International Journal on Robotics, Automation and Sciences

Hybrid Graph-Recurrent Architectures for Spatio-Temporal Motion Analysis and Fall Forecasting

Asyafa Ditra Al Hauna* and Masanori Fukui

Abstract **– Falls constitute a predominant etiology of injury across demographic spectra and represent a significant contributor to morbidity, mortality, and escalating healthcare expenditures in Australia. Mitigating such incidents necessitates integrating intelligent surveillance systems capable of preemptive detection and rapid intervention. To support these goals, our study advances a deep learning model designed to anticipate and classify fall-related human motion by modeling the intricate spatiotemporal interdependencies inherent in bodily dynamics. The proposed approach is developed and evaluated exclusively using the TelUP dataset. Among our exploration on graph-recurrent models, GCN-GRU operationalizes a synergistic representation of spatial posture and temporal progression to forecast imminent falls with heightened precision. Empirical validation reveals a marked performance improvement over extant baselines, attaining an accuracy of 97.40%, an F1-score of 96.89%, a mean per-joint position error (MPJPE) of 113.4 pixels, and a mean per-joint velocity error (MPJVE) of 64.66 pixels. Supplementary analyses concerning computational complexity and comparative visual evaluation corroborate the model's superior efficacy relative to prevailing architectures, notably outperforming the LSTM benchmark. Future work will evaluate the proposed approach across additional public datasets and real-world deployment settings to assess robustness while accounting for inference-intervention latency under variations in physical environments.**

Keywords— Deep Learning, Fall Detection, Gait Analysis, Human Motion Forecasting, Multi-task Learning, Pose Estimation, Spatio-temporal modelling.

I. INTRODUCTION

Ambulation constitutes a fundamental motor competence, acquired early in human ontogeny to facilitate spatial navigation and environmental interaction [1]. During this process, the neuromuscular and perceptual systems operate synergistically to enable the unconscious avoidance of obstacles and the mitigation of imminent hazards. Yet, as modern life imposes increasing cognitive and sensory demands, individuals frequently engage in concurrent activities while walking, such as consuming food or processing digital communications, overloading attentional resources and attenuating situational awareness. This cognitive multitasking compromises anticipatory motor control, diminishing one's capacity to foresee and avert destabilizing perturbations that may precipitate a fall [2]. Individuals, regardless of their age has have the potential to fall once the equilibrium system fails to sustain postural stability [3]. The immediate aftermath of a fall inaugurates the so-called "golden hour," a term in clinical medicine denoting the temporally critical phase after a traumatic injury when prompt medical attention could dramatically improve survival and recovery outcomes [4]. Consequently, the poptimizeessment of fall risk and the predictive modeling of human gait dynamics are indispensable for preempting fall incidents and optimizing the

*Corresponding Author, email: asyafaditra@gmail.com ORCID: https://orcid.org/0009-0005-3098-5386
Asyafa Ditra Al Hauna is with Department of Informatics, Telkom University Purwokerto, Indonesia. (e-mail: alditra@student.telkomuniversity.ac.id)
Masanori Fukui is with Department of Information Engineering, Mie University, 1577 Kurimamachiya-cho Tsu, Mie, 514-8507, Japan. (e-mail: fukui@info.mie-u.ac.jp).

International Journal on Robotics, Automation and Sciences (2026) 8, 2:62-73
https://doi.org/10.33093/ijoras.2026.8.2.8
Manuscript received: 01 Feb 2026 | Revised: 27 Mac 2026 | Accepted: 29 Apr 2026 | Published: : 31 Jul 2026
Published by MMU PRESS. URL: http://journals.mmupress.com/ijoras

timeliness and efficacy of medical response when prevention fails.

According to the Australian Institute of Health and Welfare (AIHW), fall-related injuries constituted the predominant cause of injury-induced mortality in Australia during the 2022–23 period, culminating in 6,698 recorded fatalities and representing approximately 43% of all hospitalized injury cases. The economic ramifications of such incidents were equally profound, with an estimated financial burden of nearly $5 billion imposed upon the national healthcare system. This pattern persisted into 2023–24, wherein fall injuries dominated the spectrum of injury hospitalizations, encompassing 248,211 admissions, again accounting for roughly 43% of all injury-related hospitalizations. The primary etiological mechanisms underlying these events were mechanical perturbations such as slipping, tripping, and stumbling [5], underscoring the interplay between human locomotor instability and environmental contingencies that collectively render falls a persistent public health concern.

Recent interdisciplinary innovations have emerged in response to the ostensibly mundane yet profoundly consequential issue confronting nations worldwide. Among the most distinguished endeavors is the deployment of sensor-based technologies for fall-risk assessment, particularly those integrated into wearable systems such as inertial measurement units (IMUs), pressure sensors, and electromyography (EMG) devices [6], [7]. Despite their utility, these wearable modalities exhibit notable limitations. They often require multiple heterogeneous sensors to various body regions, commonly the wrist or other strategic anatomical sites, or personal items intimately associated with the user [8], [9]. Environmental fluctuations in temperature and humidity, alongside inadvertent alterations in sensor configuration, can induce signal distortions, necessitating frequent recalibration to sustain optimal accuracy [10]. Furthermore, the dependency on battery maintenance imposes an additional layer of inconvenience for users. Therefore, a growing number of researchers have gravitated their work into AI-supported cameras to provide a less intrusive, but more scalable approach to fall detection and risk assessment.

Developing AI-supported camera systems necessitates sophisticated modeling grounded in advanced computational techniques. Prior research has extensively explored human motion forecasting using recurrent-based architectures such as Gated Recurrent Unit (GRU), Long Short-Term Memory network (LSTM), and Recurrent Neural Network (RNN) [11], [12], [13], [14]. However, these studies capture temporal dependencies while overlooking spatial correlations critical for accurate human pose representation. Building upon this limitation, and leveraging the TelUP dataset containing the latest available human motion data, we propose enhanced models that integrates spatial and temporal dependencies. Drawing inspiration from preceding spatio-temporal frameworks [15], [16] and building upon research of Widi *et al.*, our methodology for graph-recurrent models involves the independent integration of Graph Convolutional Networks (GCNs) and Graph Attention Networks (GATs) with recurrent layers (RNN, LSTM, GRU). These models expected to forecast future human motion and detect potential falls. Our work tackles two core challenges: (1) forecasting future 2D human poses from historical keypoint sequences, and (2) classifying whether a predicted motion sequence is leading to a fall, enabling timely intervention during the critical golden hour.

II. Related Works

The research endeavors primarily converge on the intricate problem of human motion prediction, particularly elucidating spatio-temporal dynamics. While a substantial body of prior research has addressed human motion modeling, a notable lacuna persists concerning the nuanced significance of localized spatial features. The predominant paradigm in these works often leverages recurrent neural network architectures. Despite their limited explicit consideration of spatial attributes, these contributions have advanced the temporal modeling frontier by refining conventional RNN structures, enhancing their simplicity, scalability, and efficacy in short-term and long-term prediction horizons [17]. Further sophisticated advancements in RNNs have been directed towards modeling human pose, meticulously capturing latent hierarchical structures within temporal pose sequences. This approach exhibits a remarkable capacity to discern the frequency of specific human body part movements over time.

However, the efficacy of recurrent layers, such as RNNs and LSTMs, is not universally assured in human motion prediction. In specific contexts, classical methodologies, exemplified by the Kalman filter – a recursive algorithm that estimates the internal state of a linear dynamic system from a series of noisy observations – have demonstrably outperformed their neural counterparts [11]. It is axiomatic that diverse datasets, inherently fraught with noise and idiosyncratic behavioral patterns, present formidable modeling challenges. In this vein, recent work by Widi *et al.*, introduced a novel human fall dataset, the TelUP dataset. This seminal contribution provides a fresh empirical foundation and proposes baseline models, including multilayer perceptron, RNN, and LSTM, specifically engineered to capture temporal dependencies among human keypoints.

Recognizing the prevailing absence of methodologies that systematically account for spatial dependencies in Widi *et al.*,'s study, our current work directly addresses this critical oversight. Previous attempts to leverage graph representations for human keypoints to capture spatial and temporal features concurrently have yielded promising performance [16]. Interestingly, the integration of recurrent layers with graph-behavioral data derived from human poses has shown considerable promise, significantly outperforming a majority of pre-defined evaluation metrics benchmarked against other models tackling similar challenges [15].

III. OBJECTIVES

A. Problem Definition

The initial definition introduces the human keypoint network G, which is modeled as an unweighted, undirected graph $G = (V, A)$. The graph represents the topological structure of the human keypoint network. Individual keypoints are conceptualized as nodes, with V representing the set of nodes denoted as $V = \{v1, v2, \dots, v_N\}$, where N symbolizes the number of nodes (keypoints). An edge between two keypoints is established based on their positional connection, pertinent to the human body part they represent. The connectivity between keypoints is formally expressed through an adjacency matrix A, $A \in \mathrm{R}^{N \times N}$. This binary matrix contains only values of 0 or 1, where 0 indicates the absence of a connection and 1 signifies an existing link.

The second definition pertains to the feature matrix, denoted as $X \in \mathrm{R}^{N \times P}$. Here, P signifies the number of node attribute features, specifically a collection of 2D keypoints acquired over a defined number of frames. These 2D keypoints are managed using a sliding window technique, thoroughly explicated in the Windowing section.

$$\left(Y^{(1)}, Y^{(2)}\right) = f\left(G; (X_{t-n}, \dots, X_t)\right) \quad (1)$$

$$\hat{Y}^{(1)} \in \mathrm{R}^{15 \times 17 \times 2} \quad (2)$$

$$\hat{Y}^{(2)} \in [0, 1] \quad (3)$$

Ultimately, the challenge of spatio-temporal human motion forecasting and classification is conceptualized as learning a mapping function f given a human keypoint network G and a feature matrix X. This function subsequently computes 2D human keypoints beyond time T and classifies the motion as fall or non-fall, as depicted in (1). Here, n represents the length of the temporal time series, while t denotes the length of the time series that requires prediction. The first output as shown in (2) denoted as $\hat{Y}^{(1)}$ is represented as a tensor in $\mathrm{R}^{15 \times 17 \times 2}$. This dimensionality arises from the temporal and spatial configuration of the input representation, which will be elaborated in the Windowing section. Specifically, it encapsulates 15 sequential frames, each comprising 17 human joints parameterized by their 2D spatial coordinates. The second output $\hat{Y}^{(2)}$ as shown in (3) serves as a binary classification indicator, where a value of 0 corresponds to non-fall motion and a value of 1 denotes fall motion.

B. Overview

In this section, the detail application of the study is discussed. As illustrated in Figure 1, the initial step involves extracting 2D keypoints from the raw data and normalizing these features. A sliding window technique is then employed to generate supervised learning samples. Afterwards, subject-wise data splitting and graph data creation are performed. The resulting graph data, categorized as either training or testing according to the experimental setup, is then fed into a specific hybrid model to assimilate knowledge encompassing continuous-time series and spatial features. Finally, we derive evaluation analysis from the classifier and forecaster separately integrated with the hybrid model.

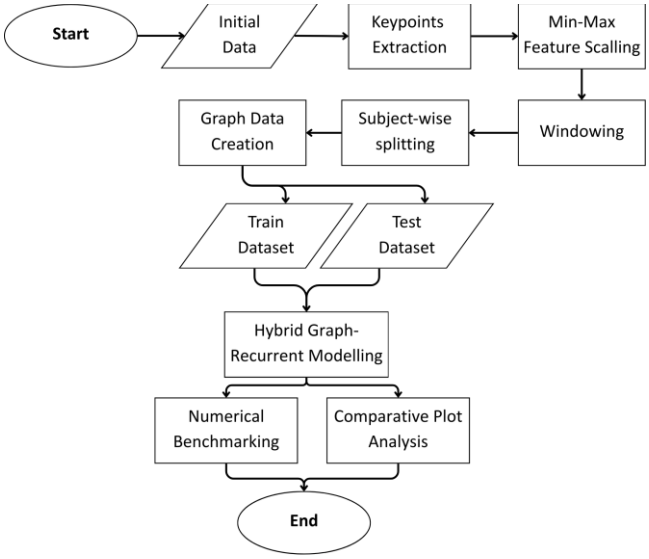


FIGURE 1. Research Flow

C. Dataset

The TelUP human fall dataset comprises video collections featuring six subjects, each performing twelve distinct motions equally divided between fall and non-fall activities [12]. Videos are segmented by activity. Fall videos also contain non-fall frames, capturing continuous activity from pre-fall to the end of the controlled fall. The dataset maintains a standardized resolution of 1920x1080 pixels, recorded with a static camera using RGB color channels in MP4 format. The frame rates vary, with several videos recorded at 30 fps and others at 60 fps. The dataset is annotated by hand to mark the initiation of a human fall, followed by automated annotation for the subsequent frames in the video.

IV. RESEARCH METHOD

The initial stage in our work resembles the established data preprocessing pipeline in Widi *et al*. study. Our strategic alignment relied on the premise that their successful approach to data preparation, which demonstrably led to significant LSTM performance, may benefit our proposed hybrid models. The divergence from ours lies in our unique approach to graph data creation. This specialized process is exclusively developed to accommodate the input requirements of our graph-recurrent architectures. The innovation allows our model to leverage both the temporal processing efficacy of the recurrent layers and the spatial recognition afforded by the graph-based layers.

A. Keypoints Extractions

This initial stage involves rendering and decomposing the video data into individual frames, leveraging OpenCV. Subsequently, we downscale the frame size from their original 1920×1080 pixel resolution to 500×500 pixels. This resizing serves two primary purposes: to mitigate the computational overhead associated with the YOLOv11 model and to normalize the spatial dimensions of each frame. Finally, each normalized frame is input into the

YOLOv11 architecture to estimate and extract 17 distinct keypoints. Figure 2 shows that these keypoints are localized across various human anatomical landmarks, including the eye, ear, nose, shoulder, elbow, wrist, hip, knee, and ankle. Given that each keypoint is represented by 2D cartesian coordinates (x, y), we denotes the aggregate output from YOLOv11 that encompasses all extracted keypoints as $\mathrm{k} \in \mathrm{R}^{17\times2}$.

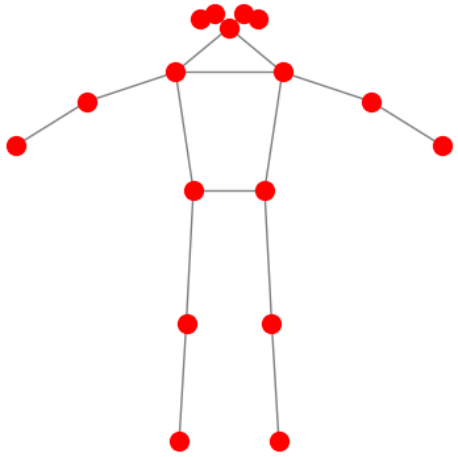


FIGURE 2. Single frame keypoints extraction.

B. Min-Max Feature Scalling

The 2D Cartesian coordinates are normalized by dividing them by the new frame size, specifically 500 for both the x and y axes, to achieve a similar numerical range of coordinates between 0 and 1. This feature scaling step is applied prior to training for promoting faster and more stable convergence within our hybrid model and preserves equitable feature importance. The operation for the i-th keypoint is expressed in (4), where x_i prime and y_i prime represent the normalized coordinates derived from the original coordinates x_i and y_i.

$$(x_i^{'}, y_i^{'}) = \left(\frac{x_i}{500}, \frac{y_i}{500}\right)_{i=1}^{17} \quad (4)$$

C. Windowing

Developing a model for forecasting future sequences with a specific time step entails a shifted manner of target synthesis relative to the source. To this end, we gain an advantage from the sliding window technique for creating sorted windows containing fixed-size frame sequences [17]. We illustrate the implementation of the technique in Figure 3. Here, we express the input frames as W_{in} and the output frames as W_{out}, representing the input expectation and output. In this case, we establish the lengths of W_{in} and W_{out} by 15, resulting in 15 consecutive frames within a single window. The L_T denotes the last window of the time series. Thus, the shape of a window for input data and output data is $\mathrm{R}^{15\times17\times2}$. The window is advanced with a stride of 5 frames, producing a 10-frame overlap between successive windows. This overlapping scheme is adopted to maintain temporal continuity and explicitly model dependencies across neighboring segments, rather than processing each window as an isolated sample. This enables each model to capture subtle temporal variations that evolve gradually over time. The aforementioned ideas are used to generate segmented data for forecasting tasks. We assign a one-hot encoded label to each specific window by drawing the class mode from the 15 frames within that window for building the classification ability of the model.

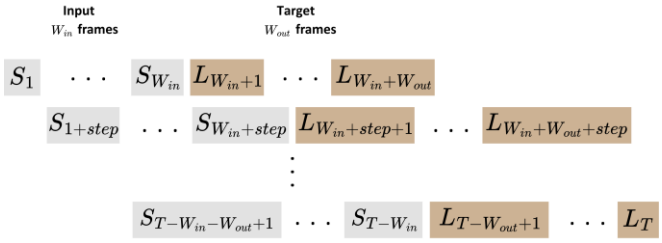


FIGURE 3. Sliding window application on the input and target data

D. Subject-wise Splitting

Upon generating the serialized dataset, a subset of the data is extracted based on its subject to facilitate splitting the training and test datasets, thereby preventing data leakage. The training dataset encompasses a collection of windows representing human keypoints from subjects 1, 2, 3, and 4. The remaining subjects, specifically subjects 5 and 6, are designated as the source for the test dataset [12]. During this phase, an analysis is also conducted to determine the number of sliding windows produced by the preceding stage for both the training and test datasets, which provides a more profound understanding of the distribution of these sliding windows across human activity, as well as fall and non-fall categories.

E. Graph Data Construction

Developing graph-recurrent models incorporating graph-based layers necessitates the construction of graph-structured data [18]. We construct a single graph, $G_K = (V_K, A_K)$ from a sliding window, K. Let $V_K = \{k_i\}_{i=1}^{17}$ represents a set of 17 human keypoints, and $A_K \in \{0,1\}^{N\times N}$ is an adjacency matrix. A_K defines the edge linkages between keypoints based on human body part relevance and symmetrical relationships, where $(A_K)_{ij} = 1$ if the i-th and j-th joints are linked, and $(A_K)_{ij}$ = 0 otherwise. Furthermore, we manage the feature matrix by reshaping W_{in} that originally $W_{in} \in \mathrm{R}^{15\times17\times2}$ into $X \in \mathrm{R}^{17\times30}$. Figure 4 depicts a particular constructed graph with unfolded node attributes, containing 15 distinct consecutive x and y coordinates for a single node. Ultimately, the results of this stage are a series of graph data as shown in Figure 5.

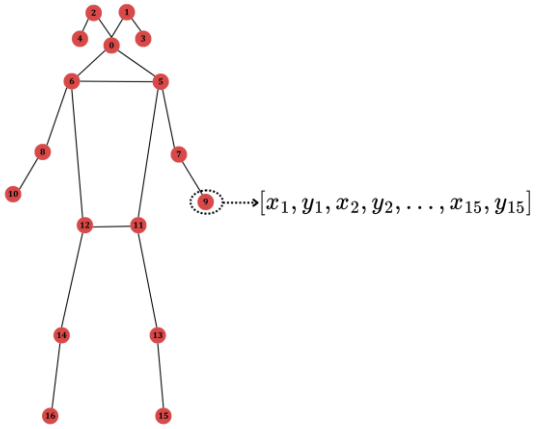


FIGURE 4. Established connection between nodes and specific node attribute

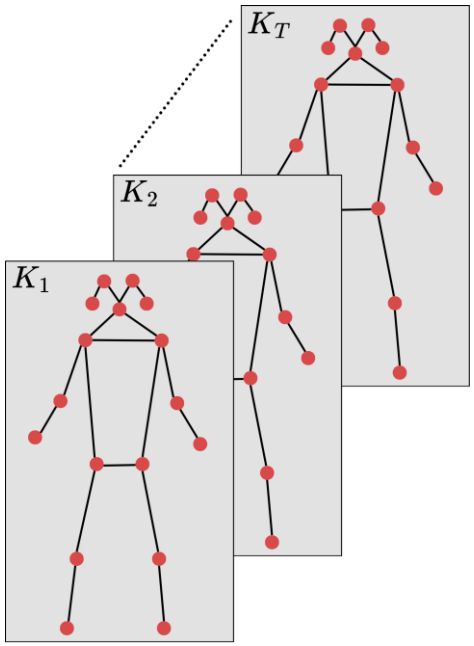


FIGURE 5. Graphical representation of the constructed graph data series.

F. Graph-Recurrent Modelling

The model architecture proposed in this study is illustrated in Figure 6. Encapsulating the intricacy spatial dependence of human keypoints with graphs involves considering the neighboring connection across nodes. We instigate this case by a graph-based layer. The layer is instantiated as either a GCN or a GAT. As a graph represents a single window that comprise 17 human keypoints over 15 consecutive frames, the graph-based layer enables the model to aggregate spatial information based on established keypoint connections across human body. A GCN layer performs neighborhood aggregation using a normalized adjacency matrix [19]. Representation update of each node is accomplished by computing a weighted sum of its own features and those of its neighbors. Graph structure and shared trainable parameters are the key factors which determined the weights. In comparison, a GAT layer introduces an attention mechanism. The mechanism assigns learnable importance coefficients to neighboring nodes [20]. This advancement allows the model to focus on more informative connections throughout feature aggregation. Rectified Linear Unit (ReLU) is a function that introduces non-linearity to the output at the conclusion of the graph-based layer operation. The dropout layer is employed to improve the robustness of our models by reducing overfitting.

Temporal dependencies are subsequently addressed using a recurrent layer (RNN or LSTM or GRU). The fundamental distinction among them resides in their respective memory control. A RNN layer manage an input sequence by recursively updating a hidden state [21]. The current state is computed as a nonlinear transformation of the current input and the preceding hidden state. This operation facilitates the propagation of information temporally through the utilization of shared and learnable parameters. LSTM introduces three exclusive gates (input, forget, and output gates) to mitigate the risks of vanishing and exploding gradients as commonly experienced by RNN [22]. Unlike the LSTM, GRU collates the input and forget gates into a single update gate, eliminating the explicit cell state and allowing hidden states to convey temporal information directly [23]. Thus, the computational demand of GRU is theoretically lesser than LSTM [24]. The linear layer in our model functions as a linear transformation to obtain classification and forecasting output by reducing the feature dimensionality.

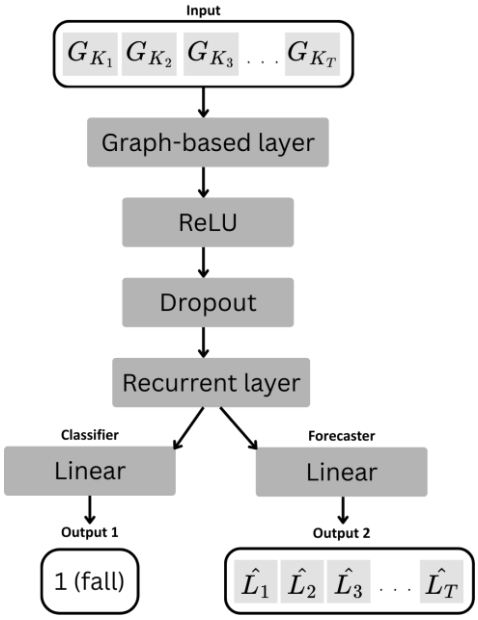


FIGURE 6. Model Architecture.

G. Loss Functions

Addressing the predefined multitasks requires simultaneously improving the forecasting and classifying abilities of the model. Consequently, we enforce the model to minimize the Mean Squared Error (MSE) between the estimated keypoint sequence $\hat{\mathbf{L}} = \{\hat{L}_1, \hat{L}_2, \ldots, \hat{L}_{W_{out}}\}$ and its corresponding ground truth sequence $\mathbf{L} = \{L_1, L_2, \ldots, L_{W_{out}}\}$. This expression is shown in (5). Concurrently, we calculate the Cross-Entropy Loss as in (6). In this context, y_i

represents the ground truth, with values of zero or one indicating non-fall and fall, while $\hat{y}_i$ is expected as normalized probabilities that sum to 1. The cumulative loss $\mathcal{J}$ in (7) constitutes the ultimate objective and subsequently backpropagate through the model facilitated by optimization algorithms.

$$\mathcal{J}_1 = \frac{1}{N}\sum_{i=1}^{N} || \mathbf{L}_i - \hat{\mathbf{L}}_i ||^2 \quad (5)$$

$$\mathcal{J}_2 = -\sum_{i=1}^{C} y_i \log(\hat{y}_i) \quad (6)$$

$$\mathcal{J} = \mathcal{J}_1 + \mathcal{J}_2 \quad (7)$$

V. Experimental Setup

A. Model Training Settings

The input and output window sizes were both set to 15. The model configuration used a batch size of 512 and the ADAM optimizer with a learning rate of 0.001. The architecture illustrated in Figure 6 was trained for 2000 epochs. The core design consisted of one graph-based layer and one recurrent layer with a hidden dimension of 67. The graph attention mechanism used eight attention heads. A dropout value of 0 was applied to the GCN-MLP and GAT-MLP components for the proposed model while a value of 0.1 was used for the other models. Dropout was set to 0.1 for the remaining models because a modest degree of stochastic regularization consistently improved performance on test data and indicated better robustness and generalization. By contrast, GCN-MLP and GAT-MLP achieved the best results without dropout, as they are more sensitive to regularization. Another factor is that feature suppression caused by dropout can hinder information propagation and destabilize already sparse representations. All experiments were implemented using PyTorch and PyTorch Geometric. Computations were conducted on a machine equipped with an AMD Ryzen 7 CPU and an NVidia GeForce RTX 4090 GPU.

B. Evaluation Metrics

We adopt several evaluation metrics to reveal our model performance. Each metric evaluates model performance from diverge facets. We adhere to prior research in human pose forecasting and estimation by calculating the mean per-joint position error (MPJPE) [25] and mean per-joint velocity error (MPJVE) [17]. This type of evaluation unfolds the error deduce by the model once predicting corresponding joints. The MPJPE is defined by:

$$E_{\mathrm{MPJPE}} = \frac{1}{N}\left\| \mathbf{L}_{W_{out}} - \hat{\mathbf{L}}_{W_{out}} \right\| \quad (8)$$

The MPJPE is defined as the average of the squared Euclidean distance between $\mathrm{L}_{W_{out}}$ and $\hat{\mathrm{L}}_{W_{out}}$, which represents the ground truth and predicted coordinates for specific W_{out} frames within the sliding window sequence. The MPJVE is calculated by drawing the L2-norm of the motion velocity, which is derived from the first derivative of joint coordinates or one-frame gap of joint position between the ground truth and the prediction. MPJVE examines the average motion velocity difference across frames. MPJVE is expressed as follows:

$$E_{\mathrm{MPJVE}} = \frac{1}{N}\left\| \mathbf{V}_{W_{out}} - \hat{\mathbf{V}}_{W_{out}} \right\| \quad (9)$$

$$\mathbf{V}_{W_{out}} = \mathbf{L}_{W_{out}} - \mathbf{L}_{W_{out}-1} \quad (10)$$

where $\mathbf{V}_{W_{out}}$ and $\hat{\mathbf{V}}_{W_{out}}$ denote velocities computed between frame $\mathbf{L}_{W_{out}-1}$ and $\mathbf{L}_{W_{out}}$.

Evaluating the classifier of our model demands a confusion matrix for demonstrating model performance in classifying fall and non-fall human motion. The confusion matrix summarizes the relationship between the actual ground truth and predicted labels through four components: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) [26]. These components are utilized to derive advanced metric scores, specifically accuracy, precision, recall, and F1-score. These metrics are defined as follows:

$$\mathrm{Accuracy} = \frac{\mathrm{TP+TN}}{\mathrm{TP+FP+TN+FN}} \quad (11)$$

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP+FP}} \quad (12)$$

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP+FN}} \quad (13)$$

$$\mathrm{F1} = 2 \times \frac{\mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}} \quad (14)$$

Accuracy quantifies the proportion of correct model predictions. Precision assesses the rate at which predicted fall class instances are genuinely members of the fall class. Recall evaluates the proportion of actual fall class instances that were accurately predicted. Given that precision and recall can exhibit an inverse relationship upon model re-tuning, the F1-score provides a harmonic mean between these two metrics.

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) serves as a robust metric, extending the insights derived from a confusion matrix. Specifically, the AUC-ROC represents the integral of the True Positive Rate (TPR) plotted against the False Positive Rate (FPR) [27]. The TPR gauges the proportion of actual fall motions that are correctly identified by the model. Conversely, the FPR quantifies the proportion of non-fall motions that are erroneously classified as fall motions. Mathematically, the metrics is defined as:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{P}} \quad (15)$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{N}} \quad (16)$$

$$\mathrm{AUC} = \sum_{k=1}^{n-1} \frac{(\mathrm{FPR}_{k+1} - \mathrm{FPR}_k) \cdot (\mathrm{TPR}_k + \mathrm{TPR}_{k+1})}{2} \quad (17)$$

Here, P and N serve as total fall motions and total non-fall motions. The Area Under the Curve (AUC) is calculated using the trapezoidal rule, which approximates the area beneath the ROC curve by

summing the areas of trapezoids formed by consecutive points on the curve.

Regardless of the aforementioned metrics suggest significance score, they are insufficient to outline the effectiveness of our model in explaining the variability of the data. The coefficient of determination (R-squared) addresses this deficiency by interpreting the proportion of the variance in the ground truth that is predictable from the features [28]. R-squared can be defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{m}(\mathbf{L}_i - \hat{\mathbf{L}}_i)^2}{\sum_{i=1}^{m}(\mathbf{L}_i - \overline{\mathbf{L}})^2} \quad (18)$$

In the given formula, $\mathbf{L}_i$ represents the actual ground truth of the$i$-th sequence, while $\hat{\mathbf{L}}_i$denotes the predicted i-th sequence. $\overline{L}$ signifies the mean of the actual ground truth.

Model complexity is assessed by quantifying the computational effort required for inference and training. Multiply–Accumulate Operations (MACs) and Floating-Point Operations (FLOPs) serve as feasible metrics for this task. A MAC operation represents a fundamental arithmetic calculation within a neural network, comprising one multiplication followed by one addition, such as the dot product of input data and weights. FLOPs extend upon MACs by summarizing not only multiplication-addition operations but also subtraction and division. Consequently, the total number of MACs in a model indicates the dominant operations necessary for output generation. In contrast, the total number of FLOPs provides a standardized measure that accounts for all operations executed by the model. The total parameter of the model is also calculated to suggest the number of weights and biases that perform the operations. The MACs, FLOPs, and parameters of our model are computed using state-of-the-art Calflops, an exclusive Python library for theoretically estimating Pytorch-based model complexity [29].

VI. Results and Discussions

A. Windowing

During the keypoint extraction phase, YoloV11 did not encounter any extraction failures. It indicates that all keypoints derived from both the training and test datasets were successfully obtained and do not contain any zero vectors, which were implemented as a prophylactic measure against extraction anomalies.

TABLE 1. Generated data by sliding windows.

TelUP Dataset			
Usage	**Motion Type**	**Windows**	**Total**
Training	Fall	1921	2600
	Non-fall	679	
Testing	Fall	1202	1729
	Non-fall	527	

After the windowing stage, a detailed analysis of the data distribution was conducted to assess the balance of input data for the proposed model. The sequential windows stratified by motion type were enumerated for both training and test datasets. As presented in Table 1, the aggregate number of train data windows totaled 2600, comprising 1921 instances of fall motions and 679 instances of non-fall motions. The disparity between fall and non-fall motions is significant with a difference of 1242 windows, indicating an imbalanced distribution. Conversely, the test dataset consisting of 1729 windows exhibits a trailing margin with 1202 fall motion windows and 527 non-fall motion windows, resulting in a difference of 675. Nevertheless, this distribution of windows proved adequate for model training, ultimately contributing to noteworthy performance metrics.

B. Performance Benchmarking

TABLE 2. Forecast and classification loss.

Models	MSE	Cross-entropy loss
MLP	0.0099	0.9139
RNN	0.0117	0.6607
LSTM	0.0121	0.4589
GCN-MLP	0.0112	0.5479
GAT-MLP	0.0127	0.7237
GCN-RNN	0.0110	0.4034
GCN-LSTM	0.0108	0.3819
GCN-GRU	**0.0057**	**0.0816**
GAT-RNN	0.0109	0.8274
GAT-LSTM	0.0101	0.5681
GAT-GRU	0.0101	0.8238

TABLE 3. MPJPE and MPJVE of 2D joint positions in pixels.

Models	MPJPE	MPJVE
MLP	131.5	71.94
RNN	162.6	71.7
LSTM	168.6	61.48
GCN-MLP	170.0	**59.27**
GAT-MLP	167.6	82.23
GCN-RNN	149.6	73.4
GCN-LSTM	151.6	71.7
GCN-GRU	**113.4**	64.66
GAT-RNN	142.2	72.9
GAT-LSTM	141.6	65.87
GAT-GRU	131.2	70.84

Among the experiments on graph-recurrent architecture, GCN-GRU achieves the lowest MSE and cross-entropy loss. Such losses indicate superior temporal prediction precision and discriminative capability in motion classification tasks relative to the baseline models proposed in a previous study. These results suggest early implications that the incorporation of GCN with GRU effectively captures both spatial dependencies among joints and temporal dynamics across frames.

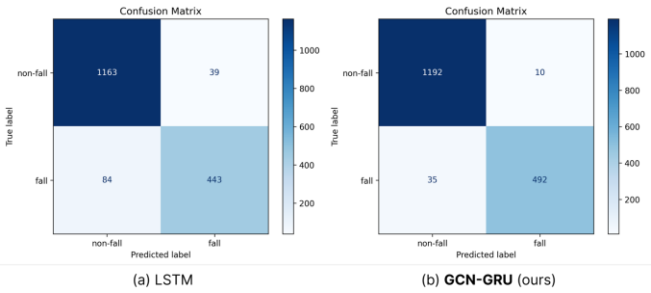
Table 3 presents a comparative analysis of MPJPE and MPJVE against ground truth test data by benchmarking our proposed model with the baseline models. The GCN-GRU model achieves a notably superior MPJPE of 113.4 pixels. The score demonstrates a significant reduction of 18.1 pixels compared to the MLP as the lowest MPJPE producer. The score also hints at an enhanced capacity of the model to discern and leverage spatial dependencies,

mainly attributable to the integration of a GCN layer within its architecture. The GCN layer facilitates the prediction of joint keypoints that exhibit a closer correspondence to the ground truth, thereby theoretically improving both forecasting accuracy and classification performance. In contrast, the GCN-MLP model attains the highest MPJPE of 170.0 pixels, suggesting greater distortion in the forecasted keypoints relative to the ground truth.

The GCN-MLP achieves the lowest MPJVE by 59.27 pixels. The score shows modest improvement by 2.21 pixels compared to the LSTM. This outcome suggests that achieving the lowest MPJPE does not necessarily correspond to the lowest MPJVE. This case occurred on the GCN-GRU. The model demonstrates a marginally higher MPJVE than other graph-recurrent architectures. Such a score suggests that its predicted joint trajectories may exhibit slightly less temporal smoothness or increased frame-to-frame variability. Consequently, a comprehensive evaluation of both MPJPE and MPJVE reveals that GCN-GRU substantially enhances pose accuracy by effectively exploiting spatial dependencies among joints. Nevertheless, this improvement incurs a slight penalty in terms of temporal smoothness.

The classification performance of our proposed model demonstrates a significant advancement over baseline models, particularly the LSTM. As detailed in Table 4, the GCN-GRU achieves 97.40% accuracy and an F1-score of 96.89%. These metrics underscore the enhanced robustness of our model in accurately classifying both fall and non-fall motions. Furthermore, the AUC of the ROC curve is 96.26%, closely matching the F1-score. This high AUC value indicates a successful mitigation of prediction errors and a strong generalization capability. In addition, the R-squared score of the model substantially outperforms the LSTM baseline by over 20%, indicating that the model has a stronger ability to explain the variability in the test data than the LSTM network.

Unsignificant improvement is encountered by GCN-LSTM as an advancement to the LSTM. The model scored 91.65% on F1-score and 67.25% on R^2, improving by 0.26% and 0.82% compared to LSTM performance. This case indicates that incorporating GCN with the LSTM layer has a small impact on performance. Such a case, yet worse also experienced by GAT-LSTM. The model resulted in decreases of 2.36% and 7.92% in F1-score and R^2 from 89.03% and 58.51%. The contrary case is revealed by the advancement of the RNN. Incorporation of the GCN and GAT individually into the



(a) LSTM (b) **GCN-GRU** (ours)

FIGURE 7. Confusion Matrix Diagram.

TABLE 4. Models performance.

Models	Accuracy	Precision	Recall	F1-score	AUC	R^2
MLP [12]	85.08	84.78	78.83	80.90	78.82	29.58
RNN [12]	88.37	88.46	83.49	85.50	83.48	45.14
LSTM [12]	92.89	92.59	90.41	91.39	90.40	66.43
GCN-MLP	87.57	89.41	81.04	83.77	81.00	41.32
GAT-MLP	92.54	91.77	90.42	91.06	90.43	64.79
GCN-RNN	90.98	90.91	87.44	88.91	87.44	57.42
GCN-LSTM	93.06	92.51	90.85	91.65	90.85	67.25
GCN-GRU	**97.40**	**97.58**	**96.26**	**96.89**	**96.26**	**87.72**
GAT-RNN	92.37	93.09	88.75	90.54	88.76	63.97
GAT-LSTM	91.21	91.98	87.07	89.03	87.07	58.51
GAT-GRU	90.80	92.29	85.98	88.35	85.98	56.60

TABLE 5. Models Complexity.

Models	MACs		FLOPS		Parameters (K)
	Inferencing (MMACs)	Training (MMACs)	Inferencing (MFLOPS)	Training (MFLOPS)	
MLP [12]	66.978	200.933	134.021	402.063	131.46
RNN [12]	33.554	100.663	386.597	1159.790	25.64
LSTM [12]	33.554	100.663	1350.960	4052.880	88.61
GCN-MLP	**15.6979**	**47.094**	**31.657**	**94.971**	**16.80**
GAT-MLP	125.583	376.750	253.473	760.418	131.31
GCN-RNN	25.397	76.192	155.511	466.534	42.65
GCN-LSTM	25.397	76.192	471.966	1415.900	63.35
GCN-GRU	25.397	76.192	366.481	1099.440	56.45
GAT-RNN	80.233	240.697	512.157	1536.470	65.58
GAT-LSTM	80.233	240.697	1563.400	4690.210	134.12
GAT-GRU	80.233	240.697	1212.990	3638.960	111.28

RNN provides a progressive improvement in performance. The GCN-RNN performed with an accuracy of 90.98% and an AUC of 87.44%. The scores show gains of 2.61% and 3.96%, compared with the RNN for the corresponding metrics. The performance climbed further for the GAT-RNN. The model achieved 92.37% accuracy and an R^2 of 88.76%.

A more in-depth analysis of LSTM and GCN-GRU is depicted in Figure 7. In broad terms, both models encountered the same challenge in non-fall and fall classes. Most misclassifications occurred on fall data. 84 fall data were erroneously predicted as non-fall by LSTM, while the GCN-GRU predicted 35 fall data as non-fall. Aligning with prior evaluation on models' performance, GCN-GRU has an enhanced ability juxtaposed to LSTM. GCN-GRU successfully anticipates 49 fall and 29 non-fall data points more.

C. Models Complexity

Applying real-world human fall forecasting entails the urge to evaluate the computational cost under resource-constrained environments. Table 5 presents a comparative analysis of the computational requirements of our proposed model against baseline architectures during inferencing and training. The MACs and FLOPS are quantified in MegaMACs (MMACs) and MegaFLOPS (MFLOPS) units, representing the cost of a single-step training or inference process computed with a batch size 32. The GCN-MLP is recognised as the simplest model among the alternatives. This simplicity results in the lowest measures of MACs and FLOPS, not only during the inferencing stage but also throughout the training process. Although the MLP layer is straightforward in our experiment, when preceded by a more complex layer, such as GAT, the resulting model increases in complexity due to the edge-wise attention computation and aggregation operation. This phenomenon is observed in the GAT-MLP model during both the inferencing and training stages, demanding 125.583 and 376.750 MMACs, which registers as the highest consumption. Conversely, the GAT-LSTM is the computationally heaviest model in terms of FLOPS. This model possesses 134.12 K parameters and requires 1563.400 and 4690.210 MFLOPS. This model requires substantial computation because it combines an attention-based GAT layer with an LSTM layer that employs multiple gating operations.

The GCN-recurrent unit models exhibit equivalent complexity measured by MACs. The training and inference require 25.397 and 76.192 MACs. This similarity in MAC counts is attributed to the identical structure of the GCN layer. Consequently, the input size, hidden size, sequence length, and graph size are consistent across models. An additional factor is that each recurrent model utilizes the same number of dense matrix multiplications per timestep. However, the GCN-RNN, GCN-LSTM, and GCN-GRU models exhibit distinctions in FLOPS. This divergence arises from the differing operations during the backward propagation process. Since FLOPS accounts for all arithmetic operations, the variation in each recurrent layer's operation is effectively captured. As anticipated, the GCN-RNN possesses the lowest FLOPS due to its two simple primary operations: the hyperbolic tangent (tanh) activation and minimal element-wise multiplication. This trend is also observed in the GRU (with reset and update gates) and the LSTM (with input, forget, and output gates), where their distinct gating mechanisms show a positive correlation with higher FLOPS counts.

D. Comparative Graphical Anaylsis

Comparative trajectories of the forecasted keypoints are visualized against their empirical counterparts to obtain a more comprehensive understanding of the quantitative behavior of MPJPE and MSE. The evaluation was conducted for the LSTM model, the highest-performing baseline, and our proposed architecture. Figure. 8 illustrates the keypoint predictions extracted from the initial frame within the 36^{th} sliding window sequence, encompassing two distinct kinematic patterns: forward-fall and backward-fall motions. The 36^{th} sliding window was deliberately selected following empirical analysis wherein the fall motion prominently emerged. As evidenced in the figure, our model demonstrates a markedly closer spatial alignment of predicted keypoints to the ground truth relative to the LSTM. The predicted skeletal configurations exhibit a high degree of morphological congruence in prone and supine orientations, indicating a more adept capacity to capture the holistic dynamics of human motion. Meanwhile, the LSTM model produces noticeably perturbed estimations, wherein the prone posture during forward-fall sequences degenerates into a push-up-like position, and the supine posture during backward-fall sequences is erroneously anticipated as a pre-supine transitional state. The discrepancy indicates that our model achieves an approximate equilibrium between spatial configuration and temporal progression, facilitating a more coherent encoding of spatiotemporal dependencies.

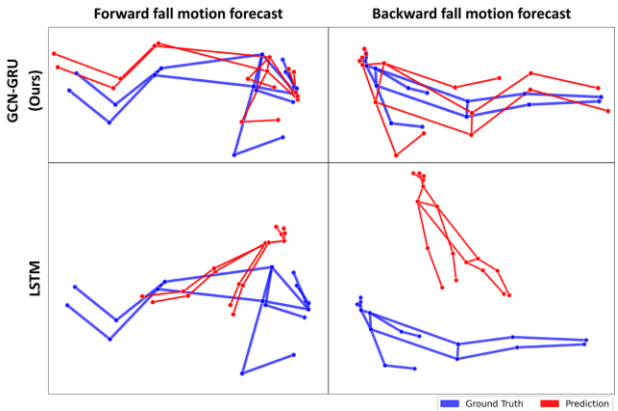


FIGURE 8. Prediction comparison in a frame of ours vs LSTM.

The evaluation to multi-frame keypoint predictions is illustrated in Figure 9, to validate the preceding one-frame forecasting analysis further. The figure visualizes results from the initial frame of the 0^{th}, 9^{th}, 18^{th}, 27^{th}, 36^{th}, 45^{th}, 54^{th}, and 63^{th} sliding windows. These intervals were determined based on a predefined frame stride of 135 between selected

windows, given each window comprises 15 consecutive frames. The rationale behind this configuration was to capture the transition from pre-fall to fall postures to provide a more apparent contrast in model behavior across different motion phases. As depicted in Figure 9, our model consistently produces keypoint forecasts that closely align with the ground truth despite the MPJVE value being marginally higher than that of the LSTM. Conversely, the LSTM model exhibits increasing distortion once the fall motion sequence commences, beginning at the fourth plot and persisting through the eighth. This degradation underscores the instability of LSTM in modeling rapid kinematic transitions. However, despite a slightly elevated MPJVE, our model demonstrates a more robust capacity to capture and reproduce the nuanced dynamics of human motion.

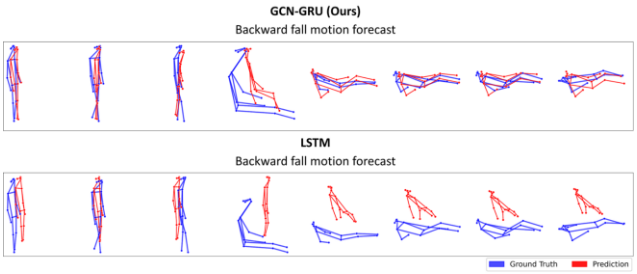


FIGURE 9. Prediction comparison in frames of ours vs LSTM

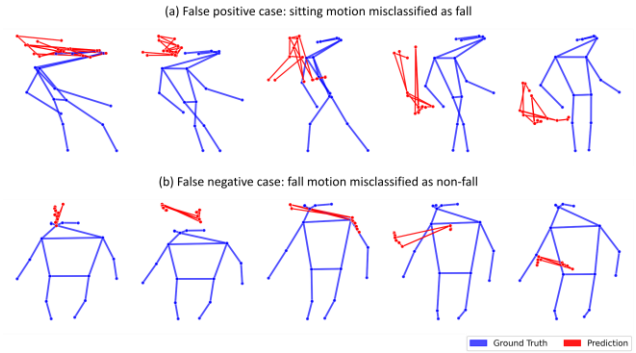


FIGURE 10. Misclassifications in sitting motion sequences using the GCN-GRU

Fall detection is a safety-critical application. Evaluating only successful predictions is insufficient. False alarms and missed falls can both have practical consequences. For this reason, Figure 10 presents representative misclassification cases produced by the GCN-GRU model. The input window consists of 15 consecutive frames, visualized with a stride of three to produce five representative skeletal comparisons.

Figure 10(a) shows a false-positive case. In this case, the GCN-GRU model misclassified a sitting motion sequence as a fall. This failure is important because quick sitting can resemble a fall-like activity. The body moves downward rapidly during quick sitting. The torso and hip positions may also become similar to the transition phase of a fall. However, the predicted skeletons deviate substantially from the ground-truth skeletons. This indicates that the model failed to preserve the correct spatial body configuration during the controlled sitting motion.

Figure 10(b) shows the opposite failure mode. In this case, a fall-related motion was misclassified as non-fall. The model did not adequately capture the rapid onset of the falling motion. The predicted pose trajectory remained more cluttered than the non-fall posture. These qualitative results show that the proposed model still experiences failure cases despite its strong quantitative performance. The errors are more likely to occur when the motion transition is abrupt. They may also occur when sitting and falling share similar short-term skeletal patterns.

VII. CONCLUSION

This study delves into the nuanced domain of motion forecasting and the dichotomous classification of fall versus non-fall events, all predicated upon historical frame-based 2D human keypoints extracted from a meticulously controlled video dataset. Despite the imbalanced nature of the dataset, our model demonstrates higher robustness than LSTM. Specifically, the proposed GCN–GRU model attains lower MPJPE, reduced forecast and classification loss, and improved performance across extended evaluation metrics, including confusion matrix, AUC, and R^2. Moreover, architecture of the GCN-GRU is notably lightweight compared to the LSTM as the best performer baseline model from previous study. This signifies lower computational cost without sacrificing accuracy and robustness. Comparative graphical analyses visually confirm that the GCN-GRU produces more accurate forecasts, particularly when predicting holistic human poses against the ground truth. These findings collectively validate the efficacy of the GCN–GRU model among the hybrid graph-recurrent models in capturing the intrinsic spatial–temporal dependencies that underlie complex motion patterns. Nevertheless, the observed misclassification cases should still be considered when interpreting the model's reliability in safety-critical fall-detection scenarios.

This study has several limitations as important directions for future investigation. First, broader external validation across countries, populations, and physical environments remains necessary. Fall patterns and fall-like activities may vary according to age, health condition, mobility status, room layout, camera position, lighting, and monitoring context. Consequently, the present findings should not yet be interpreted as evidence of universal generalizability across home, clinical, rehabilitation, or aged-care environments. Second, this study focuses on the model-development component. End-to-end delay from inference latency on edge-computing devices to intervention was not evaluated. Advanced work on model development will further address dataset imbalance through advanced resampling or cost-sensitive learning strategies, while exploring both short- and long-term forecasting paradigms. Feature engineering for 2D keypoints will be investigated to improve the robustness of the model across diverse real-world scenarios.

ACKNOWLEDGEMENT

The authors would like to express our gratitude to Telkom University for its support during this research. We also thank the peer reviewers for their valuable feedback. The authors acknowledge the anonymous reviewers whose insightful comments and suggestions significantly improved the quality and readability of this paper.

FUNDING STATEMENT

There is no funding agencies supporting the research work.

AUTHOR CONTRIBUTIONS

Asyafa Ditra Al Hauna: Conceptualization, Data Curation, Visualization, Methodology, Writing – Original Draft Preparation, Writing – Review & Editing;

Masanori Fukui: Administration, Supervision.

CONFLICT OF INTERESTS

No conflict of interests were disclosed.

ETHICS STATEMENTS

Ethical approval was not applicable to this research since it did not involve human participants, animals, or sensitive data.

DECLARATION OF AI-ASSISTED TECHNOLOGIES IN THE WRITING PROCESS

During the preparation of this work, Grammarly was used to reduce grammatical errors and to clarify complex content for readers. After utilizing this tool, the manuscript was reviewed and edited as needed and took full responsibility for the content of the publication. Grammarly was not used to generate any scholarly content in this article.

REFERENCES

[1] D. Basak, A.A. Anthony, N. Banerjee, S. Rath, S. Chatterjee, K.D. Soni, N. Sharma, T. Ogawa, G. O'Reilly, J. Attergrim, M.G. Wärnberg and N. Roy, "Mortality from fall: A descriptive analysis of a multicenter Indian trauma registry," *Injury*, vol. 53, no. 12, pp. 3956-3961, 2022. DOI: https://doi.org/10.1016/j.injury.2022.09.048

[2] P.H. Pelicioni, L.L. Chan, S. Shi, K. Wong, L. Kark, Y. Okubo and M.A. Brodie, "Impact of mobile phone use on accidental falls risk in young adult pedestrians," *Heliyon*, vol. 9, no. 8, pp. e18366, 2023. DOI: https://doi.org/10.1016/j.heliyon.2023.e18366

[3] G.S. Kim, M.K. Park, J.J. Lee, L. Kim, J.Y. Lee and N. Kim, "Situational and environmental risk factors associated with home falls among community-dwelling older adults: Visualization of disparities between actual and perceived risks," *Geriatric Nursing*, vol. 62, pp. 221-228, 2025. DOI: https://doi.org/10.1016/j.gerinurse.2025.02.006

[4] K.J. Burdick, A.P. Coulter and M. Tirabassi, "Prehospital Transport Time and Outcomes for Pediatric Trauma: A National Study," *Journal of Surgical Research*, vol. 292, pp. 144-149, 2023. DOI: https://doi.org/10.1016/j.jss.2023.07.041.

[5] Australian Institute of Health and Welfare, "Falls." Accessed: Jan. 28, 2026. [Online]. Available: https://www.aihw.gov.au/reports/injury/falls

[6] T.E. Lockhart, R. Soangra, H. Yoon, T. Wu, C.W. Frames, R. Weaver and K.A. Roberto, "Prediction of fall risk among community-dwelling older adults using a wearable system," *Scientific Reports*, vol. 11, no. 1, 2021. DOI: https://doi.org/10.1038/s41598-021-00458-5

[7] S. Subramaniam, A.I. Faisal and M.J. Deen, "Wearable Sensor Systems for Fall Risk Assessment: A Review," *Frontiers in Digital Health*, vol. 4, 2022. DOI: https://doi.org/10.3389/fdgth.2022.921506

[8] A. Latif, H.F.A. Janabi, M. Joshi, G. Fusari, L. Shepherd, A. Darzi and D.R. Leff, "Use of commercially available wearable devices for physical rehabilitation in healthcare: a systematic review," *BMJ Open*, vol. 14, no. 11, pp. e084086, 2024. DOI: https://doi.org/10.1136/bmjopen-2024-084086

[9] J. Hu, S. Wang, Y. Shen and X. Miao, "A Wearable System for Knee Osteoarthritis: Based on Multimodal Physiological Signal Assessment and Intelligent Rehabilitation," *Sensors*, vol. 25, no. 23, pp. 7334, 2025. DOI: https://doi.org/10.3390/s25237334

[10] S. Saurav, R. Saini and S. Singh, "A dual-stream fused neural network for fall detection in multi-camera and 360 videos," *Neural Computing and Applications*, vol. 34, no. 2, pp. 1455-1482, 2022. DOI: https://doi.org/10.1007/s00521-021-06495-5

[11] A.P. Yunus, N.C. Shirai, K. Morita and T. Wakabayashi, "Comparison of RNN-LSTM and Kalman Filter Based Time Series Human Motion Prediction," *Journal of Physics: Conference Series*, vol. 2319, no. 1, pp. 012034, 2022. DOI: https://doi.org/10.1088/1742-6596/2319/1/012034

[12] A. Widiyanto, R.G. Candraningtyas, A.H.H. F.F, M. Prameswari, H. Bashiran, G.N. Surahmat, B.A. Rahmah, A.A.I.C.M. Dewi and A.P. Yunus, "TelUP Human Fall Dataset: A Motion Forecasting Study of Human Falls," *JURNAL INFOTEL*, vol. 17, no. 3, pp. 456-471, 2025. DOI: https://doi.org/10.20895/infotel.v17i3.1420

[13] B. Huang and X. Li, "GGTr: An Innovative Framework for Accurate and Realistic Human Motion Prediction," *Electronics*, vol. 12, no. 15, pp. 3305, 2023. DOI: https://doi.org/10.3390/electronics12153305

[14] A. Bharathi, R. Sanku, M. Sridevi, S. Manusubramanian and S.K. Chandar, "Real-time human action prediction using pose estimation with attention-based LSTM network," *Signal, Image and Video Processing*, vol. 18, no. 4, pp. 3255-3264, 2024. DOI: https://doi.org/10.1007/s11760-023-02987-0

[15] M. Lovanshi, V. Tiwari and S. Jain, "3D Skeleton-Based Human Motion Prediction Using Dynamic Multi-Scale Spatiotemporal Graph Recurrent Neural Networks," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 1, pp. 164-174, 2024. DOI: https://doi.org/10.1109/TETCI.2023.3318985

[16] L. Chen, R. Liu, X. Yang, D. Zhou, Q. Zhang and X. Wei, "STTG-net: a Spatio-temporal network for human motion prediction based on transformer and graph convolution network," *Visual Computing for Industry, Biomedicine, and Art*, vol. 5, no. 1, 2022. DOI: https://doi.org/10.1186/s42492-022-00112-5

[17] A.P. Yunus, K. Morita, N.C. Shirai and T. Wakabayashi, "Temporal-Spatial Time Series Self-Attention 2D & 3D Human Motion Forecasting," *2023 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 66-72, 2023. DOI: https://doi.org/10.1109/IAICT59002.2023.10205596

[18] H. Choi, G. Moon and K.M. Lee, "Pose2Mesh: Graph Convolutional Network for 3D Human Pose and Mesh Recovery from a 2D Human Pose," *Lecture Notes in Computer Science*, pp. 769-787, 2020. DOI: https://doi.org/10.1007/978-3-030-58571-6_45

[19] T. N. Kipf and M. Welling, "Semi-Supervised Classification with Graph Convolutional Networks," *Proceedings of the 5th International Conference on Learning Representations (ICLR 2017)*, 2017. DOI: https://doi.org/10.48550/arXiv.1609.02907

[20] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph Attention Networks", *International Conference on Learning Representations,* 2018. DOI: https://doi.org/10.48550/arXiv.1710.10903

[21] A. Hikmah, S. Adi and M. Sulistiyono, "The Best Parameter Tuning on RNN Layers for Indonesian Text Classification," *2020 3rd International Seminar on Research of Information Technology and Intelligent Systems (ISRITI)*, pp. 94-99, 2020. DOI: https://doi.org/10.1109/ISRITI51436.2020.9315425

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735-1780, 1997. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[23] A. Shukurlu, "Improving Deep Knowledge Tracing via Gated Architectures and Adaptive Optimization," *arXiv preprint arXiv:2504.20070*, 2025. DOI: https://doi.org/10.48550/arXiv.2504.20070

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014. DOI: https://doi.org/10.48550/arXiv.1412.3555

[25] A. Bouazizi, A. Holzbock, U. Kressel, K. Dietmayer and V. Belagiannis, "MotionMixer: MLP-based 3D Human Body Pose Forecasting," *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pp. 791-798, 2022. DOI: https://doi.org/10.24963/ijcai.2022/111

[26] D. M. W. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation", arXiv:2010.16061, 2020. DOI: https://doi.org/10.48550/arXiv.2010.16061

[27] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861-874, 2006. DOI: https://doi.org/10.1016/j.patrec.2005.10.010

[28] N.J.D. NAGELKERKE, "A note on a general definition of the coefficient of determination," *Biometrika*, vol. 78, no. 3, pp. 691-692, 1991. DOI: https://doi.org/10.1093/biomet/78.3.691

[29] Y. Ji, "calflops: A FLOPs and Params calculate tool for neural networks in PyTorch framework," 2023. [Online]. Available: https://github.com/MrYxJ/calculate-flops.pytorch