Fuzzy Support Vector Machine based Fall Detection Method for Traumatic Brain Injuries

A New Systematic Approach of Combining Fuzzy Logic with Support Vector Machine to Achieve Higher Accuracy in Fall Detection System

Mohammad Kchouri[1], Norharyati Harum[2]*, Ali Obeid[3], Hussein Hazimeh[4]
Fakulti Teknologi Maklumat Dan Komunikasi, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia[1, 2]
Psymeon CTO, Paris, France[3]
Department of Computer Science, Lebanese University, Beirut, Lebanon[4]

***Abstract*—Falling is a major health issue that can lead to both physical and mental injuries. Detecting falls accurately can reduce the severe effects and improve the quality of life for disabled people. Therefore, it is critical to develop a smart fall detection system. Many approaches have been proposed in wearable-based systems. In these approaches, machine learning techniques have been conducted to provide automatic classification and to improve accuracy. One of the most commonly used algorithms is Support Vector Machine (SVM). However, classical SVM can neither use prior knowledge to process accurate classifications nor solve problems characterized by ambiguity. More specifically, some values of falls are inaccurate and similar to the features of normal activities, which can also greatly impact the performance of the learning ability of SVMs. Hence, it became necessary to look for an effective fall detection method based on a combination of Fuzzy Logic (FL) and SVM algorithms so as to reduce false positive alarms and improve accuracy. In this paper, various training data are assigned to the corresponding membership degrees. Some data points with a high chance of falling are assigned a high degree of membership, yielding a high contribution for SVM decision-making. This does not only achieve accurate fall detection, but also reduces the hesitation in labeling the outcomes and improves the heuristic transparency of the SVM. The experimental results achieved 100% specificity and precision, with an overall accuracy of 99.96%. Consequently, the experiment proved to be effective and yielded better results than the conventional approaches.**

Keywords—Fall detection; fuzzy logic; SVM; traumatic brain injuries; wearable sensor

I. Introduction

The main goal of this paper is to introduce and propose a new fall detection method for traumatically brain-injured people. The goal is to suggest a highly accurate method that uses the advantages of both Fuzzy Logic and SVM. The main reasons behind the idea of integrating these two latter methods are their well-known limitations as standalone techniques, with SVM considered a "black box" and Fuzzy logic being especially limited for nonlinear problem solving. Falling is one of the major life-threatening problems faced by people with physical disabilities. According to World Health Organization research, falls are the second most common cause of injury-related death worldwide. Nationally, there was a 53% increase in the number of total deaths due to falls from 2000 to 2019 [1]. Health systems are significantly impacted by falls. The quality of life of elderly persons can also be significantly impacted out of the fear of falling.

An Ambient Intelligent (AI) environment can improve the lifestyle of the disabled by using different sensor technologies. With such technologies, environments might become sensitive, adaptive, and responsive to people's presence for the sake of supporting them to live independently in the environment they prefer. One important aim of assistive technology is to allow disabled people to stay in their homes as long as possible without changing their lifestyles. Smart fall detection can offer support to people with special needs, enabling them to live actively and independently both at home and in their communities. This improves the quality of their lives on the one hand, and reduces costs for their families and the entire society on the other hand.

Currently, available techniques that are used to design fall detection systems are classified into two main categories based on their sensor type: one is an ambient-based fall detection system whereas the other relies on a wearable-based fall detection system [2]. Ambient-based approaches use ambiance sensors, including acoustic sensors, vibration sensors, pressure sensors, and infrared sensors for detecting a fall event [3,4,5]. They also use single or multiple cameras in an indoor environment to track a person's movements and body shape while falling [6,7,8]. One main drawback of ambient sensors is that they limit falls to only those detected in a pre-set area, and this does not seem suitable for people's mobility.

Wearable sensor-based fall detection methods track the user's body motion using embedded sensors such as accelerometers. These methods can detect a fall when the person is wearing the sensor anywhere and at any time. Wearable sensors have several advantages, including low cost, low power consumption, and ease of use. As a result, they are commonly used to detect human falls. However, those with a single accelerometer are insufficient to offer a robust system and are susceptible to false positive fall detection; which reduces system accuracy [9].

*Corresponding Author.

Recently, a fusion system in which the process of combining multiple data sources can produce more robust measurement and accurate detection was developed. In this context, fall detection based on Inertial Measurement Units (IMUs) sensors, which are composed of an accelerometer, gyroscope, and magnetometer can be attached to the body. This fall detection technique is of special interest due to the fact that it is undetectable, widely available, inexpensive, and has a low power consumption [10,11]. Additionally, this technique can offer complementary information about the activity performed [12]. The IMU has been widely used for data collection and for differentiating a fall event from Activities of Daily Living (ADLs) based on two main approaches which are threshold-based or machine learning classification algorithms.

In a threshold-based approach, the system notifies that a fall happens if the real-time sensor data surpasses the given threshold values after comparing them. In particular, a low threshold value brings out false alarms whereas a high threshold value causes large fall missing issues. The authors in [13] show that machine learning algorithms can effectively enhance the system's performance in comparison to the threshold-based method. In machine learning approach, impressive results can be obtained using various classifiers [12]**.** An SVM is one of the most popular algorithms used in supervised machine learning for activities of recognition and classification [5]. However, labels for the dataset are needed for the employment of SVM algorithms for fall detection. In more detail, some training points of falls that are uneven and similar to being mixed with the features of ADLs can significantly impact the performance of the learning ability of SVMs. Here, it is worth mentioning that classical SVMs can never solve inaccurate and ambiguous problems. FL has the ability to mimic the human way of thinking to productively use reasoning methods that are uneven and not precise, and it has been verified to be effective in reducing the false fall detection rate [9].

SVM as a standalone algorithm, just like FL, has its own advantages and disadvantages. Merging both methods by making use of the best traits of each algorithm will certainly yield outstanding results in some applications, as in the case of fall detection. The specific contributions of this study could be summarized as follows:

- Identifying the types of disabilities that require smart fall detection as this can help in developing the desired device.
- Proposing a new fall detection method that takes advantage of both FL and classical SVM. Our proposed method uses the logical reasoning of FL by Fuzzification of all input values of the training data into fuzzy membership functions to obtain the intermediate output. Different memberships reflect various contributions to the learning of decision-making. A high degree of membership of data points, which have a significantly high chance of falling, provides a significant contribution to SVM decision-making. As a result of the functions of the fuzzy membership, an SVM output is properly produced for modeling data and driving training. This network combination reduces false positive alarms to obtain more accurate fall detection and it achieves better generalization ability that was motivated to perform much better than conventional SVM methods on smart fall detection.

The rest of this paper is structured as follows: Section II provides a review of existing papers related to fall detection systems. Section III presents a brief review of classical SVM and Fuzzy SVM basic theory. Section IV identifies the types of disabilities that require smart fall detection. Section V describes the hardware design to collect data and highlights the proposed fall detection method with a classification process. Section VI details the experimental settings, the performance evaluation, the experimental results, and the analysis for the various scenarios. Section VII summarizes the conclusion with a comment on future work.

II. Related Work

Generally, the framework of automatic fall detection systems using wearable sensors consists of the collection of sensor data, a fall detection algorithm, and an emergency alarm. Data collected by the accelerometer and gyroscope are transmitted to a microcontroller to be processed to differentiate a fall from an ADL. Recently, there has been a growing interest in identifying and detecting fall events using IMU sensors [14]. 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 as well as the body orientation before, during, and following the fall. Nevertheless, the location of the sensor can influence the performance of the system. Thus, numerous studies related to the topic of optimal sensor placement were conducted. The waist, wrist, trunk, thigh, back, ankle, foot, neck, and head represent the most common wearing positions. The authors in [15] studied fall detection by placing accelerometers on the subject's head, waist, and wrist. They reported that the most efficient positions are the waist and the head in contrast to the wrist which is not. The authors in [16] placed sensors on the trunk and thigh and reported the trunk as a better position. The authors in [17] identified that the waist location utilizing a single sensor was a suitable placement after evaluating single IMU sensors deployed at several places in the body. In conclusion, sensor location is an important factor in developing wearable sensor-based fall detection systems. Furthermore, the waist could be the best option for a wearable sensor-based fall detection system.

The classification algorithm is applied to classify ADL and several fall events. A wearable-based fall detection algorithm can be categorized into two approaches, namely: threshold-based and machine learning. Threshold-based approaches use single or multiple threshold values that can be adjusted automatically depending on motion history to classify events [15,16,18]. Due to the low computational complexity, current fall detection studies have widely used the threshold-based method. However, a high threshold must be set in order to obtain a highly accurate system. Hence, 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. This is why a suitable threshold should be set to avoid any problems.

Machine learning is a field of Artificial Intelligence. It explores how to use computers to imitate human learning activities [19]. It uses learning algorithms to extract features from raw data in order to gain new abilities, recognize current knowledge, and constantly improve performance and achievement [20]. Data processing includes pre-processing, smoothing data, and data reduction methods where the data are acquired from the sensor and then processed. This is followed by the classification stage, which either provides a prediction or a decision at the end of the process. Learning methods can be categorized into two main groups: supervised and unsupervised learning. The algorithm in supervised learning is trained using a labeled input dataset. In the case of unsupervised algorithms, there are no explicit labels associated with the training dataset, which saves time during processing [21]. Examples of algorithms used in fall detection experiments include Hidden Markov Model (HMM), K-Nearest Neighbors (K-NN), Random Forest (RF), SVM, Decision Tree, Linear Regression, Naïve Bayes, Fuzzy Inference System (FIS), and Artificial Neural Network (ANN), which have achieved significant success in detecting falls and classifying fall events from ADLs [22].

Based on the preceding, [23] successfully differentiate falls from ADLs using six machine learning classifiers, which are the K-NN classifier, Least Squares Method (LSM), SVM, Bayesian decision making (BDM), Dynamic Time Warping (DTW), and ANNs. They achieved the greatest results with the K-NN classifier and LSM; with sensitivity, specificity, and accuracy all above 99%. The authors in [24] attained the best accuracy in fall detection using the K-NNs classifier and the highest accuracy in distinguishing various falling activities using the RF classifier. However, [25] compared the applicability of RF and SVM in the development of wearable intelligent devices. The obtained results show that SVM is more suitable for the development of wearable intelligent devices. The authors in [26] used the K-NN and SVM algorithms for classifying the fused accelerometer and gyroscope data collected from smartphone sensors. They reported a classification performance of 98.32% for SVM and 97.42% for K-NN. ANNs have been greatly improved in recent times [27]. This method regularly outperforms classic machine learning 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 can be affected by too much disorienting information.

Study [13] evaluated the accuracy of these two approaches, which are threshold values and five machine learning algorithms. In fact, they concluded that five machine learning algorithms' gross production was greater than the overall performance of five algorithms based on the threshold. In addition, SVM-based classification has outperformed the five machine learning in terms of sensitivity and specificity. Since they are highly accurate in comparison with threshold-based fall detection methods, machine learning based fall detection algorithms are nowadays being widely used. The authors in [28] achieved the best accuracy performance of 99%, indicating that the system's performance in comparison to the threshold-based method can be effectively enhanced. Decision-making based on machine learning algorithms, on the other hand ensures high rates of true positives.

A. SVM-Based Fall Detection

SVM is a powerful tool in machine learning for classifying data with good generalization ability that is less computationally intensive than other algorithms like artificial neural networks, decision trees, and Bayesian networks due to their high accuracy, elegance, mathematical practicability, and simple geometric interpretation. In addition, they do not require a lot of training data to prevent overfitting [29,30].

In SVM-based fall detection, distinguishing between falling activities and non-falling ADLs is possible. For instance, [31] used a hyperplane of SVM as the separating plane to replace the traditional threshold method for the detection of falling ADLs. They used the Gaussian radial basis function to construct the kernel function with the cost parameter tuned where the constant (C) is the regularization parameter in the SVM. When adjusted, a balance between margin maximization and classification is realized [32]. The results showed that the SVM method is better than the threshold-based algorithm when the parameters of falling and non-falling ADLs are very close. The authors in [33] extracted features from data collected by the Kinect sensor followed by fall recognition by using an SVM algorithm. In [3], the authors proposed a Multi-Feature Semi-Supervised SVM framework for human fall detection to specifically handle the human fall classification problem where the Radial Basis Function (RBF) classifier is selected in SVM training on extracted features from the training samples. The authors in [34] proposed a system that detects human falls by using the audio signal from a microphone. Their system was designed in a way that models each fall or noise segment by means of a Gaussian Mixture Model (GMM). Then, the SVM classifier would be employed to classify audio segments into falls and different types of noise. Study [35] collected the body's acceleration and rotational angle data in the wearable terminal to execute the SVM algorithm. As such, the hyperplane, which separates fall events from ADL events, was introduced. RBF kernel function which allows nonlinear mapping in this model was selected with a penalty parameter factor (C) that was adjustable to obtain the largest gap distance. The authors in [36] employ multiple kernels of learning to distinguish difficult fall-like events. The classification performance kept improving until it became constant at a certain point when the tuned parameter (C) was increased. Nonetheless, the number of selected kernels also expanded with (C), which raised the computational cost. Another model proposed by [10] showed that an SVM classifier provides the best performance metrics when trained on a fall dataset containing simulated falls and when cross-validated with real-world falls. The classifier appears to be appropriate for further evaluation concerning real-world applications due to robust results and high accuracy, sensitivity, and specificity. The authors in [25] found that SVM is more suitable for fall detection algorithm based on multi-sensor data fusion.

The SVM can accomplish great performance in the classification and calculation processes. This can be achieved through the use of a relatively low amount of the provided learning data. However, they cannot give a comprehensible representation of where a produced output has been attained. They are similar to black-box models due to their complex structure, numerous parameters, and excessive abstraction [37]. With classical SVMs, the experimenters can never depend on their prior knowledge to process accurate classifications or solve inaccurate and ambiguous problems, which can also greatly impact the performance of the learning ability of SVMs.

B. Fuzzy Logic-based Fall Detection

Fuzzy systems are systems in which the variables have domain fuzzy sets. Such systems allow the encoding of structured, empirical, heuristic, or linguistic knowledge in a numerical framework. As opposed to conventional logic, FL is based on the mathematical theory of fuzzy sets, which mimics human thought and tries to reflect reality while taking into account all outcomes between 0 and 1 [38], where 1 means absolute truth and 0 means absolute falsehood. The FL consists of three main parts. The first part is Fuzzification which allows the conversion of crisp values into fuzzy membership functions. The second part is the fuzzy inference aggregation which contains all the rules and if-then conditions to control the decision-making system. In this phase, the Mamdani method, which requires finding the centroid of a two-dimensional shape by integrating it across a continuously varying function, is the most commonly used technique. The third part is the defuzzification process that converts the fuzzy sets into crisp values. Despite the availability of several defuzzification methods, the centroid technique remains the most popular among all of them [39,40].

In Fuzzy Logic-based fall detection, [41] confirmed that Fuzzy is capable of detecting a fall from real-time data as it requires minimum hardware and software specifications. Reusing the existing data, balancing the load amongst FLS devices, and cost-efficiency are some of the advantages offered by the FLS architecture to introduce flexible and smooth decisions. In their proposed method, they "fuzzify" each input value as a function of fuzzy membership, where each input contains three linguistic values: low, medium, and high. Every membership is classified as a turning point with different values. To perform their experiment, the researchers created nine rules to identify whether a fall occurred or not. Finally, they transformed a fuzzy output set into a crisp value in the defuzzification phase. The authors in [42] used FL to identify the range and type of fall, which can include the position before fall, fall direction, fall velocity, and post-fall inactivity. The authors in [43] looked beyond the traditional threshold-based approaches and implemented a fuzzy inference technique for precise decision making. They fused the data from multiple sensors and generated a value between 0 and 1, which implies the chance of a fall; thus reducing the number of false alarms. The authors in [44] initiated a new FL algorithm, which is worn on wrists to detect falls and reduce the number of false alarms. 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 is followed 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 the AND-output rule in their fall detection algorithm and a weighted average formula in defuzzification. Furthermore, since the Mamdani approach is generally accepted for the development of expert knowledge, this helps one to explain the ability in a more perceptive, human-like manner. However, this technique is not computationally effective and can be expensive due to calculating a two-dimensional form by adding it up or combining it more accurately through a function that change continuously.

III. SVM and Fuzzy SVM Basic Theory

In this section, we briefly review the basics of the theory of SVM in classification problems and fuzzy support vector machines, which are discussed by [45,46,47.48].

A. SVMs Theory

SVM is a robust classification method that was developed by Vladimir Vapnik and aims to construct a decision function that separates the data in the input space into different classes. The basis of this method is minimizing the structural risk method in order to reduce the error [32]. An optimal hyperplane must be found through maximizing the margin between classes in input space. Nonetheless, the samples close to the hyperplane are called support vectors [49].

In the input space, it is assumed that the patterns are drawn by the training points $\{x_1,\ldots,x_n\}$. If this input data are linearly separable, the hyperplane that produces the separation could be described as: $w^T x_i + b = 0$ where x is an input vector, $w = (w_1, w_2, \ldots, w_d)$ is a weight vector, d is the number of the input variables, and b is a bias that will determine the distance between the hyperplane and the origin. It can classify the points using the following equation:

$$\begin{cases} w^T x_i + b \geq 0 \; for \; y = +1 \\ w^T x_i + b \leq 0 \; for \; y = -1 \end{cases} \tag{1}$$

where $y_i \in \{-1,+1\}$ is the binary class label for a new point x_i, this output enables x_i to be classified as belonging to one of the two classes.

The goal of the support vector machine is to build the optimal hyperplane that maximizes the distance between the closest points of each class and the separation [50]**.** This is presented by solving the linear problem:

$$\begin{cases} \min \frac{1}{2} \|w\|^2 \\ y(w^T x_i + b) \geq 1, i = 1, 2, \ldots, N \end{cases} \tag{2}$$

To deal with data that are not separable cases, slack variables ε_i are introduced. This represents the misclassified sample of the corresponding margin hyperplane, where $i \in \{1,2,3,\ldots,N\}$ is an upper bound of the number of error [48]. Thus, the optimal hyperplane in a nonlinear space can be determined by:

$$minimize\ 0.5\|w\|^2 + c\sum_{i=1}^{N}\varepsilon_i$$

subject to:

$$\begin{cases} y_i(w^T x_i\ +\ b) \geq 1-\varepsilon_i\ , i=1,2,..,N \\ \qquad \varepsilon_i\ \geq 0\ , i=1,2,..,N \end{cases} \tag{3}$$

In this sense, the adjustable parameter (C) plays a major role in maximizing the margin and carefully tuning the number of misclassifications. If (C) is bigger, it makes the training of SVM fewer misclassifications and it narrows the margin. In contrast, if (C) decreases, it makes SVM disregard more training points and it widens the margin.

However, the search for a suitable hyperplane in an input space is too restrictive to be of practical use when the samples are linearly non-separable. Thus, SVM techniques utilize a group of mathematical functions known as the kernel that satisfies Mercer's theorem [51]. This can be applied to map data relevant to the feature of higher dimensions, hence seeking an optimal separating hyperplane in the feature space [29].

The main approach is to find the function that performs the mapping from the input to the feature space. By introducing the vector of Lagrange multipliers [31], the nonlinear separating hyperplane can be found as the solution of:

$$\max\sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\,\alpha_j y_i y_j K\left(x_i.x_j\right)$$

subject to:

$$\begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0, \forall\ i \\ \ 0\ \leq\ \alpha_i\ \leq\ c \end{cases} \tag{4}$$

where $k\ (x, x_i) = \langle\emptyset(x), \emptyset(x_i)\rangle$ is the dot products of the corresponding feature vectors into high dimensional space [51].

Four common types in kernel at the SVM algorithm are linear, polynomial, Gaussian RBF, and sigmoid kernel where each kernel function has a particular parameter that must be optimized to obtain the best result performance [49]. Three kernel types will be used in the experiments to compare their results to those of our proposed method. These kernel types are:

- Linear kernel: $k(x, x_i) = \langle x, x_i\rangle$.
- Polynomial kernel: $k(x, x_i) = (\langle x, x_i\rangle + 1)^d$ with degree d.
- RBF kernel: $k(x, x_i) = e^{\frac{-(\|\vec{x}-\vec{x_i}\|)^2}{2\sigma^2}}$ with adjustable width parameter σ.

B. Fuzzy SVMs Theory

SVM is a powerful tool for classifying data points that are assumed to belong to the one and only class [52]. However, as discussed previously, the effects of the training points are different. Especially for fall detection, classical SVM can neither use prior knowledge to process accurate classifications nor solve problems characterized by inaccuracy and ambiguity. In more specific terms, some values of falls are inaccurate and similar to the features of normal activities, which can also greatly impact the performance of the learning ability of SVMs. Furthermore, some training points no longer exactly belong to one of the two classes. For example, 80% belong to the class of falls and 20% to the class of ADLs. These points are critical and may cause false positive alarms which reduce the accuracy of the system.

To do that, the points that have a low potential for falling or normal activity will be assigned to lower membership functions. Otherwise, a high chance of falls will be assigned with a high degree of membership function. In this sense, each training point is fuzzified into a membership function. This fuzzy membership $\mu_i \in \{0,1\}$ is considered to be the attitude of the corresponding training point toward one class in the classification problem whereas the value $(1-\mu_i)$ is considered to be the attitude of meaninglessness. So, the idea of SVM will be expanded and combined with a fuzzy membership function to make it a Fuzzy SVM.

The term μ_i is introduced as a membership vector for each training point x_i. Thus, the optimal hyperplane problem is then regarded as the solution to:

$$min\ 0.5\|w\|^2 + c\sum_{i=1}^{N}\boldsymbol{\mu_i}\ \boldsymbol{\varepsilon_i}$$

subject to:

$$\begin{cases} y_i(w^T x_i\ +\ \mathrm{b}) \geq 1-\varepsilon_i\ , i=1,2,..,N \\ \qquad \varepsilon_i\ \geq 0\ , i=1,2,..,N \end{cases} \tag{5}$$

where the term $\mu_i\ \varepsilon_i$ is a measure of error with different weighting. Hence, it can be noticed that the effect of misclassified parameter ε_i will be reduced when the membership functions μ_i are smaller. In this case, the training point x_i is treated as less important in the training. By applying Lagrange multipliers, the above problem is reformulated as:

$$\max Q(x) = \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\,\alpha_j y_i y_j K\left(x_i.x_j\right)$$

subject to:

$$\begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0, \forall\ i \\ \ 0\ \leq\ \alpha_i\ \leq\ s_i c \end{cases} \tag{6}$$

Solving problem (6), dual of (5), is the same for classical SVM with a slight difference. As a consequence, this is the basic theory of Fuzzy SVM.

IV. Clinical Study

A clinical study was conducted to identify the disabled group that needs fall detection. Patients and specialists in healthcare centers were interviewed to collect data on health and welfare. The interviewer was a physiotherapy officer in Lebanon, and the interviewees were people with special needs living in the sample households. A survey was distributed to a group of physically disabled people who suffered traumatic brain injuries. Each case had a different kind of disability, such as Cerebral Vascular Accident (CVA), Cerebral Palsy (CP), Meningitis, or Guillain-Barre. This survey included information on the installation of sensors based on the preference of disabled people. As a result, patients with brain injuries were asked to fill out a survey with a special focus on the duration of their disability, the reason for disability, and the number of falls in the last 6–12 months.

In Fig. 1, it is illustrated that 65% of the studied cases were patients with CVA, 25% were patients with CP, 6% were patients with Meningitis, and 4% were patients with Guillain-Barre. In Fig. 2, which studies the number of falls in each case, it is revealed that CVA cases aged above 55 ran into 83 falls, which is 37%, while CVA cases aged under 55 ran into 60 falls, ranking a lower percentage, which is 27%. In general, CVAs ranked 64%, indicating the highest percentage of falls. In addition, CP cases marked 59 falls, indicating the second-highest percentage of falls, which is 26%, whereas other cases marked 24 falls, showing the lowest percentage of falls, which is 10%. Thus, it was noticed that people with disabilities related to traumatic brain injuries such as CVA, CP, Guillain-Barre, and Meningitis have a high frequency of falls. Each respondent gave a different answer regarding the position of the fall detection sensor.

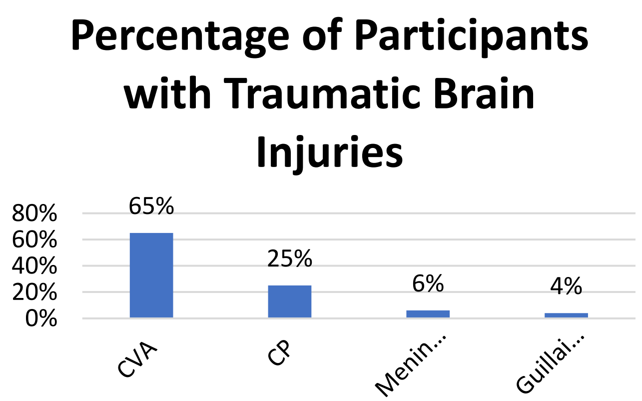

Fig. 1. Percentage of Participants with Traumatic Brain Injuries.

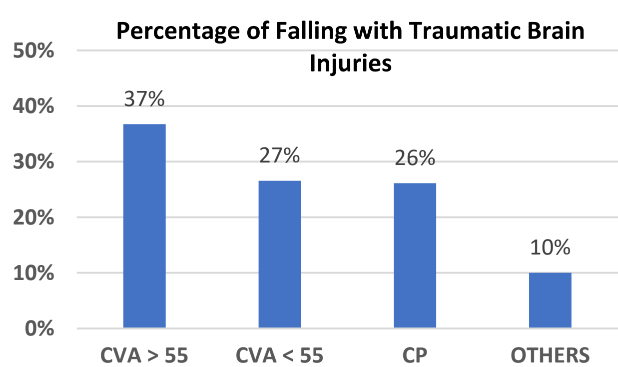

Fig. 2. Percentage of Falling for Participants with Traumatic Brain Injuries.

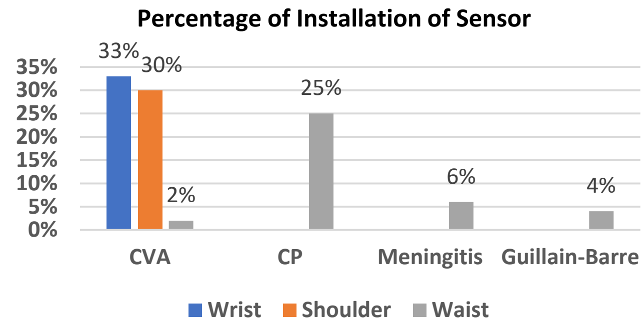

Fig. 3. Preferred Position of the Sensor.

Fig. 3 shows that 37% of the CP, Meningitis, and Guillain-Barre cases suggested placing the sensor on the waist in a way that allows it to be balanced right between the upper part of the body and its lower part. On the other hand, CVA cases had different answers, each according to their age. Those aged 55 and above (33% of the CVA cases) wanted to install the sensor on the wrist to quickly detect a fall, while people under the age of 55 (30%) wanted to install it on the shoulder.

In this paper, we are going to install a sensor for brain injury cases on the waist since it is the most fixed point of the body and is needed to maintain joint stability. Thus, it can track any movement easily.

V. Proposed Method

A prototype of the wearable device is designed using hardware in the form of a small-sized IMU sensor. The digital output of a 9-axis motion tracker by the IMU module is accessible by the I2C communication protocol (Inter-Integrated Circuit). This module is based on the MPU-9250, which assists in detecting activity changes, determining the slope of the object on which the sensor is mounted, generating acceleration, and expressing the angle and rotation about each axis in 3D space. It also achieves targets with low power consumption and robustness during the short duration of dynamic accelerations. Data gathered by the sensor are defined by an Arduino Uno microcontroller that operates at a voltage of 3.3/5 volts and is used 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. The Baud rate is set to 9600 bits per second for serial communication between the Arduino board and the MPU-9250. The acquired data are transferred via a low-energy Bluetooth interface to the classification part. The Bluetooth module HC-05 is used for wireless communication between the Arduino Uno and MPU-9250. Bluetooth technology is a suitable choice for a lot of applications in daily life as it provides a reliable connection and low power consumption [53].

This paper suggests a new method to detect falls through the effective combination of FL and SVM (Fuzzy SVM). The input matrix consists of a 9-axis accelerometer, gyroscope, and magnetometer to collect multiple human body data points at the same time, including human body acceleration, rotational velocity, and displacement along the three directions. The next step involves smoothing the data collected and computing the standard deviation of each of the nine axes from the IMU sensor in 3D space. After that, the magnitude of standard deviation features is extracted to indicate abnormal activity in the preprocessing phase. The building of the FL model is formed by using a trapezoidal membership function along with the input dataset to obtain the intermediate output. SVM, with selected kernels, will use the high degree of membership function of the three inputs to determine whether a fall has occurred or not and to obtain a confidential decision. Additionally, and to summarize, hyperparameters, including sensitivity, specificity, accuracy, and precision, were monitored during the methods' evaluations. As for the 9-axis sensor, all nine parameters (projections of acceleration, gyroscope, and magnetometer into three-dimensional space) were included in the model's inputs with different correlation indices.

A. Data Smoothing

A simple moving average filtering method is used for smoothing noisy raw data in order to obtain clearer data and state estimation of signals from human activities [25]. This can be obtained in (7) from the mathematical definition of a vector x:

$$y(n) = \frac{1}{windowSize\ (x(n)+x\ (n-1)+\cdots+x(n-(windowSize-1))} \quad (7)$$

where, $y(n)$ is the current output, $x(n)$ is the current input, $x\ (n-1)$ is the previous input, etc.; noting that n is the length of the window size.

In Fig. 4, the signal in the original data is a pulse buried in random noise. In Fig. 5, this signal is filtered with $n = 5$ point moving average filters. The noise level becomes lower when the number of points in the filter increases. The optimal solution for this problem is the moving average filter, which gives the lowest noise possible for a given edge sharpness. The smoothing action of the moving average filter reduces the amplitude of the random noise when $N = 5$. Averaging the raw data leads to smoothing out the incidental peaks.

B. Feature Extraction

The acceleration value of human movement for specifying the changing rate of human motion can be calculated by using standard deviation. This function is sensitive to fall detection and can detect sudden tilt changes [54]. Standard deviation is useful for distinguishing static from dynamic activity and for identifying dynamic activity. Assumingly, a human fall is of high acceleration whereas walking is regarded as a low-acceleration activity [55,56].

The standard deviation is an index of how near the individual data points cluster around the mean. If we called each data point $\boldsymbol{x_i}$, an index of dispersion, it would be represented in (8):

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{i=N}(x_i - \bar{x})^2} \quad (8)$$

where $i = 1,..,N$ is an index of data sample, N is the number of data sample, and $\bar{x}$ represents the sample mean.

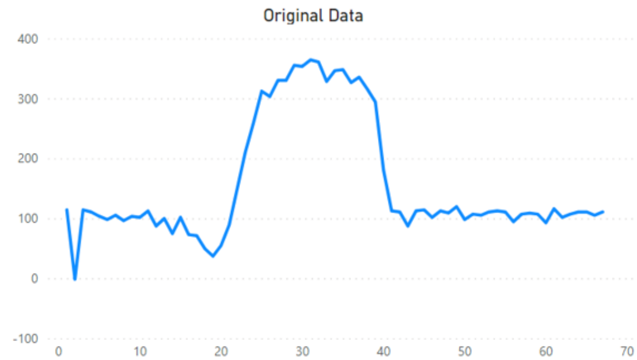

Fig. 4. Original Data by Moving Average Filter.

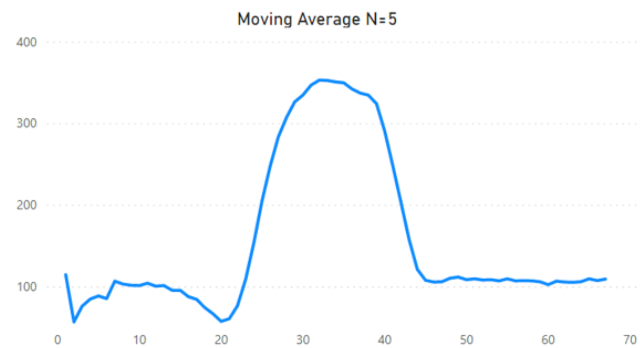

Fig. 5. Smoothing Raw Data by Moving Average Filter.

We computed the standard deviation of each of the nine axes depending on whether they were an accelerometer, a gyroscope, or a magnetometer raw data. After that, three features were extracted from the IMU sensor, which are:

The norm of a standard deviation of the acceleration $|\boldsymbol{\sigma_A}|$ is calculated in (9):

$$|\boldsymbol{\sigma_A}| = \sqrt{\sigma_{Ax}{}^2 + \sigma_{Ay}{}^2 + \sigma_{Az}{}^2} \quad (9)$$

where σ_{Ax}, σ_{Ay}, and σ_{Az} are the standard deviation along the directions of x, y, and z axes of the acceleration raw data that are represented by A_x, A_y, and A_z respectively.

The norm of a standard deviation of the rotation $|\boldsymbol{\sigma_G}|$ is calculated in (10):

$$|\boldsymbol{\sigma_G}| = \sqrt{\sigma_{Gx}{}^2 + \sigma_{Gy}{}^2 + \sigma_{Gz}{}^2} \quad (10)$$

where σ_{Gx}, σ_{Gy}, and σ_{Gz} are the standard deviation along the directions of x, y, and z axes of the rotation raw data that are represented by G_x, G_y, and G_z respectively.

The norm of a standard deviation of the magnetometer $|\boldsymbol{\sigma_M}|$is calculated in (11):

$$|\boldsymbol{\sigma_M}| = \sqrt{\sigma_{Mx}{}^2 + \sigma_{My}{}^2 + \sigma_{Mz}{}^2} \quad (11)$$

where σ_{Mx}, σ_{My}, and σ_{Mz} are the standard deviation along the directions of x, y, and z axes of the magnetometer raw data that are represented by M_x, M_y, and M_z respectively.

Thus, when a standard deviation method is applied, it can differentiate between an actual fall and other activities.

This method can compute a sudden change in acceleration in zero gravity. If this standard deviation value has a high changing trend, it indicates unusual activity. In Fig. 6, a fall might be shown to arise if this value has a high changing rate. As a result, the human motion after this is re-examined to check if the human body has no movement. In this way, it can be ensured that it is the actual fall.

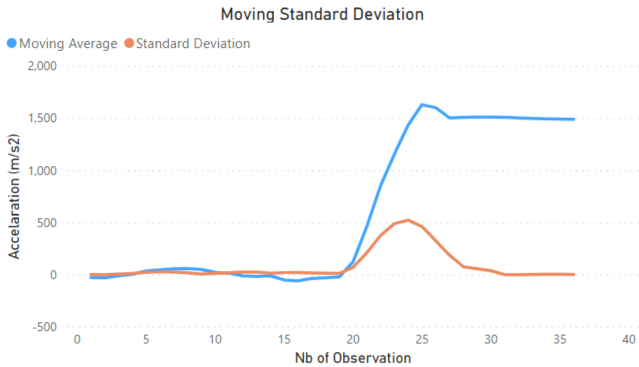

Fig. 6. Example of Standard Deviation for Accelerating Raw Data.

C. Generate Fuzzy Membership Function

A membership function is a function that is responsible for defining how each point in the input space is mapped to a membership value (or degree of membership). A fuzzy set F in X will be explained as a group of ordered pairs, as expressed below:

$$F = \{x, \mu_F(x) | x \in X\} \tag{12}$$

In particular:

- $\mu_F(x)$ is the degree of membership of x in F.
- The boundaries of a fuzzy set F is the set of all $x \in X$ such that $0 < \mu_F(x) < 1$.
- X is the input space and its elements are denoted by x.

In this paper, an MPU–9250 sensor device in an IoT-enabled environment is used to effectively recognize fall events by examining the impact of the body on the ground in addition to the body orientation prior to, during, and following the fall. Next, the proposed method fuzzily analyses these inputs and produces an output as a crisp value between 0 and 1, which signifies the possibility of a fall. FL is used to solve such classification and decision problems without a clear threshold boundary. When a specific value does not completely belong to a certain category, the membership function is used to measure it. The process of converting the logical input value into the membership of each set (normal, medium, high) is called fuzzification.

Three input features, $|\sigma_A|$, $|\sigma_G|$, and $|\sigma_M|$, are introduced to build the suggested approach in order to get the intermediate output. These three linguistic variables for every three inputs are represented in (13):

$$f_{|\sigma_A|} = f_{|\sigma_G|} = f_{|\sigma_M|} = \{NORMAL, MEDIUM, HIGH\} \tag{13}$$

In this paper, trapezoidal Membership Function (MF) was considered as this type is most frequently used, very flexible, and a small amount of data is needed to define it. The trapezoidal function guarantees the existence of a certainty interval in the fuzzification [40].

Each membership is configured with specific values as specification points based on our experimental test to balance sensitivity and specificity. An FL model will be built based on selectable membership functions of the input datasets to get the intermediate output.

The general form of high trapezoidal MF for the fuzzy set $|\sigma_A|$, in terms of the degree of membership, could be defined in (14):

$$\mu_{\mathbf{high}\,|\sigma_A|} = \begin{cases} 0, \text{for x } < 145 \\ \frac{x-145}{160-145}, \text{for } 145 \le x \le 160 \\ 1, \text{for x} \ge 160 \end{cases} \tag{14}$$

The general form of high trapezoidal MF for the fuzzy set $|\sigma_G|$, in terms of the degree of membership, could be defined in (15):

$$\mu_{\mathbf{high}\,|\sigma_G|} = \begin{cases} 0, \text{for x } < 12 \\ \frac{x-12}{15-12}, \text{for } 12 \le x \le 15 \\ 1, \text{for x} \ge 15 \end{cases} \tag{15}$$

The general form of high trapezoidal MF for the fuzzy set $|\sigma_M|$, in terms of the degree of membership, could be defined in (16):

$$\mu_{\mathbf{high}\,|\sigma_M|} = \begin{cases} 0, \text{for x } < 30 \\ \frac{x-30}{35-30}, \text{for } 30 \le x \le 35 \\ 1, \text{for x} \ge 35 \end{cases} \tag{16}$$

Fig. 7 illustrates the membership of the magnitude of a standard deviation of the acceleration in a plan. The normal magnitude is assigned to 80, the medium magnitude is distributed between 95 and 145, and the maximum is defined as 160. Therefore, we put a minimum value that a fall can happen at 145 and consider it as the medium acceleration value.

Fig. 8 depicts memberships of the magnitude of a standard deviation of the rotation in a plan. The normal magnitude is assigned to 6, the medium magnitude is distributed between 9 and 12, and the maximum is defined as 15. Therefore, we put the minimum value that a fall can happen at 12 and consider it the medium angle value.

Fig. 9 represents memberships of the magnitude of a standard deviation of the displacement in a plan. The normal magnitude is assigned to 20, the medium magnitude is distributed between 25 and 30, and the maximum is defined as 35. Therefore, we put a minimum value that a fall can happen at 30 and consider it the medium magnetometer value.

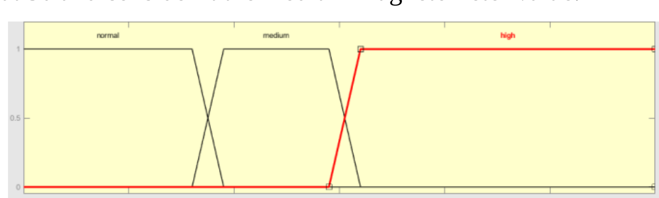

Fig. 7. Membership Function for the Input 1: $|\sigma_A|$.

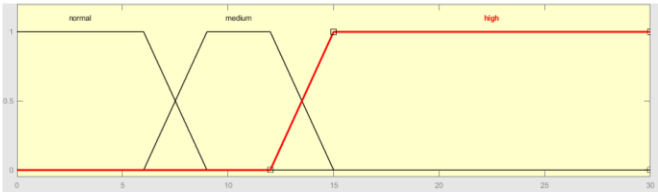

Fig. 8. Membership Function for the Input 2: $|\boldsymbol{\sigma_G}|$.

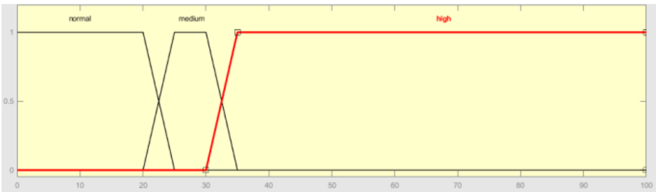

Fig. 9. Membership Function for the Input 3: $|\boldsymbol{\sigma_M}|$.

The output of such FL analysis is a block formed from a high degree of membership based on these three inputs. This means that the event will be considered to have a high degree of membership if the magnitude of the standard deviation of the acceleration is greater than 145, if the magnitude of the standard deviation of the rotation is greater than 12, and if the magnitude of the standard deviation of the magnetometer is greater than 30. The intermediate output will be used as an SVM input with selectable kernels to reach the decision. This will help provide an initial decision bias on whether an event of falling likelihood is high. SVM will use the high degree of membership function of the three inputs to determine whether a fall has occurred or not. However, data points that have a potential normal or medium chance of falling are assigned lower membership degrees.

VI. Experimental Scenario

The prototype was attached at the waist to collect real-time motion data since this is the most fixed position of the body and is needed to provide joint stability and efficiently track any movement. Unlike wired systems, wireless data collection allows users to perform movements more fluently. Four volunteers were chosen to participate in a simulated falling event where they implemented the fall activities on 30 cm thick mats to prevent injuries. 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. In this paper, different subcategories and characteristics of falls in five directions (forward, backward, left, right, and vertical falls) and normal activities including (walking, sitting, and stumbling while walking) were examined in the experiments to achieve our goal. The average duration of each trial was about 40 seconds. Each fall type was repeated more than ten times in a total of 100 trials.

The dataset was read from the CSV file and was implemented with the assistance of the LIBSVM library for MATLAB (version R2018a). Three predictors were introduced where they showed the value of each nine-axis of the trapezoidal membership function of the magnetometer, accelerometer, and gyroscope. Two classes of responses, -1 and 1, were designated to sort the non-falling and falling events, respectively. To make model predictions and estimate how accurate a predictive model will be when implemented, a cross-validation model was used. 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. We trained an SVM classifier in the supervised machine learning 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. We used linear, cubic, quadratic, and RBF SVM kernel functions and standardized the features.

In this paper, the kernel function will be specified as a Medium Gaussian or RBF kernel, a Quadratic kernel, and a Cubic kernel. In order to tune the SVM classifier, the kernel scale was picked by different scales. Next, we adjusted the box constraint level with diverse values for every kernel scale to reach the self-confidential decision. The adjustable parameter (C) usually plays a vital role between cautiously tuning the number of mistakes and maximizing the margin. Therefore, the increase in the box constraint level might decrease the number of support vectors, but it can also increase the time of training, make the training of SVM have fewer misclassifications, and narrow the margin. On the other hand, it allows the SVM to discount more training points and it widens the margin. The evaluation of the Fuzzy SVM network included dividing the datasets from model training (75%) and real-time data for testing new data (25%).

A. *Experimental Results Metrics*

In the final phase, to show the effectiveness of the proposed method, the confusion matrix plot was calculated between the model predictions and the ground truth labels in order to check each class's performance and to know how the current classifier performed in every class [57]. Classification algorithm performance has traditionally been evaluated using a range of performance criteria, which are presented in Table I.

TABLE I. Descriptions of Parametric Evaluation

	Detected Fall	**Undetected Fall**
Fall Occurrence	True Positive (TP)	False Negative (FN)
Fall Unoccurence	False Positive (FP)	True Negative (TN)

In this paper, we compute the following common machine learning classification metrics to assess the classifier's effectiveness in our evaluation. Sensitivity and specificity are calculated by (17) and (18), respectively. Accuracy could be defined by (19), which is the ratio of all samples that the classifier correctly classified [58]. The precision metric could be described by (20), which evaluates the number of correct positive predictions made. Hence, low precision might be an indication of a large number of FP.

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP + FN}} \tag{17}$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN +FP}} \tag{18}$$

$$\text{Accuracy} = \frac{\text{TP +TN}}{\text{TP + TN+FP+FN}} \tag{19}$$

$$\text{Precision } = \frac{\text{TP}}{\text{TP + FP}} \quad (20)$$

B. Experimental Results of Proposed Method without Fuzzy MFs

This section reports the preliminary results of the fall detection system. We used the Gaussian RBF kernel for training and adjusted the optimal regularization parameter with diverse values for every kernel scale. The box constraint level, denoted as the soft margin penalty (C), was varied with different values for each kernel scale in this experiment to provide increased flexibility.

Table II shows the experimental results of the proposed method without employing Fuzzy MFs with RBF kernel function. The percentage of accuracy ranges between 97.78 and 97.96, the percentage of sensitivity ranges between 93.39 and 95.21, whereas the percentage of specificity ranges between 98.91 and 99.45; with precision between 96.96 and 98.43. The boldfaced numbers reveal the best-obtained results in the current classifier that were used later to compare with employed Fuzzy MFs to show the improvement of the proposed method.

In Table III, the experimental results indicate that the best model was selected in the Quadratic kernel, where the percentages of accuracy and sensitivity are 97.78 and 94.71, respectively. As for specificity and precision, the best results were obtained in the cubic kernel, which reached 99.03 and 97.26, respectively.

TABLE II. EXPERIMENTAL RESULTS OF THE PROPOSED METHOD WITHOUT FUZZY MFS FOR GAUSSIAN RBF KERNEL FUNCTION

K=3	*Accuracy %*	*Sensitivity %*	*Specificity %*	*Precision %*
C = 1	97.82	93.39	**99.45**	**98.43**
C = 2	97.82	94.38	99.09	97.44
C = 3	97.87	94.71	99.03	97.28
C = 4	97.91	94.88	99.03	97.29
C = 5	97.91	94.88	99.03	97.29
C = 6	97.87	94.71	99.03	97.28
C = 7	97.87	94.71	99.03	97.28
C = 8	97.91	94.88	99.03	97.29
C = 9	97.82	94.88	98.91	96.96
C = 10	97.82	94.88	98.91	96.96
K=2	***Accuracy %***	***Sensitivity %***	***Specificity %***	***Precision %***
C = 1	97.78	94.38	99.03	97.27
C = 2	97.82	94.71	98.97	97.12
C = 3	97.91	95.04	98.97	97.13
C = 4	97.87	94.88	98.97	97.12
C = 5	97.78	94.71	98.91	96.95
C = 6	97.82	94.88	98.91	96.96
C = 7	97.91	95.04	98.97	97.13
C = 8	**97.96**	**95.21**	98.97	97.13
C = 9	97.91	95.21	98.91	96.97
C = 10	97.87	95.04	98.91	96.96

TABLE III. EXPERIMENTAL RESULTS WITHOUT FUZZY MFS FOR LINEAR, CUBIC, AND QUADRATIC KERNEL FUNCTIONS

Kernel function	*Accuracy %*	*Sensitivity %*	*Specificity %*	*Precision %*
Linear SVM	97.47	93.88	98.78	96.6
Cubic SVM	97.64	93.88	**99.03**	**97.26**
Quadratic SVM	**97.78**	94.71	98.91	96.95

C. Experimental Results of Proposed Method with Fuzzy MFs

Table IV shows the experimental results of the proposed method by employing Fuzzy MFs with RBF kernel function. The percentage of accuracy ranges between 99.82 and 99.96, the percentage of sensitivity ranges between 99.63 and 99.81, whereas the percentage of specificity ranges between 99.88 and 100; with precision between 99.63 and 100. The boldfaced numbers reveal the best-obtained results in the current classifier by employing Fuzzy MFs to show the improvement of the proposed method.

In Table V, the experimental results of Fuzzy SVM 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 the kernel function. An overall score of 100% for specificity and precision and 99.81% for sensitivity was obtained by using the new method.

TABLE IV. EXPERIMENTAL RESULTS OF THE PROPOSED METHOD WITH FUZZY MFS FOR GAUSSIAN RBF KERNEL FUNCTION

K=3	*Accuracy %*	*Sensitivity %*	*Specificity %*	*Precision %*
C = 1	99.91	99.63	100	100
C = 2	99.87	99.81	99.88	99.63
C = 3	99.82	99.63	99.88	99.63
C = 4	99.82	99.63	99.88	99.63
C = 5	99.82	99.63	99.88	99.63
C = 6	99.87	99.63	99.94	99.81
C = 7	99.91	99.63	100	100
C = 8	**99.96**	**99.81**	**100**	**100**
C = 9	**99.96**	**99.81**	**100**	**100**
C = 10	**99.96**	**99.81**	**100**	**100**
K=2	***Accuracy %***	***Sensitivity %***	***Specificity %***	***Precision %***
C = 1	99.91	99.63	100	100
C = 2	99.91	99.63	100	100
C = 3	99.91	99.63	100	100
C = 4	**99.96**	**99.81**	**100**	**100**
C = 5	**99.96**	**99.81**	**100**	**100**
C = 6	**99.96**	**99.81**	**100**	**100**
C = 7	**99.96**	**99.81**	**100**	**100**
C = 8	**99.96**	**99.81**	**100**	**100**
C = 9	**99.96**	**99.81**	**100**	**100**
C = 10	**99.96**	**99.81**	**100**	**100**

TABLE V. EXPERIMENTAL RESULTS OF THE PROPOSED METHOD WITH FUZZY MFS FOR LINEAR, CUBIC, AND QUADRATIC KERNEL FUNCTIONS

Kernel function	*Accuracy %*	*Sensitivity %*	*Specificity %*	*Precision %*
Linear SVM	99.56	99.44	99.59	89.70
Cubic SVM	99.96	99.81	100	100
Quadratic SVM	99.96	99.81	100	100

In Table VI, we compared the effectiveness of our proposed fall detection method when we employed the Fuzzy MFs. The overall performance in terms of accuracy, sensitivity, specificity, and precision was increased by 2%, 4.6%, 0.55%, and 1.57%, respectively.

A fall detection system should avoid acquiring FP and FN results to obtain more reliable results. The experimental results demonstrate that the proposed method with fuzzy membership reduced false alarms and achieved accurate fall detection with better performance than traditional SVM.

TABLE VI. EXPERIMENTAL RESULTS OF THE PROPOSED FUZZY SVM VS. CONVENTIONAL METHOD

Proposed method	*SVM WITH FUZZY MFs*			
Kernel function	***Accuracy %***	***Sensitivity %***	***Specificity %***	***Precision %***
Linear SVM	99.56	99.44	99.59	89.70
Cubic SVM	99.96	99.81	100	100
Quadratic SVM	99.96	99.81	100	100
Gaussian RBF	99.96	99.81	100	100
Proposed method	*SVM **WITHOUT** FUZZY MFs*			
Kernel function	***Accuracy %***	***Sensitivity %***	***Specificity %***	***Precision %***
Linear SVM	97.47	93.88	98.78	96.6
Cubic SVM	97.64	93.88	99.03	97.26
Quadratic SVM	97.78	94.71	98.91	96.95
Gaussian RBF	97.96	95.21	99.45	98.43

VII. CONCLUSION

In this paper, we introduced a new hybrid method integrating FL and SVM as a powerful technique for fall detection. The obtained results based on the defined environment showed a significant improvement in the accuracy of detection as compared to standalone, independent methods. Moreover, this paper examined the ways of identifying, expressing, and inspecting the relationship between disabled people suffering from traumatic brain injuries and falling events. Detecting falls accurately in time can reduce the severe consequences, especially since it can improve the quality of life of people with disabilities by promoting their independence. Besides, the prototype we examined was based on the Arduino platform, with the MPU-9250 sensor forming the part that was fixed on the patient's body and that wirelessly connects over the IoT platform via a low-energy Bluetooth interface. Thus, a 9-axis of accelerometer, gyroscope, and magnetometer data formed the input matrix. The data were then smoothed using the moving average method to obtain clearer data and reduce the amplitude of the random noise. The standard deviation of each of the nine axes from the IMU sensor, depending on whether it was an accelerometer, gyroscope, or magnetometer, raw data in the x, y, and z planes in 3D space, was then computed. After that, the magnitude of standard deviation features was extracted in a way that makes it possible to differentiate between an actual fall and other activities, and thus to indicate abnormal activity in the pre-processing phase. Our proposed fall detection method is based on the effective combination of FL and SVM algorithms. In Fuzzy SVM, a fuzzy membership is given to each data point and set by expert experience. Different memberships reflect various contributions made to the decision-making learning process. SVM uses a high degree of membership functions extracted from selected features to automatically generate a model of the data and drive training, which enhances generalization ability and makes the heuristic obviousness of the traditional SVM efficient. The overall performance of the proposed fall detection system in terms of sensitivity and accuracy were 99.81% and 99.96%, respectively. This experiment achieved the maximum specificity and precision of 100% and, accordingly, proved to be effective and yielded better results than the conventionally used approaches. In other words, there was clear evidence that combining FL and SVM detects falls more accurately and performs better in reducing the effects of false positive alarms. This type of work is common abroad, yet it is still novel in Lebanon. So this study is considered a pioneering experience in fall detection systems for disabled patients. Furthermore, knowing that fall event datasets for this type of patients are not currently available in the literature, further trials must be conducted to simulate and refine the fall detection system and evaluate it outside the workplace. The application can be tested in real-world scenarios involving various types of disability. Our continuing research includes integrating fall detection algorithms into a smartphone that would be worn around the waist. This will be very useful to improve the robustness of our proposed method.

ACKNOWLEDGMENT

This research is funded by Fakulti Teknologi Maklumat dan Komunikasi, Universiti Teknikal Malaysia Melaka.

REFERENCES

[1] Step safely: strategies for preventing and managing falls across the life-course. Geneva: World Health Organization; 2021. Licence: CC BY-NC-SA 3.0 IGO.

[2] N. Pannurat, S. Thiemjarus, and E. Nantajeewarawat, "Automatic fall monitoring: A review," Sensors (Switzerland), vol. 14, no. 7, pp. 12900–12936, 2014, doi: 10.3390/s140712900.

[3] Liu, Z. Jiang, X. Su, S. Benzoni, and A. Maxwell, "Detection of human fall using floor vibration and multi-features semi-supervised SVM," Sensors (Switzerland), vol. 19, no. 17, Sep. 2019, doi: 10.3390/s19173720.

[4] L. Ren and Y. Peng, "Research of fall detection and fall prevention technologies: A systematic review," IEEE Access, vol. 7. 2019. doi: 10.1109/ACCESS.2019.2922708.

[5] Z. Chen and Y. Wang, "Infrared–ultrasonic sensor fusion for support vector machine–based fall detection," J. Intell. Mater. Syst. Struct., vol. 29, no. 9, pp. 2027–2039, 2018, doi: 10.1177/1045389X18758183.

[6] R. Espinosa, H. Ponce, S. Gutiérrez, L. Martínez-Villaseñor, J. Brieva, and E. Moya-Albor, "A vision-based approach for fall detection using multiple cameras and convolutional neural networks: A case study using the UP-Fall detection dataset," Comput. Biol. Med., vol. 115, 2019, doi: 10.1016/j.compbiomed.2019.103520.

[7] S. Sathyanarayana, R. K. Satzoda, S. Sathyanarayana, and S. Thambipillai, "Vision-based patient monitoring: a comprehensive review of algorithms and technologies," J. Ambient Intell. Humaniz. Comput., vol. 9, no. 2, 2018, doi: 10.1007/s12652-015-0328-1.

[8] N. Harum, Z. Z. Abidin, W. Shah, and A. Hassan, "Implementation of Smart Monitoring System with Fall Dectector for Elderly using IoT Technology," Int. J. Comput., vol. 17, no. 4, pp. 243–249, 2018.

[9] P. V. Er and K. K. Tan, Wearable solution for robust fall detection. INC, 2020. doi: 10.1016/b978-0-12-818546-9.00004-x.

[10] J. Alizadeh, M. Bogdan, J. Classen, and C. Fricke, "Support vector machine classifiers show high generalizability in automatic fall detection in older adults," Sensors, vol. 21, no. 21, pp. 1–14, 2021, doi: 10.3390/s21217166.

[11] T. Özdemir, "An analysis on sensor locations of the human body for wearable fall detection devices: Principles and practice," Sensors (Switzerland), vol. 16, no. 8, 2016, doi: 10.3390/s16081161.

[12] P. Tsinganos and A. Skodras, "On the comparison of wearable sensor data fusion to a single sensor machine learning technique in fall detection," Sensors (Switzerland), vol. 18, no. 2, 2018, doi: 10.3390/s18020592.

[13] O. Aziz, M. Musngi, E. J. Park, G. Mori, and S. N. Robinovitch, "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," Med. Biol. Eng. Comput., vol. 55, no. 1, 2017, doi: 10.1007/s11517-016-1504-y.

[14] M. Safeea and P. Neto, "Minimum distance calculation using laser scanner and IMUs for safe human-robot interaction," Robot. Comput. Integr. Manuf., vol. 58, 2019, doi: 10.1016/j.rcim.2019.01.008.

[15] M. Kangas, A. Konttila, P. Lindgren, I. Winblad, and T. Jämsä, "Comparison of low-complexity fall detection algorithms for body attached accelerometers," Gait Posture, vol. 28, no. 2, 2008, doi: 10.1016/j.gaitpost.2008.01.003.

[16] K. Bourke, J. V. O'Brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," Gait Posture, vol. 26, no. 2, 2007, doi: 10.1016/j.gaitpost.2006.09.012.

[17] H. Ponce, L. Martínez-Villaseñor, and J. Nuñez-Martínez, "Sensor location analysis and minimal deployment for fall detection system," IEEE Access, vol. 8, pp. 166678–166691, 2020, doi: 10.1109/ACCESS.2020.3022971.

[18] N. Otanasap, "Pre-impact fall detection based on wearable device using dynamic threshold model," Parallel Distrib. Comput. Appl. Technol. PDCAT Proc., vol. 0, pp. 362–365, 2016, doi: 10.1109/PDCAT.2016.083.

[19] H. Wang, C. Ma, and L. Zhou, "A brief review of machine learning and its application," Proc. - 2009 Int. Conf. Inf. Eng. Comput. Sci. ICIECS 2009, 2009, doi: 10.1109/ICIECS.2009.5362936.

[20] M. Rahmani et al., "Machine learning (Ml) in medicine: Review, applications, and challenges," Mathematics, vol. 9, no. 22. 2021. doi: 10.3390/math9222970.

[21] Y. Hsieh, K. C. Liu, C. N. Huang, W. C. Chu, and C. T. Chan, "Novel hierarchical fall detection algorithm using a multiphase fall model," Sensors (Switzerland), vol. 17, no. 2, 2017, doi: 10.3390/s17020307.

[22] L. Palmerini, J. Klenk, C. Becker, and L. Chiari, "Accelerometer-based fall detection using machine learning: Training and testing on real-world falls," Sensors (Switzerland), vol. 20, no. 22, pp. 1–15, Nov. 2020, doi: 10.3390/s20226479.

[23] T. Özdemir and B. Barshan, "Detecting falls with wearable sensors using machine learning techniques," Sensors (Switzerland), vol. 14, no. 6, pp. 10691–10708, 2014, doi: 10.3390/s140610691.

[24] Hussain, F. Hussain, M. Ehatisham-Ul-Haq, and M. A. Azam, "Activity-Aware Fall Detection and Recognition Based on Wearable Sensors," IEEE Sens. J., vol. 19, no. 12, 2019, doi: 10.1109/JSEN.2019.2898891.

[25] D. Pan, H. Liu, D. Qu, and Z. Zhang, "Human Falling Detection Algorithm Based on Multisensor Data Fusion with SVM," Mob. Inf. Syst., vol. 2020, 2020, doi: 10.1155/2020/8826088.

[26] Şengül, E. Ozcelik, S. Misra, R. Damaševičius, and R. Maskeliūnas, "Fusion of smartphone sensor data for classification of daily user activities," Multimed. Tools Appl., vol. 80, no. 24, pp. 33527–33546, 2021, doi: 10.1007/s11042-021-11105-6.

[27] Z. A. Kakarash, S. H. T. Karim, and M. Mohammadi, "Fall Detection Using Neural Network Based on Internet of Things Streaming Data," UHD J. Sci. Technol., vol. 4, no. 2, pp. 91–98, Oct. 2020, doi: 10.21928/uhdjst.v4n2y2020.pp91-98.

[28] T. De Quadros, A. E. Lazzaretti, and F. K. Schneider, "A Movement Decomposition and Machine Learning-Based Fall Detection System Using Wrist Wearable Device," IEEE Sens. J., vol. 18, no. 12, 2018, doi: 10.1109/JSEN.2018.2829815.

[29] Y. Zhang and L. Wu, "Classification of fruits using computer vision and a multiclass support vector machine," Sensors (Switzerland), vol. 12, no. 9, pp. 12489–12505, 2012, doi: 10.3390/s120912489.

[30] Rescio, A. Leone, and P. Siciliano, "Supervised expert system for wearable MEMS accelerometer-based fall detector," J. Sensors, vol. 2013, 2013, doi: 10.1155/2013/254629.

[31] S. H. Liu and W. C. Cheng, "Fall detection with the support vector machine during scripted and continuous unscripted activities," Sensors (Switzerland), vol. 12, no. 9, pp. 12301–12316, 2012, doi: 10.3390/s120912301.

[32] R. Amami, D. Ben Ayed, and N. Ellouze, "Practical Selection of SVM Supervised Parameters with Different Feature Representations for Vowel Recognition," vol. 7, no. May, pp. 418–424, 2015, doi: 10.4156/jdcta.vol7.issue9.50.

[33] W. Min, L. Yao, Z. Lin, and L. Liu, "Support vector machine approach to fall recognition based on simplified expression of human skeleton action and fast detection of start key frame using torso angle," IET Comput. Vis., vol. 12, no. 8, 2018, doi: 10.1049/iet-cvi.2018.5324.

[34] X. Zhuang, X. Zhou, M. A. Hasegawa-Johnson, and T. S. Huang, "Real-world acoustic event detection," Pattern Recognit. Lett., vol. 31, no. 12, 2010, doi: 10.1016/j.patrec.2010.02.005.

[35] T. Zhen, L. Mao, J. Wang, and Q. Gao, "Wearable preimpact fall detector using SVM," 2016. doi: 10.1109/ICSensT.2016.7796223.

[36] Shahzad and K. Kim, "FallDroid: An Automated Smart-Phone-Based Fall Detection System Using Multiple Kernel Learning," IEEE Trans. Ind. Informatics, vol. 15, no. 1, pp. 35–44, 2019, doi: 10.1109/TII.2018.2839749.

[37] Lee and S. Kim, "Black-Box Classifier Interpretation Using Decision Tree and Fuzzy Logic-Based Classifier Implementation," Int. J. Fuzzy Log. Intell. Syst., vol. 16, no. 1, pp. 27–35, 2016, doi: 10.5391/ijfis.2016.16.1.27.

[38] M. H. Azam, M. H. Hasan, S. Jadid, A. Kadir, and S. Hassan, "Prediction of Sunspots using Fuzzy Logic : A Triangular Membership Function-based Fuzzy C-Means Approach," Int. J. Adv. Comput. Sci. Appl., vol. 12, no. 2, pp. 357–362, 2021.

[39] B. Pękala, T. Mroczek, D. Gil, and M. Kepski, "Application of Fuzzy and Rough Logic to Posture Recognition in Fall Detection System," Sensors, vol. 22, no. 4, Feb. 2022, doi: 10.3390/s22041602.

[40] M. Casal-Guisande, A. Comesaña-Campos, I. Dutra, J. Cerqueiro-Pequeño, and J. B. Bouza-Rodríguez, "Design and Development of an Intelligent Clinical Decision Support System Applied to the Evaluation of Breast Cancer Risk," J. Pers. Med., vol. 12, no. 2, 2022, doi: 10.3390/jpm12020169.

[41] B. Pandya, A. Pourabdollah, and A. Lotfi, "Comparative Analysis of Real-Time Fall Detection Using Fuzzy Logic Web Services and Machine Learning," Technologies, vol. 8, no. 4, p. 74, 2020, doi: 10.3390/technologies8040074.

[42] Fernández-Caballero, M. V. Sokolova, and J. Serrano-Cuerda, "Lateral inhibition in accumulative computation and fuzzy sets for human fall pattern recognition in colour and infrared imagery," Sci. World J., vol. 2013, 2013, doi: 10.1155/2013/935026.

[43] S. Moulik and S. Majumdar, "FallSense: An Automatic Fall Detection and Alarm Generation System in IoT-Enabled Environment," IEEE Sens. J., vol. 19, no. 19, 2019, doi: 10.1109/JSEN.2018.2880739.

[44] Zhang, M. Alrifaai, K. Zhou, and H. Hu, “A novel fuzzy logic algorithm for accurate fall detection of smart wristband,” Trans. Inst. Meas. Control, vol. 42, no. 4, pp. 786–794, Feb. 2020, doi: 10.1177/0142331219881578.

[45] C. F. Lin and S. De Wang, “Fuzzy support vector machines,” IEEE Trans. Neural Networks, vol. 13, no. 2, pp. 464–471, 2002, doi: 10.1109/72.991432.

[46] E. Spyrou, G. Stamou, Y. Avrithis, and S. Kollias, “Fuzzy support vector machines for image classification fusing MPEG-7 visual descriptors,” IET Semin. Dig., vol. 2005, no. 11099, pp. 23–30, 2005, doi: 10.1049/ic.2005.0706.

[47] H. Tang, Y. Liao, F. Sun, and H. Xie, “Fuzzy support vector machine with a new fuzzy membership function,” Hsi-An Chiao Tung Ta Hsueh/Journal Xi’an Jiaotong Univ., vol. 43, no. 7, pp. 40–43, 2009.

[48] C. Yang, S. K. Oh, B. Yang, W. Pedrycz, and Z. W. Fu, “Fuzzy quasi-linear SVM classifier: Design and analysis,” Fuzzy Sets Syst., vol. 413, pp. 42–63, 2021, doi: 10.1016/j.fss.2020.05.010.

[49] M. A. Nanda, K. B. Seminar, D. Nandika, and A. Maddu, “A comparison study of kernel functions in the support vector machine and its application for termite detection,” Inf., vol. 9, no. 1, 2018, doi: 10.3390/info9010005.

[50] Fleury, M. Vacher, and N. Noury, “SVM-based multimodal classification of activities of daily living in health smart homes: Sensors, algorithms, and first experimental results,” IEEE Trans. Inf. Technol. Biomed., vol. 14, no. 2, pp. 274–283, 2010, doi: 10.1109/TITB.2009.2037317.

[51] E. A. Zanaty and A. Afifi, “Support Vector Machines (SVMs) with Universal Kernels,” Appl. Artif. Intell., vol. 9514, 2011, doi: 10.1080/08839514.2011.595280.

[52] Sulistiana and M. A. Muslim, “Support Vector Machine (SVM) Optimization Using Grid Search and Unigram to Improve E-Commerce Review Accuracy,” J. Soft Comput. Explor., vol. 1, no. 1, pp. 8–15, 2020.

[53] Akshay and M. Sumanth, “Bluetooth based home automation by using arduino.,” Grad. Res. Eng. Technol., vol. 7, no. 02, pp. 37–42, 2021, doi: 10.47893/gret.2021.1116.

[54] M. Tolkiehn, L. Atallah, B. Lo, and G. Z. Yang, “Direction sensitive fall detection using a triaxial accelerometer and a barometric pressure sensor,” Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. EMBS, no. August 2011, pp. 369–372, 2011, doi: 10.1109/IEMBS.2011.6090120.

[55] Q. Li and J. A. Stankovic, “Detection for Falls with Different Activity Levels,” Wirel. Heal. 11, 2011.

[56] H. Gjoreski, M. Luštrek, and M. Gams, “Accelerometer placement for posture recognition and fall detection,” Proc. - 2011 7th Int. Conf. Intell. Environ. IE 2011, no. August, pp. 47–54, 2011, doi: 10.1109/IE.2011.11.

[57] Zhang and A. S. Zhang, “Real-Time Wildfire Detection and Alerting with a Novel Machine Learning Approach: A New Systematic Approach of Using Convolutional Neural Network (CNN) to Achieve Higher Accuracy in Automation,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 8, pp. 1–6, 2022, doi: 10.14569/IJACSA.2022.0130801.

[58] Saha, H. K. Tripathy, F. Masmoudi, and T. Gaber, “A Machine Learning Model for Personalized Tariff Plan b ased on Customer ’ s Behavior in the Telecom Industry,” Int. J. Adv. Comput. Sci. Appl., vol. 13, no. 10, pp. 171–184, 2022.