. The rule-based algorithm is constructed by including all the conditions specified in the nodes of the decision tree as in Fig. 13. The extracted rules are utilized to fix the threshold for critical event identification. The pruned decision tree with height 2 is generated with root node AZ.

The rule-based algorithm requires threshold values to determine the state transition using accelerometer values obtained from the sensors in the vehicle and the helmet of the rider. Fig. 13 is the result of the J48 decision tree algorithm executed on the dataset. The circles represent the internal nodes, which include parameters in the dataset that were used to make classification decisions. The remaining nodes are leaf nodes, which indicate a group of data points that belong to a single class. The system can classify the data point using the path with decision conditions. The accelerometer value measured along the Z-axis is denoted by AZ. The target class consists of three-leaf nodes: Normal Riding, Fall-like, and Fall. The pruned decision tree describes that when the AZ value is greater than 6.998, it is classified as normal riding. Similarly, the data point is classified as fall if the AZ value is less than or equal to 3.976. Fall-like is defined when the AZ value is less than or equal to 6.998 and greater than 3.976. To classify the accelerometer values into three states, these decision conditions are implemented as a rule-based algorithm. As a result, the tri-axial accelerometer values

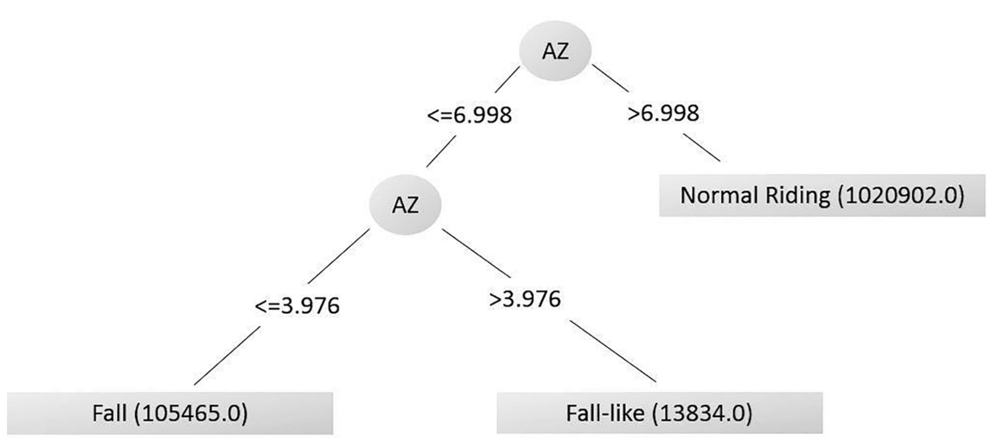

Fig. 13. Pruned decision tree generated by J48 decision tree algorithm.

Algorithm: Adaptive Sequence Window (ASW)

```
ASW ( Buffer S)
Precondition: Initially the buffer of size l_max is filled with sequences of states. K is the number
              of state transitions.
Post condition: returns 1, if fall state remains till the buffer is full; returns 0 if normal state
              remains till the buffer is full
Begin
P=1; l_w =l_min; previous_state = S1; k=0;
current_state =S[P];
While (P < l_w)
{
        if (current_state != previous_state)
        {
            k++;
            if (current_state == S1)
            {
                l_new = a_1 l_w - a_2 k;
                l_max=l_new;
                if (l_new < l_min)
                        l_w =l_min;
                else
                        l_w = l_new;
            }
            else if (current_state == S2)
                        l_w =l_min;
            else
                    Set P=1; l_w =l_max;
        }
        else
        {
            l_w =l_w +1;
            if (l_w == l_max)
                    Set P=1; return 0;
        }
        S[P]=current_state;
        P++;
        previous_state= current_state;
        if (P== l_w && current_state == S3)
            return 1;
}
end
```

Fig. 14. Adaptive Sequence Window algorithm.

measured by the sensor unit on the PTW and helmet are used to determine the state condition of the vehicle and the rider. The following rules are extracted from the decision tree as in Fig. 13.

1. AZ $>$ $\delta 2$ defined to be in class " Normal riding"
2. $\delta 1$ $<$ AZ$<=$ $\delta 2$ defined to be in class "Fall like"
3. AZ$<=$ $\delta 1$ defined to be in class "Fall"

where the threshold values $\delta 1 = 3.976$ and $\delta 2 = 6.998$.

Therefore the automatic critical event (Fall) identification is carried out in the OBD unit mounted on PTW by the above-defined rules. As the rules are very simple, the incident detection time will be very less.

3.2. Accident detection using adaptive sequence window (ASW) algorithm

Fall detected at any instant cannot be considered as an accident, due to the uncertain and highly unstable nature of PTW design. Continuous monitoring of the sequence of state transitions over a while is mandatory to confirm the occurrence of an accident. The challenge involved in accident detection is differentiating the different types of incidents precisely that can happen during PTW rides. If the algorithm is not devised properly, then actions like skidding, tipping, simple hit, and sliding can also be detected as an accident which leads to many false-positive cases. Therefore an ASW algorithm is designed to monitor the state sequence continuously since the PTW engine starts. ASW algorithm has a window called buffer (*S*), which is used to store the sequence of states of length (l_w). Initially, minimum (l_{min}) and maximum (l_{max}) values are fixed to define the size of the window. The length of the sequence in the buffer is adaptive based on the state transitions. The system considers 4 possible state transitions across three states namely normal state (S1), fall-like state (S2), and fall state (S3) within the sequence stored in the buffer. The state transitions are as follows:

Case 1. (*S1 → S2 (Normal → Fall-like)*) This case is more likely to occur whenever the rider losses his control over the vehicle and reaches the fall-like state from normal riding conditions. The probability of moving to a fall state and bouncing back from a fall-like to a normal state is equal. After the normal to fall-like condition, it is crucial to observe the transition towards a fall state. Therefore this transition will reduce the value of l_w and enable the close monitoring of the ride. This case can occur on the much-degraded track, leaning on a curve and due to sudden deceleration.

Case 2 S2 → S3 (Fall-like → Fall). The most crucial case is the transition from fall-like to fall. It happens within a fraction of a second. To precisely identify the state transition, reset the buffer. Hence every millisecond the value sensed by the accelerometer is continuously monitored before it gets accumulated on the buffer, which enables the system to identify the fallen state as soon as it occurs. So, the buffer is resized to minimum capacity and window length is thus reduced.

Case 3 S3 → S2 (Fall → Fall-like). After a fall, if the rider regains the stability or when the co-commuters help the rider to bring back to the normal state, then this transition will occur. After this transition, the value of *lw* is increased slowly till l_{max}, because still, it is uncertain that the fall may likely occur after this transition. This case is more likely to occur after a simple fall or slight tip over.

Case 4 S2 → S1 (Fall-like → Normal). This is a positive scenario similar to case 3, where the rider started moving in a normal ride after a skid or leaning on the curve. After this transition, the value of l_w is doubled because it attains the normal state and the vehicle is in control. To avoid the sudden raise in l_w, the raise in the buffer length is limited by the number of state transitions within the current transition time 't'. Also, when the number of state transitions increases, it, in turn, increases the probability of accident occurrence. Because an increasing number of state transitions indicate the loss of control of the rider over the vehicle or instability in riding. In both cases the chances of an accident occurring is high.

Based on the above four state transitions the ASW algorithm is proposed which can adaptively change the l_w to detect the occurrence of accidents accurately as depicted in Fig. 14.

If the PTW engine is switched on, the DSS decides the vehicle's and rider's condition depending on the rules. The sequence of states for the vehicle and the rider are created separately during the journey. The series is fed into the ASW algorithm as input. When the buffer is full, the ASW algorithm returns the state of the vehicle and rider. '0′ signifies no fall, while '1′ denotes a fall that occurs as a result of an accident. The algorithm fixes the length of the window depending on the current state (current_state) through the tuning parameter 'k' which reduces the response time to save the victim. If state transition occurs before reaching the maximum limit of the buffer, then the buffer is reset and the 'k' value is incremented. It is logically evident that the state of vehicle and rider can never be fall-like for a long time, vehicle and rider may either fall or return to a stable position very quickly. Depends on the individual riding the PTW, the time taken to either return to normal or fall from a fall-like state is undecidable. ASW algorithm thereby makes the system stable to decide the state by choosing an optimal value for the weights (a_1 and a_2) of the parameters l_w and k, to adaptively change the window size. The ASW algorithm is implemented and the results are validated with a sample set of a sequence taken from the dataset after the classification of states. The number of state transitions during the complete ride can also be obtained from the ASW algorithm. It helps us in understanding the riding patterns also. The algorithm eliminates false positives by waiting for the buffer window to fill before disclosing the current state after each transition. When another transition happens until the buffer window is complete, the condition is overlooked because it means that the rider is actively trying to keep the vehicle steady. The length of the window is identified using Equation (14) and is proved below. Corollaries 1 and 2 defines the probability $P(S_j \neq S_i|S_i)$- the probability of going to a new state S_j which is different from the existing state S_i.

Corollary 1. During PTW rides, $P(S2/S1) = 1$; where $P(S2|S1)$ is defined as the probability of the vehicle's current state is fall-like given that the vehicle is previously in a normal state.

Corollary 2. During riding, if the rider reaches fall like state, the rider can either reach the normal or fall states. Both these transitions are equally likely to occur. Hence,

$$P(S3|S2) = P(S1|S2)$$

$P(S3|S2)$ is defined as the probability of occurrence of fall after fall-like state; similarly $P(S1|S2)$ is defined as the probability of occurrence of normal after fall-like state.

Theorem. The length of the window is determined using $l_{new} = a_1 l_w - a_2 k$. The Change in the buffer length will reduce the time taken by the ASW algorithm to detect the occurrence of an accident, thereby reducing the response time.

Proof. As the states are stored in a buffer, the orientation of the vehicle and the rider can be decided once the buffer reaches its maximum capacity. So, the length of the buffer window (l_w) is directly proportional to the accident detection time.

Let R be the maximum allowable accident detection time by the system, and S be the minimum accident detection time. Then,

$$l_{min} = S \tag{10}$$

$$l_{max} = R \tag{11}$$

where l_{min} and l_{max} are the minimum and maximum buffer window length.

Whenever a state transition occurs to a new state, the current state is checked to adjust the window length.

After Case 1 and Case 3, the current state is a fall-like state, by corollary 2 the vehicle/rider will attain either normal or fall state very quickly, So l_w is set to l_{min} until the current state becomes a normal state.

In case 4, the transition reaches S2 here the previous state is ignored because S1 can be attained only from S2 from corollary 1. In this scenario, the rider regains his/her stability, so l_w can be increased, but due to the unstable nature of the PTW, it cannot be certain that once S1 is reached it is in a safe state. Hence l_{new} can be obtained by multiplying a factor 'a_1′ with the current window length l_w.

$$l_{new} = a_1 {}^* l_w \tag{12}$$

The increase in the number of state transitions (k) during the ride indicates the unsteadiness of the rider, which in turn increases the probable likelihood of an accident. Hence the factor 'k' is considered as one of the controlling variables while deciding l_w. Due to the varying travel time of each ride, the influence of 'k' on changing the window length is characterized by a factor a_2. The values of coefficients a_1 and a_2 play a vital role in deciding the rate of reducing the window length after a transition from FL to NR, therefore it cannot be fixed as constant.

Equation (12) is rewritten as,

$$l_{new} = a_1 {}^* l_w - a_2 {}^* k \tag{13}$$

$$l_{new} = a^T (l_w - k) \tag{14}$$

The value of 'a' can be found by solving the following equation using linear programming,

$$\text{Minimize}: a_1 {}^* l_w - a_2 {}^* k \tag{15}$$

$$\text{Subject to:} \quad \begin{array}{c} l_{min} \leqslant l_w \leqslant l_{max}; \\ l_w \geqslant 0; \\ l_{min} \geqslant 0; \\ k \geqslant 0; \\ l_{new} \leqslant l_{max}; \\ l_{new} \geqslant l_{min}; \end{array}$$

The intersection point of vertical and horizontal intercept of Equation (15) with the defined maximum and minimum value yields the

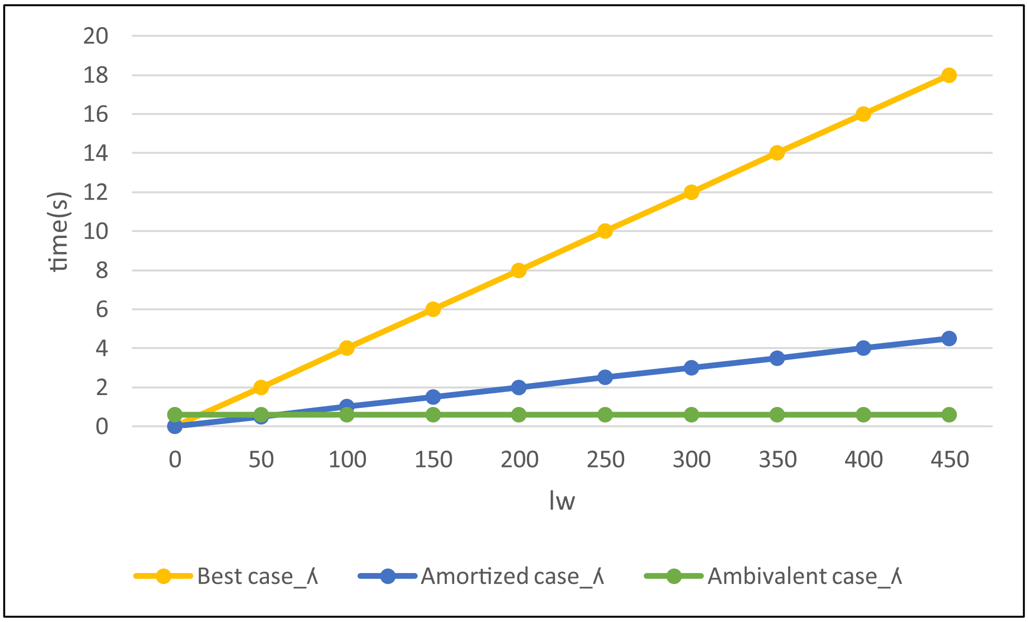

Fig. 15. Rate of change of detection time for different scenario.

optimal value for l_{new}. The accident detection time is directly proportional to the running time of the ASW algorithm which is O(l_w).

Let ʎ be the time taken by the ASW algorithm to assess the state of the vehicle using accelerometer values.

$$\lambda = \frac{l_w}{f} \tag{16}$$

where f is the frequency or sensing rate of the sensor. On substituting Equation (10) in Equation (16),

$$\lambda = \frac{R^*t}{f} \tag{17}$$

The ASW algorithm has a maximum detection time as a prerequisite. It is then modified depending on its characteristics during each ride. The maximum time bound for accident detection is defined by Equation (17). The following is how the proof is verified using mathematical induction.

Base case: At the time ‘t’ assume that the value of lw is n; if n = f, then ʎ=1.

Inductive cases:

Best case: A ride with no state transition or when the rider maintains the equilibrium state during the ride is defined as the best case for the study. In the best case, the length of the buffer window increases slowly as per the ASW algorithm. So the value of l_w and ʎ is given as,

$$l_w = n + \delta \tag{18}$$

$$\lambda = \frac{n+\delta}{f} \tag{19}$$

Amortized case: A ride with more numbers transition frequently is defined as the worst case for the study. In the worst case, the length of the buffer window changes more often with a decreasing trend. So the value of l_w and ʎ is given as,

$$l_w = n - \delta \tag{20}$$

$$\lambda = \frac{n-\delta}{f} \tag{21}$$

Ambivalent case: Whenever the current state is a fall-like state, as per corollary 2, the next state is equally likely to be either fall or normal. So the ASW algorithm shrinks the window size to the minimum bound. Also, the duration of the fall-like state is very minimal when compared to the normal and fall states. Equation (22) and (23) gives the window length and detection time for this scenario.

$$l_w = S^*t \tag{22}$$

$$\lambda = \frac{S^*t}{f} \tag{23}$$

During the PTW ride, the ASW algorithm continuously adjusts the buffer window length and detection time for each type of transition, as seen in Fig. 14.

On ordering the detection time (ʎ) for each case,

$$\frac{S^*t}{f} \leqslant \frac{n-\delta}{f} \leqslant \frac{n}{f} \leqslant \frac{n+\delta}{f} \leqslant \frac{R^*t}{f} \tag{24}$$

Equation (24) is depicted diagrammatically in Fig. 15 for sample values of l_w. Thus choosing an optimal window length adaptively with states of vehicle and rider as per the three scenarios explained above, will enable the system to detect the occurrence of accidents accurately within the maximum time limit set for the detection system. Response time of a traffic road accident is defined as the summation of accident detection time and ambulance travel time. Ambulance travel time is dynamic as per road traffic conditions, which cannot be controlled, while the accident detection time can be reduced by using the ASW algorithm.

3.3. Severity of Accident:

Different types of accidents occur on the PTW such as crashes, hits, collisions, sliding, rollover, and so on. Irrespective of the type of accident, the victim requires medical assistance based on the severity of the accident. However determining the accident severity in a PTW accident is very difficult as it purely depends on the individual, i.e. the rider, who handles the situation. Hence the computation of the accident severity score is uncertain. The many crash severity index assessments are defined by various authors in the existing dataset with the crash characteristics using machine learning algorithms. The score estimated through these learning algorithms cannot be used in the real-time scenario as the incidents don’t depend only on the vehicle-dependent parameters but it includes environmental parameters, which are highly unpredictable. And the co-commuters also play a vital role in estimating

Table 1
Severity of accident based on fall state.

Fall_Vehicle	Fall_rider	Severity of accident
0	0	–
0	1	Moderately injured
1	0	Not injured
1	1	Severe

Table 2
Severity of accident based on fall condition and pulse rate.

Fall_Vehicle	Fall_rider	Abnormal pulse rate	Severity of accident
0	0	0	0
0	0	1	X
0	1	0	low
0	1	1	high
1	0	0	low
1	0	1	low
1	1	0	low
1	1	1	high

the crash severity index. PTW may collide with other PTW or heavy vehicles or any roadside infrastructure, the crash severity level may change in each case. As all the parameters are highly dynamic and unpredictable, determining crash severity index automatically using machine learning algorithms without crash characteristics will result in a high false-positive rate (Wei, Shu, Huang, Taylor, & Chen, 2017; Wu, Sasidharan, Thor, & Chena, 2018; Chong, Abraham, & Paprzycki, 2004). Victim analysis adds more evidence towards the occurrence of the accident and concludes the accident severity class. The proposed methodology aims at categorizing the accident severity in two classes namely, high and low by considering the pulse rate of the rider. The real-time monitoring of all the physiological parameters of the rider is not feasible, especially after an accident/crash/hit, the pulse rate of a rider is monitored as it can be easily measured through sensors attached in the smart helmet.

The pulse rate of a normal adult may vary between 60 and 100 beats per minute (bpm) (University, 2019). All the telemedicine systems use this criterion to monitor the patients remotely. But fixing such a hard threshold common for all riders is not a good choice, because riders of all ages may or may not be with health ailments that deviate from this criterion. Hence, the proposed system determines the Normal Pulse Rate (NPR) of the rider dynamically on every ride starting from the beginning of the ride. The pulse rate of the rider is monitored and averaged only during the normal state as shown in Equation (25).

$$NPR = \left(\sum_{i=1}^{l_w} PR_i\right)/l_w \tag{25}$$

where l_w is the number of observations during the state S1.

$\Delta PR = |NPR - CPR|$ (26)

Whenever there occurs the state transition, the Current Pulse Rate (CPR) is sensed through a sensor. If the absolute difference (ΔPR) between NPR and CPR is greater than the threshold (Malamed, 2015) then it is referred to as abnormality in the pulse rate of the rider. The severity of the accident is defined in Table 1 based on the fall state identification of PTW and rider. The no-fall state of the vehicle or rider is encoded as 0 and the fall is encoded as 1.

Along with the rider's pulse rate, vehicle's and rider's fall/no-fall conditions are used to categorize the severity of the accident, as seen in Table 2.

In Table 2, 'X' in the second row indicates that the change in the pulse rate of the rider is not caused by the accident, it may be due to the health ailment of the rider. Each case represented in Table 2 is set as a hypothesis and justified using natural deduction and derived rules of predicate logic. The following predicate symbols are associated with the hypothesis:

Premises:

Let p: Fall of the vehicle
q: Fall of the rider
r: Abnormal pulse rate
s: Severity of accident

Table 3
Severity of accident with predicate logic expression.

Hypothesis. No	Fall_Vehicle	Fall_rider	Abnormal pulse rate	Severity of accident	Predicate logic
1	0	0	0	0	–
2	0	0	1	X	–
3	0	1	0	low	¬pˆqˆ¬r →¬s
4	0	1	1	high	¬pˆqˆr → s
5	1	0	0	low	pˆ¬qˆ¬r →¬s
6	1	0	1	low	pˆ¬qˆr →¬s
7	1	1	0	low	pˆqˆ¬r →¬s
8	1	1	1	high	pˆqˆr → s

Table 4
List of premises.

Number	Premise	Description
1	pˆq	Both vehicle and rider fall
2	pˆq → s	If both vehicle and rider falls, then severity will be high
3	¬pˆ¬q	Both vehicle and rider did not fall
4	¬pˆ¬q →¬s	If both vehicle and rider did not fall, then severity is low

The negation of the above declarative statements indicates either the value '0' or low (i.e.) ¬p, ¬q, ¬r, and ¬s.

In the first two cases, no-fall of vehicle and rider with normal/abnormal pulse rate is ignored because it does not have an impact on the severity of the accident. The following statements are considered as premises (Table 4) to prove the hypothesis given in the form predicate login in table 3.

The following are the justification table for the hypotheses 3 and 4 defined in table 3 using predicate logic rules.

Hypothesis 3. ***¬pˆqˆ¬r →¬s***

Proof.

Number	*Statements*	*Justification*
1	*¬pˆqˆ¬r*	*Assumption*
2	*¬pˆ¬q*	*Premise*
3	*¬pˆ¬q →¬s*	*Premise*
4	*¬s*	*→e 2,3*
5	*¬pˆ¬qˆ¬r →¬s*	*→i, 1–4*

where → e denotes implication elimination rule ; →i denotes implication introduction rule.

Hypothesis 4. ***¬pˆqˆr → s***

Proof.

Number	*Statements*	*Justification*
1	*¬pˆqˆr*	*Assumption*
2	*pˆq*	*Premise*
3	*pˆq → s*	*Premise*
4	*S*	*→e 2,3*
5	*¬pˆqˆr → s*	*→i, 1–4*

Table 5
Dichotomized portion of the combined dataset.

AX	AY	AZ
−0.379	0.022	9.82
−0.379	0.022	9.773
−0.379	0.046	9.773
−0.382	0.096	9.773
−0.385	0.046	9.772
−0.385	0.094	9.77
…	…	…
0.765	−0.001	6.202
0.765	−0.037	6.142
0.765	−0.037	6.107
0.765	−0.006	6.107
0.765	−0.006	5.941
0.563	−0.001	5.842
…	…	…
−2.682	0.01	1.317
−2.73	0.01	1.317
−2.874	0.01	1.317
−3.078	0.01	1.317
−3.281	0.01	1.437
−3.426	0.01	1.509
−3.545	0.01	1.509
−3.545	0.01	1.521

Similarly, hypotheses 5 to 8 can be proved using the natural deduction rules of first-order predicate logic (Huth & Ryan, 2004). Thus, Table 3 is proved and validated using predicate logic.

4. Results and discussion

4.1. Dataset Description

Based on the observation of realistic falls, different scenarios are made in the controlled environment, the dataset was collected with 7 parameters by Abderrahamane boubezoul et al in 2019. The features in the published dataset are accelerometer and Gyroscope readings with respect to X, Y, and Z-axis at a different instance of time for 11 different scenarios like acceleration on a curve, acceleration on a straight line, fall in a curve, fall in the roundabout, fall-like in Manoeuvre1, fall-like in Manoeuvre2, fall-like in Manoeuvre3, fall on a slippery straight road section, fall with a leaning of the motorcycle, harsh breaking on a straight line, much-degraded track (Boubezoul, Dufour, Bouaziz, & Espié, 2019). Since the dataset is unlabelled the proposed system first categorizes the data into three classes as Normal, Fall-Like, and Fall. The other classes indicating the type of ejection with different angles are not considered as they may increase the complexity of the system. The accelerometer values from all 11 experiments are extracted and combined to make a single dataset. A portion of the combined dataset is shown in Table 5. This combined dataset is further utilized for state classification.

The unlabelled dataset is clustered to group the data points into normal, fall-like, and fall using the K-means clustering algorithm because it results in a good outcome in cluster validation measures rather than hierarchical and PAM clustering.

4.2. K-means Clustering

The experimental setup considers 1,140,201 data instances and the dataset is split into a 70:30 ratio for clustering training and testing. The elbow plot is drawn to validate the selection of 3 cluster centers, as depicted in Fig. 16. The K means clustering algorithm with k = 3 clustered the data with an accuracy of 71.24%.

After clustering the data instances are labeled as normal, fall-like, and fall. The number of instances and the total spread of different clusters are shown in Table 6. The data points with normal classes are high compared to the other two classes.

4.3. Performance analysis of classifiers

Naïve Bayes, J48 decision tree algorithm, Artificial neural networks (ANN), and Recurrent neural network (RNN) are used to classify the labeled dataset. The performance of the classifiers are measured in terms of precision, recall, F-measure, Matthew's correlation co-efficient, ROC curve, and precision-recall curve (PRC) are compared in Table 7 to choose the best classifier. The entropy calculated using the tanh function in the decision tree reduces the convergence time to 1.31 s. Fig. 17 shows the entropy value obtained from a decision tree algorithm using information gain with Shannon's entropy function (Entropy_1) and tanh function (Entropy_2).

The precise difference identification between fall-like and fall states is very crucial. Based on the measures listed in Table 7, the classifier

Table 6
Clustering results on combined dataset.

Cluster	Number of instances	Spread on total	Label
Cluster 0	112019	10%	Fall
Cluster 1	435875	38%	Fall-like
Cluster 2	592307	52%	Normal

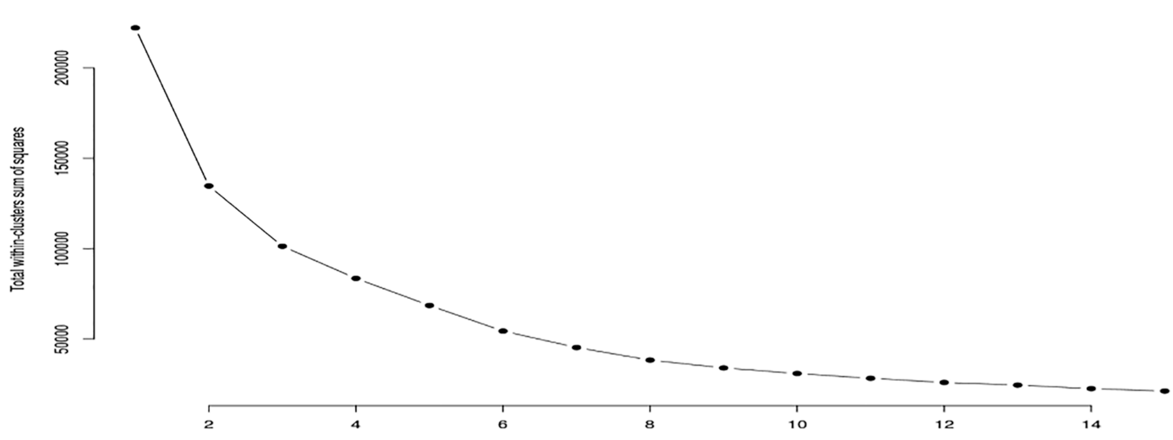

Fig. 16. Elbow plot for K means algorithm.

Table 7
Class wise detailed performance evaluation of machine learning algorithms.

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
Naïve Bayes	0.985	0.000	0.995	0.985	0.990	0.989	0.999	0.998	Fall
	0.998	0.006	0.999	0.998	0.998	0.985	1.000	1.000	Normal
	0.909	0.004	0.756	0.909	0.826	0.827	0.998	0.734	Fall-Like
J48 Decision tree (uses tanh to calculate entropy)	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Fall
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Normal
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Fall-Like
ANN	0.999	0.000	1.000	0.999	1.000	0.989	0.999	0.998	Fall
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Normal
	0.991	0.009	0.988	0.985	0.995	0.968	0.987	0.983	Fall-Like
RNN	0.995	0.045	0.996	0.991	0.9934	0.991	0.998	0.999	Fall
	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000	Normal
	0.997	0.005	0.998	0.998	0.998	0.984	0.999	0.999	Fall-Like

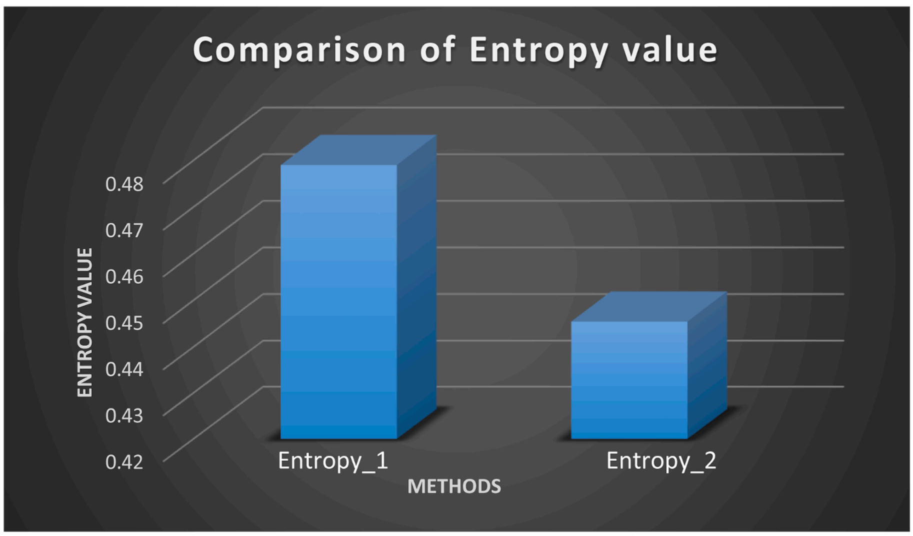

Fig. 17. The comparative results of entropy in ID3 and modified entropy function.

Table 8
A dichotomized portion of the labelled dataset.

AX	AY	AZ	state
−0.379	0.022	9.82	Normal
−0.379	0.022	9.773	Normal
−0.379	0.046	9.773	Normal
−0.382	0.096	9.773	Normal
−0.385	0.046	9.772	Normal
−0.385	0.094	9.77	Normal
…	…	…	…
0.765	−0.001	6.202	Fall-like
0.765	−0.037	6.142	Fall-like
0.765	−0.037	6.107	Fall-like
0.765	−0.006	6.107	Fall-like
0.765	−0.006	5.941	Fall-like
0.563	−0.001	5.842	Fall-like
…	…	…	…
−2.682	0.01	1.317	Fall
−2.73	0.01	1.317	Fall
−2.874	0.01	1.317	Fall
−3.078	0.01	1.317	Fall
−3.281	0.01	1.437	Fall
−3.426	0.01	1.509	Fall
−3.545	0.01	1.509	Fall
−3.545	0.01	1.521	Fall

with high accuracy, less convergence time, and low computational cost is selected as a best classifier. Thus, the J48 decision tree algorithm using a modified entropy function with stratified cross-validation is selected as the best classifier. Though neural network classifiers perform well in the classification of states it takes more training time due to high convergence time and high computational cost.

The sample dataset classified using a decision tree algorithm with a modified entropy function is shown in Table 8.

Due to the unstable nature of the PTW, state transition can happen at any instant of time. ASW algorithm in the DSS at OBD monitors the state transitions through the buffer window of length l_w and decides the occurrence of an accident.

4.4. The outcome of ASW algorithm

The ASW algorithm is designed to change the buffer window size adaptively concerning the riding behavior. The algorithm initializes its maximum and minimum value whenever a new ride starts. Irrespective of the rider's profile, the algorithm adapts itself to the characteristics of the riding style as soon as the engine is switched ON. The state estimation utilizing the ASW technique is validated using the data drawn for each example. The sample results of the changes in buffer window length are shown in Fig. 18.

The length of the buffer window grows and shrinks according to the four cases as depicted in Fig. 18(a) where the transition occurs from normal to fall-like in which, the window length decreases from 900 to minimum value 90. Fig. 18(b) shows the transition that occurs from fall-like to normal, after which the new window length is calculated as per the equation (15) and l_{max} is updated accordingly. The transition that occurs from fall-like to fall is shown in Fig. 18(c). After this transition, the buffer length is increased to an updated maximum value to reduce the false-positive rate of fall detection. Fig. 18(d) shows the transition

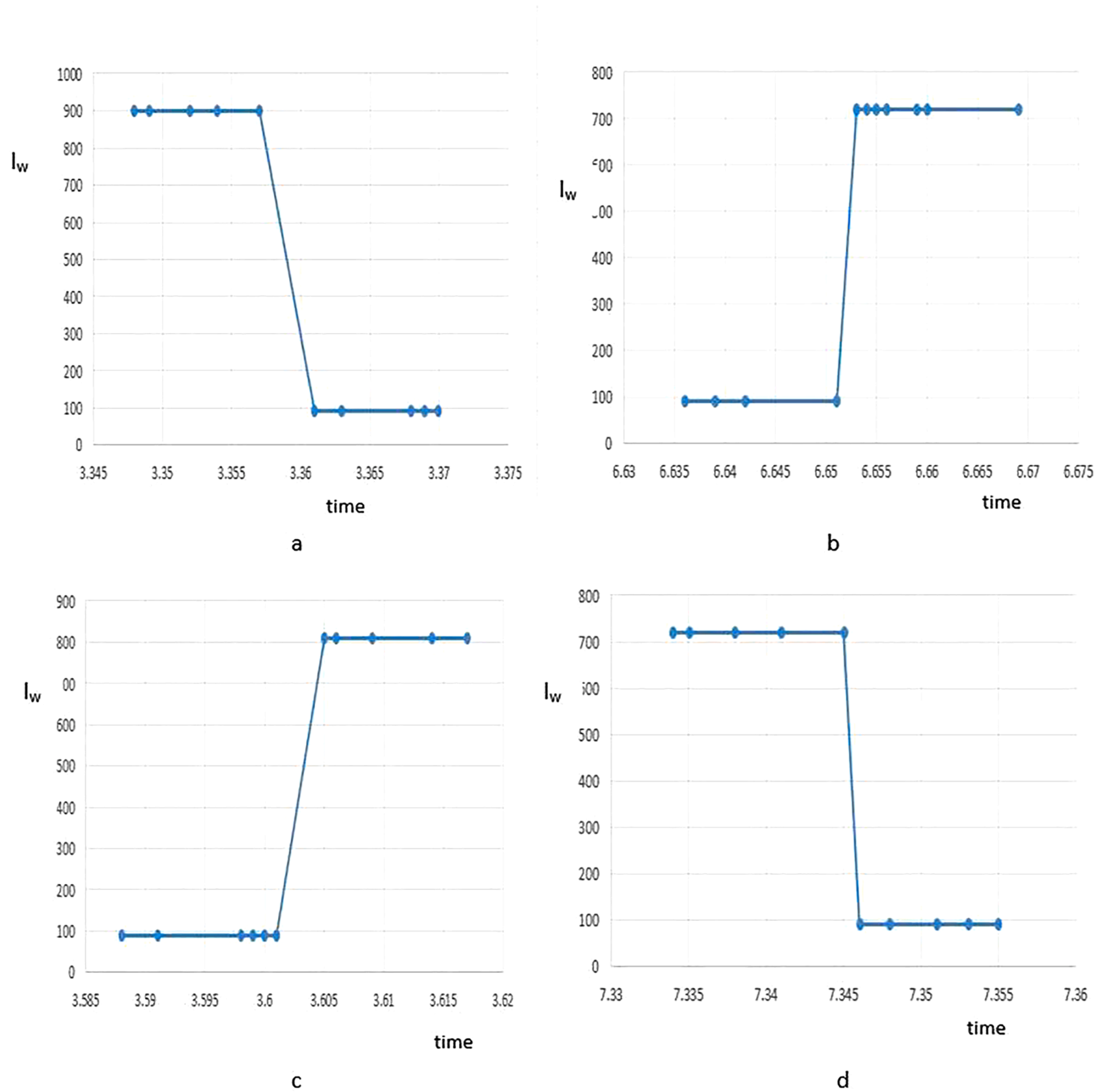

Fig. 18. Adaptive sequence window algorithm for accident detection (a) Normal to Fall-Like (b) Fall-Like to Normal (c) Fall-Like to Fall (d) Fall to Fall-Like.

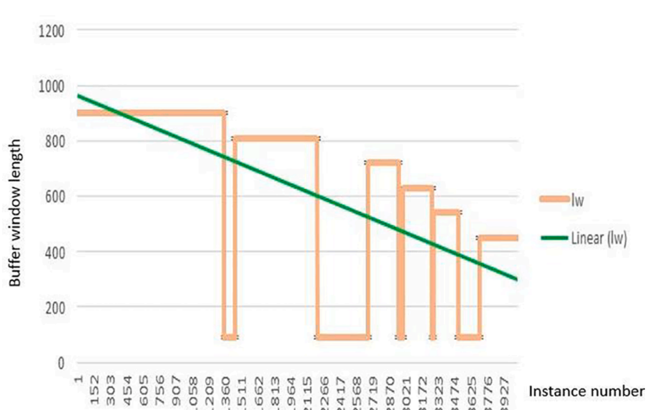

Fig. 19. Adaptively changing length of the window.

that occurs from fall to fall-like where the window length is decreased to the minimum value to render the minimum detection time because it may lead to a falling state quickly as explained in corollary 2.

The ASW algorithm is executed on the dataset recorded through the experiment, where an accelerometer is mounted on the vehicle and the rider performs a sequence of the events like lean over a turn, a hit fall, and again a lean. The adaptive changes in the window length (l_w) by the ASW algorithm are depicted in Fig. 19. A fixed linearly decreasing function to update the window length is also plotted on the experimental data to demonstrate the adaptive nature of the proposed ASW algorithm. The trend line of the linearly decreasing function of lw is represented by Linear(lw). The ASW algorithm performs the fall detection with a window length of $l_{max} = 900$ and $l_{min} = 90$ is shown in Fig. 19. The window length changes for each state transition according to the ASW algorithm. The adaptive change enables the system to differentiate between real fall and fall-like conditions. As the experimental values include all four cases, the adaptive nature of the window for all four cases is observed.

The impact of including the number of state transitions to decide the window length is illustrated in Fig. 20. It shows that when the number of transitions is increasing, the window length shrinks in accordance with

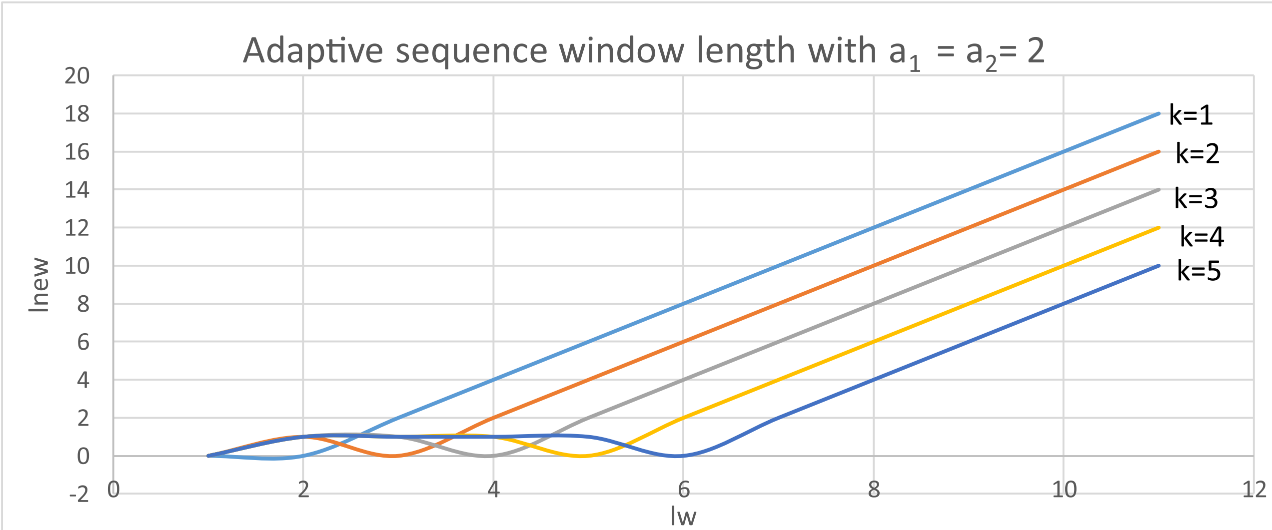

Fig. 20. Illustrative plot for new window length using different values of k.

Table 9
Specifications of the experimental setup.

Number	Parameter	Condition / unit
1.	Vehicle Model	Hero splendor
2.	Vehicle Weight	119 kg
3.	Number of riders	1
4.	Rider details: weight/ sex/age	85 kg/Male/31
5.	Time duration of the ride	20 min
6.	Sensor Model	BMI120
7.	Acceleration range	±16 g
8.	Output data rate	1.6 kHz
9.	Sensitivity	2048 LSB/g
10.	Resolution	16 bit

the current window length. The sample values are drawn for l_w to show the difference in estimating new window length as the value of '*k*' increases. If the ride is being continued with a fewer number of transitions, then there is no need to evaluate the state of the vehicle and rider more often. The ride is considered to be unstable when the '*k*' value increases.

To verify the correctness of the proposed algorithms and the time taken to detect the critical event (Fall), an experimental setup is carried out with a tri-axial accelerometer (BMI120) mounted on the body of the PTW. The sensor collection rate is 407 Hz which is down sampled to a sampling frequency of 45 Hz. The data is collected during the sequence events such as lean over a right turn, a hit fall, and fall on a right turn. The hit fall is simulated with an intended hit on the obstacle directly with a fall. The characteristics of the vehicle and sensor used in the experiment are listed in Table 9. The results of state estimation obtained using a rule-based algorithm and ASW algorithm are plotted in Fig. 21.

The recorded raw data is presented in Fig. 21(a), where gFx, gFy, and gFz represent the accelerometer values on the X, Y, and Z-axis. The existing threshold-based fall detection is executed and its result is depicted in Fig. 21(b), where $|g|$ represents the norm value of tri-axial accelerometer values. The proposed rule-based algorithm is used to classify the states with the extracted threshold from the decision tree. The outcome of the rule-based algorithm is depicted in Fig. 21(c) with 0 represents normal, 1 represents fall-like, and −1 represents fall state. The ASW algorithm is initialized with a maximum detection time of 5 min and a minimum detection time of 0.5 min and executed on the recorded experimental values. Hence, the initial window has a minimum sequence length of 90 and a maximum sequence length of 900. Fig. 21 (d) depicts the result of fall detection using the ASW algorithm with the state estimated using the proposed rule-based algorithm. The instances 4700 to 7800 represent fall over right turn during the experimental ride. It is found that instances from 6335 to 7660 have both fall-like and fall states by the rule-based algorithm, while ASW did not report it. A state transition to fall-like occurs before the within the window length range, because the fall phase is minimal. The actual hit fall on the experimental setup covers the instance range from 13,041 to 14502. The fallen state is identified and continued from instance number 13,346 to 14,556 by rule-based algorithm, where the fall is identified and reported by ASW at the instance 13645. The lag of 600 instances is accounted, due to three state transitions encountered during the journey before the fall. Similarly, during the fall on the right turn, the proposed ASW algorithm detects the fall correctly within the maximum fall detection time. Hence, the proposed algorithm outcomes are verified with experimental ground truth states.

The fall detection of the proposed ASW algorithm is compared with the existing threshold-based fall detection (Amin, Jali, & Reaz, 2012; Boubezoul, Espié, Larnaudie, & Bouaziz, 2013; Baramy, Singh, Jadhav, Javir, & Tarleka, 2016; Priyanka, Darshini, Shavi, & Begum, 2018) on the recorded values of the conducted experiment. The comparative result of fall state estimation is illustrated in Fig. 22. The states are represented numerically 0 for normal riding, 1 for fall-like, and −1 for fall state. The threshold-based fall detection ($|g|$) results in more number state transition before and after the events, while the ASW algorithm provides fall detection efficiently in accordance with the ground truth value with a minimal time lag.

The fall and accident detection time of the proposed sytem is compared with an existing system designed for crash dection by Gelmini et al. An automatic crash detection system (ACDS) was developed using a tri-axial accelerometer sensor mounted on the body of the vehicle (Gelmini et al., 2019). The proposed system's fall detection and accident confirmation time are compared with ACDS. Accelerometer values of six different sequence of different experimental setups from the dataset published by boubezoul et al (Boubezoul, Dufour, Bouaziz, & Espié, 2019) is used to compare the detection time of the proposed system with an existing system. The fall and accident detection time of the existing and proposed system are presented in the Table 10.

The sequence of accelerometer values collected during a fall in a curve, fall on a slippery straight road section, fall with a leaning of the motorcycle, fall on a much-degraded track, fall-like in Manoeuvre1, and fall in Manoeuvre2 are selected to compare the detection time of existing and proposed system (Automatic accident detection system). The minimum and maximum waiting time for the ASW algorithm is fixed as 60 and 120 s respectively.

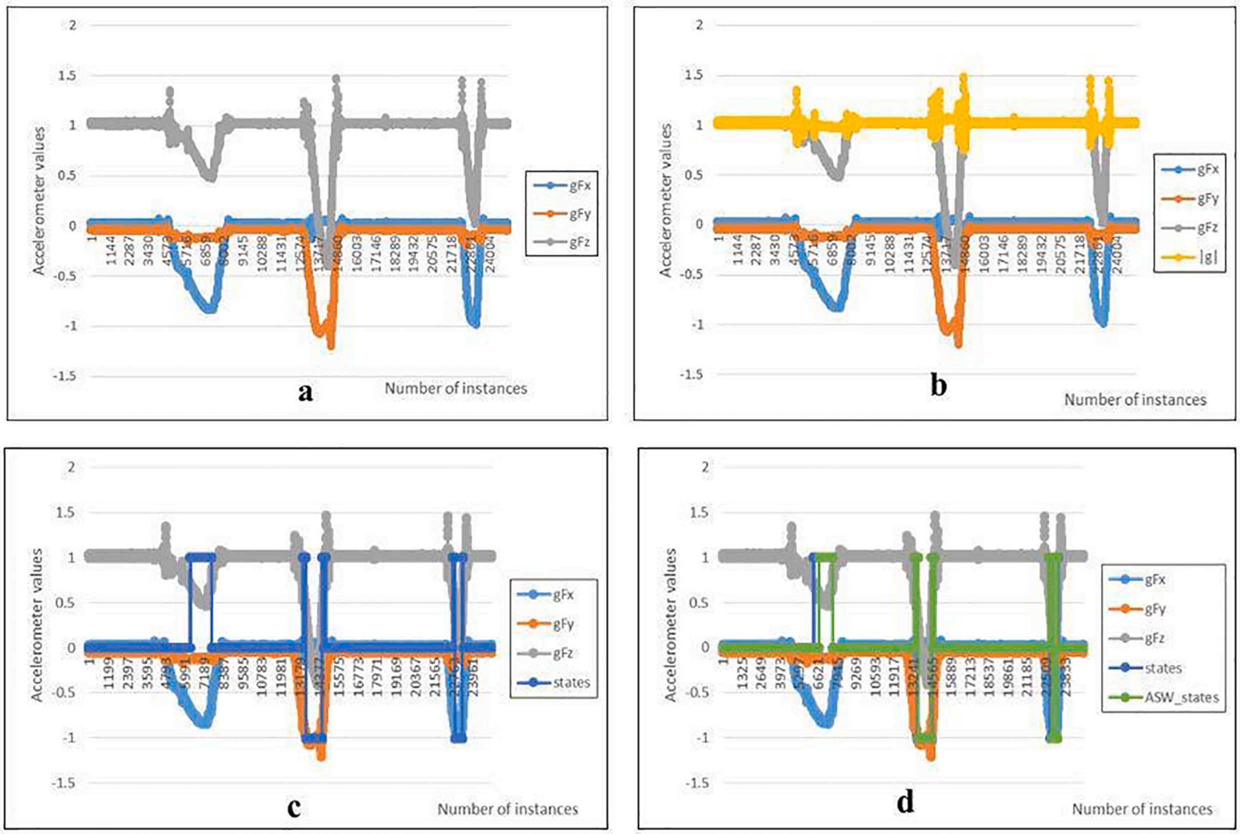

Fig. 21. State estimation on the sequence of the event right turn, a hit fall, and fall on a right turn using the proposed ASW algorithm a) accelerometer reading b) state sequence c) state estimation using ASW algorithm.

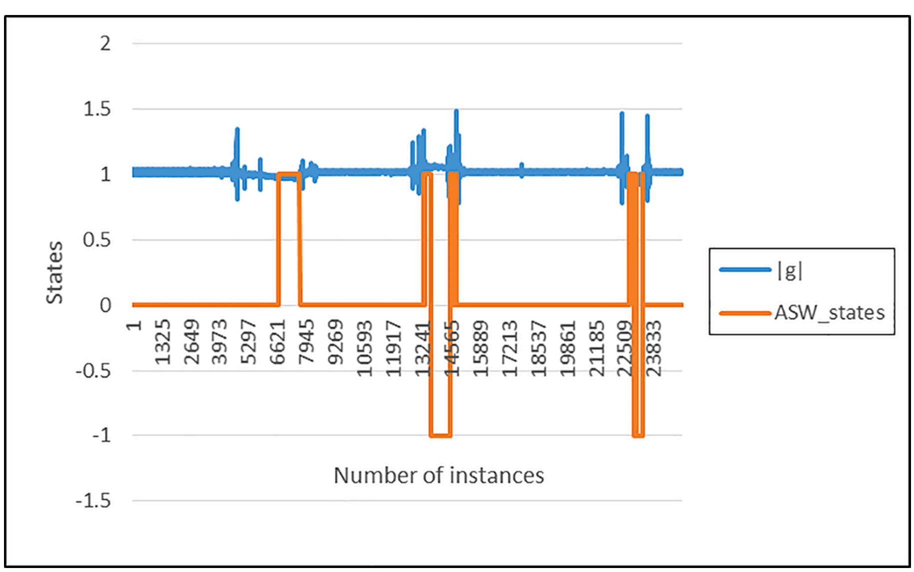

Fig. 22. Outcome of the proposed ASW algorithm and threshold-based state identification.

The accident detection time is the summation of fall detection time and the waiting time taken to confirm the fall as an accident. The proposed system identified the occurrence of an accident in 67 s for sequence 4(fall on a much-degraded track) because the buffer length was shrunk due to more number of transitions in poor road condition, but the proposed system takes 93 s to detect fall for sequence 1(Fall in a curve) due to less state transition before a fall. In sequence 6, it is an induced fall set up, the existing system reported it as a fall with the minimum waiting time and confirms it as an accident, while the proposed system did not report it as an accident because the duration of fall is less than the buffer length. This scenario will happen when the rider regains strength after a fall to proceed a normal ride.

Hence, the proposed automatic accident detection system for PTW provides the accurate result with less detection time when compared to the existing system.

Table 10
Comparison of fall and accident detection time in existing and proposed system.

Min_window_length = 60 Max_window_length = 120 t_wait = 60 s	Nature of the sequence	Fall detection (in seconds)		Accident detection time (in seconds)	
		Existing System (ACDS)	Proposed System (Automatic accident detection system)	Existing System (ACDS)	Proposed System (Automatic accident detection system)
Sequence 1	Fall in a curve	60	90	120	93
Sequence 2	Fall on a slippery straight road	60	84	120	87
Sequence 3	Fall with a leaning of the motorcycle	60	73	120	76
Sequence 4	Fall on a much degraded track	60	64	120	67
Sequence 5	fall-like in Manoeuvre1	**Fall not detected**	**Fall not detected**	NA	NA
Sequence 6	fall in Manoeuvre 2	60	86	120	**Accident not detected**

5. Conclusion

This research work aims at automatic detection of PTW accidents within very less time using parameters such as accelerometer value and pulse rate of the rider. In this regard, a simple and cost-effective OBD unit is designed, with a decision support system. A rule-based algorithm for state identification, ASW algorithm for fall detection, and severity assessment are required to implement such a system in an efficient and low-cost OBD unit with less memory space for all types of PTW. The first algorithm is used for the state identification of the vehicle and the rider. The rules used in this algorithm are extracted from the decision tree algorithm with modified entropy calculation. The second algorithm, the adaptive sequence window algorithm, is used to confirm the occurrence of an accident in terms of the fall of the vehicle and the rider by continuously monitoring the state of the vehicle and rider through a list of state sequences with linear time complexity of O(l_w). The length of the sequence is adaptive as per the state transition that occurs between three states namely, Normal, Fall-like, and Fall. Finally, the fall identification of the vehicle and rider with a pulse rate of the rider is used to categorize the accident in terms of severity. The irregular condition of the rider's pulse rate is also dynamically determined by monitoring the pulse rate from the start of the ride. First-order predicate logic is used to verify the classification of the accident severity as high or low. An automatic accident detection system for PTW is thus designed using three proposed algorithms deployed on the OBD unit.

CRediT authorship contribution statement

A. Jackulin Mahariba: Conceptualization, Software, Data curation, Writing – original draft. **Annie Uthra R.:** Investigation, Visualization, Supervision, Resources. **Golda Brunet Rajan:** Methodology, Formal analysis, Validation, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Stephens, A. N., Brown, J., Rome, L., Baldock, M. R. J., Fernandes, R., & Fitzharris, M. (2017). The relationship between Motorcycle Rider Behaviour Questionnaire scores and crashes for riders in Australia. *Accident Analysis & Prevention*, 202–212.

Ali, A., & Eid, M. (2015). An automated system for Accident Detection. *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, (pp. 1608-1612). Pisa: IEEE.

Amelie. (2018, June 01). *Digital health chronicles: Aclimed.* Retrieved from Aclimed company website: https://www.alcimed.com/en/alcim-articles/digital-health-chronicles-portable-sensors-patient-monitoring/.

Amin, M. S., Jali, J., & Reaz, M. B. (2012). *Accident Detection and Reporting System using GPS, GPRS and GSM Technology. IAPR* (pp. 640–643). Malaysia: IEEE.

Atalar, D., & Thomas, P. (2019). Powered two-wheeler crash scenario development. *Accident Analysis and Prevention*, 198–206.

Baramy, J., Singh, P., Jadhav, A., Javir, K., & Tarleka, M. S. (2016). ACCIDENT DETECTION & ALERTING SYSTEM. *International Journal of Technical Research and Applications e-ISSN, 2320–8163*, 8–11.

Boubezoul, A., Dufour, F., Bouaziz, S., & Espié, S. (2019). Dataset on Powered Two wheelers Fall and critical events detection. *Data in brief*, 1–6.

Boubezoul, A., Espié, S., Larnaudie, B., & Bouaziz, S. (2013). A simple fall detection algorithm for powered two wheelers. *Control Engineering Practice, 21*, 286–297.

5th International Congress - Sustainability of Road Infrastructures (pp. 881-890). Italy: Procedia - Social and Behavioral Science..

Cheng, A. S., Ng, T. C., & Lee, H. C. (2013). A comparison of the hazard perception ability of accident-involved and accident-free motorcycle riders. *Accident Analysis and Prevention, 43*, 1464–1471.

Chong, M., Abraham, A., & Paprzycki, M. (2004). Traffic Accident Data Mining Using Machine Learning Paradigms. *Fourth International Conference on Intelligent Systems Design and Applications (ISDA'04)* (pp. 415-420). Hungary: ISDA.

Elliott, M., Armitage, C., & Baughan, C. (2007). Using the theory of planned behaviour to predict observed driving behaviour. *The British journal of social psychology / the British Psychological Society.*, 69–90.

Fernandes, B., Alam, M., Gomes, V., Ferreira, J., & Oliveira, A. (2016). Automatic accident detection with multi-modal alert system. *Vehicular Communications*, 1–11.

Gelmini, S., Strada, S., Tanelli, M., Saravesi, S., & Tommasi, C. D. (2019). Automatic crash detection system for two-wheeled vehicles: design and experimental validation. *9th IFAC International Symposium on Advances in Automotive control* (pp. 498–503). France: IFAC PaperOnline.

Gibson, R. M., Amira, A., Ramzan, N., Casaseca-de-la-Higuera, P., & Pervez, Z. (2016). Multiple comparator classifier framework for accelerometer-basedfall detection and diagnostic. *Applied Soft Computing*, 94–103.

Grassi, A., Barbani, D., Baldanzini, N., Barbieri, R., & Pierini, M. (2018). Belted Safety Jacket: a new concept in Powered Two-Wheeler passive safety. *AIAS 2017 International Conference on Stress Analysis, AIAS* (pp. 573–593). Pisa, Italy: Structural Integrity Procedia.

Huth, M., & Ryan, M. (2004). Propositional logic. In M. Huth, & M. Ryan (Eds.), *Logic in Computer science- Modelling and reasoning about systems* (pp. 1–40). Cambridge: Cambridge University Press.

Jeong, H., Jang, Y., Bowman, J., & Masoud, N. (2018). Classification of motor vehicle crash injury severity: A hybrid approach for imbalanced data. *Accident Analysis and Prevention, 120*, 250–261.

Khalil, U., Javid, T., & Nasir, A. (2017). Automatic road accident detection techniques: A brief survey. *International Symposium on Wireless Systems and Networks (ISWSN)* (pp. 1-6). Lahore: IEEE Xplore.

Kobayashi, Y., & Takumi, M. (2013). Crash detection method for motorcycle airbag system with sensors on the front fork. *Proceedings of the 23rd International Technical Conference on the Enhanced Safety of Vehicles (ESV)*. Seoul, South Korea: ESV.

Li, Y., & Bai, Y. (2008). Development of crash-severity-index models for the measurement of work zone risk levels. *Accident Analysis and Prevention, 40*, 1724–1731.

Melchera, V., Diederichs, F., Maestre, R., Hofmann, C., Nacenta, J.-M., Gent, J. v., . . . Žagar, B. (2015). Smart vital signs and accident monitoring system for motorcyclists embedded in helmets and garments for advanced eCall emergency assistance and health analysis monitoring. *6th International Conference on Applied Human Factors and Ergonomics (AHFE 2015) and the Affiliated Conferences, AHFE 2015* (pp. 3208 – 3213). Las vegas , USA: Procedia Manufacturing, Elsevier.

Moore, D., Schneider, W., Savolainen, T., & Farzaneh, M. (2011). Mixed logit analysis of bicyclist injury severity resulting from motor vehicle crashes at intersection and non-intersection locations. *Accident Analysis and Prevention, 43*, 621–630.

Priyanka, Darshini, P., Shavi, V. G., & Begum, S. (2018). Two-Wheeler Safety System for Accident Prevention, Detection and Reporting. *International Journal Of Engineering And Computer Science ISSN:2319-7242, 7*(3), 23680-23682. doi:10.18535/ijecs/v7i3.03.

SaveDrives. (2016, june 17). Retrieved from SAVE DRIVES: http://www.savedrives.com/index.html.

Sharon, N., & Natassia, G. (2015). Do not blame the driver: A systems analysis of the causes of road freight crashes. *Accident Analysis and Prevention*, 141–151.

Sumer, N., Ozkan, T., & Lajunen, T. (2006). Asymmetric relationship between driving and safety skills. *Accident Analysis & Prevention*, 703–711.

Tanga, J., Lianga, J., Hana, C., Lib, Z., & Huanga, H. (2019). Crash injury severity analysis using a two-layer Stacking framework. *Accident Analysis and Prevention*, 226–238.

University, T. J. (2019, April 05). *Health Home :Conditions and Dieseases*. Retrieved from John hopkins Medicine: https://www.hopkinsmedicine.org/health/conditions-and-diseases.

Wei, X., Shu, X., Huang, B., Taylor, E. L., & Chen, H. (2017). Analyzing Traffic Crash Severity in Work Zones under Different Light Conditions. *Hindawi. Journal of Advanced Transportation*, 1–10.

White, J. T. (2011). WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones. *Mobile Network Applications*, 285–303.

Wu, K.-F., Sasidharan, L., Thor, C. P., & Chena, S.-Y. (2018). Crash sequence based risk matrix for motorcycle crashes. *Accident Analysis and Prevention, 117*, 21–31.