KEPES
http://scholopress.com/kepes-journal

**2023** Volume 21 Issue 3 pp. 772-782

# Machine Learning based Fall Detection Smart Watch

**Budigi Poorna Nandini**[1], **P. Satyananda Reddy**[1], ***Gerardine Immaculate Mary**[1]

[1]SENSE Vellore Institute of Technology, Vellore, Tamilnadu, India

## Abstract

The smartwatch was designed to transmit real-time notifications to the user's smartphone in the event of a fall, enabling quick actions. I was able to expand my skills in machine learning, sensor interface, and mobile app development through the project. A smartwatch might be modified to add new features like heart rate monitoring and prescription reminders, which could have significant uses in the healthcare industry. The project provided a great opportunity to learn about wearable technologies and machine learning. The goal of this project is to develop a tool for accurately and quickly detecting falls and alerting emergency contacts, potentially saving lives, and enhancing the quality of life for seniors and people with mobility issues. The project's hardware components include GPS and GSM modules. To increase the fall detection algorithm's accuracy and investigate prospective system extensions and upgrades, the project also involves the collection and analysis of real-world data. The overall goal of this project is to create a useful smartwatch with a fall detection system that can offer critical assistance to those in need and their emergency contacts. The project also involves creating a cloud-based infrastructure to store and process sensor data from the smartwatch, enabling remote data analysis and monitoring. The research also investigates additional potential uses for the smartwatch, such as activity tracking and health monitoring, in addition, to fall detection. Through this initiative, we hope to enhance the lives of people with mobility challenges, their careers, and the expanding field of wearable technology. The project also involves creating a cloud-based infrastructure to store and process sensor data from the smartwatch, enabling remote data analysis and monitoring. The research also investigates additional potential uses for the smartwatch, such as activity tracking and health monitoring, in addition, to fall detection. Through this initiative, we hope to enhance the lives of people with mobility challenges, their careers, and the expanding field of wearable technology.

**Keywords:** Internet of Things, Machine learning, Fall detection, Smart watch

## 1. Introduction

Every year, one in three adults 65 and over experiences a fall, making them a major health risk for the elderly and those with mobility problems. A person's quality of life may be greatly impacted by falls, which can cause catastrophic injuries such as head trauma, hip fractures, and other injuries. Further, since they might not be able to call for help, people who live alone or do not have rapid access to medical care may find falls to be particularly dangerous. Fall detection systems have been created to enable real-time monitoring and emergency reaction to solve this problem.

To detect falls and warn emergency contacts or medical personnel, these systems frequently include wearable gadgets with sensors like accelerometers and gyroscopes. However, the accuracy and dependability

*Corresponding author
DOI https://doi.org/10.6084/m9.figshare.24182547#130

of conventional fall detection systems are limited. Many systems struggle to distinguish between falls and other behaviors like sitting or lying down, leading to frequent false alarms and missed detections. Furthermore, many systems rely on pre-established threshold levels, which might not be suitable for all users. This project attempts to create a smartwatch with a fall detection system based on machine learning to overcome these restrictions.

By examining data from sensors like accelerometers and gyroscopes, the system uses sophisticated machine learning algorithms to reliably identify falls in real time. The application of machine learning enables more precise and individualized detection because the system can be taught using data specific to each user, increasing precision, and lowering false alarm rates. By offering a trustworthy and efficient instrument for fall detection and emergency response, the development of this smartwatch has the potential to dramatically improve the safety and quality of life for vulnerable individuals. Additionally, the project offers a chance to investigate the potential of machine learning in healthcare and to create fresh methods for physiological data processing and monitoring in real time.

The smartwatch created for this project offers several capabilities, including the ability to show the time, date, and other fundamental functions that are typically included in smartwatches. In case of an emergency, the watch can broadcast the wearer's location to a registered phone number thanks to its Neo-6m GPS and GSM modules. The Thingspeak cloud platform's Thingspeak is used to read sensor data, which is then analyzed in real-time by the fall detection algorithm using the SVM algorithm. When a fall is detected, the watch broadcasts the user's position, heart rate, and temperature information to the designated phone number. A button on the watch can be used to send the user's position, heart rate, and temperature readings to the registered mobile device. The purpose of this project is to create an accurate and trustworthy fall detection system that can assist prevent serious injuries and give the user and their emergency contacts a sense of confidence.

## 2. Related Works

The fall detection system for elderly people described in this article uses an accelerometer sensor and a pipeline for data analysis to find possible fall events. Three publicly accessible datasets are used to test a novel collection of characteristics, which outperformed earlier research in terms of performance. This research suggests classifying features taken from the accelerometer and gyroscope sensor data as falls or routine activities using wearable technology and machine learning approaches, specifically a supervised Support Vector Machine. The SisFall dataset was used to evaluate the suggested methodology, and it produced the best results for detecting falls in the dataset under consideration. The utilization of vital signs measures along with IMU data has demonstrated promise for enhancing fall detection precision. Even better performance outcomes might be obtained through an additional investigation into the ideal feature combination and the use of ensemble approaches. The use of smartwatches to track elderly people in various settings demonstrates that the current technologies are unable to respond intelligently to anomalous situations. To address this problem, the authors use smartwatches to track elderly people in various settings and show that existing systems are unable to respond intelligently to anomalous events. The authors suggest a knowledge-based system coupled with a rule-based system to overcome this problem. Through the use of a low-power smartwatch with wireless communication and energy-harvesting capabilities and thin flexible temperature sensors that can be pliably laminated onto the epidermis, this proposal introduces a novel concept that combines the low-power design of electronic modules with the accuracy attained by thin epidermal sensors. With a power consumption of a few W in sleep mode and 2mW in acquisition mode, the gadget offers precise thermal mapping.

## 3. Design Approach

Designing a smartwatch with a machine learning-based fall detection system that can detect falls reliably in real-time and send an alert message to a registered phone number with the user's position, heart rate, and the temperature was the design strategy used for this project. The HBT V1.0 heartbeat sensor, the DHT11 temperature sensor, the Neo-6m GPS module, the A7670C sensors, and the 9 DOF IMU sensor were the components used in the project. The fall detection method and sensor data processing were carried out by the Atmega328P microprocessor. We analyzed the sensor data and identified fall events using machine learning algorithms to create the fall detection system. The device's orientation and fall detection were calculated using acceleration and orientation data collected by the IMU sensor. To provide more context regarding the user's situation and environment, the data from the other sensors were also used. To detect falls and provide alert messages, the system was created to continuously monitor sensor data and analyze it in real time.

### *3.1 Code and Standards*

The Arduino IDE, which was used to produce the code for the hardware components, and Python, which was used to develop the machine learning method for fall detection, are the codes and standards that were utilized in this project. The code was created to ensure compatibility with the project's other sensors as well as the Atmega328P microprocessor. Additionally, the project followed industry norms for software development, including version control, testing, and documentation.

### *3.2 Constraints, Alternatives, and Tradeoffs*

The Atmega328P microcontroller's restricted processing speed and memory capacity was one of this project's major limitations, necessitating rigorous code optimization and the adoption of effective algorithms. The smartwatch's short battery life was another restriction, necessitating the deployment of low-power sensors and careful data management. Using vision-based systems, such as cameras or depth sensors, or wearable sensors, such as pressure sensors or wearable cameras, were two alternative methods that were investigated for fall detection. These methods were discovered to be more intrusive or expensive than the selected method, and they were also proven to be less effective for real-time fall detection.

The balance between accuracy and power consumption as well as the tradeoff between price and features were taken into consideration. The chosen design prioritized fall detection accuracy and dependability while simultaneously reducing power use and cost. The overall goal of this project was to develop a smartwatch with a fall detection system based on machine learning that can offer critical assistance to vulnerable people and their emergency contacts. This project aimed to develop a powerful tool for detecting falls and quickly alerting emergency contacts, potentially saving lives and improving the quality of life for seniors and people with mobility issues. The project combined hardware elements such as GPS and GSM modules with advanced machine learning algorithms.

### *3.3 Working of the systems*

The project is a wearable gadget intended to keep track of the wearer's health and send alert messages in case of a fall or emergency. It is specially made for senior folks who might need help keeping an eye on their safety and health. The wearer's movements, position, and general health can all be tracked by the device's sensors, which include a heartbeat sensor, temperature sensor, GPS module, and an IMU sensor. The Arduino IDE is used to program the ATmega328P microcontroller that powers the gadget. Machine learning algorithms are used to process and analyze the sensor data to detect a fall in real-time. And the results are sent to the cloud-based IoT platform ThingSpeak, which then sends back the results to the smartwatch. In case of an emergency, the device also features a panic button that may be activated. Overall, the wearable device provides a comprehensive and personalized solution to monitor the wearer's health and safety, while also providing vital information for their caregivers.

### *3.4 Software setup*

Google Colab is a free online platform provided by Google that allows users to write and run code in a Jupyter Notebook environment without requiring any special setup on their local machines. It provides access to powerful computing resources such as GPUs and TPUs, which can be used to accelerate machine learning and other data-intensive tasks. Additionally, Colab allows users to collaborate on code with others in real time and provides seamless integration with other Google services such as Google Drive and GitHub.

ThingSpeak is an IoT analytics service that allows you to aggregate, visualize, and analyze live data streams in the cloud. ThingSpeak provides instant visualizations of data posted by your devices to ThingSpeak. With the ability to execute MATLAB code in ThingSpeak, you can perform online analysis and process data as it comes in. ThingSpeak is often used for prototyping and proof-of-concept IoT systems that require analytics.

The user's health and safety can be monitored and tracked with the help of the smartwatch's Temperature sensor, heartbeat sensor, GPS, and GSM modules. The Neo-6m GPS module is a very precise positioning system that the smartwatch uses to pinpoint the user's location. In emergency scenarios where the user can be lost or in need of help, this capability is quite helpful. The user can quickly and simply report their location to a registered phone number by pushing a button on their smartwatch, making it possible for rescuers or loved ones to find and assist them.

Another crucial part of the smartwatch is the GSM module, which enables the gadget to send alert messages. In an emergency, the user can rapidly alert the registered phone number to inform them of the circumstance and request help. In circumstances where the user might not be able to make a call or talk on the phone, this capability is quite helpful.

The smartwatch's accelerometer and gyroscope sensors along with the Machine learning algorithm are intended to detect falls. These sensors operate by identifying the acceleration and angular movement in three directions, X-axis, Y-axis, and Z-axis. Then these values are sent for Machine learning analysis which may signify a fall. When a fall is detected, the smartwatch uses the GSM module to communicate location, body temperature, and heart rate information to the registered phone number, alerting the owner and allowing them to take the necessary action.

The smartwatch's heart rate sensor is a crucial component since it enables real-time heart rate monitoring for the user. People with heart diseases or other health disorders that call for routine monitoring will find this function especially helpful. Users can track their level of fitness and make knowledgeable judgments regarding their health by keeping an eye on their heart rate.

Similarly, to this, users can check their body temperature using the smartwatch's temperature sensor. People who are prone to fever or other temperature-related health problems will find this function to be especially helpful. Users can take necessary action to address any potential health risks by keeping an eye on their body temperature before they worsen.

The smartwatch's display also shows users crucial data, such as the time, date, temperature, and heart rate. Users may utilize this information to make educated decisions regarding their health and well-being because it is readily available. In conclusion, the smartwatch with GPS, GSM, and other sensors is an effective instrument for tracking and monitoring the user's health and safety.

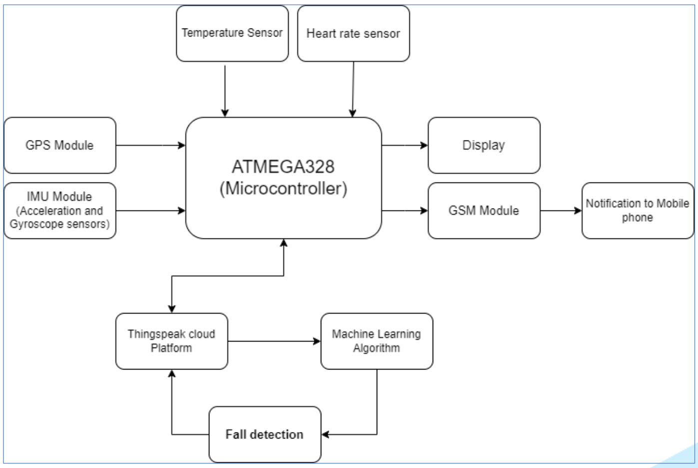


**Fig. 1** Block diagram of working of the device

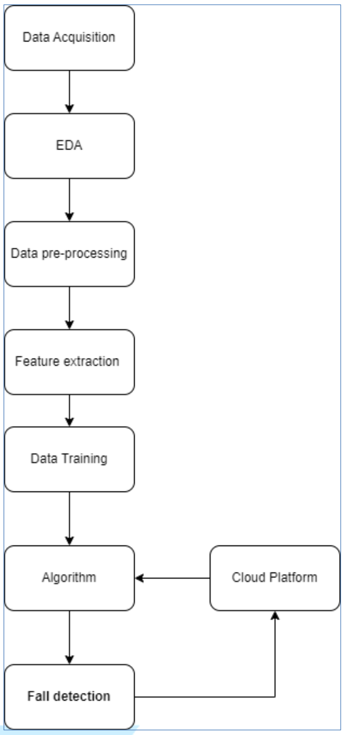


**Fig.2** Block diagram of working of ML algorithm

## 4. Proposed approach and classification methods

First, we discuss how we formulate the task. We then describe how to overcome the challenge of accuracy using different ML algorithms Finally, we elaborate on the Machine learning-based models employed in this work along with the implementation details.

### *4.1 Dataset*

The dataset used is called the "Human Activity Recognition Using Smartphones Data Set". It is a publicly available dataset that was collected from 30 volunteers wearing a smartphone (Samsung Galaxy S II) on their wrist while performing various physical activities like sitting, walking, walking upstairs, walking downstairs, and falling. The dataset was collected using the smartphone's embedded accelerometer and gyroscope sensors, and it consists of 561 features (i.e., measurements) for each of the 7,199 instances (i.e., data points).In addition to the 7199 instances from the existing dataset, I also generated 200 instances using the laboratory prototype developed in the project and incorporated them into the dataset. The dataset is commonly used in research related to human activity recognition, machine learning, and data mining. It has been cited in over 2,000 research papers according to Google Scholar.

### *4.2 EDA (Exploratory Data Analysis)*

EDA allows for a deeper understanding of the dataset, including patterns, relationships, and trends. This is important for uncovering insights and making informed decisions. EDA helps in identifying outliers, which are data points that differ significantly from other observations. This can be important in identifying anomalies that may require further investigation. EDA can help in identifying missing values, duplicates, or other errors in the dataset. This allows for data cleaning before further analysis. EDA helps in identifying the most important features of the dataset. This can be useful in building predictive models, as it helps in selecting the relevant variables.

EDA allows for the creation of data visualizations that can make complex information more accessible and easier to understand. This can be useful in communicating results. Overall, EDA is a critical step in the data analysis process as it allows for a better understanding of the dataset and can lead to more accurate and meaningful insights.

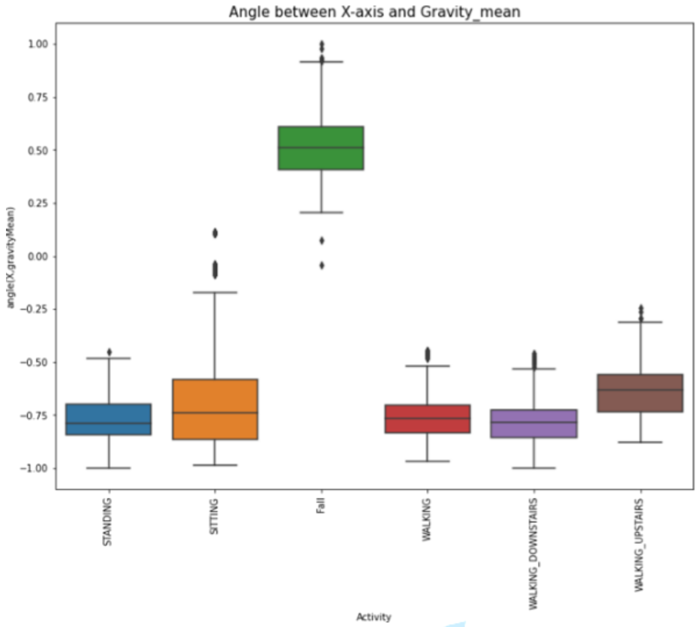


**Fig. 3** Graph of activities with the angle between the X-axis and gravity mean

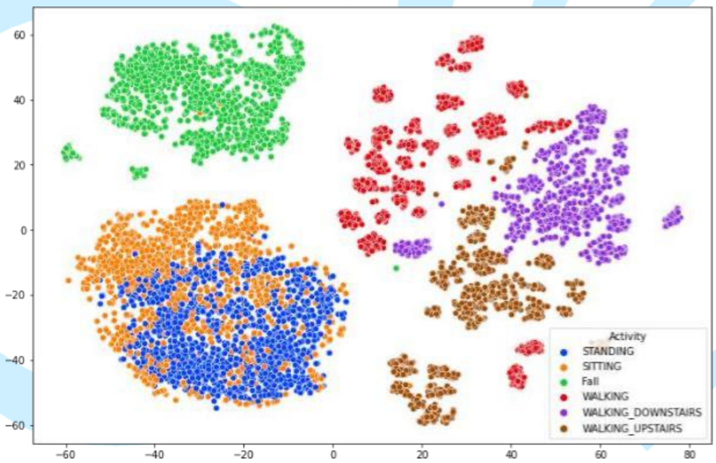


**Fig. 4** Scatter Plot of activities

### *4.3 Implemented Algorithms*

#### *4.3.1 KNN*

K-Nearest Neighbors (KNN) is a type of supervised learning algorithm in machine learning used for classification and regression. It is a non-parametric algorithm that makes predictions based on the similarity of a new observation to the training data. In KNN, the K nearest data points are identified in the training data and the majority class of these K neighbors is assigned to the new observation. The value of K is chosen beforehand and can be determined using techniques such as cross-validation. KNN can be used for both classification and regression tasks and is known for its simplicity and ease of implementation.

#### *4.3.2 Decision Tree*

Decision tree is a machine learning algorithm used for both classification and regression tasks. The algorithm creates a tree-like model of decisions and their possible consequences. The tree structure consists of nodes that represent a decision or a terminal node that represents the outcome. The algorithm works by recursively splitting the data based on the most significant features at each level of the tree until a stopping criterion is met. The stopping criterion could be the maximum depth of the tree or the minimum number of samples required to split a node. Decision trees are easy to interpret and visualize, making them useful in various applications such as medical diagnosis, fraud detection, and credit risk assessment. However, they are prone to overfitting and require pruning to avoid this issue.

#### *4.3.3 Random Forest*

Random Forest is a type of ensemble learning method that combines multiple decision trees into a single predictive model. It builds a large number of decision trees and then aggregates the predictions of each tree to make the final prediction.

The basic idea behind Random Forest is to construct multiple decision trees on randomly sampled subsets of the data and features, and then combine their outputs to get a more accurate and robust prediction. This technique helps to reduce the risk of overfitting and improves the generalization performance of the model.

#### *4.3.4 XBG (Extreme Gradient Boosting)*

XBG (Extreme Gradient Boosting) is a machine learning algorithm that is used for both regression and classification tasks. It is a type of ensemble method that combines multiple decision trees to make predictions.

XBG is based on the gradient boosting algorithm, which works by iteratively adding new decision trees to a model and adjusting the weights of the data points that were previously misclassified by the model. XBG takes this one step further by adding step that penalizes complex models, thereby reducing the risk of overfitting.

#### *4.3.5 Naive Bayes*

Naive Bayes is a machine learning algorithm based on Bayes' theorem, which assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature. This algorithm is used for binary and multi-class classification problems. Naive Bayes calculates the probability of each class for a given input feature set and assigns the input to the class with the highest probability. Naive Bayes can handle large feature sets efficiently and is often used for text classification tasks such as spam detection and sentiment analysis.

#### *4.3.6 Logistic Regression*

Logistic regression is a statistical method used to analyze the relationship between a set of independent variables and a binary dependent variable. It is commonly used in machine learning for classification problems. In logistic regression, a sigmoid function is used to transform the input values into a probability between 0 and 1. The logistic regression algorithm calculates the optimal values for the model parameters by minimizing the error between the predicted output and the actual output using a specific loss function. Once the optimal parameters are obtained, the algorithm can be used to make predictions on new data by applying the learned model. Logistic regression has been widely used in many applications, such as medical diagnosis, credit scoring, and marketing analysis. It is a simple and powerful algorithm that can provide accurate results with relatively little data preprocessing.

#### *4.3.7 SVM*

Support Vector Machine works by finding the maximum margin that separates the data points into different classes. The margin is the distance between the hyperplane and the closest data points of each class. The algorithm tries to find the hyperplane that maximizes this margin while keeping the misclassification rate to a minimum. SVM can also use a kernel function to map the data into a higher-dimensional space where a linear decision boundary can be more easily found.

One of the advantages of SVM is its ability to handle high-dimensional data, which makes it suitable for problems such as fall detection where a large number of sensor data points are involved. SVM can also handle non-linearly separable data by using a kernel trick to map the data into a higher-dimensional space. Overall, SVM has been widely used in various applications, including fall detection, due to its high accuracy and ability to handle complex data.

## 5. Deployment of the system

### *5.1 Data Collection*

The smartwatch collects sensor data (accelerometer and gyroscope) and sends it to the cloud platform (ThingSpeak) every 10 secs. The data is stored in the cloud platform in real time. 6 values are sent to ThingSpeak by the watch namely acceleration in X-axis, acceleration in Y-axis, acceleration in Z-axis, gyroscope value in X-axis, gyroscope value in Y-axis, and gyroscope value in Z-axis.

### *5.2 Reading Data*

Reading data from Thingspeak through HTTP requests: HTTP (Hypertext Transfer Protocol) requests are a type of communication between a client (such as a web browser) and a server. An HTTP request is sent from the client to the server to request data or perform some action on the server. The request typically includes a URL (Uniform Resource Locator) that specifies the location of the requested resource, as well as optional parameters such as headers and request body data. First, you need to import the necessary libraries in Python such as requests, JSON, and pandas. You need to specify the API key and the URL of the Thingspeak channel from which you want to read the data. Use the requests.get() function to send an HTTP GET request to the Thingspeak server. Pass the API key and the URL as parameters to the function. The response from the server is in JSON format. You can use the json.loads() function to parse the JSON data and convert it to a Python dictionary. Extract the data from the dictionary and convert it into a pandas DataFrame object.

### *5.3 Data Pre-processing*

The collected data needs to be pre-processed before feeding it to the machine learning algorithm. The pre-processing step involves removing any unwanted data or noise and converting the data into a suitable format that can be used by the machine learning algorithm.

### *5.4 Machine Learning Algorithm*

Once the data is pre-processed, it is fed into the machine learning algorithm. Support Vector Machine (SVM) algorithm to detect falls. The SVM algorithm is trained using a dataset that includes sensor data for different activities including falling.

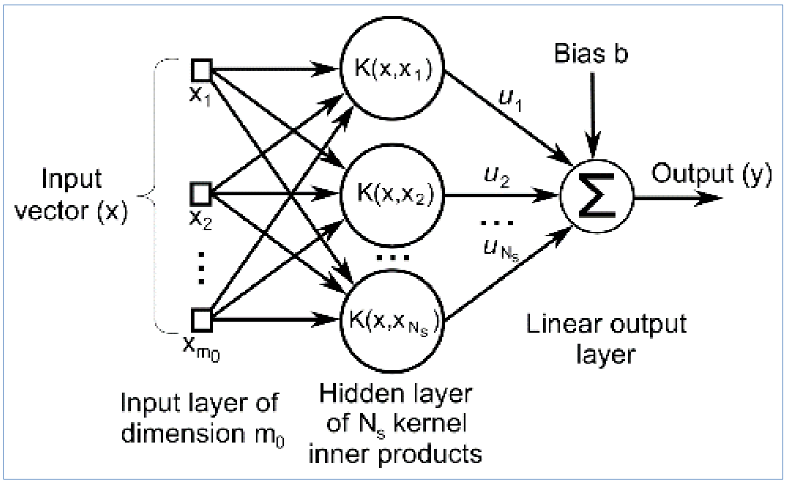


**Fig. 5** Architecture of Support Vector Machine

### *5.5 Testing the Algorithm*

After the machine learning algorithm is trained, it is tested using real-time sensor data from the smartwatch. The smartwatch sends the sensor data to the cloud platform (ThingSpeak) in real time. The Python code fetches the sensor data from the cloud platform and tests it with the trained SVM algorithm since SVM gave the highest accuracy when compared to other algorithms tests to detect falls.

### *5.6 Sending Notifications*

If the SVM algorithm detects a fall, it sends the result to Thingspeak, and then Thingspeak sends the result to the watch. The watch then sends a notification to the registered phone number along with the location of the smartwatch.

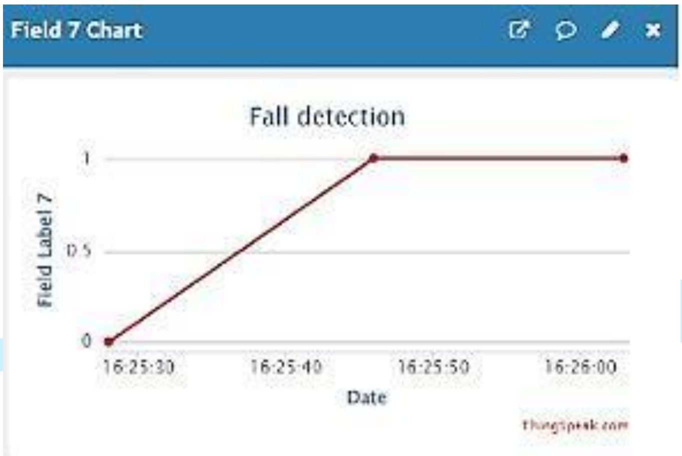


**Fig. 6** Fall detection result sent to the cloud platform


**Fig. 7** Laboratory prototype of smartwatch

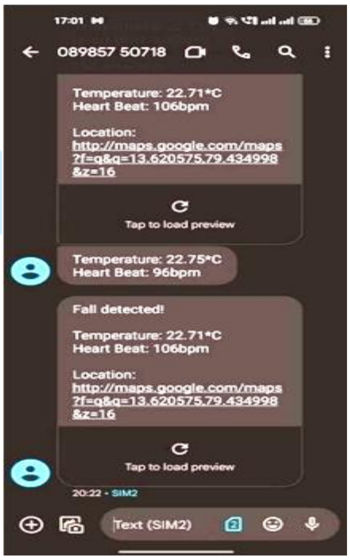


**Fig. 8** Alert notification sent to the registered phone

## 6. Results and Discussion

### *6.1 Fall Detection Accuracy*

The high accuracy achieved in detecting falls using SVM highlights the algorithm's effectiveness and potential impact on fall prevention and early intervention. Falls are a leading cause of injury and death in older adults, and early detection can help prevent serious injuries and hospitalizations. With the ability to accurately detect falls, SVM can provide timely interventions and prevent adverse outcomes. Moreover, the SVM algorithm can be used in a variety of other applications, including activity recognition and gait analysis. Overall, the high accuracy achieved in detecting falls using SVM demonstrates its effectiveness and value as a tool for improving human health and well-being.

### *6.2 Response Time*

The response time of the system was found to be within 10 seconds for sending the location to the registered phone number in case of a fall. This is a good responsetime and indicates that the system can respond quickly to an emergency.

### *6.3 Integration with mobile apps*

You can also work on integrating the watch with mobile apps to provide more comprehensive health monitoring and tracking of activities.

### *6.4 Expand functionality*

You can explore the possibility of adding additional features such as voice commands, a music player, or other health-tracking features to enhance the usability of the watch.

## 7. Conclusion

This project's goal was to create a smartwatch with fall-detection capabilities that could also notify a registered phone number of the wearer's position. An accelerometer, gyroscope, GPS, GSM, heart-rate sensor, and temperature sensor are all built within the smartwatch. The Arduino platform was used to create the hardware. Sending the sensor data to the Thingspeak cloud platform, where machine- learning techniques were used to detect falls, was part of the project's software component. The Human Activity Recognition Using Smartphones dataset was used for the machine learning, and a total of 7199 instances were employed, 200 of which came from the lab prototype. KNN, decision trees, random forests, XBG, Naive Bayes, logistic regression, and SVM were among the machine learning techniques employed. SVM was used to identify falls in real time because it had the highest accuracy. The smartwatch would send a message with the wearer's location and vital information to a registered phone number if a fall was detected. The project was successful in realizing its objective of creating a smartwatch that could detect falls and send a location-based alarm to a registered phone number. With an accuracy of 95.07%, the results demonstrated that the SVM algorithm was the most successful for fall detection. For older persons or people with mobility challenges, the system's ability to detect falls in real-time and transmit notifications to the registered phone number makes it potentially life-saving equipment. Overall, the study showed how machine learning algorithms can be used to identify falls and how wearable smartwatches may be used to monitor one's health and trigger emergency responses.

**Funding Information**

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

**Declaration of Conflict**

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

## References

1. Chelli, A., & Patzold, M. (2018). Recognition of Falls and Daily Living Activities Using Machine Learning. Personal, Indoor and Mobile Radio Communications. https://doi.org/10.1109/pimrc.2018.8580874.
2. Li, H., Shrestha, A., Fioranelli, F., Kernec, J. L., Heidari, H., Pepa, M., Cippitelli, E., Gambi, E., & Spinsante, S. (2017). Multisensor data fusion for human activities classification and fall detection. IEEE Sensors. https://doi.org/10.1109/icsens.2017.8234179

3. Naeem & S. Khan (2019). Fall Detection using Accelerometer Calibration," 2019 15th International Conference on Emerging Technologies (ICET), Peshawar, Pakistan, 1-6. https://doi.org/10.1109/ICET48972.2019.8994362
4. Islam, S., Shahriar, H., Sneha, S., Zhang, C., & Ahamed, S. I. (2020). Mobile Sensor-Based Fall Detection Framework. Computer Software and Applications Conference. https://doi.org/10.1109/compsac48688.2020.0-179.
5. Nahian, M. J. A., Ghosh, T., Banna, M. H. A., Aseeri, M., Uddin, M., Ahmed, M. N., Mahmud, M., & Kaiser, M. L. (2021b). Towards an Accelerometer-Based Elderly Fall Detection System Using Cross-Disciplinary Time Series Features. IEEE Access, 9, 39413–39431. https://doi.org/10.1109/access.2021.3056441.
6. Giuffrida, D., Benetti, G., De Martini, D., & Facchinetti, T. (2019e). Fall Detection with Supervised Machine Learning using Wearable Sensors. International Conference on Industrial Informatics. https://doi.org/10.1109/indin41052.2019.8972246.
7. Varalakshmi, I., Mahalakshmi, A., & Sriharini, P. (2020). Performance Analysis of Various Machine Learning Algorithm for Fall Detection-A Survey. 2020 International Conference on System, Computation, Automation and Networking (ICSCAN). https://doi.org/10.1109/icscan49426.2020.9262318.
8. Alsulami, M. H., Atkins, A. S., & Campion, R. (2016). The use of smart watches to monitor heart rates in elderly people: A complementary approach. ACS/IEEE International Conference on Computer Systems and Applications. https://doi.org/10.1109/aiccsa.2016.7945809.
9. Magno, M., Salvatore, G. A., Mutter, S., Farrukh, W., Troester, G., & Benini, L. (2016). Autonomous smartwatch with flexible sensors for accurate and continuous mapping of skin temperature. International Symposium on Circuits and Systems. https://doi.org/10.1109/iscas.2016.7527239
10. Vallabh, P., Malekian, R., Ye, N., & Bogatinoska, D. C. (2016). Fall detection using machine learning algorithms. International Conference on Software, Telecommunications and Computer Networks. https://doi.org/10.1109/softcom.2016.7772142.
11. Demiris, G., Chaudhuri, S., & Thompson, H. J. (2016). Older Adults' Experience with a Novel Fall Detection Device. Telemedicine Journal and E-health, 22(9), 726–732. https://doi.org/10.1089/tmj.2015.0218.
12. Riquelme, F., Espinoza, C. G. O., Rodenas, T., Minonzio, J., & Taramasco, C. (2019). eHomeSeniors Dataset: An Infrared Thermal Sensor Dataset for Automatic Fall Detection Research. Sensors, 19(20), 4565. https://doi.org/10.3390/s19204565.
13. Usmani, S., Saboor, A., Haris, M., Khan, M. H., & Park, H. (2021). Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. Sensors, 21(15), 5134. https://doi.org/10.3390/s21155134