


## Smart Fall Detection by Enhanced SVM with Fuzzy Logic Membership Function

**Mohammad Kchouri**

(Universiti Teknikal Malaysia Melaka, Melaka, Malaysia,  
P031820017@student.utem.edu.my)

**Norharyati Harum\***

(Universiti Teknikal Malaysia Melaka, Melaka, Malaysia  
 <https://orcid.org/0000-0003-0068-6025>, norharyati@utem.edu.my)

**Hussein Hazimeh**

(Al Maaref University, MIDIL Lab and Lebanese University, Faculty of Sciences Beirut,  
Lebanon  
hussein.hazimeh@ul.edu.lb)

**Ali Obeid**

(Psymeon CTO, Paris, France,  
aobeyd@psymeon.net)

**Abstract:** Falling is a critical issue for disabled people, and it leads to potentially serious injuries and death. Smart fall detection is a technology that depends on sensors and auxiliary devices that seek to improve the quality of life and enhance the lifestyle of disabled people. So far, the most widely used fall prediction methods collect data from inertial measurement unit (IMU) sensors. In addition, they use thresholds to identify falls based on artificial experiences or machine learning (ML) algorithms. Nonetheless, these approaches still require extensive classification and calibration. In this paper, we suggest a new technique to detect falls by combining Fuzzy Logic (FL) and Support Vector Machine (SVM). The FL model is built by using a fuzzy membership function along with the input dataset to obtain the intermediate output. Because combining these two algorithms is not an easy task, we leverage SVM with a kernel comprised of a fuzzy membership function and thus build a new model known as FSVM. Besides, the hyperplane of the SVM is used as the separating plane to replace the traditional threshold method for detecting falling Activities of Daily Living (ADLs) on a comprehensive dataset containing simulated falling ADLs, non-falling ADLs, and scripted ADLs, including falling ADLs and unscripted ADLs performed by volunteers with our designed device. The results show that no false-positive rate had been triggered, and 100% specificity was achieved for ADL. An overall accuracy of about 99.87% in detecting the fall function was obtained. Furthermore, the overall sensitivity of 100% with no false negative rate obtained was achieved by implementing the proposed method. The attained results validate that our introduced method can effectively learn from features extracted from a multiphase fall model.

**Keywords:** fall detection, fuzzy membership function, SVM, kernel function, acceleration, IMU sensor

**Categories:** I.2, I.2.1

**DOI:** 10.3897/jucs.91399

\*Corresponding author

## 1 Introduction

Falling is one of the major life-threatening problems faced by disabled people. According to a report by the World Health Organization<sup>1</sup>, every year, 684 000 people die and an estimated number of 172 million suffer from short-or long-term disability. These numbers are expected to rise further in the coming decades as the life expectancy of the world's population increases. With this demographic shift, combined with the often fatalistic view that falls are an inevitable part of life (especially as we age), an intelligent fall detection system that is reliable and cost-effective must be considered as an option to assist people with special needs, especially the elderly, who may face serious injuries and might, in some cases, die as a result of falling.

Various studies have been conducted to investigate the fall detection problem [Roshni Thanka et al., 2022]. However, despite these studies, several limitations are still not addressed so far. One major problem which is very common among fall detection systems is related to low accuracy or high false positives [Gnanadesigan et al., 2022]. A false alarm is considered by fall detection systems with a single sensor, which means that other information might be needed to improve the accuracy of the system. The main objective of this fall detection method is to achieve higher performance than the existing state-of-the-art methods.

Recently, there has been a growing interest in identifying and detecting fall events using IMU sensors, which combine triaxial accelerometers, triaxial gyroscopes, and triaxial magnetometers [Safeea and Neto, 2019] for data collection, and set thresholds to identify a fall based on artificial experience or ML algorithms. However, a suitable threshold should be set in order to avoid any problems [Yacchirema et al., 2018]. Certainly, Threshold-Based (TB) approaches have a poor recognition capability due to high false alarms.

ML algorithms can effectively enhance the system's performance in comparison to the threshold-based method [Aziz et al., 2017]. Nonetheless, this necessitates collecting a huge amount of data and is easily affected by noise and outliers; which causes an escalation in the computing difficulty and system value [Yang et al., 2021]. Therefore, ML algorithms distinguish poorly between falls and activities of daily living; and none of them is commonly accepted.

SVMs is one of the most regularly used algorithms; however, they will not be able to correctly spot the corrupted data since an SVM is greatly sensitive to outliers. Furthermore, the outputs need to be labeled in advance. Classical SVM can neither use prior knowledge to process accurate classifications nor solve problems characterized by inaccuracy and ambiguity.

The relationship between SVMs and fuzzy systems will be highlighted to avoid the classical issue of dimensionality, which appears when a high-dimensional data problem is addressed, thus resulting in high generalization performance.

This paper provides a deeper analysis of the novel FSVM fall detection method, highlighting its advantages over existing methods. Specifically, the authors describe the two main steps of the proposed method, including the trapezoidal membership function

---

<sup>1</sup> Step safely: strategies for preventing and managing falls across the life-course. Geneva: World Health Organization; 2021

and SVM, which work together to determine the degree of membership of each input value from the sensor features through Fuzzification. Additionally, the researchers present a thorough evaluation of the proposed method by introducing new evaluation metrics and comparing it with previous studies. Moreover, this paper explicitly addresses the limitations of SVM without a fuzzy membership function in fall detection systems.

To ensure the robustness of the proposed model, this study includes a description of the wearable device prototype used for hardware implementation, as well as an examination of the activities of daily living and their fall subcategories during the experimental phase. Overall, this paper presents a compelling case for the superiority of the FSVM approach in fall detection and provides valuable insights into the design of wearable devices for health monitoring applications.

The rest of this paper is organized as follows: Section 2 provides an overview of the related work of fall detection-based algorithms. Section 3 highlights the main contributions of the proposed fall detection method. The proposed method is comprehensively explained in Section 4. The experimental settings and the criteria are detailed in Section 5. The findings and the results for the various scenarios are detailed in Section 6. The conclusion by pointing out the important observations and guidelines for potential future directions are discussed in Section 7.

### Table of Notations

Notation	Definition
MA	Moving Average
SD	Standard Deviation
$ A_{SD} $	Euclidean Norm of the SD Acceleration
$ G_{SD} $	Euclidean Norm of the SD Rotation
$ M_{SD} $	Euclidean Norm of the SD Magnetic Field
MF	Membership Function
$\mu_F(x)$	Degree of Membership

## 2 Related Work

Currently, available techniques that are used to design fall detection systems are classified into three categories: (1) the vision-based method, (2) the ambient-based method, and (3) the wearable-based method. As such, there are several pros and cons for each one of these approaches.

The vision-based approach uses single or multiple cameras in an indoor environment to track a person's movements and body shape while falling [Espinosa et al., 2019, Sathyanarayana et al., 2018], which are then processed using an image processing technique [Harum et al., 2018]. The main advantage of the vision-based approach is that the person does not have to wear any extra devices for fall detection. However, such approaches continue to face difficulties and limitations in terms of pervasive detection, affordability, and acceptability.

Ambient-based approaches use ambience sensors which include floor vibrational data [De Miguel et al., 2017], or audio data and passive infrared (PIR) sensors for detecting a fall event [Ren and Peng, 2019]. These technologies allow environments to be sensitive, adaptive, and responsive to the presence of people to support them in living independently in their preferred environment.

The wearable device-based methods require the subjects to put on some devices or garments with embedded sensors such as magnetometers, gyroscopes, and accelerometers to track the user's body motion or posture [Kuncan et al., 2022, Roshni Thanka et al., 2022]. The data collected by the inertial sensors are used as motion signals to analyze the state of movement [Kuncan et al., 2019, Wu et al., 2020]. Data collected by the accelerometer and gyroscope are transmitted to a microcontroller to be processed to differentiate a fall from an ADL. The fusion of inertial sensor-based wearable systems can be effectively used to recognize fall events by examining the impact of the body on the ground and the body orientation prior to, during, and following the fall. Nevertheless, the location of the sensor and the external noise can influence the performance of the system.

The classification algorithm is applied to classify ADL and several fall events. A wearable-based fall detection method can be categorized into two approaches, namely: (1) TB, and (2) ML. TB approaches use single or multiple threshold values to classify events. For instance, [Bourke et al., 2007] proposed a TB fall-detection algorithm using a bi-axial gyroscope sensor. [Kangas et al., 2008] used a TB fall detection algorithm with a built in triaxial accelerometer attached at the wrist, head, and waist to show the optimal placement for wearable sensor-based fall detection. Accordingly, they reported that the waist is the most efficient position, while the wrist is not. Due to the low computational complexity, current fall detection studies have widely used the fixed threshold-based method. However, a low threshold value brings out false alarms, whereas a high threshold value causes large-fall missing issues.

ML is a subset of Artificial Intelligence techniques. It uses statistical methods to help in improving machines' experience, thereby allowing computers to learn new knowledge and abilities, identify existing knowledge, and continuously enhance efficiency and achieve self-improvement methods. It extracts patterns from raw data automatically [Rahmani et al., 2021]. In particular, ML algorithms include Hidden Markov Model (HMM), SVM, Decision Tree, Linear Regression, Naïve Bayes, K-Nearest Neighbors (K-NN), Random Forest (RF), and (FL), which have been widely used for decades in various fields like healthcare, and subsequently achieved significant success [Ahmad et al., 2018]. In the ML-based approach, different types of falls and ADL patterns are trained by a learning algorithm. The event is later on classified by an evaluation algorithm.

[Hussain et al., 2019] conducted experiments with three ML algorithms: KNNs, SVM, and RF. They attained the best accuracy in fall detection using the KNNs classifier, and the highest accuracy in distinguishing various falling activities using the RF classifier. [Aguiar et al., 2014] proposed a smartphone-based detection system by using an accelerometer sensor embedded in the device. Additionally, they checked three ML algorithms, namely the Decision Tree, the K-NN, and the Naive Bayes, but among those algorithms, the Decision Tree has proven to have the best performance. Numerous recently published papers discuss different aspects of fall detection techniques based on the combination of TB and ML-based algorithms.

Recently, Neural Networks have been greatly improved in recent times [Collado-Villaverde et al., 2020, Kakarash et al., 2020]. This method regularly outperforms classic ML algorithms in terms of learning ability. However, the model of the Neural Network is highly dependent on the quality and quantity of the training datasets and might largely lead to disorienting information. [Yodpijit et al., 2017] used accelerometer and gyroscope motion sensors to detect the fall and focused on TB and Artificial Neural Network algorithm to distinguish between ADL and falls to minimize the number of false-positive outcomes. Nonetheless, the method based on the fusion or merging of numerous data sources introduces unnecessary information, which causes an escalation in the computing difficulty and system value. Furthermore, [De Quadros et al., 2018] evaluated the accuracy of these two approaches. They achieved the best accuracy performance of 99% where the system's performance in comparison to the TB method can be effectively enhanced.

The paper authored by [Kchouri et al., 2022] is a comprehensive exploration of the theory of Support Vector Machines (SVM) and Fuzzy SVM in the context of classification problems. The authors conducted a clinical study to investigate the relationship between falling events and disabled individuals with traumatic brain injuries.

SVMs are feed-forward networks that focus on building an optimal hyperplane that acts as a decision space where the margin separating positive and negative examples is maximized [Vapnik, 1999]. The solution of the optimal hyperplane can be written as a combination of a few input points that are called support [Lin and Wang, 2002]. The SVM finds the optimal hyperplane by using a group of mathematical functions known as the kernel [Lee and Kim, 2016]. The most vital are linear kernel function, polynomial kernel function, and Radial Basis Function (RBF) kernel function [Pan et al., 2020]. For instance, an SVM-based pre-impact fall detector was presented by [Zhen et al., 2016], who selected an RBF kernel which allows nonlinear mapping in this model. A fall detection system based on an Android and accident alarm application was developed by [Shahzad and Kim, 2019]. All kernels were constructed by [Shahzad and Kim, 2019] to distinguish difficult fall-like incidents. Nonetheless, the number of selected kernels also expanded with the regulation parameter ( $c$ ), which increased the computational cost.

The SVMs even achieve notable performance in problems regarding classification. However, they cannot give a comprehensible representation of where a produced output has been attained. A limitation of the SVMs is that they act as Black Boxes [Castro et al., 2007]. The Fuzzy SVM approach presented by [Kchouri et al., 2022] outperformed traditional SVM in fall detection accuracy by reducing false alarms, which was achieved through the integration of a fuzzy membership function. Meanwhile, the main problem is choosing the parameters that SVM and kernel tricks need. An example of such is the regularization parameter ( $c$ ), which is an unintuitive parameter that plays a major role in maximizing the margin and carefully tuning the number of misclassifications. Furthermore, specific points in the training data are placed on the wrong side. These are called outliers, and in such a case, classical SVM will not be able to correctly spot the corrupted data since SVM is greatly sensitive to outliers.

According to [Zadeh, 2008], FL is an extension of Boolean logic based on the mathematical theory of fuzzy sets, which is a generalization of the classical set theory. The Fuzzy is bounded with a degree of membership between  $[0, 1]$ , where 1 means absolute truth and 0 means absolute falseness [Castañeda et al., 2022]. The FL consists

of four parts while the rule base contains all the rules and if-then conditions to control the decision-making system. The fuzzification allows converting crisp numbers into fuzzy sets. The Inference Engine determines the degree of match between fuzzy input and the rules based on the matching percentage. The most commonly used fuzzy inference technique is called the Mamdani method, which requires finding the centroid of a two-dimensional shape by integrating it across a continuously varying function. Finally, there is the defuzzification process that converts the fuzzy sets into crisp values. FL enables the identification of falls and categorizes them into one possible form of fall. [Fernández-Caballero et al., 2013] used FL to identify the range and type of fall, which includes the position before fall, fall direction, fall velocity, and post-fall inactivity. [Moulik and Majumdar, 2019] proposed a *FallSense* system prototype that combined the data from multiple sensors and generated a value between 0 and 1, which implies the chance of a fall. For a fuzzy inference technique, the Mamdani model was considered. To solve the problem of rowdy sensor data and reduce the number of false alarms, [Zhang et al., 2020] initiated a new FL algorithm, which is worn on wrists to detect falls. The fall detection system has three major phases, which are data sampling, data processing, and fuzzy classification. In the three stages, a typical FL procedure was done by fuzzily setting all input values into fuzzy membership functions, executing all relevant rules to calculate the fuzzy output functions, and defuzzifying the fuzzy output functions to get output values. They used Mamdani's minimum operation and weighted average formula in defuzzification. [Pekala et al., 2022] developed a novel hybrid approach based on fuzzy and rough sets that applies to reducing the number of rules in a set while preserving the inference process's efficiency. However, different defuzzification methods present different results in the evaluation.

FL could reduce the limitations in parameter evaluation to introduce flexible and smooth decisions. However, fuzzy systems are systems with domain fuzzy sets of variables, which present the potential to emulate the human way of thinking to productively use reasoning methods that are uneven and imprecise. Nonetheless, they do not produce the most preferred features of adaptation and learning, especially for fall detection. In addition, fuzzy systems do not have the capability of ML as well as neural network type pattern recognition and more. This paper introduced the FL-enhanced classical SVM classifier (FSVM) in order to respond to the aforementioned problems. Throughout the process, we benefited from the experience of other experts who worked on constructing fuzzy membership functions, utilized SVM with selected kernels, and accordingly adopted a fuzzy SVM in order to reach a decision.

### 3 FSVM Combination Performance Analysis

Our analysis of the FSVM combination was motivated by the following question: "Why does the combination of FSVM perform much better than traditional SVM methods on smart fall detection systems?" The results were interesting especially that many researchers gave special attention to this problem. In general, the most commonly used fall methods employ kinematic sensors like gyroscopes, accelerometers, vision cameras, and vibration sensors. Such methods are usually used to collect kinematic information while observing a specific fall event, and then can set thresholds via experience or SVM algorithms to determine whether a fall has truly occurred or not. Nonetheless, there are two main drawbacks to using these conventional methods.

The first problem is that there is no alignment between the sensitivity and the specificity of demining the threshold. Hence, if an examiner seeks a highly accurate system, then a high threshold must be set. However, with a high threshold, there will be some lags in the system and, subsequently, some missed falls. In contrast, with an excessively low threshold, there will be some misjudgments and frequent false alarms.

Second, while using SVM fall detection algorithms, the experimenter needs to label the output type manually, which requires intensive work and consumes a lot of time. In this sense, it is worth mentioning that with classical SVMs, the experimenters can never depend on their prior knowledge to process accurate classifications or solve inaccurate and ambiguous problems.

To solve these problems, an FSVM, employs SVM by Fuzzification of all input values of the training data into fuzzy membership functions. This will help in providing an initial decision regarding whether an event of falling likelihood is high. The output of such FL analysis as a block will be fed into an SVM to reduce false alarms and achieve accurate fall detection.

This FSVM is often utilized as a means to overcome the challenges of threshold setting and conventional SVM. Thus, with the functions of the fuzzy membership, an SVM output is automatically generated. This does not only minimize the hesitation in labeling the results but also improves the heuristic transparency of the SVM. As far as learning is concerned, the function of SVMs can be enhanced if we apply FL since it allows us to handle uncertain information and hence supports the SVM's ability to self-learn.

#### **4 Materials and Methods**

A combination of physical components was used to construct a prototype of the wearable device in Figure 1. MPU-9250 with nine-axis motion (Gyroscope + Accelerometer + Magnetometer) was utilized to collect the motion data that occurs while the patient is falling. In this paper, the Bluetooth module HC-05 and Arduino Uno are used for hardware implementation. Correspondingly, the sensor node can be customized specifically for applications of detecting falls and thus can help detect a change in activity, generate acceleration and angular rate output data, and digitize the magnetometer outputs with a full-scale range. It attains targets with low power consumption and robustness during the short duration of dynamic accelerations. Using MPU-9250, the slope of the object that the sensor is mounted on can be found, and the angle and rotation about each axis can be expressed. Gathering the measured values of the 9-axis acceleration sensor can realize a wide variety of applications to achieve the research objective.

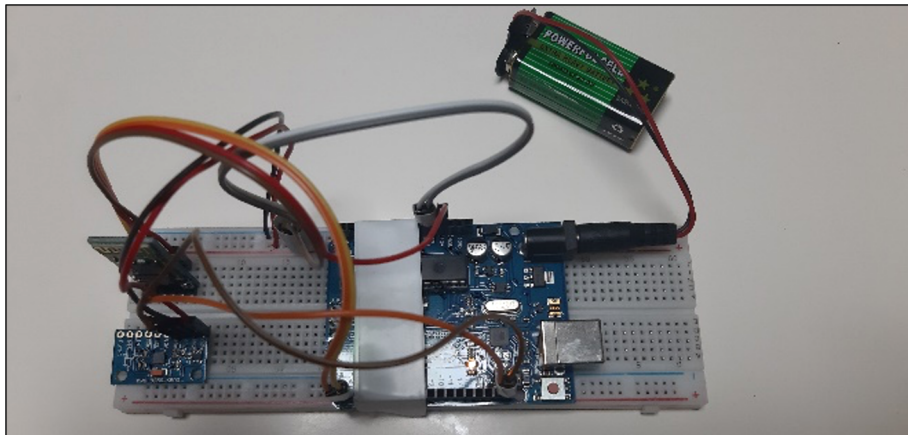


Figure 1: System prototype

The Arduino is a microcontroller-based open-source electronic prototyping board that can be designed with an easy-to-use Arduino IDE. One of the most well-known boards in the Arduino family is the UNO. The Bluetooth module HC-05 is used for wireless communication between the Arduino Uno and MPU-9250.

A 9-axis motion tracking IMU module is attached by the I2C communication protocol (Inter-Integrated Circuit) to the board. An MPU-9250 can provide us with information about yaw angle, pitch angle, and roll angle. In Figure 2, we used an Arduino UNO Library for MPU-9250 to read the accelerometer, the gyroscope, and the magnetometer, as well as the internal temperature and the Tait Bryan angle like pitch roll and yaw.

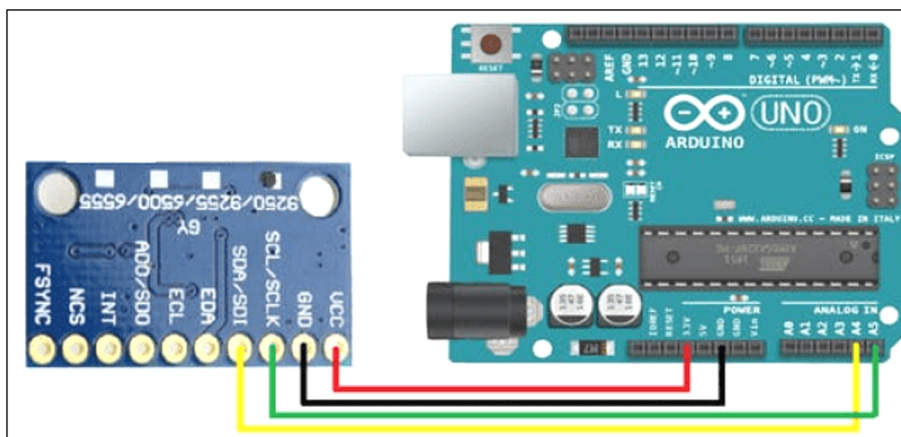


Figure 2: MPU connections with Arduino UNO



In this paper, two software tools are used: The Arduino Integrated Development Environment (IDE) - which is used to receive and transmit data serially from the Arduino board to the IMU sensor - and the Bluetooth terminal application. The entire programming for the proposed system is done by using the Arduino IDE tool. The Baud rate is set to 9600 bits per second for serial communication between the Arduino board and the MPU-9250.

A specific serial sniffer called PuTTY, which is saved to a CSV file, is used to gather raw data. The IMU raw data is first smoothed to ensure that no information is lost. A preprocessing phase follows the collection of data from the sensor. Useless information or noise gets withdrawn from the raw data. In this paper, the input matrix will be made up of 9-axis Accelerometer, Gyroscope, and Magnetometer data. Thus, the 9-axis was used to extract SD features to detect a sudden change in the acceleration in zero-gravity and to identify the fall range. This range incorporates position before the fall, fall direction, fall velocity, and post-fall inactivity. This operation is processed by the fuzzification of all input values of the training data into fuzzy membership functions.

In FSVM, a fuzzy membership is given to each data point. Different memberships reflect various contributions to the learning of decision-making. When the dimension of the input space is very high, the Fuzzy SVM network will be evaluated by splitting the datasets from model training and real-time data for testing and validating the fall detection. Figure 3 shows the main blocks constituting the newly proposed method. This will help in providing an initial decision biasing whether an event of falling likelihood is high. The output of such FL analysis as a block will be fed into an SVM to reduce false alarms and achieve accurate fall detection.

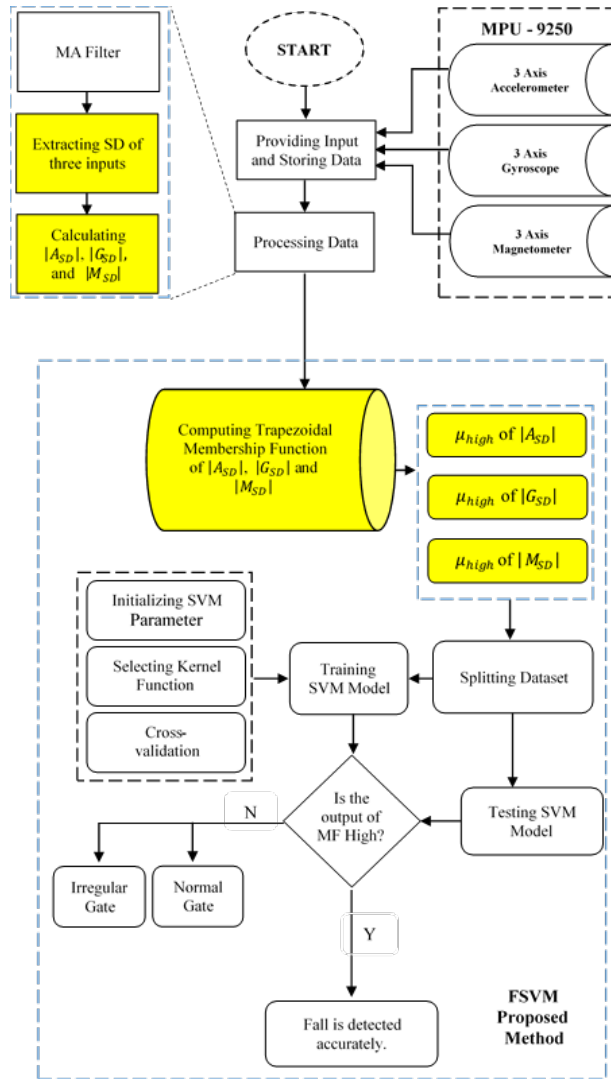


Figure 3: Fall detection proposed algorithm; a combination of two main steps: trapezoidal membership function and SVM

#### 4.1 Data Processing

The IMU raw data are first smoothed to ensure that no information is lost. A simple MA algorithm is used for smoothing noisy raw data since it is the most frequently used filter due to its ease of understanding and use. When MA is applied, the effect of an outlier or accidental spike that can lead to false detection of an event is worked on to be reduced.

Alternatively, this paper introduces how  $SD$  is used, which provides information about the changing rate of human motion and human shape [Worrakulpanit and Samanpiboon, 2014]. Assumingly, a human fall is of high acceleration, whereas walking is regarded as a low-acceleration activity. The acceleration value of human movement for specifying the changing rate of human motion can be calculated by using  $SD$ . Thus, when an  $SD$  method is applied to nine axes, it can differentiate between an actual fall and other activities.

In our approach, three antecedents are computed from the MPU-9250:

$|A_{SD}|$ , which is calculated in equation 1:

$$|A_{SD}| = \sqrt{Ax_{SD}^2 + Ay_{SD}^2 + Az_{SD}^2} \quad (1)$$

Where  $Ax_{SD}$ ,  $Ay_{SD}$  and  $Az_{SD}$  represents the  $SD$  of the acceleration along three axes to detect a sudden change in the acceleration in the zero-gravity and impact states as the most explicit method to detect falls.

In addition, we introduced  $|G_{SD}|$ , which is calculated in equation 2:

$$|G_{SD}| = \sqrt{Gx_{SD}^2 + Gy_{SD}^2 + Gz_{SD}^2} \quad (2)$$

where  $Gx_{SD}$ ,  $Gy_{SD}$  and  $Gz_{SD}$  represents the  $SD$  of the rotation speed along three axes to measure changes in orientation and rotational velocity.

$|M_{SD}|$ , was introduced and represented in equation 3:

$$|M_{SD}| = \sqrt{Mx_{SD}^2 + My_{SD}^2 + Mz_{SD}^2} \quad (3)$$

where  $Mx_{SD}$ ,  $My_{SD}$  and  $Mz_{SD}$  represents the  $SD$  of the magnetic field along three axes to measure the direction and displacement among the three axis.

#### 4.2 Generate Fuzzy Membership Function

The FL system takes the inputs into fuzzy membership functions and evaluates the output based on ambiguity and imprecise information to get a crisp output value between 0 and 1, which signifies more chances of a fall. The FL operation is processed by Fuzzification which involves converting the logical input value from the features of the sensor into the degree of membership of each input variable.

The input spaces are represented by the sets  $|A_{SD}|$ ,  $|G_{SD}|$ , and  $|M_{SD}|$  respectively. The corresponding ranges of these sets are represented by the closed intervals  $I_{|A_{SD}|}$ ,  $I_{|G_{SD}|}$ , and  $I_{|M_{SD}|}$  respectively, where  $I_{|A_{SD}|} = [0; 300]$ ,  $I_{|G_{SD}|} = [0; 30]$  and  $I_{|M_{SD}|} = [0; 100]$ .

The basic structure of the proposed fuzzy system is formulated based on these intervals. After that, this research considered three linguistic values or fuzzy sets 'LOW', 'MEDIUM', and 'HIGH', for each input.

In this paper, trapezoidal MFs are considered for all the fuzzy sets in  $f_x \forall x \in \{|A_{SD}|, |G_{SD}|, |M_{SD}|\}$ , as this type of membership functions is the most frequently used due to its flexibility, and the small amount of data needed to define it. In particular,  $\mu_F(x)$  is the degree of membership of  $x$  in  $F$ . It is easy to pick out the suitable fuzzy memberships in a given problem. First, the lower bound of fuzzy memberships has to be defined, then the main property of data set must be selected. After that, a connection between this property and fuzzy memberships should be made. A trapezoidal MF and the input dataset also need to be used to build the suggested approach in order to get the intermediate output. The motivation for selecting a

trapezoidal membership function is that piecewise, it is linear and trapezoidal; hence allowing the expression of vague information caused by descriptive events that appear to be falling ones but they are not. This is accomplished by objectively converting such events into numerical variables.

The general form of *medium* trapezoidal MF for all the fuzzy sets in  $f_x$  in terms of the degree of membership could be defined in equation 4:

$$\mu_{medium}(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{x-a}{b-a}, & \text{for } a \leq x \leq b \\ 1, & \text{for } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{for } c \leq x \leq d \\ 0, & \text{for } x > d \end{cases} \quad (4)$$

Membership values for each input value using a trapezoidal membership function are computed in equation 5:

$$f(x, a, b, c, d) = \max \left( \min \left( \frac{x-a}{b-a}, 1, \frac{d-x}{d-c} \right), 0 \right) \quad (5)$$

The x-axis of the generalized trapezoidal MF corresponding to a fuzzy set  $F$  represents the input space, whereas the y-axis shows  $\mu_F(x)$ .  $(a; 0)$ ,  $(b; 1)$ ,  $(c; 1)$ , and  $(d; 0)$  are the coordinates of the four vertex of this generalized trapezoid. When there are different fuzzy sets, the values of  $a$ ,  $b$ ,  $c$ , and  $d$  will be different. This difference also occurs with the input and output spaces.

The general form of *normal* trapezoidal membership functions for all the fuzzy sets in  $f_x$  in terms of the degree of membership could be defined in equation 6:

$$\mu_{normal}(x) = \begin{cases} 1, & \text{for } x \leq c \\ \frac{d-x}{d-c}, & \text{for } c \leq x \leq d \\ 0, & \text{for } x > d \end{cases} \quad (6)$$

The general form of *high* trapezoidal membership functions in terms of the degree of membership could be described in equation 7:

$$\mu_{high}(x) = \begin{cases} 0, & \text{for } x < a \\ \frac{x-a}{b-a}, & \text{for } a \leq x \leq b \\ 1, & \text{for } x \geq b \end{cases} \quad (7)$$

Table 1 summarizes the specification turning points of the membership functions considered in this work for all the fuzzy sets in  $x$  based on our experimental test to balance sensitivity and specificity.

<i>Predictors</i>	<i>Interval</i>	<i>Fuzzy set</i>	<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>
$ A_{SD} $	0 - 300	<i>normal</i>	0	0	80	95
		<i>medium</i>	80	95	145	160
		<i>high</i>	145	160	300	300
$ G_{SD} $	0 - 30	<i>normal</i>	0	0	6	9
		<i>medium</i>	6	9	12	15
		<i>high</i>	12	15	30	30
$ M_{SD} $	0 - 100	<i>normal</i>	0	0	20	25
		<i>medium</i>	20	25	30	35
		<i>high</i>	30	35	100	100

Table 1: Turning points of membership functions

In Figure 4, the memberships are drawn in a plan, and each membership is bounded with specific values as turning points based on our experimental test to balance sensitivity and specificity. When the dimension of the input space is very high, the Fuzzy SVM network will be evaluated by splitting the datasets from model training and real time data for testing and validating the fall detection. FL model will be built based on selectable membership functions and the input datasets to get the intermediate output. After that, the intermediate output will be used as an SVM input with selectable kernels to reach the decision.

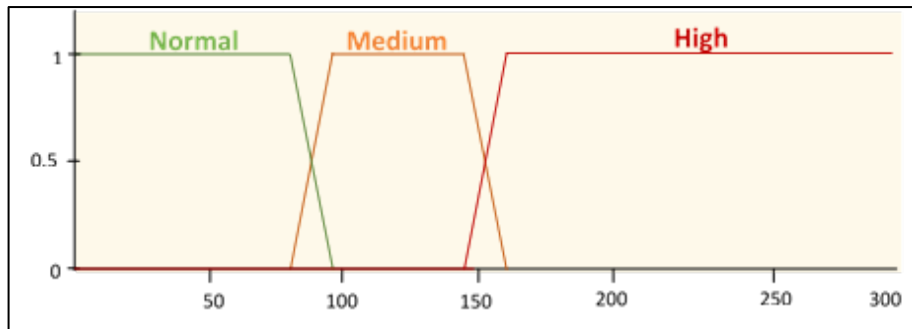


Figure 4: Membership function for acceleration input

## 5 Experimental Scenario

The prototype was installed at the waist to acquire real-time motion data, as this region is the most suitable location since it is the most fixed point of the body and is needed to provide joint stability and to easily track any movement. For safety reasons, three volunteers were chosen to participate in a simulated falling event where they implemented the fall activities on 25 cm thick mats. These volunteers have a healthy body, are aged between 25 and 35 years, weigh between 70 and 100 kg, and are 1.68 to 1.94 m tall. Detecting falls took place indoors. Table 2 summarizes the experiment

details of activities of daily living and fall subcategories that were examined. In a total of 100 trials, each fall type was repeated more than ten times.

Event	Description
<b>FALL</b>	Falling forward while walking
	Falling backward while walking
	Falling laterally while walking
	Falling vertically while walking
<b>Activities of Daily Living</b>	Walking slowly
	Walking quickly
	Sitting
	Stumbling while walking

Table 2: Characteristics of activities for experiments.

The data set were loaded with the assistance of the MATLAB workspace (version R2018a). Three predictors were introduced to show the value of each 9-axis of the trapezoidal MF of the magnetometer, accelerometer, and gyroscope. Two classes of responses, -1 and 1, were designated to sort the non-falling and falling events, respectively. The data set file type is comma-separated values (.csv), imported in the file section from the Classification Learner Tab. Table 3 summarizes the settings of the novel FSVM fall detection method.

Name	Settings of Proposed Method
Fuzzy SVM Fall Detection Proposed Method	Number of predictors: 3
	Number of observations: 2250
	Number of classes: 2
	Response: Event
	T= readtable ('FSVM_FallDetection.csv') Trapezoidal membership function of the 9-axis and response variables is numeric indicating the presence or absence of the fall event. Mistaking a fall as "-1" has more serious repercussions than wrong positives characterized as "1."

Table 3: Settings of FSVM proposed method

To do this analysis, we propose the following workflow in Figure 5: (i) selecting and loading the data, (ii) choosing a classifier option, (iii) training the classifier, (iv) accessing the classifier performance, (v) and exporting the classifier.

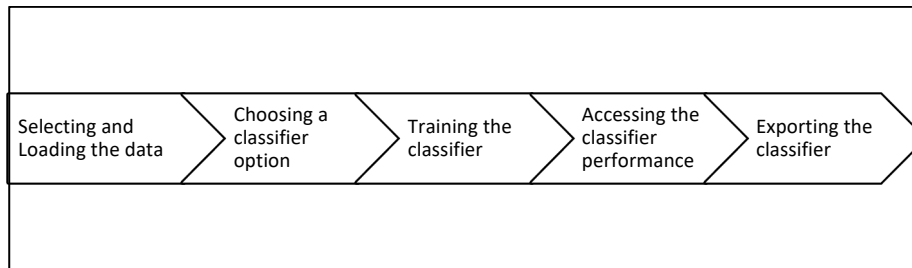


Figure 5: Training classification process

The default option is 5-fold cross-validation to partition the data set into 5 folds. This helps protect against overfitting and examines the predictive accuracy of the fitted models. To train the model, discover the data, and assess the results set, a classification learner was used to categorize the dataset using all present SVMs classifier in the supervised ML model. This was done by providing a known input data set (2250 observations) in addition to known replies to the data that comprise two classes: -1 indicating data points of the non-falling type, and +1 indicating data points of the falling type.

The kernel function of SVM was specified as:

- Medium Gaussian or RBF kernel
- Quadratic kernel
- Cubic kernel

The kernel scale mode was selected by different kernel scales to tune the SVM classifier in the Gaussian model type. In this work, the box constraint level was adjusted with diverse values from 1 to 10 for every kernel scale, respectively. Next, the suitable kernel norm was applied to calculate the Gram matrix. For more flexibility, the box constraint parameter in the dual equations was used as a hard "box" constraint, and in primal equations, it was used as the soft-margin penalty identified as C.

Additionally, the input feature space included the 9-input parameters from accelerometer, gyroscope and magnetometer; each of which is a matrix of three vectors including the x-component, the y-component and the z-component; all as functions of time.

## 6 Experimental Results of the Proposed FSVM Method

After training classifiers models, all models were compared based on:

- (1) **Sensitivity or (True Positive Rate)**, that was used to spot falls, is demonstrated in equation 8 in the form of the number of True Positives (TP) divided by the number of True Positives (TP) and the number of False Negatives (FN):

$$\text{Sensitivity} = TP / (TP + FN) \quad (8)$$

While True Positives (TP) show the number of falls that were detected correctly, False Negatives (FN) show the number of times when a fall was not detected by the system.

- (2) **Specificity or (True Negative Rate)** is utilized to calculate the system's potential for distinguishing non-fall daily activities from fall events. The

specificity is portrayed in equation 9 in the form of the number of True Negatives (TN) divided by the number of True Negatives (TN) and False Positives (FP):

$$\text{Specificity} = TN / (TN + FP) \quad (9)$$

While True Negatives (TN) depict how many times the system successfully distinguished between non-fall events and fall events, False Positives (FP) represent the number of times the system failed to do so.

- (3) **Accuracy score**, which specifies the percentage of correctly classified observations, is calculated in equation 10 by the ratio of the number of correct predictions to the total number of predictions made for a dataset:

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN) \quad (10)$$

- (4) **Precision metric**, which evaluates the number of correct positive predictions made, is computed in equation 11 by the number of True Positives (TP) divided by the number of True Positives (TP) and False Positives (FP). However, a large number of False Positives (FP) can be indicated by low precision.

$$\text{Precision} = TP / (TP + FP) \quad (11)$$

Tables 4 and 5 compare the performance of FSVM, which has a fuzzy trapezoidal MF. We used the Gaussian RBF kernel for training and adjusted the optimal regularization parameter with diverse values from 1 to 10 for every kernel scale to get a better correct classification rate, where  $k$  denotes the number of nearest neighbor points. Table 6 compares the performance of FSVM, which has a fuzzy trapezoidal membership function with the cubic and quadratic kernel.

The experimental results of FSVM with Gaussian RBF Kernel Function ( $k = 3$ ) indicate that the percentage of accuracy is more than 99.87, the percentage of sensitivity ranges between 99.63 and 99.81, whereas the percentage of specificity and precision is 100 for  $C = 4$  till  $C = 7$ .

	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>	<i>Accuracy</i> %	<i>Sensitivity</i> %	<i>Specificity</i> %	<i>Precision</i> %
<i>C=1</i>	533	1714	1	2	99.87	99.63	99.94	99.81
<i>C=2</i>	533	1714	1	2	99.87	99.63	99.94	99.81
<i>C=3</i>	533	1714	1	2	99.87	99.63	99.94	99.81
<i>C=4</i>	533	1715	0	2	99.91	99.63	100	100
<i>C=5</i>	533	1715	0	2	99.91	99.63	100	100
<i>C=6</i>	533	1715	0	2	99.91	99.63	100	100
<i>C=7</i>	534	1715	0	1	99.96	99.81	100	100
<i>C=8</i>	534	1714	1	1	99.91	99.81	99.94	99.81
<i>C=9</i>	534	1714	1	1	99.91	99.81	99.94	99.81
<i>C=10</i>	534	1713	2	1	99.87	99.81	99.88	99.62

Table 4: Experimental results of FSVM with Gaussian RBF kernel function ( $k = 3$ )



The experimental results of FSVM with Gaussian RBF Kernel Function ( $k = 2$ ) indicate that the percentage of accuracy is 99.91 for  $C = 1$  till  $C = 4$  and 99.96 for  $C = 5$  till  $C = 10$ , the percentage of sensitivity is 99.63 for  $C = 1$  till  $C = 4$ , and 99.81 for  $C = 5$  till  $C = 10$ , whereas the percentage of specificity and precision is 100 for all.

	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>	<i>Accuracy</i> %	<i>Sensitivity</i> %	<i>Specificity</i> %	<i>Precision</i> %
<i>C = 1</i>	533	1715	0	2	99.91	99.63	100	100
<i>C = 2</i>	533	1715	0	2	99.91	99.63	100	100
<i>C = 3</i>	533	1715	0	2	99.91	99.63	100	100
<i>C = 4</i>	533	1715	0	2	99.91	99.63	100	100
<i>C = 5</i>	534	1715	0	1	99.96	99.81	100	100
<i>C = 6</i>	534	1715	0	1	99.96	99.81	100	100
<i>C = 7</i>	534	1715	0	1	99.96	99.81	100	100
<i>C = 8</i>	534	1715	0	1	99.96	99.81	100	100
<i>C = 9</i>	534	1715	0	1	99.96	99.81	100	100
<i>C = 10</i>	534	1715	0	1	99.96	99.81	100	100

Table 5: Experimental results of FSVM with Gaussian RBF Kernel function ( $k = 2$ )

The experimental results of FSVM with Cubic and Quadratic Kernel Function indicate that the percentage of accuracy is 99.96, the percentage of sensitivity is 99.81, whereas the percentage of precision and specificity is 100 for both types of SVM.

	<i>TP</i>	<i>TN</i>	<i>FP</i>	<i>FN</i>	<i>Accuracy</i> %	<i>Sensitivity</i> %	<i>Specificity</i> %	<i>Precision</i> %
<i>Cubic</i>	534	1715	0	1	99.96	99.81	100	100
<i>Quadratic</i>	534	1715	0	1	99.96	99.81	100	100

Table 6: Experimental results of FSVM with Cubic and Quadratic kernel function

It is worth mentioning here that FSVM with Gaussian RBF Kernel Function ( $k = 2$ ) and FSVM with Cubic and Quadratic Kernel Function almost have the same levels of accuracy, sensitivity, and specificity. An overall score of 100% for specificity and precision with 99.81% for sensitivity was obtained by using our method. Moreover, we obtained an overall accuracy of about 99.96% of the fall detection events.

In Table 7, the experimental results demonstrate that the FSVM with a new fuzzy membership reduced false alarms and achieved accurate fall detection with better performance in reducing the effects of outliers and a lower false positive rate than some already existing methods.

	<b>PROPOSED METHOD FSVM</b>			<b>EXISTING METHOD</b>		
	<i>Accuracy</i>	<i>Sensitivity</i>	<i>Specificity</i>	<i>Accuracy</i>	<i>Sensitivity</i>	<i>Specificity</i>
	%	%	%	%	%	%
<b>Cubic SVM</b>	99.96	99.81	100	97.64	93.88	99.02
<b>Quadratic SVM</b>	99.96	99.81	100	97.86	94.86	98.96
<b>Gaussian RBF</b>	99.96	99.81	100	97.77	94.54	98.96

Table 7: Experimental results of proposed FSVM and conventional method

On the ROC curve in Figure 6, the False Positive Rate (FPR) and the True Positive Rate (TPR) for the trained classifier are seen. The current classifier performance is seen on the marker on the plot. This marker displays the values of the (TPR) and the (FPR) for the currently selected classifier.

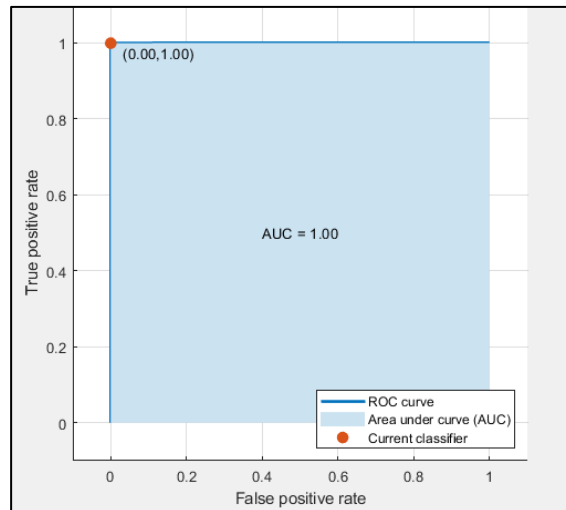


Figure 6: ROC curve of the proposed method

The right angle to the plot’s top left is the best result, having no wrongly classified points. The measure of the classifier’s overall quality is the area below the curve number. A better performance was indicated by a larger area below the curve values. After that, the best model was exported to the workspace to predict a new data set for models including Principal Component Analysis (PCA).

The evaluation of the Fuzzy SVM network included dividing the datasets from model training (75%) and testing new data (25%) for the validation of fall detection. Similar experiments for the device's prototype fall detection were conducted. In the non-fall activities, no false positive rate had been triggered where 100% specificity was

achieved for these activities. An overall accuracy of about 99.87% in detecting fall function was obtained for all activities. Furthermore, the overall sensitivity of 100% with no false negative rate obtained was achieved by implementing the proposed method.

For more evaluation, the F1 score, which is 0.99 was calculated in equation 12 to find a balance between precision and sensitivity:

$$F1\ Score = 2TP / (2TP + FP + FN). \quad (12)$$

To evaluate the proposed FSVM of prior works related to the fall detection system, Table 8 conveys an effective performance, in terms of sensitivity, specificity, and accuracy. The present study reached the maximum specificity of 100% and consequently achieved a more reliable system and more accurate results in comparison to previously existing models. This proves that the performance of the proposed method was much better as compared to previously used methods.

	[Shahzad and Kim, 2019]	[Pan et al., 2020]	[Zhang et al., 2020]	[Peçkala et al., 2022]	FSVM proposed method
<b>Algorithms</b>					
	SVM	SVM	FL	FL and Rough Logic	FL and SVM
<b>Accuracy %</b>	97.8	NA	98.36	96.9	99.96
<b>Sensitivity %</b>	99.5	96.67	100	96.2	99.81
<b>Specificity %</b>	95.2	97	95.10	87.8	100

Table 8: Comparison of Accuracy, Sensitivity, and Specificity of prior works related to fall detection system

## 7 Discussion and Conclusion

The problems raised by fall incidents are critical for people with disabilities, especially elders. Detecting falls accurately in time can reduce the severe consequences, especially since it can improve the quality of their lives. In this paper, we proposed a novel fall detection method to distinguish falling events from non-falling events by combining FL with SVM which was not explored before. In FSVM, a fuzzy membership is given to each data point. Different memberships reflect various contributions to the learning of decision-making. FSVM can effectively use human prior knowledge by the logical reasoning, association, and perception capabilities of FL to initially set fuzzy rules and then use SVM algorithms to model the data and drive training. To achieve our goal, we examined a prototype based on the Arduino platform with an IMU sensor, a power unit, and a Bluetooth 4.0 module. This formed the part fixed on the patient's body or

embedded in a wearable device that is wirelessly connected over the IoT platform. Furthermore, the use of the 9-axis acceleration to detect and report the data has several advantages that help achieve the research objective. In addition, we proposed an MA filter and SD to smooth noisy values early and to use only useful data at the signal preprocessing stage. The overall performance in terms of sensitivity, specificity, accuracy, and precision were 99.81%, 100%, 99.96%, and 100%, respectively. The present study reached the maximum specificity and precision of 100% and consequently achieved a more efficient system and more accurate results in comparison to previously existing studies. As far as the F1 score is concerned, based on the results presented in this study, we can conclude that our binary classification model achieved an F1 score of 0.99, indicating excellent precision and recall in distinguishing between positive and negative classes. Though this high F1 score suggests that our model is performing very well, it is important to note that the F1 score alone does not provide a complete picture of model performance. For this reason, we also evaluated our model according to other metrics such as accuracy, AUC-ROC, and the precision-recall curve. Accordingly, we found that our model performed well on these metrics as well. Here, it is worth noting that further evaluation of our model on hold-out test sets or through cross-validation is necessary to ensure that the high F1 score is not due to overfitting. Moreover, other factors such as model complexity, interpretability, and real-world feasibility should be considered when assessing the overall effectiveness of the model. Additionally, our study demonstrates that our binary classification model achieved excellent performance on multiple metrics, including the F1 score. While the high F1 score suggests that our model is performing well, further evaluation and consideration of other factors are necessary to draw definitive conclusions about its effectiveness.

The improvement results showed the effectiveness of the present approach by comparing it with the previous conventional methods. As some limitations of this newly proposed method regarding the stability and overall precision of fall event detection may appear with time, new variants of this method should be introduced and examined to overcome these limitations. More specifically, this can be done by including different classical machine learning techniques and/or developing more operation research methods such as heuristics and meta-heuristics.

As a future work, a closed loop between research and applied biomedical engineering will be formed. In this sense, further trials must be conducted to simulate and collect data to detect other events like spasms for different types of patients or injuries. Furthermore, the application can be tested in normal routine usage involving the elderly population. Such recorded data on cloud storage will be very useful for further refinement of the new method to be patient-friendly.

### Acknowledgements

This work is supported by Universiti Teknikal Malaysia Melaka (UTeM) under grant PJP/2021/FTMK/S01827.

### References

[Aguiar et al., 2014] Aguiar, B., Rocha, T., Silva, J., & Sousa, I. (2014). Accelerometer-based fall detection for smartphones. *IEEE MeMeA 2014 - IEEE International Symposium on Medical Measurements and Applications, Proceedings*. <https://doi.org/10.1109/MeMeA.2014.6860110>

- [Ahmad et al., 2018] Ahmad, M. A., Teredesai, A., & Eckert, C. (2018). Interpretable machine learning in healthcare. *Proceedings - 2018 IEEE International Conference on Healthcare Informatics, ICHI 2018*. <https://doi.org/10.1109/ICHI.2018.00095>
- [Aziz et al., 2017] Aziz, O., Musngi, M., Park, E. J., Mori, G., & Robinovitch, S. N. (2017). A comparison of accuracy of fall detection algorithms (threshold-based vs. machine learning) using waist-mounted tri-axial accelerometer signals from a comprehensive set of falls and non-fall trials. *Medical and Biological Engineering and Computing*, 55(1). <https://doi.org/10.1007/s11517-016-1504-y>
- [Bourke et al., 2007] Bourke, A. K., O'Brien, J. V., & Lyons, G. M. (2007). Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait and Posture*, 26(2). <https://doi.org/10.1016/j.gaitpost.2006.09.012>
- [Castañeda et al., 2022] Castañeda, M., Merayo, M. G., & Boubeta-puig, J. (2022). MODELIFY : A Model-driven Solution for Decision Making based on Fuzzy Information. *Journal of Universal Computer Science*, 28(5), 445–474. <https://doi.org/10.3897/jucs.76030>
- [Castro et al., 2007] Castro, J. L., Flores-Hidalgo, L. D., Mantas, C. J., & Puche, J. M. (2007). Extraction of fuzzy rules from support vector machines. *Fuzzy Sets and Systems*, 158(18), 2057–2077. <https://doi.org/10.1016/j.fss.2007.04.014>
- [Collado-Villaverde et al., 2020] Collado-Villaverde, A., Cobos, M., Muñoz, P., & Barrero, D. F. (2020). A simulator to support machine learning-based wearable fall detection systems. *Electronics (Switzerland)*, 9(11), 1–15. <https://doi.org/10.3390/electronics9111831>
- [De Miguel et al., 2017] De Miguel, K., Brunete, A., Hernando, M., & Gambao, E. (2017). Home camera-based fall detection system for the elderly. *Sensors (Switzerland)*, 17(12). <https://doi.org/10.3390/s17122864>
- [De Quadros et al., 2018] De Quadros, T., Lazzaretti, A. E., & Schneider, F. K. (2018). A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device. *IEEE Sensors Journal*, 18(12). <https://doi.org/10.1109/JSEN.2018.2829815>
- [Espinosa et al., 2019] Espinosa, R., Ponce, H., Gutiérrez, S., Martínez-Villaseñor, L., Brieva, J., & Moya-Albor, E. (2019). A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset. *Computers in Biology and Medicine*, 115. <https://doi.org/10.1016/j.compbiomed.2019.103520>
- [Fernández-Caballero et al., 2013] Fernández-Caballero, A., Sokolova, M. V., & Serrano-Cuerda, J. (2013). Lateral inhibition in accumulative computation and fuzzy sets for human fall pattern recognition in colour and infrared imagery. *The Scientific World Journal*, 2013. <https://doi.org/10.1155/2013/935026>
- [Gnanadesigan et al., 2022] Gnanadesigan, N. S., Dhanasegar, N., S, J., Ramachandran, M., Muthusamy, S., Thangavel, G., & Veerappan, P. (2022). An efficient empirical mode decomposition based feature extraction model for human activity recognition of elderly people using machine learning algorithms. 1–18. <https://doi.org/https://doi.org/10.21203/rs.3.rs-1899339/v1>
- [Harum et al., 2018] Harum, N., Abidin, Z. Z., Shah, W., & Hassan, A. (2018). IMPLEMENTATION OF SMART MONITORING SYSTEM WITH FALL DETECTOR FOR ELDERLY USING IOT TECHNOLOGY. *International Journal of Computing*, 17(4), 243–249.
- [Hussain et al., 2019] Hussain, F., Hussain, F., Ehatisham-Ul-Haq, M., & Azam, M. A. (2019). Activity-Aware Fall Detection and Recognition Based on Wearable Sensors. *IEEE Sensors Journal*, 19(12). <https://doi.org/10.1109/JSEN.2019.2898891>

- [Kakarash et al., 2020] Kakarash, Z. A., Karim, S. H. T., & Mohammadi, M. (2020). Fall Detection Using Neural Network Based on Internet of Things Streaming Data. *UHD Journal of Science and Technology*, 4(2), 91–98. <https://doi.org/10.21928/uhdjst.v4n2y2020.pp91-98>
- [Kangas et al., 2008] Kangas, M., Konttila, A., Lindgren, P., Winblad, I., & Jämsä, T. (2008). Comparison of low-complexity fall detection algorithms for body attached accelerometers. *Gait and Posture*, 28(2). <https://doi.org/10.1016/j.gaitpost.2008.01.003>
- [Kchouri et al., 2022] Kchouri, M., Harum, N., Obeid, A., & Hazimeh, H. (2022). Fuzzy Support Vector Machine based Fall Detection Method for Traumatic Brain Injuries. *International Journal of Advanced Computer Science and Applications*, 13(11), 302–314.
- [Kuncan et al., 2022] Kuncan, F., Kaya, Y., Yiner, Z., & Kaya, M. (2022). A new approach for physical human activity recognition from sensor signals based on motif patterns and long-short term memory. *Biomedical Signal Processing and Control*, 78, 103963. <https://doi.org/10.1016/j.bspc.2022.103963>
- [Kuncan et al., 2019] Kuncan, F., Kaya, Y., & Kuncan, M. (2019). A novel approach for activity recognition with down-sampling 1D local binary pattern features. *Advances in Electrical and Computer Engineering*, 19(1). <https://doi.org/10.4316/AECE.2019.01005>
- [Lee and Kim, 2016] Lee, H., & Kim, S. (2016). Black-Box Classifier Interpretation Using Decision Tree and Fuzzy Logic-Based Classifier Implementation. *The International Journal of Fuzzy Logic and Intelligent Systems*, 16(1), 27–35. <https://doi.org/10.5391/ijfis.2016.16.1.27>
- [Lin and Wang, 2002] Lin, C. F., & Wang, S. De. (2002). Fuzzy support vector machines. *IEEE Transactions on Neural Networks*, 13(2), 464–471. <https://doi.org/10.1109/72.991432>
- [Moulik and Majumdar, 2019] Moulik, S., & Majumdar, S. (2019). FallSense: An Automatic Fall Detection and Alarm Generation System in IoT-Enabled Environment. *IEEE Sensors Journal*, 19(19). <https://doi.org/10.1109/JSEN.2018.2880739>
- [Pan et al., 2020] Pan, D., Liu, H., Qu, D., & Zhang, Z. (2020). Human Falling Detection Algorithm Based on Multisensor Data Fusion with SVM. *Mobile Information Systems*, 2020. <https://doi.org/10.1155/2020/8826088>
- [Pekala et al., 2022] Pekala, B., Mroczek, T., Gil, D., & Kepski, M. (2022). Application of Fuzzy and Rough Logic to Posture Recognition in Fall Detection System. *Sensors*, 22(4). <https://doi.org/10.3390/s22041602>
- [Rahmani et al., 2021] Rahmani, A. M., Yousefpoor, E., Yousefpoor, M. S., Mehmood, Z., Haider, A., Hosseinzadeh, and challenges. In *Mathematics* (Vol. 9, Issue 22). <https://doi.org/10.3390/math9222970>
- [Ren and Peng, 2019] Ren, L., & Peng, Y. (2019). Research of fall detection and fall prevention technologies: A systematic review. In *IEEE Access* (Vol. 7). <https://doi.org/10.1109/ACCESS.2019.2922708>
- [Roshni Thanka et al., 2022] Roshni Thanka, M., Juliet, S., Bijolin Edwin, E., & Raahul John, R. (2022). Deep Learning-Based Big Data Analytics Model for Activity Monitoring of Elderly People. *Lecture Notes in Electrical Engineering*, 589–599. [https://doi.org/10.1007/978-981-19-2177-3\\_55](https://doi.org/10.1007/978-981-19-2177-3_55)
- [Safeea and Neto, 2019] Safeea, M., & Neto, P. (2019). Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction. *Robotics and Computer-Integrated Manufacturing*, 58. <https://doi.org/10.1016/j.rcim.2019.01.008>

- [Sathyanarayana et al., 2018] Sathyanarayana, S., Satzoda, R. K., Sathyanarayana, S., & Thambipillai, S. (2018). Vision-based patient monitoring: a comprehensive review of algorithms and technologies. *Journal of Ambient Intelligence and Humanized Computing*, 9(2). <https://doi.org/10.1007/s12652-015-0328-1>
- [Shahzad and Kim, 2019] Shahzad, A., & Kim, K. (2019). FallDroid: An Automated Smart-Phone-Based Fall Detection System Using Multiple Kernel Learning. *IEEE Transactions on Industrial Informatics*, 15(1), 35–44. <https://doi.org/10.1109/TII.2018.2839749>
- [Vapnik, 1999] Vapnik, V. N. (1999). An overview of statistical learning theory. In *IEEE Transactions on Neural Networks* (Vol. 10, Issue 5). <https://doi.org/10.1109/72.788640>
- [Worrakulpanit and Samanpiboon, 2014] Worrakulpanit, N., & Samanpiboon, P. (2014). Human Fall Detection Using Standard Deviation of C-Motion Method. *Journal of Automation and Control Engineering*, 2(4), 388–391. <https://doi.org/10.12720/joace.2.4.388-391>
- [Wu et al., 2020] Wu, T., Gu, Y., Chen, Y., Wang, J., & Zhang, S. (2020). A Mobile Cloud Collaboration Fall Detection System Based on Ensemble Learning. *ASSETS 2020 - 22nd International ACM SIGACCESS Conference on Computers and Accessibility*. <https://doi.org/10.1145/3373625.3417010>
- [Yacchirema et al., 2018] Yacchirema, D., De Puga, J. S., Palau, C., & Esteve, M. (2018). Fall detection system for elderly people using IoT and Big Data. *Procedia Computer Science*, 130, 603–610. <https://doi.org/10.1016/j.procs.2018.04.110>
- [Yang et al., 2021] Yang, C., Oh, S. K., Yang, B., Pedrycz, W., & Fu, Z. W. (2021). Fuzzy quasi-linear SVM classifier: Design and analysis. *Fuzzy Sets and Systems*, 413, 42–63. <https://doi.org/10.1016/j.fss.2020.05.010>
- [Yodpijit et al., 2017] Yodpijit, N., Sittiwanchai, T., & Jongprasithporn, M. (2017). The development of Artificial Neural Networks (ANN) for falls detection. *2017 3rd International Conference on Control, Automation and Robotics, ICCAR 2017*. <https://doi.org/10.1109/ICCAR.2017.7942757>
- [Zadeh, 2008] Zadeh, L. A. (2008). Is there a need for fuzzy logic? *Information Sciences*, 178(13). <https://doi.org/10.1016/j.ins.2008.02.012>
- [Zhang et al., 2020] Zhang, H., Alrifai, M., Zhou, K., & Hu, H. (2020). A novel fuzzy logic algorithm for accurate fall detection of smart wristband. *Transactions of the Institute of Measurement and Control*, 42(4), 786–794. <https://doi.org/10.1177/0142331219881578>
- [Zhen et al., 2016] Zhen, T., Mao, L., Wang, J., & Gao, Q. (2016). Wearable preimpact fall detector using SVM. *Proceedings of the International Conference on Sensing Technology, ICST*. <https://doi.org/10.1109/ICSensT.2016.7796223>