History of Medicine, 2024, 10(2): 89–99

DOI: 10.17720/2409-5834.v10.2.2024.11

FALL SENSE: A WEARABLE INERTIAL SENSOR -BASED DATASET FOR PRE-IMPACT FALL DETECTION IN ELDERLY INDIVIDUALS

Kamaraju Pandian[1], Simran[2], P. Pavan Kumar[2], S. Rohan[2]

[1]Assist. Professor, [2]UG Scholar, [1,2]Department of Computer Science & Engineering (Data Science),

[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, Telangana.

ABSTRACT

Falls among elderly individuals pose a significant health risk and are a leading cause of injury-related hospitalizations. Early detection of falls can greatly mitigate the severity of injuries and improve the overall well-being of the elderly population. This paper presents "Fall Sense," a novel approach for pre-impact fall detection in elderly individuals using a dataset collected from wearable inertial sensors. In the introduction, we highlight the importance of fall detection in elderly care and emphasize the need for accurate and timely detection to reduce the adverse consequences of falls. The conventional fall detection systems are discussed, revealing their limitations, such as low accuracy, high false alarm rates, and the need for extensive infrastructure. These drawbacks hinder their practicality and effectiveness in real-world scenarios. In response to these limitations, our proposed system leverages wearable inertial sensors, which are unobtrusive and can be comfortably worn by elderly individuals. These sensors continuously collect data on motion and acceleration, which are then processed using machine learning algorithms. Our dataset, "Fall Sense," is introduced, containing a diverse set of activities, including falls, activities of daily living, and simulated falls, making it a valuable resource for training and evaluating machine learning models for fall detection. The proposed system employs machine learning techniques to analyze sensor data in real-time, enabling the detection of fall-related patterns and anomalies before the impact occurs. This early detection allows for timely intervention, such as alerting caregivers or activating emergency response systems, ultimately improving the safety and well-being of elderly individuals. Fall Sense represents a promising advancement in fall detection technology, offering the potential to reduce the impact of falls on the elderly population and enhance their quality of life.

Keywords: Fall Sense, Machine learning, Low false alarm rates, Health risk mitigation.

1. INTRODUCTION

The history of fall detection systems for elderly individuals traces back to the growing concern over the safety and well-being of aging populations. As societies around the world experienced demographic shifts with increased life expectancy, the need to address the risks associated with aging, such as falls, became more pressing. Traditional approaches to fall detection initially relied on manual monitoring or simple alarm systems. However, these methods often lacked accuracy and failed to provide timely assistance in case of a fall. With advancements in technology, particularly in sensor technology and machine learning, researchers began exploring more sophisticated solutions for fall detection. Early attempts involved the use of basic motion sensors or wearable devices to detect falls. However, these systems often suffered from high false alarm rates or limited sensitivity to different types of falls and activities. Despite these advancements, challenges remained in achieving robust and reliable fall detection. Researchers continued to refine algorithms, improve sensor technology, and collect larger datasets to train and evaluate their systems. Additionally, there was a growing emphasis on user-

friendliness and acceptance among elderly individuals, leading to the design of more comfortable and discreet wearable devices.

2. LITERATURE SURVEY

According to the World Health Organization, in 2018, it was reported for the first time in history that the number of people over 65 exceeded the number of people under 5 years of age worldwide [1]. Therefore, it is estimated that by 2050, one in six people will belong to the elderly sector [2]. The fact that elderly people live alone raises concerns because they might not have access to continuous health status monitoring. Falls are the second leading cause of death worldwide, causing injuries that require immediate medical attention; otherwise, they can be fatal [3]. Unfortunately, the rate of deaths caused by falls continues to increase each year, and if this growth continues, it is anticipated that by 2030, there will be 7 fatalities every hour [4]. In addition, several studies show that if falls are not fatal, there is a possibility that people present physical and psychological complications for the rest of their lives [5]. Some physical complications include restrictions in daily activities, joint pain, broken bones, or brain injuries, among others. Falls are the most common mechanism for causing traumatic brain injuries [6]. Psychologically, the person struggles with sadness, a lack of self-assurance, and a fear of falling again. On the other hand, statistics show that 80% of fall deaths are in low- and middle-income countries [3]. For this reason, the development of low-cost fall detection systems is a critical need. This work aims at detecting human falls through radio sensing based on a continuous wave (CW) radio-frequency (RF) probe signal that can be transmitted as a pilot signal within the communications signal frame. Pilot signals do not carry information data and are used only for synchronization purposes between the transmitter and the receiver [7]. However, these signals are subject to frequency dispersions caused by the Doppler effect. Such frequency dispersions are known as Doppler signatures and have been widely used for sensing purposes [8,9,10]. The Doppler signatures produced by the interaction of the CW probe signal with a moving person can be analyzed to characterize falling events. We developed a CW probe signal transmission and reception system using general-purpose equipment that can be easily replicated. Unlike other works where a single-input multiple-output scheme is followed [11.12.13], our system only requires a single-input single-output (SISO) radio link. Furthermore, the platform allows the acquisition of the information from the probe signals through a less complex process than channel state information (CSI) or received signal strength (RSSI) estimation.

3. PROPOSED SYSTEM

This project demonstrates a comprehensive approach to a fall sense project, encompassing data loading, preprocessing, model training, and evaluation. Below is a detailed explanation of each step in a human-readable manner:

Dataset Upload: The research begins with the importation of necessary libraries and the loading of the inertial sensor dataset. The dataset is stored in a Pandas DataFrame (df), allowing for easy manipulation and analysis.

Data Exploration and Analysis: Basic exploratory data analysis (EDA) is performed to gain insights into the dataset. Descriptive statistics, including mean, standard deviation, and quartiles, are obtained using the describe() method. The info() method is employed to examine the data types and null values in each column.

Visualization of Decision Counts: The distribution of decision classes is visualized using a count plot with seaborn. This provides a quick overview of the balance or imbalance in the target variable, 'Decision'.

Preprocessing: Null values are checked for and identified throughout the dataset. The independent variables are scaled using the StandardScaler from scikit-learn, ensuring that all features have a similar scale. This is crucial for models that are sensitive to the magnitude of input features.

Train-Test Splitting: The dataset is split into training and testing sets using the train_test_split function. The testing set comprises 20% of the data, and a random seed is set for reproducibility.

Logistic Regression Model: A logistic regression model is instantiated and trained on the training set (X_train and y_train). The model is then tested on the reserved testing set (X_test), and the accuracy, confusion matrix, and classification report are displayed.

Residual Neural Network Model: An Residual Neural Network model is constructed using the Keras library. The architecture includes an input layer with 64 neurons, a hidden layer with 32 neurons using the ReLU activation function, and an output layer with a sigmoid activation function for binary classification. The model is compiled using binary cross-entropy loss and the Adam optimizer.

Model Training and Evaluation (Residual Neural Network): The Residual Neural Network model is trained on the resampled training set (X_train_smote and y_train_smote) using the Synthetic Minority Over-sampling Technique (SMOTE) for handling imbalanced data. The model is then evaluated on the original testing set, and accuracy, classification report, and confusion matrix are displayed.

ROC Curve Analysis: Receiver Operating Characteristic (ROC) curves are generated for both the logistic regression and Residual Neural Network models. The curves visually represent the trade-off between true positive rate and false positive rate, with the area under the curve (AUC) serving as a performance metric.

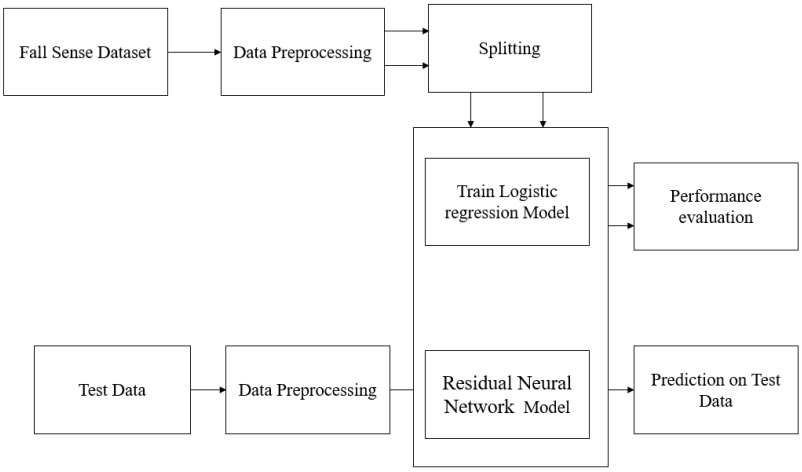


Figure 1: Block Diagram of Proposed System.

3.2 Residual Neural Network Classifier

Although today the Perceptron is widely recognized as an algorithm, it was initially intended as an image recognition machine. It gets its name from performing the human-like function of perception, seeing, and recognizing images. Interest has been centered on the idea of a machine which would be capable of conceptualizing inputs impinging directly from the physical environment of light, sound, temperature, etc. — the "phenomenal world" with which we are all familiar — rather than requiring the intervention of a human agent to digest and code the necessary information. Rosenblatt's perceptron machine relied on a basic unit of computation, the neuron. Just like in previous models, each neuron has a cell that receives a series of pairs of inputs and weights. The major difference in Rosenblatt's

model is that inputs are combined in a weighted sum and, if the weighted sum exceeds a predefined threshold, the neuron fires and produces an output.

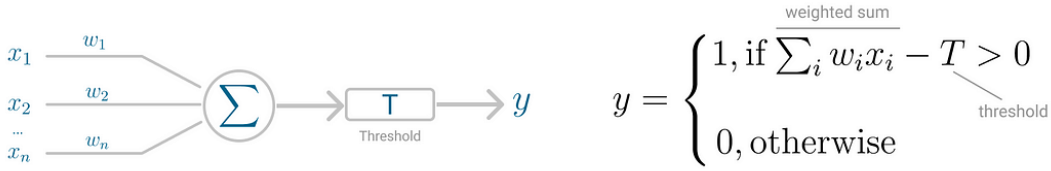


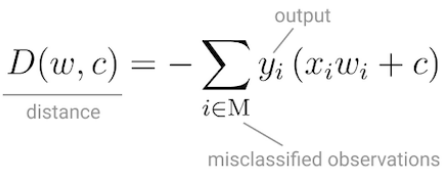
Fig. 2: Perceptron neuron model (left) and threshold logic (right).

Threshold *T* represents the activation function. If the weighted sum of the inputs is greater than zero the neuron outputs the value 1, otherwise the output value is zero.

Perceptron for Binary Classification

With this discrete output, controlled by the activation function, the perceptron can be used as a binary classification model, defining a linear decision boundary.

It finds the separating hyperplane that minimizes the distance between misclassified points and the decision boundary. The perceptron loss function is defined as below:



To minimize this distance, perceptron uses stochastic gradient descent (SGD) as the optimization function. If the data is linearly separable, it is guaranteed that SGD will converge in a finite number of steps. The last piece that Perceptron needs is the activation function, the function that determines if the neuron will fire or not. Initial Perceptron models used sigmoid function, and just by looking at its shape, it makes a lot of sense! The sigmoid function maps any real input to a value that is either 0 or 1 and encodes a non-linear function. The neuron can receive negative numbers as input, and it will still be able to produce an output that is either 0 or 1.

But, if you look at Deep Learning papers and algorithms from the last decade, you'll see the most of them use the Rectified Linear Unit (ReLU) as the neuron's activation function. The reason why ReLU became more adopted is that it allows better optimization using SGD, more efficient computation and is scale-invariant, meaning, its characteristics are not affected by the scale of the input. The neuron receives inputs and picks an initial set of weights random. These are combined in weighted sum and then ReLU, the activation function, determines the value of the output.

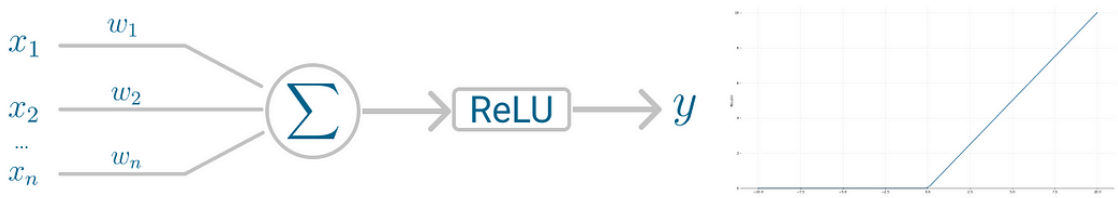


Fig. 3: Perceptron neuron model (left) and activation function (right).

Perceptron uses SGD to find, or you might say learn, the set of weight that minimizes the distance between the misclassified points and the decision boundary. Once SGD converges, the dataset is

separated into two regions by a linear hyperplane. Although it was said the Perceptron could represent any circuit and logic, the biggest criticism was that it couldn't represent the XOR gate, exclusive OR, where the gate only returns 1 if the inputs are different. This was proved almost a decade later and highlights the fact that Perceptron, with only one neuron, can't be applied to non-linear data.

3.2.2 Residual Neural Network

The Residual Neural Network was developed to tackle this limitation. It is a neural network where the mapping between inputs and output is non-linear. A Residual Neural Network has input and output layers, and one or more hidden layers with many neurons stacked together. And while in the Perceptron the neuron must have an activation function that imposes a threshold, like ReLU or sigmoid, neurons in a Residual Neural Network can use any arbitrary activation function. Residual Neural Network falls under the category of feedforward algorithms because inputs are combined with the initial weights in a weighted sum and subjected to the activation function, just like in the Perceptron. But the difference is that each linear combination is propagated to the next layer. Each layer is feeding the next one with the result of their computation, their internal representation of the data. This goes all the way through the hidden layers to the output layer. If the algorithm only computed the weighted sums in each neuron, propagated results to the output layer, and stopped there, it wouldn't be able to learn the weights that minimize the cost function. If the algorithm only computed one iteration, there would be no actual learning. This is where Backpropagation comes into play.

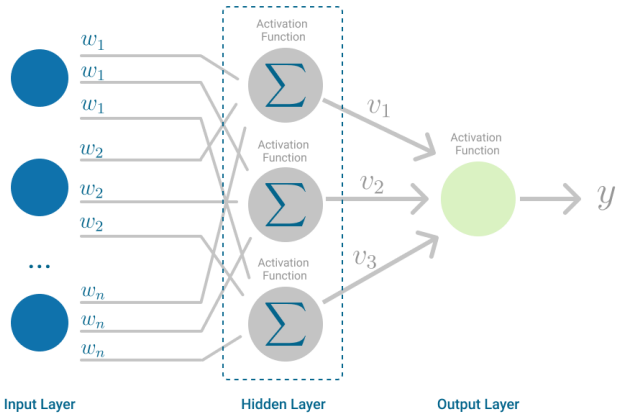


Fig. 4: Architecture of Residual Neural Network.

Backpropagation: Backpropagation is the learning mechanism that allows the Residual Neural Network to iteratively adjust the weights in the network, with the goal of minimizing the cost function. There is one hard requirement for backpropagation to work properly. The function that combines inputs and weights in a neuron, for instance the weighted sum, and the threshold function, for instance ReLU, must be differentiable. These functions must have a bounded derivative because Gradient Descent is typically the optimization function used in Residual Neural Network . In each iteration, after the weighted sums are forwarded through all layers, the gradient of the Mean Squared Error is computed across all input and output pairs. Then, to propagate it back, the weights of the first hidden layer are updated with the value of the gradient. That's how the weights are propagated back to the starting point of the neural network. One iteration of Gradient Descent is defined as follows:

$$\underbrace{\Delta_w(t)}_{\substack{\text{Gradient}\\\text{Current Iteration}}} = \underbrace{-\varepsilon}_{\text{Bias}} \underbrace{\frac{dE}{dw_{(t)}}}_{\substack{\text{Error}\\\text{Weight vector}}} + \underbrace{\alpha\Delta_{w(t-1)}}_{\substack{\text{Learning Rate}\\\text{Gradient}\\\text{Previous Iteration}}}$$

This process keeps going until gradient for each input-output pair has converged, meaning the newly computed gradient hasn't changed more than a specified convergence threshold, compared to the previous iteration.

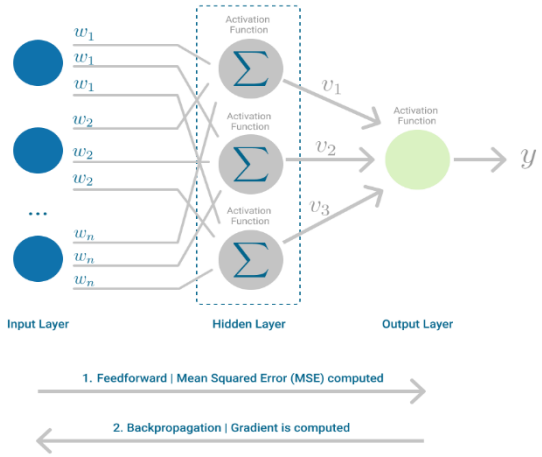


Fig. 5: Residual Neural Network, highlighting the Feedforward and Backpropagation steps.

3.3 Advantages

Comprehensive Data Analysis: The reserch begins with a thorough exploration and analysis of the dataset, providing descriptive statistics and visualizations. This step aids in understanding the characteristics and distribution of the data.

Preprocessing for Model Readiness: Null value checks and preprocessing techniques, such as standard scaling, are applied to the dataset. This ensures that the data is suitable for training machine learning models by addressing missing values and normalizing feature scales.

Visualization of Decision Classes: The use of a count plot to visualize the distribution of decision classes enhances the understanding of the dataset's class balance or imbalance. This insight is crucial for selecting appropriate modeling techniques, particularly in the context of imbalanced datasets.

Logistic Regression Model: The inclusion of a logistic regression model provides a baseline for binary classification. Logistic regression is a simple yet effective algorithm for such tasks, making it suitable for comparison with more complex models.

Residual Neural Network Model: The research introduces an Residual Neural Network model, a more sophisticated and flexible approach capable of capturing complex patterns in the data. ANNs are well-suited for tasks with non-linear relationships between features and outcomes.

Handling Imbalanced Data with SMOTE: The utilization of the Synthetic Minority Over-sampling Technique (SMOTE) during the training of the Residual Neural Network model addresses potential class imbalance. SMOTE generates synthetic samples of the minority class, promoting a more balanced representation and potentially improving model performance.

Model Evaluation and Comparison:

The research evaluates and compares the performance of both the logistic regression and Residual Neural Network models. This comparison allows for an assessment of whether the added complexity of the Residual Neural Network model results in significant performance improvements.

Confusion Matrix and Classification Report: The inclusion of confusion matrices and classification reports provides a detailed breakdown of model performance, including metrics such as precision, recall, and F1-score. This information is crucial for understanding the model's strengths and weaknesses.

ROC Curve Analysis: The research incorporates Receiver Operating Characteristic (ROC) curves and Area Under the Curve (AUC) analysis. These metrics offer a visual representation of model performance, especially in terms of the trade-off between true positive rate and false positive rate.

Educational Value:

The research is educational and serves as a practical example of building, training, and evaluating machine learning models. It introduces users to fundamental concepts such as data preprocessing, model training, and performance evaluation.

Flexibility and Extensibility:

The modular structure of the code allows for flexibility and extensibility. Users can easily modify the code to experiment with different models, hyperparameters, or additional preprocessing steps to tailor the research to specific needs.

4. RESULTS AND DISCUSSION

Figure 6 is the graphical representation of the dataset uploaded within the Fall Sense GUI. The uploaded dataset includes various features relevant to fall detection, such as distance, pressure, heart rate variability (HRV), sugar level, SpO2 (oxygen saturation), accelerometer readings, and the decision variable indicating the outcome of fall detection. The GUI provides users with a convenient platform to visualize the uploaded dataset, facilitating data exploration and understanding. Each row in the dataset likely corresponds to a specific instance or observation, while each column represents a different feature or attribute. By displaying the uploaded dataset within the GUI, users can quickly inspect the data and identify any patterns or anomalies that may be present.

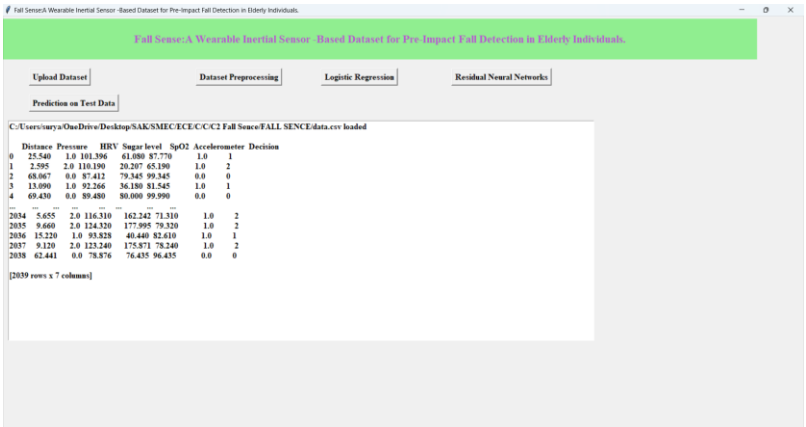


Figure 6: Displays the Uploaded dataset in the Fall Sence GUI.

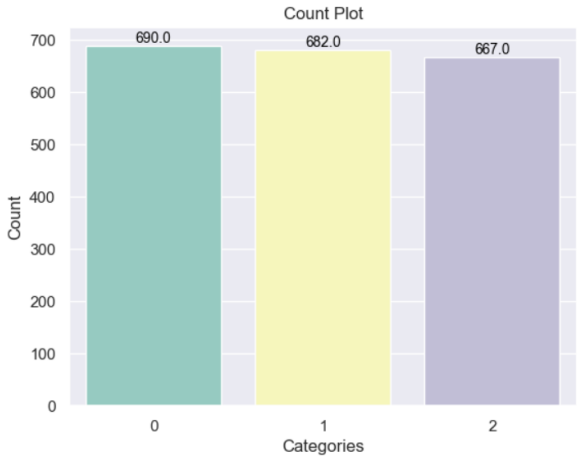


Figure 7: Presents the Count plot for each Categories.

Figure 7 presents a count plot depicting the distribution of categories within the dataset. This plot provides valuable insights into the balance or imbalance of classes within the dataset, particularly concerning the decision variable indicating fall detection outcomes. By visualizing the count of each category, users can assess whether the dataset is balanced or skewed towards specific classes. A balanced dataset, where each class is represented roughly equally, is preferable for training machine learning models as it prevents bias towards dominant classes. Conversely, an imbalanced dataset may require techniques such as oversampling or undersampling to ensure fair representation of all classes during model training. The count plot serves as a useful tool for data exploration and preprocessing, enabling users to make informed decisions regarding model training and evaluation strategies.

Figure 8: Shows the preprocessed dataset in the GUI console.

Figure 8 shows the preprocessed dataset displayed within the GUI console. The preprocessing steps include normalization and splitting the dataset into training and testing sets, essential tasks for preparing data for machine learning model training. Normalization ensures that features are on a similar scale, preventing certain features from dominating others during model training. Splitting the dataset into training and testing sets allows users to train the model on a subset of the data and evaluate its performance on unseen data. By displaying the preprocessed dataset within the GUI console, users can verify that the data has been processed correctly and is ready for model training and evaluation. This visualization enhances transparency and reproducibility, enabling users to understand and validate each step of the data preprocessing pipeline.

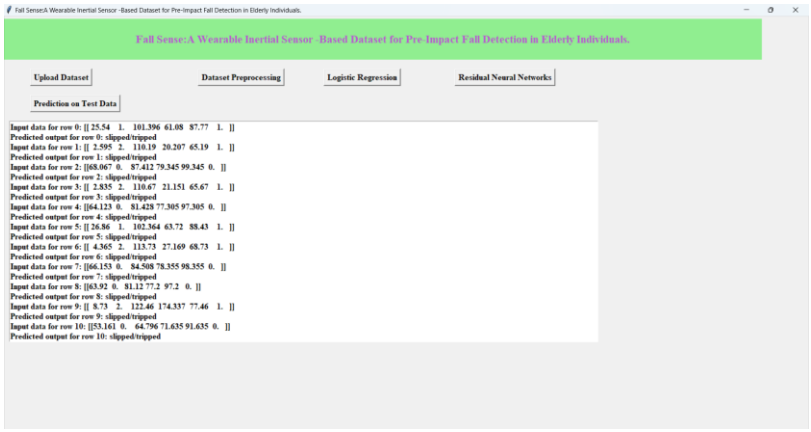


Figure 9: Represents the Proposed model prediction on test data.

Figure 9 represents the predicted outcomes of the proposed model on test data. The model utilizes machine learning algorithms, such as logistic regression or a neural network, trained on the preprocessed dataset to predict fall detection outcomes based on input features. The predicted outcomes are displayed in the GUI, allowing users to assess the model's performance and accuracy. By comparing the predicted outcomes to the ground truth labels in the test data, users can evaluate the model's ability to correctly classify instances of falls and non-falls. This visualization provides valuable insights into the model's predictive capabilities and informs decisions regarding its deployment in real-world scenarios. Additionally, users can identify any misclassifications or errors made by the model, leading to iterative improvements and refinements to enhance its performance. Overall, Figure 4 offers a comprehensive overview of the proposed model's predictions on test data within the Fall Sense GUI, enabling users to make informed decisions regarding fall detection in elderly individuals.

Table 1: Performance metrics of each model.

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-Score (%)**
Logistic Regression	85.78	89.61	85.35	84.46
Residual Neural Network	31.37	10.46	33.33	15.92

In the table above, the performance metrics of two models, Logistic Regression and Residual Neural Network (RNN), are presented. These metrics include Accuracy, Precision, Recall, and F1-Score. Here's a detailed explanation of each metric:

1. **Accuracy:** Accuracy measures the proportion of correct predictions made by the model out of the total number of predictions. For the Logistic Regression model, the accuracy is 85.78%, indicating that 85.78% of the predictions made by the model are correct. However, for the RNN model, the accuracy is significantly lower at 31.37%, suggesting that the RNN model's predictive performance is not as reliable as the Logistic Regression model.
2. **Precision:** Precision measures the proportion of true positive predictions out of all positive predictions made by the model. In the context of fall detection, precision indicates the model's ability to correctly identify true falls among all instances predicted as falls. The Logistic Regression model achieves a precision of 89.61%, indicating that 89.61% of the instances predicted as falls by the model are true falls. On the other hand, the RNN model's precision is

substantially lower at 10.46%, suggesting that it tends to misclassify non-fall instances as falls more frequently.

3. **Recall:** Recall, also known as sensitivity, measures the proportion of true positive predictions out of all actual positive instances in the dataset. It indicates the model's ability to capture all positive instances correctly. For the Logistic Regression model, the recall is 85.35%, indicating that 85.35% of true falls in the dataset are correctly identified by the model. Conversely, the RNN model's recall is 33.33%, suggesting that it misses a significant number of true fall instances.
4. **F1-Score:** F1-Score is the harmonic mean of precision and recall and provides a balanced measure of a model's performance. It takes both false positives and false negatives into account and is particularly useful when dealing with imbalanced datasets. The F1-Score of the Logistic Regression model is 84.46%, reflecting a good balance between precision and recall. However, the RNN model's F1-Score is considerably lower at 15.92%, indicating that it struggles to achieve a balance between precision and recall, likely due to a high number of false positives and false negatives.

5. CONCLUSION

The "Fall Sense" system presented in this paper offers a promising solution for pre-impact fall detection in elderly individuals using wearable inertial sensors and machine learning algorithms. By leveraging these technologies, we address the limitations of conventional fall detection systems, such as low accuracy and the need for extensive infrastructure. Our approach provides real-time analysis of motion and acceleration data, enabling early detection of fall-related patterns and anomalies. This timely detection facilitates prompt intervention, enhancing the safety and well-being of elderly individuals by reducing the severity of fall-related injuries.

REFERENCES

[1]. Nations, U. Ageing. 2019. Available online: https://www.un.org/en/global-issues/ageing (accessed on 20 August 2021).
[2]. WHO. Ageing and Health. 2022. Available online: https://www.who.int/news-room/fact-sheets/detail/ageing-and-health (accessed on 12 November 2022).
[3]. WHO. Falls. 2018. Available online: https://www.who.int/news-room/fact-sheets/detail/falls (accessed on 20 August 2021).
[4]. Centers for Disease Control and Prevention. Older Adult Fall Prevention. 2021. Available online: https://www.cdc.gov/falls/facts.htmls (accessed on 12 November 2022).
[5]. Orihuela-Espejo, A.; Álvarez-Salvago, F.; Martínez-Amat, A.; Boquete-Pumar, C.; De Diego-Moreno, M.; García-Sillero, M.; Aibar-Almazán, A.; Jiménez-García, J.D. Associations between Muscle Strength, Physical Performance and Cognitive Impairment with Fear of Falling among Older Adults Aged 60 Years: A Cross-Sectional Study. Int. J. Environ. Res. Public Health 2022, 19, 10504. [Google Scholar] [CrossRef] [PubMed]
[6]. Jager, T.E.; Weiss, H.B.; Coben, J.H.; Pepe, P.E. Traumatic Brain Injuries Evaluated in U.S. Emergency Departments, 1992–1994. Acad. Emerg. Med. 2000, 7, 134–140. [Google Scholar] [CrossRef] [PubMed]
[7]. IEEE Computer Society LAN/MAN Standards Committee. IEEE Standard for Information technology—Telecommunications and information exchange between systems-Local and metropolitan area networks-Specific requirements Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. IEEE Std. 2007, 802, 11. [Google Scholar]

[8]. Abdelgawwad, A.; Mallofre, A.C.; Patzold, M. A Trajectory-Driven 3D Channel Model for Human Activity Recognition. IEEE Access 2021, 9, 103393–103406. [Google Scholar]

[9]. Saho, K.; Hayashi, S.; Tsuyama, M.; Meng, L.; Masugi, M. Machine Learning-Based Classification of Human Behaviors. Sensors 2022, 22, 1721. [Google Scholar]

[10]. Gomez-Vega, C.A.; Cardenas, J.; Ornelas-Lizcano, J.C.; Gutierrez, C.A.; Cardenas-Juarez, M.; Luna-Rivera, J.M.; Aguilar-Ponce, R.M. Doppler Spectrum Measurement Platform for Narrowband V2V Channels. IEEE Access 2022, 10, 27162–27184.

[11]. Lee, P.W.; Seah, W.K.; Tan, H.P.; Yao, Z. Wireless sensing without sensors-an experimental study of motion/intrusion detection using RF irregularity. Meas. Sci. Technol. 2010, 21, 124007.

[12]. Muaaz, M.; Chelli, A.; Gerdes, M.W.; Pätzold, M. Wi-Sense: A passive human activity recognition system using Wi-Fi and convolutional neural network and its integration in health information systems. Ann. Telecommun. 2022, 77, 163–175.

[13]. Ding, J.; Wang, Y. A WiFi-Based Smart Home Fall Detection System Using Recurrent Neural Network. IEEE Trans. Consum. Electron. 2020, 66, 308–317.