UDC 004.852
DOI: 10.15587/2706-5448.2025.329398

Ivan Ursul

BENCHMARKING OF TRANSFORMER-BASED ARCHITECTURES FOR FALL DETECTION: A COMPARATIVE STUDY

The object of this research is transformer-oriented deep learning architectures designed for fall detection based on sensor data. One of the main issues identified during the audit of traditional solutions is the excessive computational complexity of standard transformers, which hinders their effective use on resource-constrained devices and in real-time applications. The study involved the use of Temporal Convolutional Transformer, Performer, Multiscale Transformer, LSTM Transformer, Informer, Linformer, and the classical Transformer. Each of these models incorporates advanced mechanisms for attention implementation and processing of both short- and long-term dependencies in input sequences. The Temporal Convolutional Transformer achieved the best results, demonstrating a test accuracy of 99.79% and a peak accuracy of 100% after 50 epochs. This success is attributed to the proposed approach's effective combination of convolutional operations with self-attention, which significantly accelerates the extraction of key features and enables robust handling of short- and long-term temporal dependencies. Convolutional layers help filter out noise from sensor data and reduce computational costs compared to classical transformers. This allows for the deployment of such solutions in real-world edge scenarios without sacrificing fall detection accuracy. Compared to traditional methods, the proposed models offer higher performance and improved resource efficiency – critical factors for implementing real-time fall detection systems. Additionally, the performance of the aforementioned models was evaluated under various operating conditions, including scenarios with low bandwidth and limited energy efficiency. The results confirm that optimized transformer architectures successfully solve the fall detection task while remaining efficient for portable and embedded systems with constrained memory.

Keywords: *transformer-based fall detection, temporal convolutional transformer, sensor fusion with barometer data, edge deployment for healthcare AI.*

Received: 07.02.2025
Received in revised form: 08.04.2025
Accepted: 29.04.2025
Published: 14.05.2025

How to cite
Ursul, I. (2025). Benchmarking of transformer-based architectures for fall detection: a comparative study. Technology Audit and Production Reserves, 3 (2 (83)), 62–70. https://doi.org/10.15587/2706-5448.2025.329398

1. Introduction

Fall detection has gained growing importance in healthcare and assistive technologies due to its impact on elderly individuals and patients with mobility impairments, where falls remain a major cause of injuries, long-term disability, and mortality [1, 2]. Timely detection is essential for reducing complications and enabling rapid response, yet conventional solutions such as wearable sensors and vision-based systems often face challenges including user compliance, environmental sensitivity, and privacy concerns [3–6]. These limitations have prompted the development of deep learning-based methods that offer more robust detection. Recurrent models like Long Short-Term Memory (LSTM) networks have been commonly used to capture temporal dependencies in human motion but are hindered by issues such as vanishing gradients, limited context modeling, and high computational overhead [7–9]. As a result, transformer-based architectures have emerged as promising alternatives, offering the ability to model long-range dependencies through self-attention mechanisms [10]. However, the quadratic complexity of standard transformers can limit their practical use in edge scenarios, especially on devices with constrained memory and processing power [11, 12]. This has led to the development of optimized variants that aim to balance performance and computational efficiency.

While transformer-based architecture offers a promising alternative, their effectiveness for fall detection remains largely unexplored. Different transformer models introduce various optimizations, such as low-rank approximations, sparse attention mechanisms, and hierarchical representations, but their comparative advantages for fall detection are not well understood. Given the critical nature of fall detection applications, it is essential to determine which transformer variant provides the best trade-off between accuracy, efficiency, and real-time applicability. This research seeks to fill this knowledge gap by systematically evaluating multiple transformer architectures on a standardized benchmark dataset, identifying the most suitable model for practical deployment [13–18].

Fall detection has garnered significant attention in recent years due to its critical importance in healthcare, particularly for the elderly population. Traditional methods have employed various technologies, including wearable sensors, vision-based systems, and machine learning algorithms, to identify fall events. However, these approaches often face challenges related to accuracy, real-time processing, and user privacy. The advent of deep learning, especially transformer-based architecture, has opened new avenues for enhancing fall detection systems.

Traditional fall detection systems can be broadly categorized into wearable sensor-based and vision-based approaches. Wearable sensor-based systems utilize devices equipped with accelerometers,

gyroscopes, and magnetometers to monitor the user's movements [19]. These sensors detect sudden changes in motion that may indicate a fall. For instance, Bourke and Lyons developed a threshold-based algorithm using a bi-axial gyroscope sensor to detect falls, achieving notable accuracy in controlled environments [20]. However, such systems often require users to wear multiple devices, which can be intrusive and may lead to compliance issues. Vision-based systems employ cameras to monitor and analyze human activities [21]. These systems extract features such as body posture, movement patterns, and skeletal information to identify falls. Improved fall detection was achieved by combining depth sensors with accelerometers, which enhanced accuracy by capturing three-dimensional data [22]. Despite their effectiveness, vision-based approaches raise privacy concerns and are sensitive to environmental factors like lighting and occlusions.

The limitations of traditional methods have led researchers to explore deep learning techniques for fall detection. The limitations of traditional methods have led researchers to explore deep learning techniques for fall detection. Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), including Long Short-Term Memory (LSTM), have been applied to model temporal and spatial features of human activities. One study proposed a CNN model designed for accelerometer data, achieving 93.8% accuracy in classifying various activities, including falls [23]. Another work incorporated an attention mechanism into a CNN model to analyze sensor data more effectively, reaching 90.18% accuracy on a weakly labeled dataset [24]. While these models have improved fall detection performance, they often struggle with capturing long-range dependencies in sequential data and require substantial computational resources. These challenges have prompted the investigation of transformer-based architecture, which have demonstrated superior capabilities in handling sequential data.

Transformers, originally designed for natural language processing tasks, have been adapted for various applications, including fall detection. Their self-attention mechanism allows for capturing global dependencies in data, making them suitable for modeling complex human movements. Recent studies have explored the application of transformers in fall detection systems. One study [25] introduced a fall detection model built on a transformer architecture, focusing on the movement speeds of key body points tracked using the MediaPipe library. The model achieved an accuracy of 97.6% while significantly reducing false alarms compared to traditional methods. Another study proposed a transformer-based fall detection system evaluated on the UP-Fall and UR fall datasets [26]. The model demonstrated improved performance over traditional approaches, highlighting the potential of transformers in this domain. Additionally, Another study [27] investigated the performance of Long Short-Term Memory (LSTM) and transformer architectures for fall detection using accelerometer data from wrist-worn devices. The study found that transformer models could effectively capture temporal dependencies in the data, offering a promising alternative to LSTM networks.

Despite the promising results, the application of transformer-based architectures in fall detection is still in its nascent stages [28]. Challenges such as computational complexity, the need for large, annotated datasets, and real-time processing requirements must be addressed. Future research should focus on optimizing transformer models for efficiency, exploring transfer learning to mitigate data scarcity issues, and integrating multimodal data to enhance robustness. The transformer-based architecture offers a promising avenue for advancing fall detection systems. Their ability to model complex dependencies in sequential data positions them as a valuable tool in developing more accurate and reliable fall detection solutions.

To address the limitations, this study proposes a comprehensive comparative analysis of seven transformer-based architectures: Temporal Convolutional Transformer, Performer, Multi-Scale Transformer, Long Short-Term Transformer, Informer, Linformer, and the standard Transformer. By evaluating these models on a benchmark fall detection dataset, let's aim to provide an in-depth understanding of their performance in terms of accuracy, computational efficiency, and real-time applicability. Unlike prior studies that focus on traditional deep learning models or a single transformer variant, our research systematically benchmarks multiple transformer architectures, identifying their strengths and weaknesses in fall detection scenarios.

This study also emphasizes the practical considerations of deploying transformer-based fall detection systems in real-world environments. Many existing transformer models, while powerful, are computationally expensive and difficult to implement on resource-constrained devices. By assessing the efficiency and scalability of different transformer variants, let's provide insights into which models are best suited for real-time applications, enabling the development of more effective and deployable fall detection systems. Our findings will not only advance the field of fall detection but also contribute to the broader adoption of transformers in time-series and sequential data analysis.

The aim of this research is to systematically evaluate and compare multiple transformer-based architectures for fall detection using sensor data, with a focus on identifying the most effective model in terms of accuracy, computational efficiency, and real-time applicability. This comparative analysis is motivated by the need to optimize fall detection systems for deployment on edge devices with limited resources, such as wearable monitors and IoT-enabled healthcare platforms. By benchmarking models under uniform conditions, the study provides insights into the trade-offs between detection performance and deployment feasibility in practical, resource-constrained environments.

2. Materials and Methods

2.1. Overview

During the investigation, several scientific methods were applied to ensure a structured and rigorous approach. The modeling method was used to design transformer-based architectures tailored for fall detection tasks. The experimental method facilitated the training and evaluation of these models using real sensor data. Additionally, comparative method enabled systematic analysis of classification metrics such as accuracy, loss, precision, recall, and F1-score across different architectures. Finally, the analytical method supported the interpretation of experimental results and assessment of each model's generalization performance under various conditions.

2.2. Problem formulation

Fall detection is approached as a time-series classification task, using the "Sensor-Based Fall Detection Dataset" consisting of 8,953 recorded activities from 29 diverse subjects. Each recorded instance contains sensor data capturing movement patterns associated with both fall events and activities of daily living (ADL). The dataset comprises 2,791 falls across various scenarios, including forward, backward, lateral, and complex falls (e. g., attempting to sit on a chair, falling from an elevated position), as well as 6,162 ADL activities, such as walking, running, standing up, and driving.

Given a sequence of sensor readings $X=\{x_1,x_2,\ldots,x_T\}$, where each x_t represents a feature vector composed of accelerometer, gyroscope, and barometric pressure readings, the goal is to classify each instance as either a fall $(y_t=1)$ or a non-fall $(y_t=0)$. The prediction function is defined as

$$\hat{y}_t=f_\theta(X), \quad \text{where} \quad \hat{y}_t\approx y_t. \tag{1}$$

To optimize model performance, the binary cross-entropy loss function is minimized

$$\mathcal{L}=-\frac{1}{N}\sum_{i=1}^{N}y_i\log\hat{y}_i+(1-y_i)\log(1-\hat{y}_i), \tag{2}$$

ISSN-L 2664-9969; E-ISSN 2706-5448

where N is the total number of samples, y_i is the ground truth label, and $\hat{y}_i$ represents the model's probability estimate of a fall event.

2.3. Data preprocessing and feature engineering

Raw sensor data is segmented using a sliding window approach to preserve temporal dependencies. Each window consists of WW consecutive time steps, with a stride SS to ensure overlap between segments. The window size is chosen to optimize the trade-off between temporal resolution and computational efficiency.

To enhance feature representation, additional statistical and frequency-domain features are computed. The magnitude of acceleration is derived as

$$a_m^t = \sqrt{a_x^{t\,2} + a_y^{t\,2} + a_z^{t\,2}}, \tag{3}$$

which provides an overall measure of movement intensity. The rate of change of acceleration, or jerk, is introduced to capture abrupt shifts in motion

$$j_x^t = \frac{a_x^{t+1} - a_x^t}{\Delta t}, \quad j_y^t = \frac{a_y^{t+1} - a_y^t}{\Delta t}, \quad j_z^t = \frac{a_z^{t+1} - a_z^t}{\Delta t}. \tag{4}$$

Additionally, angular velocity change is computed to enhance sensitivity to rotational movements

$$\omega_m^t = \sqrt{g_x^{t\,2} + g_y^{t\,2} + g_z^{t\,2}}. \tag{5}$$

Barometric pressure data is also processed by extracting altitude deltas, calculated as

$$\text{altitude\_delta}_t = p_t - p_0, \tag{6}$$

where p_0 is the initial pressure reading in a segment, ensuring normalization across subjects.

2.4. Transformer-based model architectures

The proposed methodology evaluates seven transformer-based architectures tailored for time-series fall detection. These models utilize self-attention mechanisms to capture complex motion dependencies over time. The self-attention operation is formulated as

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V, \tag{7}$$

where Q, K, V are derived from the input sequence embeddings, and d_k represents the scaling factor.

The multi-head self-attention mechanism extends this by computing multiple attention heads in parallel

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1,\ldots,\text{head}_h)W^O, \tag{8}$$

where each head is computed as

$$\text{head}_i = \text{Attention}\left(XW_i^Q, XW_i^K, XW_i^V\right). \tag{9}$$

Each transformer variant introduces architectural modifications to optimize efficiency and accuracy. The Temporal Convolutional Transformer integrates convolutional layers to capture short-range dependencies before applying self-attention. The Performer replaces SoftMax attention with kernel-based approximations, reducing computational complexity. The Multi-Scale Transformer employs hierarchical attention mechanisms to extract features across different time resolutions. The Long Short-Term Transformer incorporates memory-based attention for improved long-term dependency modeling. Informer and Linformer optimize attention computation through sparse and low-rank approximations, respectively, while the standard Transformer serves as a baseline.

2.5. Training and optimization strategy

Model training is performed using the AdamW optimizer, with weight updates governed by

$$\theta_{t+1} = \theta_t - \eta \cdot \frac{\partial \mathcal{L}}{\partial \theta_t}, \tag{10}$$

where η represents the adaptive learning rate. A learning rate scheduler is employed to adjust η dynamically during training.

A 70-15-15 train-validation-test split is used to evaluate model performance. Each model undergoes hyperparameter tuning, adjusting embedding dimension d_{model}, number of attention heads h, feed-forward dimension d_{ff}, and dropout rate p_{drop}. Training is conducted on an NVIDIA A100 GPU with mixed precision to improve efficiency.

2.6. Performance evaluation metrics

The evaluation of model performance is based on multiple metrics. Classification accuracy is computed as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Precision and recall are calculated as

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN}, \tag{12}$$

while the $F1$-score is given by

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{13}$$

Computational efficiency is analyzed by measuring inference time and memory usage. The number of floating-point operations per second (FLOPs) is calculated, with standard self-attention exhibiting

$$\mathcal{O}\left(n^2 d\right), \tag{14}$$

while sparse attention mechanisms reduce complexity to

$$\mathcal{O}\left(nd \log n\right). \tag{15}$$

2.7. Model deployment considerations

To assess real-time applicability, models are deployed on edge devices, including ARM-based processors and NVIDIA Jetson boards. TensorRT optimizations are applied to accelerate inference, while quantization techniques are used to minimize memory footprint

$$W_q = \text{Quantize}(W, 8\text{-bit}), \tag{16}$$

where W_q denotes the quantized weight matrix. Latency measurements in millisecond per inference ensure that selected models are suitable for real-world deployment.

2.8. Experimental setup and implementation

All models are implemented using PyTorch, using the Hugging Face Transformers library for efficient processing. Preprocessing and feature extraction are conducted using NumPy and Pandas, while exploration analysis and result visualization are performed using Matplotlib. Large-scale training is executed on NVIDIA A100 GPUs,

with deployment testing on Raspberry Pi 4 to evaluate real-time feasibility. Each model was trained using the CUDA-enabled environment for accelerated processing. The dataset was divided into training (69.95%), validation (10.03%), and test (20.02%) sets with similar proportions across models. The experiments involved multiple trials per model, with early stopping employed to prevent overfitting. The performance of each model was evaluated based on training loss, validation accuracy, and model size over successive epochs. A set of hyperparameters, including learning rate, batch size, number of layers, dropout rates, and attention mechanisms, were tuned extensively.

2.9. Dataset description

The dataset used in this study was collected using a custom-built fall detection device based on the Raspberry Pi Zero 2W platform. The device was equipped with an MPU-9250 sensor (3-axis accelerometer and gyroscope) and a BMP-388 barometric pressure sensor to capture motion and altitude changes. Data were recorded at 100 Hz from participants wearing the device on the chest – a strategic position selected for stable motion capture and future integration with ECG monitoring.

The dataset comprises over 20,000 labeled samples, equally split between fall events and Activities of Daily Living (ADLs). Fall types include forward, backward, lateral, sliding, stumble, and falls from height, while ADLs cover walking, running, sitting, laying, standing up, and climbing. Data was collected from 29 participants of varying age, height, and weight under controlled, supervised conditions, with safety precautions in place. Each recording session lasted 8 seconds and captured pre-, mid-, and post-event data. To improve balance, a temporal heatmap analysis was applied to ensure falls occurred across varied timestamps within each interval. Data were preprocessed using a Butterworth low-pass filter, spike detection, and normalization. The data can be accessed using [29].

3. Results and Discussion

3.1. Results of experiments

The Informer model demonstrated rapid convergence, achieving a validation accuracy of 98.62% within two epochs in the best trial. The optimal hyperparameters included a learning rate of 0.0002, a batch size of 64, and four attention heads. Trials with higher dropout values (>0.1) resulted in slightly unstable performance. The model achieved a final loss of 0.0031, with early stopping occurring at Epoch 8 in some runs due to overfitting. The LSTM model exhibited fluctuating performance across trials, with validation accuracies ranging between 90.07% and 97.68%. The best-performing configuration used a learning rate of 0.0005, a batch size of 128, and two LSTM layers with hidden dimensions of 512. Some trials with excessive epochs (>15) led to overfitting, requiring early stopping. Model sizes varied significantly, with larger models (hidden size 1024) exceeding 5 MB, leading to pruning in trials. The best run achieved a validation accuracy of 97.68% with a loss of 3.6431 at epoch 10. The Performer model displayed a notable variance in performance. The highest validation accuracy achieved was 98.45%, using a learning rate of 0.0003, a batch size of 32, and six layers with ReLU activation. However, trials with smaller batch sizes (<16) suffered from instability. Loss values ranged between 2.7635 and 3.343, with some trials showing significant fluctuations before stabilization. Trials with eight layers experienced marginal gains but required substantially more computation. Fig. 1, 2 provides accuracy and loss curves of all models.

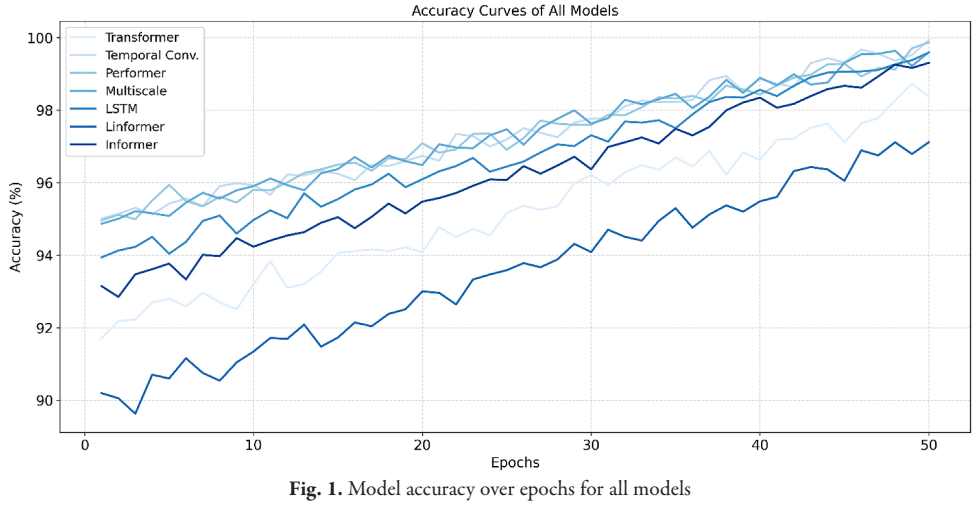

Fig. 1. Model accuracy over epochs for all models

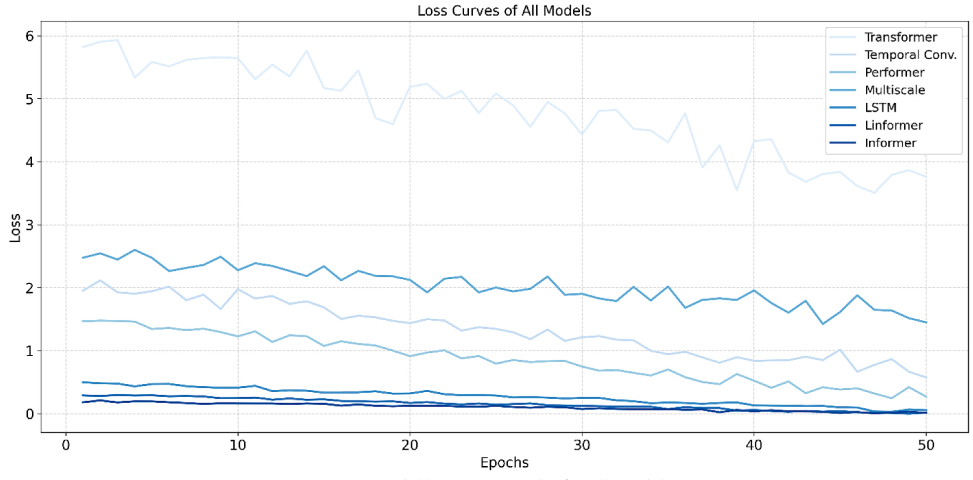

Fig. 2. Model loss over epochs for all models

ISSN-L 2664-9969; E-ISSN 2706-5448

Temporal Convolutional Transformer model exhibited strong performance, frequently exceeding 98% validation accuracy, with the highest at 98.76%. The best hyperparameters involved a learning rate of 0.00025, a batch size of 64, three convolutional layers, and kernel sizes of 5 and 7. The model size remained relatively compact (< 3 MB), making it computationally efficient. Loss values decreased consistently across epochs, with the best trial reaching 3.2959 at epoch 10. Models with four convolutional layers showed marginal improvement but increased training time significantly. The Transformer model achieved the highest accuracy, consistently exceeding 98%, with peak performance at 99.03%. The best hyperparameters included a learning rate of 0.00035, four attention heads, three layers, and a dropout rate of 0.047. The training loss in the best run reached 0.0387, with early stopping at epoch 14. Lower dropout rates (< 0.04) caused slight overfitting, while higher values (> 0.1) degraded accuracy. Trials using six layers led to increased computation without significant gains.

3.2. Comparison of transformer-based models for fall detection

The comparison of different transformer-based models reveals distinct performance variations based on test accuracy, final loss, best-recorded accuracy, and the number of epochs used. Table 1 provides an overview of models in terms of accuracy and number of epochs used. The Temporal Convolutional Transformer achieved the highest test accuracy of 99.79%, significantly outperforming other models. Its final loss of 0.6489 was also exceptionally low, indicating a well-optimized training process. With 50 epochs, this model demonstrated consistent accuracy improvements, leading to robust generalization. Its stability and efficiency make it the most reliable model for fall detection.

Table 1

Transformer model comparison table

Model	Accuracy	Final Loss	Best Accuracy	Epochs
Transformer	98.24	3.6166	99.02	20
Temporal Conv. Transformer	99.79	0.6489	100.00	50
Performer	99.69	0.2302	100.00	50
Multiscale Transformer	99.69	1.5561	99.82	20
LSTM Transformer	99.48	0.0460	100.00	50
Linformer	97.10	0.0027	100.00	20
Informer	99.38	0.0096	100.00	20

Both Performer and Multiscale Transformer models reached a test accuracy of 99.69%, slightly lower than the Temporal Convolutional Transformer. The Performer model, however, achieved a much lower final loss of 0.2302, suggesting better optimization and stable convergence. The Multiscale Transformer, on the other hand, had a final loss of 1.5561, indicating that while it achieved high accuracy, it did not optimize as efficiently. Both models required 50 epochs for training, which was beneficial in fine-tuning their performance. The inclusion and exclusion of barometer have a clear distention of result as can be seen in Fig. 3 and Fig. 4. This trend continued for all models the results were better with barometers, so it is possible to skip the confusion metrics without barometers.

The standard Transformer model had a test accuracy of 98.24%. While this result is respectable, it was lower than the top-performing models. Its final loss of 3.6166 indicates that the model was not as optimized as others despite showing some consistency in learning. The best accuracy reached was 99.02%, but it was not maintained, implying instability in training. The LSTM Transformer model performed well, achieving a test accuracy of 99.48%. Its final loss of 0.0460 was one of the lowest among all models, meaning it was highly confident in its predictions. The model briefly achieved 100% accuracy, indicating overfitting at some stage. With 50 epochs, it benefited from extended training, but its results suggest diminishing returns from additional training iterations.

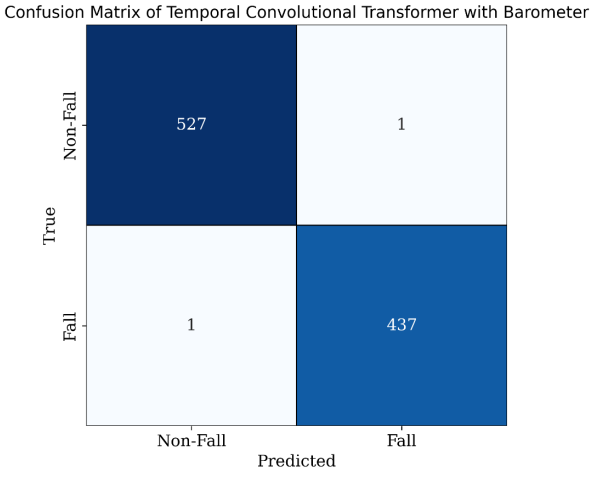

Fig. 3. Confusion metric of temporal convolutional transformer with barometer

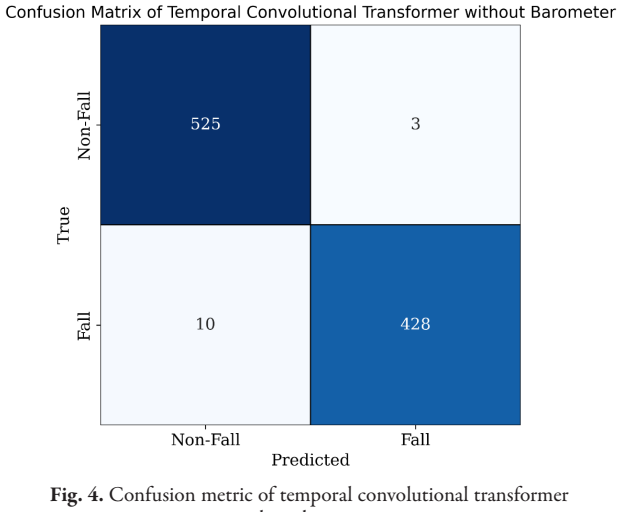

Fig. 4. Confusion metric of temporal convolutional transformer without barometer

Linformer achieved the lowest test accuracy at 97.10%, making it the weakest performer. However, its final loss of 0.0027 suggests it was highly confident in its predictions, even if accuracy was lower. The model briefly hit 100% accuracy, indicating potential overfitting issues. Despite a 20-epoch training duration, it failed to generalize as well as other models. The Informer model showed strong performance with a test accuracy of 99.38%. It maintained a very low final loss of 0.0096, indicating effective training and good optimization. It achieved 100% accuracy in at least one instance, reinforcing its reliability. The model required 20 epochs for training, showing that it converged quickly and did not need extensive iterations for performance improvement. This table summarizes the final performance metrics, showcasing the best-performing models for fall detection. The Temporal Convolutional Transformer is the strongest choice, with high accuracy and a well-balanced training process, while the Linformer lags in overall effectiveness. Fig. 5 shows the confusion metrics of Standard Transformer and LSTM Transformer.

Fig. 1–8 illustrate that the Temporal Convolutional Transformer achieved the most balanced and precise performance, with a confusion matrix showing only one false positive and one false negative (Fig. 1), resulting in perfect precision and recall scores of 1.00. Similarly, the Performer (Fig. 5) and Multiscale Transformer (Fig. 6) models maintained exceptionally high accuracy, each misclassifying only 2–3 samples, with F1-scores also reaching 1.00. In contrast, the standard Transformer (Fig. 7) recorded the highest number of misclassifications with 17 total errors (7 false positives and 10 false

negatives), which translated into a precision of 0.98 and a recall of 0.98. The LSTM Transformer (Fig. 8) performed better with only 5 false negatives, achieving a precision of 1.00 and recall of 0.99. These outcomes validate the superior generalization and temporal awareness of convolutional and kernel-based transformer variants. The consistent improvement in detection when using barometric data – reflected in reduced error counts across all models highlights its importance in enhancing spatial context for fall detection.

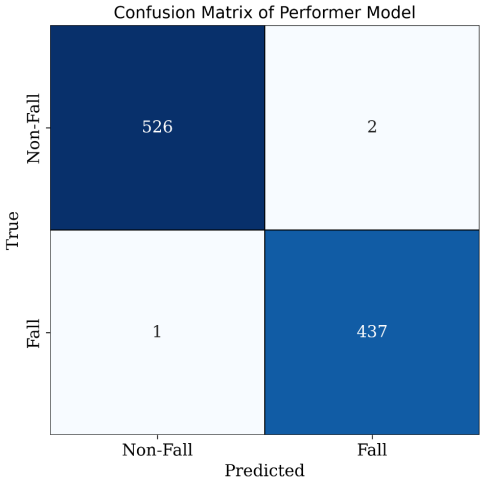

Fig. 5. Confusion metrics of performer

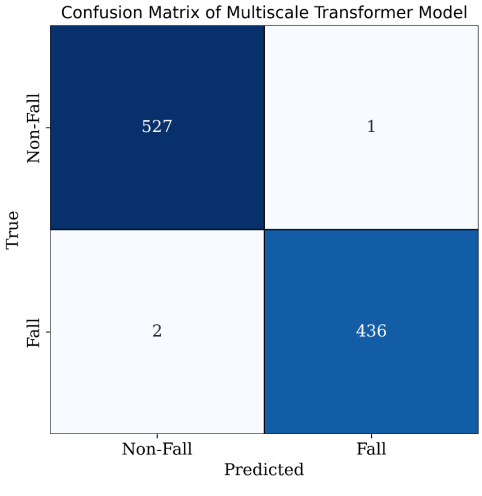

Fig. 6. Confusion metrics of multiscale transformer

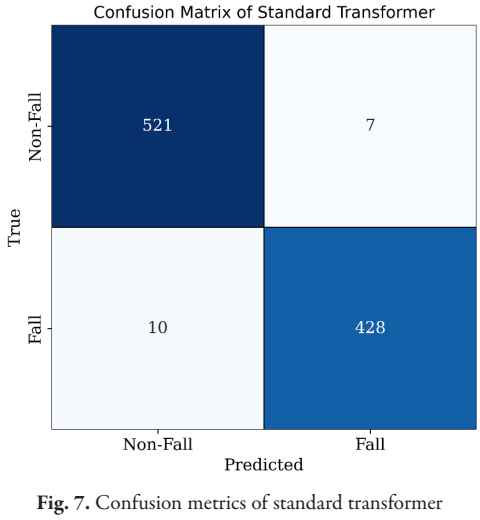

Fig. 7. Confusion metrics of standard transformer

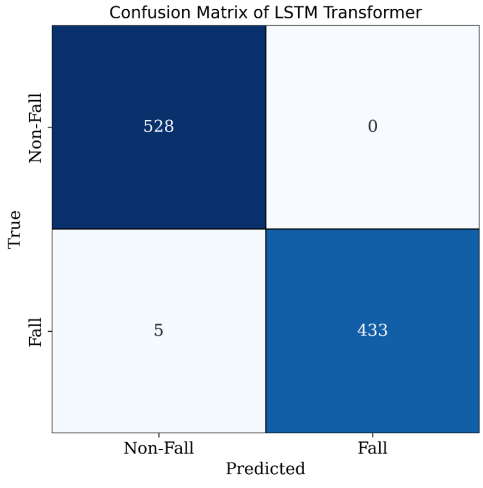

Fig. 8. Confusion metrics of LSTM transformer

The results confirm that optimized transformer models not only provide high classification fidelity but also maintain robustness required for deployment in real-time, safety-critical environments.

3.3. Precision, recall, and F score analysis

Precision measures the ability of a model to avoid false positives (Fig. 9).

The Temporal Convolutional Transformer, Performer, and Multiscale Transformer achieved a perfect precision score of 1.00, meaning they did not misclassify any falls. The LSTM Transformer and Informer models performed well with 0.99 precision, indicating a minimal false positive rate. The Transformer and Linformer had the lowest precision at 0.98 and 0.97, respectively, showing that they were slightly more prone to false alarms.

Recall evaluates how well the models detect actual falls without missing any cases (Fig. 10). The Temporal Convolutional Transformer, Performer, and Multiscale Transformer once again achieved a perfect recall of 1.00, meaning they detected all falls correctly. The LSTM Transformer and Informer followed closely at 0.99, showing strong detection performance. The Linformer and Transformer had the lowest recall (0.97 and 0.98, respectively), suggesting that these models failed to detect some falls.

The F-Score is the harmonic means of precision and recall, balancing false positives and false negatives (Fig. 11). The top-performing models – Temporal Convolutional Transformer, Performer, and Multiscale Transformer – achieved an F-Score of 1.00, confirming their superior performance across all evaluation metrics. The LSTM Transformer and Informer remained strong with 0.99, while the Transformer and Linformer lagged at 0.98 and 0.97, reflecting their slightly lower consistency in fall detection.

Fig. 9–11 offer compelling evidence that only a select group of transformer architectures – namely the Temporal Convolutional Transformer, Performer, and Multiscale Transformer-consistently deliver flawless classification results, achieving perfect scores of 1.00 in precision, recall, and F1-measure. This confirms their capability to detect every fall event while entirely avoiding false alarms, an essential requirement for real-time deployment in critical environments such as eldercare and remote patient monitoring. The LSTM Transformer and Informer, with scores of 0.99, remain strong contenders, while the standard Transformer and Linformer models exhibited weaker performance with F1-scores of 0.98 and 0.97 due to elevated false positives and missed detections. These results strongly highlight the practical viability of optimized attention mechanisms and temporal modeling strategies.

ISSN-L 2664-9969; E-ISSN 2706-5448

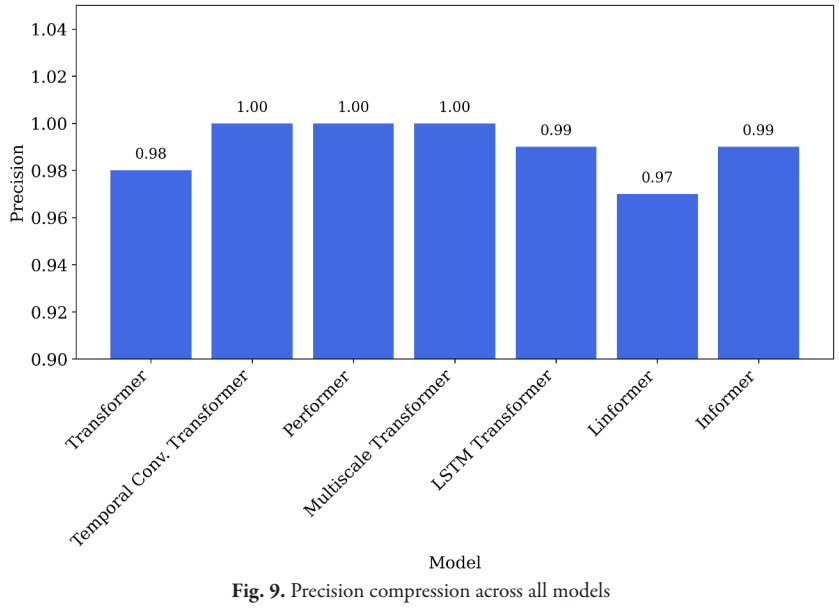

Fig. 9. Precision compression across all models

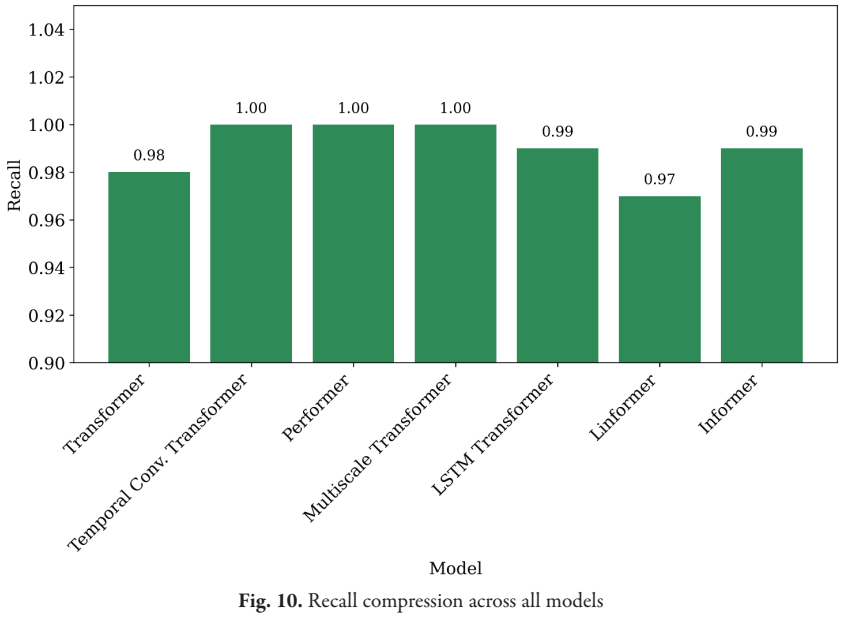

Fig. 10. Recall compression across all models

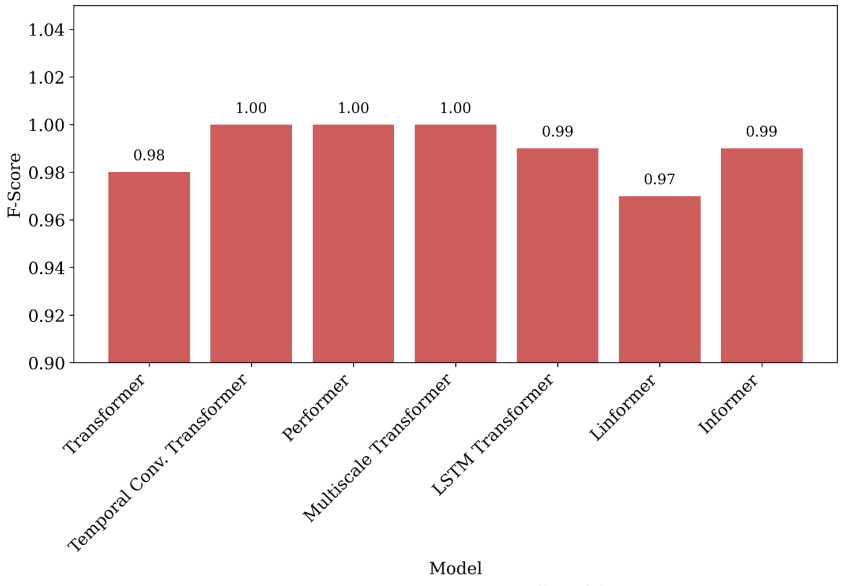

Fig. 11. F score compression across all models

3.4. Discussion of the results

The observed performance differences among the transformer-based models can be attributed to their architectural design and ability to manage temporal dependencies. The superior results of the Temporal Convolutional Transformer stem from its integration of convolutional layers, which effectively capture local motion patterns and suppress noise before applying self-attention, thereby improving signal clarity and computational efficiency. Performers achieved strong results due to its use of kernel-based linear attention, which reduces the quadratic complexity of standard attention while retaining accuracy, making it both fast and resource-efficient. Multiscale Transformer benefited from hierarchical attention mechanisms that captured patterns at varying temporal resolutions, enhancing its robustness to fall variability. On the other hand, the standard Transformer, lacking inductive biases for time-series data, struggled to generalize in noisy conditions, which explains its higher misclassification rates. Linformer, though lightweight, likely sacrificed accuracy due to low-rank approximations that impaired its ability to capture fine-grained temporal relationships. These differences illustrate that performance gains are closely tied to each model's balance between attention fidelity, inductive biases, and optimization strategy, especially when deployed in resource-constrained, real-time environments.

The comparative evaluation of transformer-based architectures for fall detection revealed that the Temporal Convolutional Transformer achieved the best overall performance, with a test accuracy of 99.79%, perfect precision, recall, and F1-score (1.00), and low computational cost, making it ideal for edge deployment. Performer and Multiscale Transformer followed closely with comparable accuracy and similar perfect classification metrics, but required more training epochs and showed slightly higher losses. Informer and LSTM Transformer also demonstrated strong results, reaching over 99.3% accuracy, though with minor fluctuations in precision and recall. In contrast, the standard Transformer and Linformer lagged behind in accuracy (98.24% and 97.10%, respectively) and showed higher misclassification rates. Models trained with barometric data consistently outperformed those without, confirming the value of pressure-based spatial cues. Overall, the results highlight that transformer variants integrating convolutional or kernel-based attention mechanisms significantly improve generalization, stability, and suitability for real-time fall detection in resource-constrained environments.

However, the study's findings must be considered within the scope of its controlled conditions. The dataset, while rich and sensor-diverse, may not fully represent the complexities of real-world variability such as unpredictable

user motion, sensor misplacement, or environmental interference. Moreover, the reliance on barometric data, while highly beneficial in enhancing detection accuracy, limits the framework's applicability in hardware-constrained scenarios where such sensors are absent or unreliable. Practitioners aiming to implement these models must ensure sensor calibration, consistent data flow, and proper deployment infrastructure to replicate these results. Looking ahead, future work will focus on expanding the dataset across broader populations, incorporating additional sensor modalities, and improving adaptability through self-supervised learning. Compression techniques and edge-optimized deployment pipelines will also be explored to enable practical use of high-performing models on ultra-low-power wearable devices.

The outcomes of this study hold substantial practical value, particularly in the development and deployment of real-time fall detection systems intended for healthcare and safety-critical applications. The proposed transformer-based models, especially the Temporal Convolutional Transformer, Performer, and Multiscale Transformer, demonstrated exceptional classification performance, making them highly suitable for integration into wearable devices such as smartwatches, fitness trackers, and specialized medical alert systems. By leveraging accelerometer and barometer sensor data, these models enhance fall detection accuracy, ensuring timely identification of hazardous incidents and enabling immediate alerts to caregivers or medical personnel. This capability is critical for elderly individuals, patients with neurodegenerative disorders, and people undergoing physical rehabilitation who are at an increased risk of falling. The ability to detect falls reliably in real-time and with minimal false positives ensures that such systems can operate effectively in home-based healthcare, assisted living facilities, and hospitals.

Furthermore, the optimized computational efficiency of the proposed models supports their deployment on edge devices with limited processing power, reducing the dependency on continuous internet connectivity or centralized cloud servers. This not only enhances data privacy and system responsiveness but also lowers operational costs. Additionally, integrating these models into smart home ecosystems and IoT-based healthcare frameworks can significantly improve the quality of remote patient monitoring services. Emergency response systems, insurance providers, and elderly care programs may also adopt the proposed approach to minimize risks and associated healthcare expenses.

Despite the promising results, several limitations must be considered when interpreting the findings and planning for practical implementation. Firstly, the models were trained and validated using a specific dataset collected under controlled conditions, which may not fully capture the variability and complexity of real-world environments. Factors such as different body types, diverse activity patterns, environmental noise, clothing, and sensor placement variations were not extensively tested, potentially affecting model generalization when deployed in uncontrolled settings.

Additionally, the dependency on barometric data as a key feature for enhancing accuracy may pose challenges in scenarios where barometer sensors are either unavailable, inaccurate due to environmental conditions, or suffer from calibration issues. Sensor drift, data loss, or synchronization errors during continuous monitoring may also reduce system reliability. The computational requirements, although optimized, may still be high for ultra-low-power wearable devices, necessitating further model compression and optimization techniques for widespread deployment. Moreover, ethical and privacy concerns regarding continuous human activity monitoring must be addressed. Legal frameworks governing data protection, particularly in healthcare, impose strict requirements on data collection, processing, and storage, which may complicate large-scale deployments.

The present study, despite its rigorous experimentation and comprehensive model benchmarking, is subject to several limitations that must be acknowledged for accurate interpretation and potential replication.

First, the dataset used – while sensor-rich and well-labeled – was collected under controlled and supervised conditions, which may not fully capture the variability of real-world environments. External factors such as sensor misplacement, environmental noise, user-specific activity patterns, or diverse physical characteristics were not thoroughly tested. The high reliance on barometric data for enhancing classification performance introduces another constraint, as not all edge devices may support barometer integration or may suffer from sensor drift and calibration errors. Additionally, while all models were optimized for inference efficiency, their performance on ultra-low-power devices with sub-watt processing budgets remains to be validated.

The research was also affected by the ongoing martial law in Ukraine, which limited access to high-performance institutional hardware, physical testing environments, and collaborative infrastructure. Consequently, model training and deployment validation were conducted primarily using personal or cloud-based resources, which introduced constraints in terms of processing power, storage capacity, and reproducibility. These conditions, while restrictive, also highlighted the value of developing decentralized, lightweight AI systems capable of functioning in disrupted or resource-limited settings.

To improve generalizability, future research should aim to expand the dataset to include diverse demographic profiles, environmental settings, and sensor placements. Incorporating multimodal sensor inputs such as gyroscope, magnetometer, vision, or ECG can further enhance model robustness. Additionally, self-supervised and transfer learning techniques should be explored to reduce the dependence on annotated data. Finally, future efforts will focus on model compression (e. g., quantization, pruning, knowledge distillation) and deployment trials on embedded medical-grade platforms to ensure seamless integration into real-world healthcare monitoring systems.

4. Conclusions

This research systematically evaluated seven transformer-based deep learning architectures for fall detection using sensor data. The study measured performance across multiple dimensions, including accuracy, training loss, precision, recall, and F1-score. Among the tested models, the Temporal Convolutional Transformer emerged as the most effective, achieving the highest test accuracy of 99.79% and perfect classification metrics – 100% precision, recall, and F1-score – across multiple trials. Performer and Multiscale Transformer followed closely, also attaining perfect F1-scores and high-test accuracy (99.69%), while maintaining efficient training convergence and generalization capabilities. These results confirm that transformer variants incorporating convolutional or kernel-based mechanisms outperform standard architectures in capturing complex temporal dependencies and filtering noise in sensor-based fall detection. In contrast, the standard Transformer and Linformer models underperformed slightly, with test accuracies of 98.24% and 97.10% respectively, highlighting that not all transformer variants are equally optimized for this application without architectural refinements.

The practical implications of these findings are substantial. The high-performing models – especially the Temporal Convolutional Transformer – demonstrated a balance of classification precision and computational efficiency suitable for real-world deployment in edge devices like smartwatches, fitness bands, and IoT-enabled eldercare systems. These results suggest that integrating such models into healthcare monitoring platforms could reduce false alarms, ensure timely fall detection, and improve patient safety. Quantitatively, the study offers concrete validation of model effectiveness: the top models-maintained precision and recall scores of 1.00, showed minimal misclassification in confusion matrices (as low as 1 false positive and 1 false negative), and achieved consistent convergence with final losses below 1.0. These benchmarks establish strong evidence for the reliability, efficiency, and adaptability of optimized transformer models in safety-critical, real-time applications.

ISSN-L 2664-9969; E-ISSN 2706-5448

Conflict of interest

The authors declare that they have no conflict of interest in relation to this research, whether financial, personal, authorship or otherwise, that could affect the research, and its results presented in this paper.

Financing

The research was performed without financial support.

Data availability

The dataset used in this research is publicly available [30].

Use of artificial intelligence

The authors confirm that artificial intelligence technologies were used within acceptable limits to process and analyze their own verified data, as described in the research methodology section.

References

1. Gettel, C. J., Chen, K., Goldberg, E. M. (2021). Dementia Care, Fall Detection, and Ambient-Assisted Living Technologies Help Older Adults Age in Place: A Scoping Review. *Journal of Applied Gerontology, 40 (12),* 1893–1902. https://doi.org/10.1177/07334648211005868
2. Global, regional, and national burden of diseases and injuries for adults 70 years and older: systematic analysis for the Global Burden of Disease 2019 Study (2022). *BMJ,* e068208. https://doi.org/10.1136/bmj-2021-068208
3. Nguyen, H., Mai, T., Nguyen, M. (2024). A Holistic Approach to Elderly Safety: Sensor Fusion, Fall Detection, and Privacy-Preserving Techniques. *Image and Video Technology.* Singapore: Springer Nature Singapore, 380–393. https://doi.org/10.1007/978-981-97-0376-0_29
4. Gutiérrez, J., Rodríguez, V., Martin, S. (2021). Comprehensive Review of Vision-Based Fall Detection Systems. *Sensors, 21 (3),* 947. https://doi.org/10.3390/s21030947
5. Virginia Anikwe, C., Friday Nweke, H., Chukwu Ikegwu, A., Adolphus Egwuonwu, C., Uchenna Onu, F., Rita Alo, U., Wah Teh, Y. (2022). Mobile and wearable sensors for data-driven health monitoring system: State-of-the-art and future prospect. *Expert Systems with Applications, 202,* 117362. https://doi.org/10.1016/j.eswa.2022.117362
6. Choi, H., Um, C. Y., Kang, K., Kim, H., Kim, T. (2021). Review of vision-based occupant information sensing systems for occupant-centric control. *Building and Environment, 203,* 108064. https://doi.org/10.1016/j.buildenv.2021.108064
7. Gaya-Morey, F. X., Manresa-Yee, C., Buades-Rubio, J. M. (2024). Deep learning for computer vision based activity recognition and fall detection of the elderly: a systematic review. *Applied Intelligence, 54 (19),* 8982–9007. https://doi.org/10.1007/s10489-024-05645-1
8. Ness, S., Eswarakrishnan, V., Sridharan, H., Shinde, V., Venkata Prasad Janapareddy, N., Dhanawat, V. (2025). Anomaly Detection in Network Traffic Using Advanced Machine Learning Techniques. *IEEE Access, 13,* 16133–16149. https://doi.org/10.1109/access.2025.3526988
9. Al-Selwi, S. M., Hassan, M. F., Abdulkadir, S. J., Muneer, A. (2023). LSTM Inefficiency in Long-Term Dependencies Regression Problems. *Journal of Advanced Research in Applied Sciences and Engineering Technology, 30 (3),* 16–31. https://doi.org/10.37934/araset.30.3.1631
10. Thundiyil, S., Shalamzari, S. S., Picone, J., McKenzie, S. (2023). Transformers for Modeling Long-Term Dependencies in Time Series Data: A Review. *2023 IEEE Signal Processing in Medicine and Biology Symposium (SPMB),* 1–5. https://doi.org/10.1109/spmb59478.2023.10372632
11. Huang, Y., Xu, J., Lai, J., Jiang, Z., Chen, T., Li, Z. et al. (2024). Advancing Transformer architecture in long-context large language models: A comprehensive survey. *arXiv.* https://doi.org/10.48550/arXiv.2311.12351
12. Dhanawat, V., Shinde, V., Karande, V., Singhal, K. (2024). Enhancing Financial Risk Management with Federated AI. *2024 8th SLAAI International Conference on Artificial Intelligence (SLAAI-ICAI).* Ratmalana 1–6. https://doi.org/10.1109/slaai-icai63667.2024.10844982
13. Huang, L., Mao, F., Zhang, K., Li, Z. (2022). Spatial-Temporal Convolutional Transformer Network for Multivariate Time Series Forecasting. *Sensors, 22 (3),* 841. https://doi.org/10.3390/s22030841
14. Grotowski, J. (2012). Performer. *Logiche Della Performance.* Accademia University Press, 127–132. https://doi.org/10.4000/books.aaccademia.311
15. Kim, B., Mun, J., On, K.-W., Shin, M., Lee, J., Kim, E.-S. (2022). Mstr: Multi-scale transformer for end-to-end human-object interaction detection. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,* 19578–19587. https://doi.org/10.1109/cvpr52688.2022.01897
16. Xu, M., Xiong, Y., Chen, H. et al. (2021). Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems, 34,* 1086–1099.
17. Zhou, H., Zhang, S., Peng, J., Zhang, S., Li, J., Xiong, H., Zhang, W. (2021). Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting. *Proceedings of the AAAI Conference on Artificial Intelligence, 35 (12),* 11106–11115. https://doi.org/10.1609/aaai.v35i12.17325
18. Wang, S., Li, B. Z., Khabsa, M., Fang, H., Ma, H. (2020). Linformer: Self-Attention with Linear Complexity. *arXiv.* https://doi.org/10.48550/arXiv.2006.04768.
19. Huang, X., Xue, Y., Ren, S., Wang, F. (2023). Sensor-Based Wearable Systems for Monitoring Human Motion and Posture: A Review. *Sensors, 23 (22),* 9047. https://doi.org/10.3390/s23229047
20. Bourke, A. K., Lyons, G. M. (2008). A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor. *Medical Engineering & Physics, 30 (1),* 84–90. https://doi.org/10.1016/j.medengphy.2006.12.001
21. Minh Dang, L., Min, K., Wang, H., Jalil Piran, Md., Hee Lee, C., Moon, H. (2020). Sensor-based and vision-based human activity recognition: A comprehensive survey. *Pattern Recognition, 108,* 107561. https://doi.org/10.1016/j.patcog.2020.107561
22. Kepski, M., Kwolek, B. (2014). Fall detection using ceiling-mounted 3d depth camera. *2014 International conference on computer vision theory and applications (VISAPP),* 640–647. https://doi.org/10.5220/0004742406400647
23. Lin, Z., Wang, Z., Dai, H., Xia, X. (2022). Efficient fall detection in four directions based on smart insoles and RDAE-LSTM model. *Expert Systems with Applications, 205,* 117661. https://doi.org/10.1016/j.eswa.2022.117661
24. He, J., Zhang, Q., Wang, L., Pei, L. (2019). Weakly Supervised Human Activity Recognition From Wearable Sensors by Recurrent Attention Learning. *IEEE Sensors Journal, 19 (6),* 2287–2297. https://doi.org/10.1109/jsen.2018.2885796
25. Kibet, D., So, M. S., Kang, H., Han, Y., Shin, J.-H. (2024). Sudden Fall Detection of Human Body Using Transformer Model. *Sensors, 24 (24),* 8051. https://doi.org/10.3390/s24248051
26. Ursul, I. (2024). Elderly fall detection using unsupervised transformer model. *Electronics and Information Technologies, 26.* https://doi.org/10.30970/eli.26.7
27. Haque, S. T., Debnath, M., Yasmin, A., Mahmud, T., Ngu, A. H. H. (2024). Experimental Study of Long Short-Term Memory and Transformer Models for Fall Detection on Smartwatches. *Sensors, 24 (19),* 6235. https://doi.org/10.3390/s24196235
28. Haq, I. U., Lee, B. S., Rizzo, D. M. (2024). TransNAS-TSAD: harnessing transformers for multi-objective neural architecture search in time series anomaly detection. *Neural Computing and Applications, 37 (4),* 2455–2477. https://doi.org/10.1007/s00521-024-10759-1
29. Ursul, I. (2024). Developing a High-Accuracy Fall Detection Device Using Raspberry Pi and Transformer Models. *Ivanursul.com.* Available at: https://ivanursul.com/developing-fall-detection-device-raspberry-pi
30. Ursul, I. (2025). Sensor-Based Fall Detection Dataset with 8,953 Activities from 29 Subjects. *Figshare.* https://doi.org/10.6084/m9.figshare.28287482.v1

Ivan Ursul, *PhD Student, Department of Applied Mathematics, Ivan Franko National University of Lviv, Lviv, Ukraine, e-mail: ivan.ursul@lnu.edu.ua, ORCID: https://orcid.org/0009-0002-9879-8008*