applied sciences

MDPI

Article

Robust Human Activity Recognition for Intelligent Transportation Systems Using Smartphone Sensors: A Position-Independent Approach

John Benedict Lazaro Bernardo [1,2,*], Attaphongse Taparugssanagorn [1], Hiroyuki Miyazaki [3], Bipun Man Pati [4] and Ukesh Thapa [4]

[1] Department of Information and Communication Technologies, School of Engineering and Technology, Asian Institute of Technology, 58 Moo 9, Klong Luang, Pathum Thani 12120, Thailand; attaphongset@ait.asia
[2] Department of Information Technology, College of Information Technology and Computing, University of Science and Technology of Southern Philippines, Cagayan de Oro City 9000, Philippines
[3] Center for Spatial Information Science, Tokyo University, Chiba 277-8568, Japan; heromiya@csis.u-tokyo.ac.jp
[4] AI Research Center, Advanced College of Engineering and Management, Tribhuwan University, Kalanki, Kathmandu 44614, Nepal; bipunmanpati@acem.edu.np (B.M.P.); ukesh.thapa@acem.edu.np (U.T.)
* Correspondence: jbl.bernardo@gmail.com

Abstract: This study explores Human Activity Recognition (HAR) using smartphone sensors to address the challenges posed by position-dependent datasets. We propose a position-independent system that leverages data from accelerometers, gyroscopes, linear accelerometers, and gravity sensors collected from smartphones placed either on the chest or in the left/right leg pocket. The performance of traditional machine learning algorithms (Decision Trees (DT), K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Classifier (SVC), and XGBoost) is compared against deep learning models (Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM), Temporal Convolutional Networks (TCN), and Transformer models) under two sensor configurations. Our findings highlight that the Temporal Convolutional Network (TCN) model consistently outperforms other models, particularly in the four-sensor non-overlapping configuration, achieving the highest accuracy of 97.70%. Deep learning models such as LSTM, GRU, and Transformer also demonstrate strong performance, showcasing their effectiveness in capturing temporal dependencies in HAR tasks. Traditional machine learning models, including RF and XGBoost, provide reasonable performance but do not match the accuracy of deep learning models. Additionally, incorporating data from linear accelerometers and gravity sensors led to slight improvements over using accelerometer and gyroscope data alone. This research enhances the recognition of passenger behaviors for intelligent transportation systems, contributing to more efficient congestion management and emergency response strategies.

Keywords: position independence; human activity recognition; multi-sensor data; nested cross-validation; random forest; XGBoost; gated recurrent unit; transformer; long short-term memory; temporal convolutional networks; intelligent transportation systems

check for updates

Citation: Bernardo, J.B.L.; Taparugssanagorn, A.; Miyazaki, H.; Pati, B.M.; Thapa, U. Robust Human Activity Recognition for Intelligent Transportation Systems Using Smartphone Sensors: A Position-Independent Approach. *Appl. Sci.* **2024**, *14*, 10461. https://doi.org/10.3390/app142210461

Academic Editor: Christos Bouras

Received: 28 September 2024
Revised: 1 November 2024
Accepted: 7 November 2024
Published: 13 November 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Human Activity Recognition (HAR) has become a pivotal technology across various domains, including transportation, active living, entertainment, and security. The integration of visual and non-visual sensors has significantly improved the efficiency and accuracy of activity monitoring and analysis. In traditional security operations, human operators rely on extensive camera networks, often leading to issues like operator fatigue and reduced effectiveness. These limitations have driven the development of automated vision-based systems, which offer greater reliability and scalability [1].

In the context of transportation, HAR is instrumental in enabling real-time traffic monitoring, optimizing routes, and predicting travel times. These advancements contribute

to the development of intelligent transportation systems designed to improve traffic flow, enhance passenger safety, and maximize vehicle efficiency, ultimately fostering more adaptive and efficient transportation networks.

HAR identifies a variety of human activities, from basic gestures to complex movements, using raw sensor data [2]. Earlier studies predominantly utilized vision-based approaches, including RGB cameras and Kinect sensors, for activity classification and analysis [3–5]. Although these methods provide precise activity recognition and are suitable for long-range monitoring, they raise significant privacy concerns and face challenges in large-scale implementation [6]. In contrast, motion sensors such as accelerometers and gyroscopes embedded in smartphones offer a cost-effective, mobile alternative that alleviates privacy issues associated with vision-based techniques. By integrating HAR with smartphone sensors and employing deep learning methods, the accuracy of activity recognition can be significantly improved.

Recent studies have demonstrated notable advancements in HAR using smartphone sensors. Palimote et al. achieved an impressive 99.06% accuracy in classifying activities such as walking, walking downstairs, walking upstairs, sitting, standing, and lying down by leveraging accelerometer and gyroscope data from a Samsung Galaxy S2. Their approach utilized an Artificial Neural Network (ANN), which proved highly effective in accurately classifying these activities based on the sensor data, showcasing the potential of ANN for robust human activity recognition systems [7]. Koping et al. introduced a flexible sensor-based HAR system utilizing smartphones and wearable devices, employing Support Vector Machines (SVM) for activity recognition [8]. Jahangiri and Rakha conducted a comparative analysis of machine learning techniques for transportation mode detection using smartphone sensors, emphasizing the effectiveness of Random Forest (RF) and SVM [9]. Similarly, Lopez et al. evaluated smartphone sensor data for travel behavior analysis, identifying limitations in data accuracy due to hardware constraints. These constraints include sensor precision, variations in sampling rates, and noise, which are inherent to smartphone hardware, as well as inconsistencies across different device models. These limitations can affect the quality of collected data and lead to propagated errors in the analysis, potentially skewing the aggregated results [10].

The integration of machine learning in HAR has traditionally depended on heuristic feature extraction methods, which often limit classification accuracy and robustness [10]. In contrast, deep learning approaches, such as Long Short-Term Memory (LSTM) networks, utilize raw data to construct complex models with multiple layers, effectively tackling challenges in sequence learning [11]. LSTM's capability to address vanishing gradient issues and manage information flow through controllable gates significantly improves accuracy compared to traditional methods.

Singh et al. utilized LSTM models to predict activities from smart home datasets, demonstrating superior performance compared to probabilistic models such as Naïve Bayes and Hidden Markov Models (HMM) [12]. Mekruksavanich and Jitpattanakul further advanced HAR in smart homes by applying accelerometer and gyroscope data with LSTM networks, achieving notable improvements in accuracy [13]. However, these studies largely focus on controlled environments with position-dependent datasets, which may not fully represent real-world scenarios where devices can be held in various positions.

Most HAR studies using smartphones rely on controlled, position-dependent datasets, which fail to capture the complexities of real-life scenarios where devices may be held in varying positions. This discrepancy can result in decreased accuracy in activity recognition. To address these challenges, it is crucial to utilize unconstrained, position-independent datasets [12,13]. Furthermore, many recent studies overlook the inclusion of sensors like gravity and linear accelerometers in smartphone-based HAR systems, despite their potential to enhance recognition accuracy.

In recent studies, HAR using smartphone sensors has shown promise in transportation and real-world activity tracking. However, the inherent challenges of position variability, sensor data diversity, and model robustness remain unresolved. This study addresses these

issues by focusing on realistic, position-independent HAR, combining machine learning and deep learning approaches for improved recognition across various user scenarios. The primary contributions of this study are as follows:

- Realistic, Position-Independent Data Collection: Our research investigates HAR using smartphones placed in multiple realistic positions—chest and leg pockets—reflecting common device placements in real-world settings. This approach addresses limitations associated with fixed-position data collection and improves the generalizability of activity recognition models.
- Enhanced Sensor Suite Utilization: In addition to accelerometers and gyroscopes, this study incorporates linear accelerometers and gravity sensors, providing a more comprehensive dataset that better captures dynamic activity patterns and improves classification accuracy. Comparative Analysis of Machine Learning and Deep Learning Models: This study evaluates both classical machine learning models (Decision Tree, K-Nearest Neighbors, Random Forest, Support Vector Classifier, and XGBoost) and advanced deep learning models (GRU, LSTM, TCN, and Transformer) for HAR.
- Performance Optimization through Overlapping Data Segmentation: The study compares non-overlapping and 50% overlapping data segmentation methods, identifying the advantages of overlapping windows for improving model accuracy by capturing transitional information.
- Application to Intelligent Transportation Systems (ITS): By demonstrating high-accuracy passenger behavior recognition, our research provides insights into practical ITS applications, contributing to enhanced traffic management, safety monitoring, and real-time emergency response systems.

These contributions establish a foundation for robust, real-world HAR implementations in intelligent transportation contexts, advancing the state of the art in activity recognition research.

The remainder of the paper is organized as follows: Section 2 describes the related works in HAR. In Section 3 provides the materials and methods of the study. The results of the study and its discussion part are then discussed in Section 4. Finally, Section 5 provides the conclusion and directions for future research.

2. Related Works and Our Contributions

2.1. Related Works

HAR plays a critical role in Intelligent Transportation Systems (ITS), enhancing safety and transportation efficiency. Recent studies emphasize the role of physiological, inertial, and environmental sensors—particularly smartphones and wearable devices—in advancing HAR [14]. This research focuses on HAR using multiple smartphone sensors to address both accuracy and applicability.

Sousa et al. [15] demonstrated that integrating gyroscope and accelerometer data improves classification accuracy by 2%, particularly for static activities, but limited improvement was noted for dynamic activities such as walking and running. Ferrari et al. [16] conducted a review indicating that, while accelerometers alone generally outperform gyroscopes, their combined usage improves overall accuracy by approximately 10%, highlighting the impact of sensor placement and orientation. Nweke et al. [17] emphasized that additional sensors, like gravity and linear accelerometers, could enhance HAR accuracy but noted that current datasets often fail to represent real-world variations in sensor positioning.

Advantages and Limitations: These studies highlight that while sensor fusion improves HAR, the focus on controlled and position-dependent datasets limits broader applicability. Our study addresses these limitations by using a comprehensive sensor suite—including accelerometers, gyroscopes, linear accelerometers, and gravity sensors—positioned more realistically, reflecting everyday user scenarios. Ouyang et al. [18] introduced a contrastive fusion learning approach specifically designed for multimodal HAR with limited data, which demonstrates improved adaptability across various sensor inputs. However, this approach is primarily designed for controlled environments with

limited data, lacking extensive testing on position-independent configurations, which our study aims to address by incorporating a broader range of smartphone-based sensors.

Additionally, Siargkas et al. [19] proposed a Transportation Mode Recognition (TMR) approach using both acceleration and location data, achieving an F1 score of 91%. Their model performed slightly better (92.3%) under varied smartphone placements but focused mainly on transportation modes and did not evaluate the potential advantages of other sensor types. Yuan et al. [20] reviewed human activity recognition methods using smartphone-based sensors, suggesting their effectiveness in data capture but did not explore the value of adding sensors like linear accelerometers or gravity sensors for enhanced recognition capabilities. In comparison, ShoesLoc by Yu et al. [21] utilized in-shoe force sensors for indoor walking path tracking with high accuracy in path estimation. This hardware-specific approach is effective but lacks general applicability in broader HAR scenarios, unlike smartphone-based HAR systems such as ours, which do not require specialized equipment.

Challenges and Our Approach: These findings underscore the need for HAR approaches that incorporate diverse sensors beyond basic accelerometers and gyroscopes. Our research addresses these gaps by including a combination of sensors, which we found to slightly improve activity classification across a range of realistic, uncontrolled scenarios. CrossHAR by Hong et al. [22] employed a hierarchical self-supervised pretraining strategy to enhance cross-dataset generalization in HAR, achieving strong generalization across diverse HAR datasets. While CrossHAR aims to improve adaptability across datasets, our work complements this goal by evaluating HAR in real-world, position-independent smartphone placements, providing robustness in uncontrolled environments.

Previous research largely relies on fixed sensor positions for high accuracy. For example, Rahn et al. [23] combined smartphone and IMU sensors at fixed positions, achieving high recognition rates across various activities. Similarly, Mekruksavanich and Jitpattanakul [13] used their Att-ResBiGRU model with fixed sensor placements, showing reliable generalization with low variance across multiple fixed positions in datasets such as PAMAP2 and REALWORLD16.

Summary of Limitations: While fixed sensor positions improve recognition accuracy in controlled environments, they reduce adaptability in real-world, variable settings. Our study extends this approach by investigating HAR using realistic smartphone placements, such as the chest and leg pockets, which are more applicable to real-world usage scenarios.

To address these constraints, position-independent HAR methodologies have been explored. For instance, Barua et al. [24] applied a 1D-CNN-LSTM model to manage intra- and inter-position sensor variability, achieving recognition rates between 68.66% and 73.64% across various device placements. Mekruksavanich and Jitpattanakul [13] further applied position-independent HAR using the Att-ResBiGRU model, achieving high F1 scores across multiple datasets, demonstrating robustness across diverse positions, albeit primarily in wearable devices.

Our Improvements: Our work builds upon these studies by testing HAR in flexible smartphone positions, aiming to expand position-independent HAR applications in transportation by utilizing multiple sensors to increase adaptability and recognition accuracy in varied scenarios.

Integrating both machine learning and deep learning methods has shown strong results in HAR. Kumar and Chauhan [14] highlighted deep learning's strength in handling complex HAR tasks, although traditional models are often favored for their lower computational requirements. They also noted the importance of temporal modeling for activity prediction, suggesting that techniques like LSTM can leverage these temporal dependencies for better activity forecasting.

Expanding Model Comparisons: Our research includes an expanded comparison of both traditional and deep learning techniques, now incorporating a wider range of models to meet the diverse demands of ITS applications. These models include Random Forest (RF) [25,26], a widely-used ensemble model that balances accuracy with low computational

demand, and XGBoost [27], known for its scalability and enhanced performance with large datasets. For sequential and complex patterns, we utilize LSTM and Temporal Convolutional Networks (TCN) [28], as well as Gated Recurrent Units (GRU) and Transformer models [29], both recognized for their efficiency in capturing dependencies in time-series data. This expanded approach enables us to capture a broader range of user behaviors in transportation scenarios, further advancing real-time HAR adaptability.

Addressing Model Limitations: Our research compares both machine learning and deep learning techniques for HAR, including DT, KNN, RF, SVC, XGBoost, GRU, LSTM, TCN, and Transformer, aiming to balance computational demands with improved accuracy in dynamic, real-world conditions typical of ITS.

2.2. Our Contributions

This research addresses critical gaps in HAR by focusing on smartphone positioning variability, comprehensive sensor data utilization, and the comparative performance of traditional machine learning and advanced deep learning models. Unlike previous studies that often rely on controlled environments and fixed sensor placements, our work investigates realistic smartphone positions and offers a thorough analysis of sensor data for improved HAR in dynamic, real-world scenarios. The key contributions of this study are as follows:

a. Realistic Smartphone Positions: Previous works, such as Rahn et al. [23] and Mekruksavanich and Jitpattanakul [13], achieved high recognition performance by using fixed sensor positions. However, these approaches fail to address the variability in smartphone placements that naturally occurs in real-world settings. Our research bridges this gap by analyzing HAR in realistic smartphone positions—chest (when in use) and leg pockets (when not in use)—offering a more robust solution to position orientation dependency. This exploration significantly enhances recognition performance in diverse, uncontrolled environments, making HAR systems more applicable to everyday use.

b. Comprehensive Sensor Data Utilization: While prior studies like Sousa et al. [15] and Ferrari et al. [16] have leveraged data from accelerometers and gyroscopes, they tend to neglect other key sensors such as the linear accelerometer and gravity sensor. Our research incorporates data from these additional sensors and compares it to the more common accelerometer-gyroscope data. Although the inclusion of these sensors yields only a marginal improvement in classification accuracy, it underscores the value of a comprehensive sensor suite, especially in scenarios requiring high precision, such as emergency detection and critical passenger behavior monitoring.

c. Comparison of Machine Learning and Deep Learning Models: Although prior research [12,13] has shown that deep learning models often outperform traditional machine learning methods in HAR tasks, comparisons in real-world, position-independent settings remain limited. Our study addresses this gap by conducting a comprehensive comparative analysis of these approaches using position-independent data. We assess a wide range of models, encompassing traditional machine learning algorithms such as DT, KNN, RF, SVC, and XGBoost, alongside advanced deep learning architectures including GRU, LSTM, TCN, and Transformers. This extensive evaluation allows us to balance considerations of computational efficiency and accuracy. The results indicate that deep learning models, with their ability to capture complex temporal dependencies, are particularly effective in dynamic, real-world ITS conditions.

d. Enhancement of Passenger Behavior Recognition: Existing research, including work by Kumar and Chauhan [14], often focuses on fixed environments and sensor placements, limiting the practical applicability of these findings in dynamic real-world scenarios. Our study advances passenger behavior recognition for intelligent transportation systems by tackling the challenges of real-world variability in smartphone positions. This contributes to improved traffic congestion management, enhanced emer-

gency response capabilities, and optimized resource utilization within public transportation networks, making HAR systems more adaptable to real-world applications.

In summary, our research offers a novel approach to Human Activity Recognition by investigating the impact of realistic smartphone positions, integrating a more comprehensive set of sensors, and conducting an in-depth comparison of machine learning and deep learning techniques. These contributions significantly advance the state of the art in HAR, providing practical solutions for real-world implementation and addressing limitations observed in prior studies.

3. Materials and Methods

This study was designed to systematically evaluate and compare the performance of various machine learning and deep learning models for HAR using sensor data from smartphones. The research specifically focused on two key factors: the influence of different data segmentation strategies—comparing non-overlapping windows with 50% overlapping windows—and the impact of sensor configurations, contrasting the use of two sensors (accelerometer and gyroscope) against four sensors (including linear accelerometer and gravity sensors). Through this comprehensive evaluation, the study aims to provide actionable insights into the optimal methodologies for accurate and efficient activity recognition in real-world scenarios, addressing both model performance and practical deployment considerations.

3.1. Data Collection

This research utilized the embedded sensors of a Samsung Galaxy A50 Android smartphone (Samsung Electronics, Suwon, South Korea)—specifically the accelerometer, gyroscope, linear accelerometer, and gravity sensor—to collect uncontrolled data from various human activities, including walking, running, and being stationary, both while using and not using the smartphone. The addition of the linear accelerometer and gravity sensor was investigated to assess whether these sensors significantly enhance classification accuracy compared to using only the accelerometer and gyroscope. The linear accelerometer and gravity sensor are derivatives of the accelerometer, providing additional information about the magnitude and direction of acceleration and gravity, respectively.

A Graphical User Interface (GUI)-based application was employed to facilitate data collection, recording each volunteer's unique identifier and the time duration for each activity (start and stop times). Data from all activities were collected at a sampling rate of 50 Hertz (Hz), capturing measurements across the X, Y, and Z axes for each of the four sensors (accelerometer, gyroscope, linear accelerometer, and gravity sensor). The gyroscope, initially collecting data at 120 Hz, was normalized to 50 Hz to ensure consistency with the other sensors.

The study involved eleven (11) volunteers, aged between 20 and 40 years, who were asked to perform a series of basic activities while holding the smartphone in different positions. These activities were categorized into two scenarios:

Scenario 1: Activities with the smartphone placed in the left or right side pockets (not in use).

Scenario 2: Activities with the smartphone held in the chest position (in use).

Each volunteer performed a sequence of activities over a period of 7 min. These activities were divided into six categories: walking, running, and stationary (sitting and standing), with the conditions of both using the smartphone and not using the smartphone in a bus stop station setting. Table 1 outlines the specific activities conducted in this study.

Table 1. Details of Experimental Activities: With and Without Smartphone Interaction.

No.	Abbreviation	Activities	Description
1	run_wos	Running without using a smartphone	(Running) [1] around the bus stop while not using the smartphones
2	stn_wos	Stationary without using a smartphone	(Standing or Sitting) around the bus stop while not using the smartphone
3	wlk_wos	Walking without using a smartphone	(Walking at 2–3 steps per seconds) around the bus stop while not using the smartphone
4	run_ws	Running using the smartphone	(Running) [1] around the bus stop while using the smartphone
5	stn_ws	Stationary using the smartphone	(Standing or Sitting) around the bus stop while using the smartphone
6	wlk_ws	Walking using the smartphone	(Walking at 2–3 steps per seconds) around the bus stop while using the smartphone

[1] Running speed is assumed to be higher than walking speed [30].

From Table 1, the walking speed in this study is standardized to two steps per second for each window sample. This rate is based on the typical gait cycle of an average, healthy individual who takes approximately two steps per second. Additionally, running is assumed to involve a higher speed compared to walking. For activities such as standing, sitting, or any other stationary actions, these were categorized as stationary actions due to the lack of movement.

The study evaluates the impact of different sensor configurations on HAR accuracy by analyzing two distinct setups:

Two-Sensor Configuration: Utilized only the accelerometer and gyroscope to establish a baseline for the effectiveness of these primary sensors in activity recognition.

Four-Sensor Configuration: Incorporated all four sensors—accelerometer, gyroscope, linear acceleration sensor, and gravity sensor—to assess whether the inclusion of additional sensor data enhances the model's activity recognition capabilities.

Table 2 presents a comparative overview of key smartphone sensors utilized in HAR. Each sensor type has distinct capabilities and limitations that affect its effectiveness in activity tracking. The sensors collectively capture detailed motion data, with accelerometers and gyroscopes being widely used in HAR for their ability to measure linear and angular motion. The addition of linear accelerometers and gravity sensors provides supplementary context, enhancing the system's accuracy by minimizing noise from changes in acceleration. This integration of multiple sensors contributes to more robust and precise activity recognition.

Table 2. Characteristics of Smartphone Sensors and Their Role in Activity Recognition.

Sensor	Measurement Type	Gravitational Effect
Accelerometer	Total acceleration (motion + gravity)	Included
Gyroscope	Rotation or angular velocity around x, y, z axes	N/A (measures rotation)
Linear Accelerometer	Acceleration due to movement (gravity removed)	Excluded
Gravity Sensor	Gravitational acceleration only	Only measures gravity

3.2. Data Processing and Sensor Configuration

The raw sensor data were preprocessed to enhance data quality and ensure consistency across measurements. This preprocessing involved noise reduction and normalization to

standardize the data before further analysis. To evaluate the impact of data segmentation strategies on model performance, we divided the data into overlapping windows of 2 s with 50% overlap and non-overlapping windows. This approach allowed for the capture of continuous sequences, which are essential for training deep learning models and provided insights into how different windowing techniques affect performance.

To ensure comprehensive analysis, the data were segmented into fixed windows of 2.56 s (128 data points per window). Each window was processed with two different approaches: one set with 50% overlap and another set without overlap. This segmentation strategy aimed to capture temporal patterns effectively and provide a robust comparison of preprocessing methods.

Following segmentation, we applied handcrafted feature extraction to each window. This approach allowed us to capture the most relevant aspects of the sensor signals, ensuring that the models could effectively differentiate between various human activities. We extracted a variety of time-domain features commonly used in HAR. The feature set for the two-sensor configuration included 42 features, while the four-sensor configuration yielded 84 features. These features were chosen based on their relevance and effectiveness in distinguishing different activities.

To provide a clear representation of the sensor measurements during a running activity, we visualize the collected data for each sensor between 80 and 85 s of the recording. This segment was selected because it captures a momentum phase of the running activity, where consistent sensor readings are observed. Figures 1–4 depict the data from different sensors during this window.

The plots provide insight into the raw sensor data for the running activity. For instance, in Figure 1, the accelerometer data shows the rapid fluctuations in acceleration along the x, y, and z axes as the volunteer propels forward during running. Figure 2 presents the gyroscope data, reflecting angular changes during body movement, while Figure 3 displays the linear acceleration measured, which isolates the acceleration component excluding gravitational effects. Figure 4 shows the gravitational component of acceleration, highlighting the orientation of the smartphone during the activity. These visualizations serve as the basis for the feature extraction process discussed in the subsequent section.

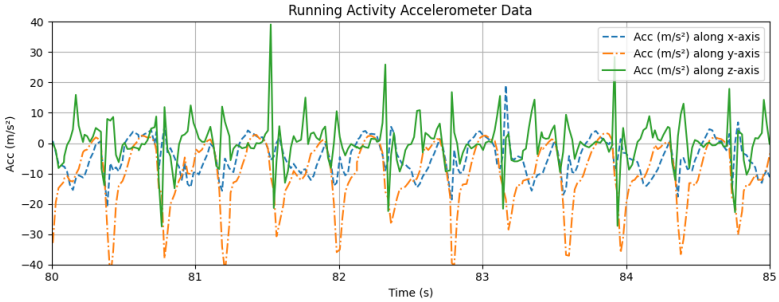


Figure 1. Running Activity Accelerometer Data: acceleration values along the x, y, and z axes recorded between 80 and 85 s.

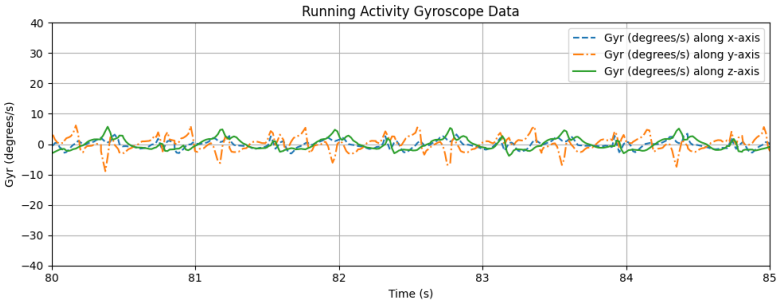


Figure 2. Running Activity Gyroscope Data: angular velocity along the x, y, and z axes recorded between 80 and 85 s.

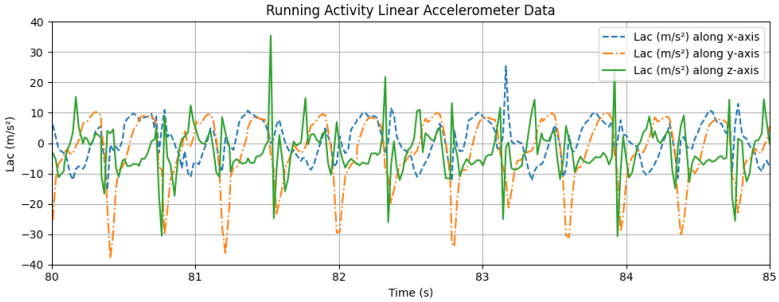


Figure 3. Running Activity Linear Accelerometer Data: linear acceleration values along the x, y, and z axes recorded between 80 and 85 s.

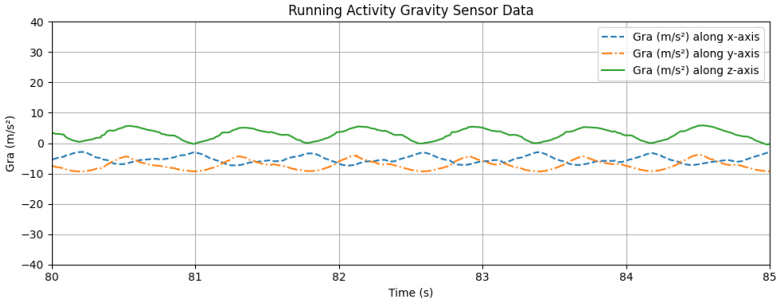


Figure 4. Running Activity Gravity Sensor Data: gravitational acceleration values along the x, y, and z axes recorded between 80 and 85 s.

Statistical Feature Analysis for Activity Recognition

Table 3 summarizes the statistical analysis of the feature vectors computed from each sliding window. The process began with raw data collection, followed by segmentation into windows corresponding to each activity. For each window, signal features were extracted and processed using statistical methods. This comprehensive feature extraction approach provides a detailed basis for the subsequent classification analysis and allows for a robust evaluation of model performance under varying sensor configurations.

In this study, several key time-domain features were extracted to capture the essential characteristics of the sensor signals over time. The **mean** (μ) provides the average value of the signal in each window, highlighting the overall level of activity:

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i, \tag{1}$$

where n represents the number of data points in the window, and x_i denotes each individual sensor reading.

Table 3. Descriptive Statistical Features for Human Activity Recognition.

No.	Statistical Feature	Description
1.	Mean	The average value of the signal over the window.
2.	Standard deviation	The standard deviation value over the window.
3.	Median absolute value	The median value over the window.
4.	Skewness	Describes the distribution's asymmetry.
5.	Maximum	The maximum value of the signal over the window.
6.	Minimum	The minimum value of the signal over the window.
7.	Kurtosis	Describes the degree of peakedness of the data.

The **standard deviation** (σ) quantifies the amount of variation in the signal, reflecting how much the values fluctuate around the mean:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2}, \tag{2}$$

Skewness assesses the asymmetry of the signal distribution, indicating whether the values tend to be higher or lower than the mean. This is particularly useful for understanding the balance of activity levels:

$$\text{Skewness} = \frac{n}{(n-1)(n-2)}\sum_{i=1}^{n}\left(\frac{x_i - \mu}{\sigma}\right)^3, \tag{3}$$

Additionally, the **maximum** ($\max(x)$) and **minimum** ($\min(x)$) values capture the range of the data, helping to identify the peak and lowest activity levels:

$$\max(x) = \max\{x_1, x_2, \ldots, x_n\}, \tag{4}$$

$$\min(x) = \min\{x_1, x_2, \ldots, x_n\}. \tag{5}$$

Lastly, **kurtosis** measures the "tailedness" or sharpness of the signal distribution. This feature helps to identify the presence of outliers or extreme values that may affect activity classification:

$$\text{Kurtosis} = \left(\frac{n(n+1)}{(n-1)(n-2)(n-3)}\sum_{i=1}^{n}\left(\frac{x_i - \mu}{\sigma}\right)^4\right) - \frac{3(n-1)^2}{(n-2)(n-3)}. \tag{6}$$

Together, these extracted features provide a rich representation of the sensor data and form the basis for further analysis in human activity recognition, allowing for robust classification models to be built based on the variability and distribution of the signal data [31].

3.3. Experimental Setup

This section outlines the experimental setup used to assess the performance of various machine learning and deep learning models for human activity recognition tasks utilizing smartphone sensor data. The hyperparameters for each model were carefully tuned to achieve optimal performance. Table 4 provides a detailed summary of the hyperparameter configurations explored for each model, enabling a systematic comparison across diverse machine learning and deep learning approaches.

The feature set was standardized using a Standard Scaler to ensure that each feature has a mean of 0 and a standard deviation of 1, transforming each feature x as follows:

$$z = \frac{x - \mu}{\sigma}, \tag{7}$$

where z is the standardized value, μ is the mean of the feature, and σ is the standard deviation. Standardization was particularly crucial for models like SVC, KNN, and neural networks (e.g., GRU and Transformer), which are sensitive to the scale of input data. Standardizing the features improves the convergence of gradient-based optimization algorithms and ensures that all features contribute equally to distance calculations in KNN and margin maximization in SVC.

Nested Cross-Validation (CV) was employed to assess the generalizability of the models to unseen data. The outer loop used five folds, while the inner loop used three folds. The input tensor shape was defined as (window size, number of features): specifically, (2,3) for the three-input configurations and (2,4) for the four-input configurations. Keras' K-Fold was used for dataset splitting, with a random state of 42 to ensure reproducibility. The dataset was split into 80% for training and 20% for testing, ensuring adequate data for

model evaluation. This approach allowed each model to train and test multiple times on different folds, maximizing the utility of the available data while avoiding overfitting.

This two-level cross-validation technique provided an unbiased estimate of model performance and minimized overfitting during hyperparameter tuning. The outer loop ensured generalization to unseen data, while the inner loop rigorously selected the best hyperparameters based on the training data. This consistent setup was applied across all models, from traditional classifiers like DT, KNN, RF, and SVC to advanced gradient-boosted models like XGBoost, and further to deep learning architectures such as GRU, LSTM, TCN, and Transformer, providing a robust evaluation across the full spectrum of models used in this study.

Table 4 illustrates the range of hyperparameters adjusted for each model to achieve optimal performance. For DT, hyperparameters such as the splitting criterion (gini or entropy), maximum depth (None, 10, 20, 30), and minimum samples per split and leaf were fine-tuned to balance model complexity and prevent overfitting.

The KNN model's parameters, including the number of neighbors and weight functions, were adjusted to enhance sensitivity to local data variations. Support Vector Classifier (SVC) was optimized by experimenting with different kernels, regularization constants, and gamma values to refine the decision boundary and handle non-linear relationships effectively.

RF was optimized with hyperparameters including the number of estimators, maximum depth, and maximum features to balance performance and computational efficiency. RF's ensemble structure and its ability to handle complex interactions in sensor data make it a robust model choice for activity recognition tasks.

XGBoost was tuned with parameters such as learning rate, maximum depth, number of estimators, and gamma. The model's gradient-boosting framework allows it to capture intricate patterns in the data, while careful tuning of these parameters ensures that it achieves high performance without overfitting.

LSTM networks required the tuning of several parameters, including the number of LSTM layers, units per layer, dropout rates, optimizers (Adam, Adamax, RMSProp, SGD), and learning rates. These adjustments were crucial for managing memory, preventing overfitting, and improving the model's ability to capture temporal dependencies in the data.

Similarly, TCN were optimized by varying the number of layers, filters, kernel sizes, and learning rates to effectively capture temporal patterns and ensure robust training dynamics.

GRU models required adjustments in the number of GRU layers, units, dropout rates, optimizers, and learning rates. The GRU's simpler structure compared to LSTM networks makes it computationally efficient while still retaining the ability to capture essential temporal dependencies in sequential data.

The Transformer model was fine-tuned by varying the number of layers, hidden units, attention heads, dropout rates, and learning rates. The Transformer's self-attention mechanism allows it to focus on relevant parts of the input sequence, making it effective in handling long-range dependencies and interactions within the activity data.

Each model's parameters were chosen through nested cross-validation to ensure they were well-suited for the data and provided generalizable performance across different configurations. The inclusion of a range of traditional and deep learning models, including DT, KNN, RF, SVC, XGBoost, GRU, LSTM, TCN, and Transformer, provided a comprehensive exploration of diverse techniques, enhancing the robustness of our activity recognition framework.

These hyperparameter optimizations were evaluated using a five-fold outer cross-validation and three-fold inner cross-validation approach, as shown in Table 5. This ensured an unbiased estimate of model performance while reducing the risk of overfitting.

Table 4. Hyperparameters Used to Tune Nine Different Models for Optimal Performance.

Model	Hyperparameters	Values
DT	criterion	[gini, entropy]
	max_depth	[None, 10, 20, 30]
	min_samples_split	[2, 5, 10]
	min_samples_leaf	[1, 2, 4]
KNN	n_neighbors	[1, 3, 5, 7, 9]
	weights	[uniform, distance]
	p	[1, 2]
RF	n_estimators	[50, 100]
	criterion	[gini, entropy]
	max_depth	[None, 4, 5]
	min_samples_split	[2, 5]
	min_samples_leaf	[1, 2]
SVC	C	[0.1, 1, 10]
	kernel	[linear, rbf]
	gamma	[scale, auto]
	class_weight	[None, balanced]
XGBoost	learning_rate	[0.01, 0.1]
	n_estimators	[50, 100]
	max_depth	[3, 6]
	min_child_weight	[1, 3, 5]
	gamma	[0, 0.1]
	reg_alpha	[0, 0.1, 1]
	reg_lambda	[0, 1, 10]
GRU	GRU Layers	[1, 2, 3]
	GRU Units	[32, 64, 128]
	Dropout Rate	[0.2, 0.3, . . . , 0.5]
	Optimizer	[Adam, Adamax, RMSProp, SGD]
	Learning Rate	[0.0001, 0.001, . . . , 0.1]
LSTM	LSTM Layers	[1, 2, 3]
	LSTM Units	[32, 64, 128]
	Dropout Rate	[0.2, 0.3, . . . , 0.5]
	Optimizer	[Adam, Adamax, RMSProp, SGD]
	Learning Rate	[0.0001, 0.001, . . . , 0.1]
TCN	TCN Layers	[1, 2, 3]
	Number of Filters	[32, 64, . . . , 128]
	Kernel Size	[2, 3, 4]
	Optimizer	[Adam, Adamax, RMSProp, SGD]
	Learning Rate	[0.0001, 0.001, . . . , 0.1]
Transformer	Transformer Blocks	[2, 4, 6, 8]
	Head Size	[8, 16, . . . , 256]
	Number of Heads	[2, 4, . . . , 16]
	FF Dim	[4, 8, . . . , 64]
	Dropout Rate	[0.1, 0.2, . . . , 0.6]
	Number of MLP Layers	[1, 2, 3]
	MLP Units	[32, 64, . . . , 256]
	MLP Dropout	[0.1, 0.2, . . . , 0.6]
	Optimizer	[Adam, Adamax, RMSProp, SGD]
	Learning Rate	[0.0001, 0.001, . . . , 0.1]

Table 5. Simulation Parameters for Machine Learning and Deep Learning Model Evaluation.

No.	Simulation Parameters	Value
1	Number of frequency for data collection	50 Hz
2	Total number of samples for each volunteer in one activity	19,200 samples
3	Total number of samples for one window	128 samples
4	Number of windows for non-overlapping case in one activity	150 windows
5	Number of windows for 50% overlapping case in one activity	300 windows
6	Window size	2.56 s
7	Total time to collect the data per activity	7 min
8	Number of nested cross-validation (outer, inner)	5, 3
9	Number of features (non-overlapping, 50%-overlapping)	42, 84

3.4. Performance Metrics and Evaluation of the Trained Model

In evaluating the efficacy of the developed models, several performance metrics are employed, each providing unique insights into the model's capability to accurately classify human activities.

Accuracy is the primary metric to evaluate the overall correctness of predictions made by the model. Accuracy represents the proportion of correctly classified instances (both positive and negative) to the total instances. This metric provides a broad indication of the model's reliability and effectiveness across all activity categories. The mathematical expression for accuracy is given by

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \tag{8}$$

Precision is another critical metric in this study, particularly when differentiating between similar activities. It indicates the proportion of correctly predicted positive observations to the total predicted positives. High precision is crucial in minimizing false positives, which could otherwise compromise the system's utility in real-world applications. The mathematical expression for precision is given by

$$Precision = \frac{TP}{TP + FP}. \tag{9}$$

Recall (Sensitivity), another critical metric, measures the model's ability to identify all relevant instances. This is especially vital for activities that are subtly distinct, such as differentiating between sitting and standing. A higher recall rate ensures that fewer actual events of interest are missed, maintaining the system's reliability across diverse scenarios. The mathematical expression for recall is given by

$$Recall = \frac{TP}{TP + FN}. \tag{10}$$

The F1 Score is used to find the balance between precision and recall. An optimal F1 Score is indicative of the robustness of the model against the potentially imbalanced nature of the dataset, ensuring consistent performance in identifying correct activities while reducing false identifications. The mathematical expression for the F1 Score is given by

$$F1\ Score = 2\left(\frac{Precision \cdot Recall}{Precision + Recall}\right). \tag{11}$$

Lastly, a confusion matrix offers a comprehensive visualization of the model's performance across all activity categories. It provides clarity on how accurately each category is predicted by displaying TP, FP, FN, and TN (Table 6). This matrix is instrumental in evaluating the model's ability to differentiate between similar activities, such as walking and running. For example, the matrix can reveal how well the model distinguishes between

these activities and help in assessing the impact of potential adjustments, such as increasing the number of sensors, on the overall performance of the system.

Table 6. Confusion Matrix for Binary Classification.

Actual \ Predicted	Positive	Negative
Positive	TP	FN
Negative	FP	TN

Collectively, these metrics facilitate a thorough evaluation of the model's performance and are essential for guiding the iterative refinement of the activity recognition system.

3.5. Methodology

In Figure 5, the research simulation integrates both traditional machine learning techniques and advanced deep learning architectures, which are elaborated upon in the following subsections.

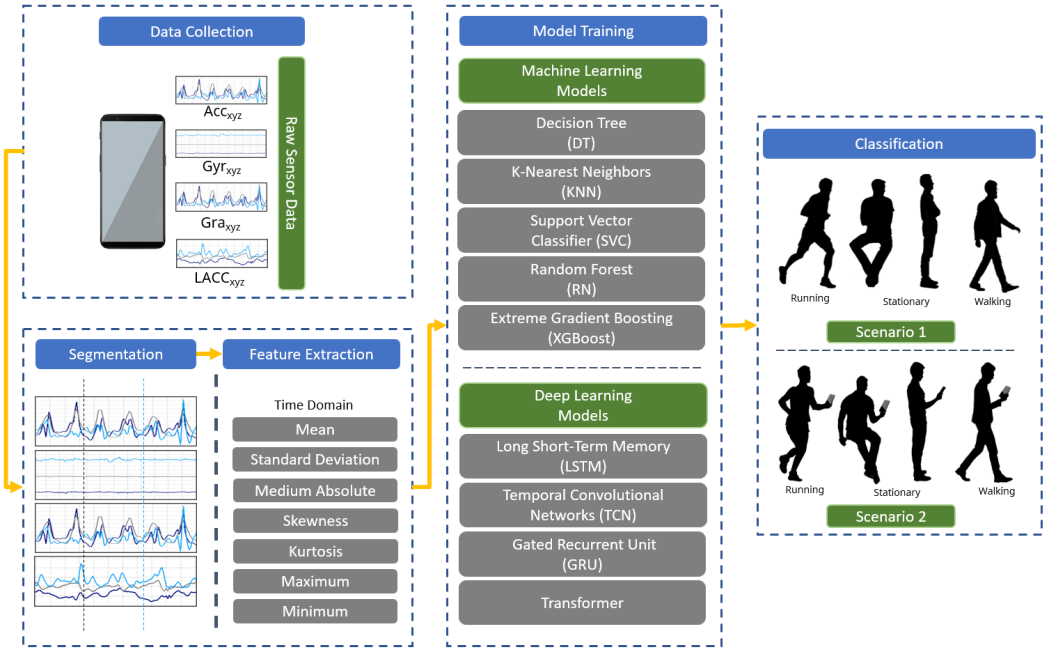


Figure 5. Methodological framework for assessing machine learning and deep learning techniques.

Figure 5 outlines the comprehensive workflow for human activity recognition using smartphone sensors, presenting the complete process from data acquisition to activity classification. Initially, raw data are collected from a variety of built-in sensors, including accelerometers, gyroscopes, linear accelerometers, and gravity sensors. These raw data undergo a crucial segmentation step to prepare them for detailed analysis, allowing for a more precise examination of different activity patterns.

Key statistical features—such as mean, standard deviation, median absolute deviation, skewness, maximum, minimum, and kurtosis—are extracted from the segmented data. These features are critical as they capture important characteristics of the signal's distribution and variability. By analyzing these features, the system can better differentiate between various physical activities, such as running, walking, or stationary positions. The extraction of these features enables the capture of subtle differences in movement patterns and ensures that the activity classification is based on robust and informative data metrics.

The extracted features are then used as inputs for a range of machine learning models. This study incorporates traditional classifiers, including DT, KNN, and SVC, as well as ensemble models such as RF and XGBoost. These models were selected for their interpretability and effectiveness in structured data classification.

Additionally, advanced deep learning architectures, including LSTM networks, TCN, GRU, and Transformers, are integrated to capture sequential dependencies and complex temporal patterns in sensor data. The combination of traditional and deep learning models allows for a thorough exploration of both conventional and contemporary approaches to activity recognition.

The models are trained to classify activities into distinct categories, including running, stationary, and walking. Their performance is evaluated across two different scenarios, providing valuable insights into their ability to generalize and perform effectively in varied contexts. This dual approach—integrating traditional machine learning techniques with cutting-edge deep learning methods—significantly improves the accuracy and reliability of the human activity recognition system. By harnessing the strengths of both methodologies, the research strives to achieve superior performance in differentiating between various physical activities, thereby advancing the capabilities of activity recognition technologies.

3.6. Model Development and Hyperparameter Optimization

In this study, we employed a comprehensive suite of machine learning and deep learning models to tackle human activity recognition, ranging from traditional algorithms such as DT, KNN, RF, SVC, and XGBoost to advanced deep learning architectures including GRU, LSTM, TCN, and Transformer.

To maximize model performance, we implemented a nested cross-validation approach. This technique involved a five-fold outer cross-validation loop for unbiased model evaluation and a three-fold inner loop dedicated to fine-tuning hyperparameters. This nested structure was critical for producing robust estimates of model performance, minimizing the risk of overfitting, and ensuring the generalizability of each model [32].

For the machine learning models, we employed grid search within the inner loop of cross-validation to systematically explore combinations of hyperparameters. Key hyperparameters optimized included the depth and split criteria for DT, the number of neighbors for KNN, the number of estimators and maximum features for RF, the regularization parameter C and kernel type for SVC, and learning rates and boosting parameters for XGBoost.

For the deep learning models, we leveraged a Hyperband tuner to explore a fine-grained range of hyperparameters, including learning rates, number of layers, number of units per layer, and dropout rates. This method allowed us to optimize complex configurations efficiently, selecting the best settings to achieve peak model performance. The Hyperband tuner's ability to dynamically allocate resources based on model performance enabled an adaptive search that significantly improved convergence times.

By rigorously optimizing each model's hyperparameters through these tailored approaches, we ensured that all models were evaluated in their most effective configurations, enhancing the reliability of our comparative analysis and the robustness of our conclusions.

3.7. Models Overview

3.7.1. Decision Tree

The Decision Tree (DT) model, a widely used non-parametric supervised learning algorithm [33], was employed in this study to classify human activities using data from multiple sensors. The core strength of DT lies in its ability to recursively partition the feature space based on decision rules, ensuring interpretable classification. However, recognizing diverse human activities from multi-sensor data presents challenges, requiring careful parameter tuning to balance model complexity and prevent overfitting.

In this study, we optimized the DT using the Gini impurity criterion to guide node splits, aiming to maximize the purity of the resulting partitions. The formula for Gini impurity is as follows:

$$Gini = 1 - \sum_{i=1}^{n} p_i^2, \tag{12}$$

where p_i is the probability of a data point belonging to class i.

Hyperparameters such as the criterion for splitting, maximum depth of the tree, minimum samples per leaf, and minimum samples required to split an internal node were optimized using the nested cross-validation and Keras Tuners as described in Section 3.6. The optimal hyperparameters for each sensor configuration are presented in Table 7.

Table 7. Optimal Hyperparameters for Decision Tree in Different Sensor Configurations.

Sensor Configuration	Criterion	Max Depth	Min Samples Leaf	Min Samples Split
2 Sensors—Non-overlapping	Entropy	5	2	2
2 Sensors—50% overlapping	Entropy	5	2	2
4 Sensors—Non-overlapping	Entropy	5	1	2
4 Sensors—50% overlapping	Gini	5	2	2

In the non-overlapping configurations, the *entropy* criterion was preferred, with a maximum depth of 5 to prevent overfitting and ensure robust pattern recognition. For 50% overlapping data, the minimum samples per split were increased to better handle the additional data points introduced by overlapping windows, addressing potential overfitting.

The switch from *entropy* to *Gini impurity* for the four-sensor 50%-overlapping setup reflects the need to handle more complex feature interactions. This adjustment allowed for better computational efficiency while maintaining high classification accuracy [26].

In summary, tuning the DT hyperparameters ensured the model could generalize effectively across sensor setups and data segmentation strategies, offering accurate classification without unnecessary complexity. The DT model proved to be a reliable classifier for human activity recognition, especially when optimized for different sensor configurations [33].

3.7.2. K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) [34] is a simple, yet effective, instance-based learning algorithm. It classifies data points based on the majority vote of their k-nearest neighbors, utilizing distance metrics like Minkowski distance. In this study, KNN was applied to human activity recognition using both non-overlapping and 50% overlapping data configurations.

The Minkowski distance, which generalizes both Euclidean and Manhattan distance, is defined as:

$$d(x_i, x_j) = \left(\sum_{k=1}^{n} |x_{ik} - x_{jk}|^p \right)^{1/p}, \tag{13}$$

where $p = 1$ for Manhattan distance and $p = 2$ for Euclidean distance.

To ensure optimal performance, we used nested cross-validation, which not only tunes hyperparameters but also validates the model, thus providing a more robust evaluation. The optimal hyperparameters across both two-sensor and four-sensor configurations (non-overlapping and overlapping) remained consistent, as shown in Table 8. Specifically, a Minkowski distance with $p = 1$ (Manhattan distance), $k = 3$ neighbors, and uniform weights yielded the best results.

Table 8. Optimal Hyperparameters for KNN in Different Sensor Configurations.

Sensor Configuration	Metric	n_neighbors	p (Minkowski)	Weights
2 Sensors—Non-overlapping	Minkowski	3	1	Uniform
2 Sensors—50% overlapping	Minkowski	3	1	Uniform
4 Sensors—Non-overlapping	Minkowski	3	1	Uniform
4 Sensors—50% overlapping	Minkowski	3	1	Uniform

KNN's simplicity in design—using local proximity to make decisions—makes it highly interpretable. Despite this, the nested cross-validation process confirmed that the

$k = 3$ setup with Manhattan distance strikes a balance between simplicity and accuracy, performing consistently well across different sensor configurations. While KNN is not as complex as deep learning models, it provides competitive results when applied in well-structured sensor setups. Its effectiveness highlights that even traditional methods can serve as reliable benchmarks in human activity recognition tasks [35].

3.7.3. Random Forest (RF)

Random Forest (RF), an ensemble learning technique developed by Breiman [26], is used to enhance classification accuracy by constructing multiple decision trees and aggregating their results. RF improves robustness against overfitting and is particularly suitable for handling high-dimensional, multi-sensor data in human activity recognition. Each tree in the forest is trained on a subset of the dataset, with the final prediction determined by a majority vote among the trees.

In RF, each decision tree classifier is denoted as $h(x, \Theta_k)$, where x represents the input feature vector and Θ_k is a random vector indicating the subset of features used for the k-th tree. The final prediction $\hat{y}$ for a classification task is given by the following:

$$\hat{y} = \text{majority_vote}\{h(x, \Theta_k)\}_{k=1}^{K}, \tag{14}$$

where K is the total number of trees in the forest. This majority voting strategy allows RF to reduce the variance in individual decision trees and improve overall classification accuracy.

Hyperparameters for RF were optimized through a nested cross-validation approach (outer = 5 folds, inner = 3 folds), ensuring robust evaluation and tuning across different configurations. The use of Keras optimizers facilitated efficient hyperparameter selection. Table 9 presents the final parameters.

Table 9. Optimal Hyperparameters for Random Forest in Different Sensor Configurations.

Sensor Configuration	Criterion	Max Depth	Min Samples Leaf	Min Samples Split	n_estimators
2 Sensors—Non-overlapping	Gini	None	2	5	100
2 Sensors—50% overlapping	Gini	5	1	5	100
4 Sensors—Non-overlapping	Entropy	5	1	2	100
4 Sensors—50% overlapping	Gini	15	2	2	50

RF's ensemble nature makes it robust to noise and variability in the sensor data, contributing to improved classification accuracy across both overlapping and non-overlapping datasets.

3.7.4. Support Vector Classifier (SVC)

Support Vector Classifier (SVC), introduced by Vapnik [36], is a powerful supervised learning method designed to find the optimal hyperplane that separates data points into distinct classes. In this research, SVC was applied to classify human activities using sensor data from smartphones, with a focus on optimizing its performance through nested cross-validation.

To effectively capture the non-linear relationships in the activity data, the Radial Basis Function (RBF) kernel was employed. The RBF kernel transforms input data into a higher-dimensional space, enabling the model to separate complex patterns in human movements. The kernel is controlled by the parameter σ, which defines its width, and was optimized through grid search.

The RBF kernel is mathematically expressed as:

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \tag{15}$$

where x_i and x_j are the input feature vectors, and σ controls the kernel's width, influencing the decision boundary.

The model's performance is highly sensitive to its regularization parameter C and kernel width σ, both of which were fine-tuned for each sensor configuration (non-overlapping and 50% overlapping). For two-sensor setups, the RBF kernel with $C = 10$ and automatic γ yielded the best results, while the four-sensor non-overlapping configuration favored a linear kernel for its ability to handle simpler decision boundaries. The 50%-overlapping configuration returned to the RBF kernel to manage more complex, overlapping data relationships.Table 10 provides a summary of the optimal hyperparameters identified for each configuration.

Table 10. Optimal Hyperparameters for SVC in Different Sensor Configurations.

Sensor Configuration	C	Class Weight	Gamma	Kernel
2 Sensors—Non-overlapping	10	None	Auto	RBF
2 Sensors—50% overlapping	10	None	Auto	RBF
4 Sensors—Non-overlapping	10	None	Scale	Linear
4 Sensors—50% overlapping	10	None	Scale	RBF

The use of nested cross-validation ensured a reliable evaluation of model performance by optimizing hyperparameters. The RBF kernel, particularly, demonstrated strength in capturing non-linearities within overlapping data, while the linear kernel proved efficient for non-overlapping configurations. These findings highlight the flexibility of SVC in handling various sensor setups and classification tasks.

3.7.5. XGBoost

XGBoost (Extreme Gradient Boosting) [27] is an advanced gradient-boosting technique known for its computational speed, scalability, and effectiveness in handling large and complex datasets. Unlike traditional boosting algorithms, XGBoost incorporates second-order derivatives (i.e., the Hessian) in the objective function, allowing for more accurate gradient-based optimization. This feature makes XGBoost particularly suitable for applications involving intricate sensor data, such as human activity recognition.

In each boosting iteration, XGBoost adds a new tree $f_m(x)$ that minimizes an objective function, defined as:

$$\text{Obj}^{(m)} = \sum_{i=1}^{n} L(y_i, \hat{y}_i^{(m-1)} + f_m(x_i)) + \Omega(f_m), \tag{16}$$

where L represents the loss function quantifying the difference between the true label y_i and the prediction $\hat{y}_i^{(m-1)}$ from the previous iteration. The term $\Omega(f_m)$ serves as a regularization component to penalize the complexity of f_m, thereby mitigating overfitting. This regularization, alongside the use of second-order gradients, is a distinctive feature that enhances XGBoost's robustness in complex classification tasks.

The hyperparameters for XGBoost were optimized using a nested cross-validation approach, with five outer folds and three inner folds, along with Keras optimizers to facilitate hyperparameter tuning. The optimized parameters for each sensor configuration are detailed in Table 11.

Table 11. Optimal Hyperparameters for XGBoost in Different Sensor Configurations.

Sensor Configuration	Gamma	Learning rate	Max Depth	n_estimators	reg_alpha	reg_lambda
2 Sensors—Non-overlapping	0.1	0.1	3	50	0.1	0
2 Sensors—50% overlapping	0.2	0.1	3	50	0	0
4 Sensors—Non-overlapping	0.2	0.1	3	50	1	0
4 Sensors—50% overlapping	0.1	0.1	3	50	0.1	0

Using nested cross-validation with Keras optimizers allowed for precise tuning of XGBoost, capturing complex interactions within multi-sensor data and yielding high accuracy across various sensor configurations. This approach ensured that the model was generalized effectively, enhancing its performance in real-world human activity recognition applications.

3.7.6. Gated Recurrent Unit (GRU)

Gated Recurrent Units (GRU) [37] are a variant of recurrent neural networks that aim to improve upon traditional LSTMs by simplifying the architecture while maintaining the capability to capture sequential dependencies. GRUs are particularly effective for HAR tasks where capturing long-term dependencies is crucial, yet computational efficiency is also a consideration.

The GRU model consists of two main gates—reset and update gates—that regulate the flow of information. The reset gate r_t controls how much of the previous information is forgotten, while the update gate z_t determines how much of the past information is carried forward. The primary equations for the GRU are given by the following:

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r), \tag{17}$$

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z), \tag{18}$$

$$\tilde{h}_t = \tanh(W \cdot [r_t \odot h_{t-1}, x_t] + b), \tag{19}$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \tag{20}$$

where r_t and z_t are the reset and update gates, respectively, and $\tilde{h}_t$ is the candidate activation. GRUs reduce the complexity of the network by combining the forget and input gates into a single update gate, which often results in faster training times compared to LSTM. Figure 6 illustrates this GRU structure, displaying the flow of data through the reset and update gates for sequential data processing.

The GRU hyperparameters, optimized using nested cross-validation (outer = 5, inner = 3), are summarized in Table 12. Temporary values include two layers, 64 GRU units, a dropout rate of 0.3, and the Adam optimizer with a learning rate of 0.0015.

Table 12. Optimal Hyperparameters for GRU in Different Sensor Configurations.

Sensor Configuration	GRU Layers	GRU Units	Dropout Rate	Optimizer/Learning Rate
2 Sensors—Non-overlapping	2	96	0.30	Adam/0.0032
2 Sensors—50% Overlapping	4	64	0.20	Adamax/0.0053
4 Sensors—Non-overlapping	4	64	0.30	Adam/0.0027
4 Sensors—50% Overlapping	3	96	0.30	Adam/0.0034

By leveraging GRU's simplified architecture, this model is able to efficiently capture both short-term and long-term dependencies, making it suitable for human activity recognition with complex sensor data. In this architecture, the reset and update gates play critical roles in managing sequential data processing. The asterisk (*) in the figure denotes multiplication, and the "−1" represents subtraction, key operations that adjust the information flow through the GRU unit.

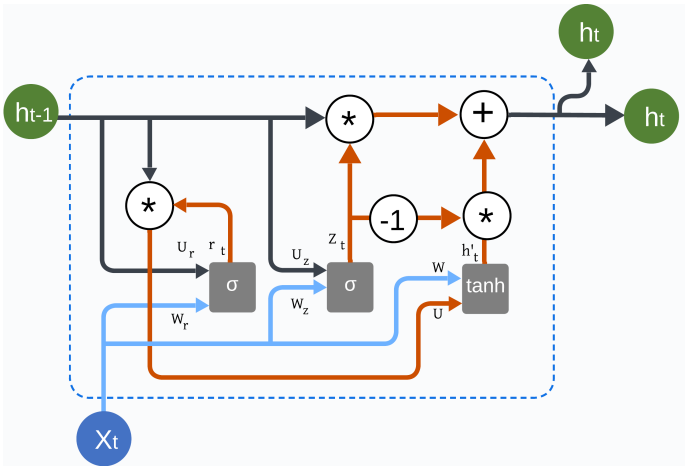


Figure 6. Architecture of a Gated Recurrent Unit (GRU) Network used in Activity Recognition. Adapted from [37], showing the flow through the reset and update gates, facilitating efficient sequential data processing.

3.7.7. Long Short-Term Memory (LSTM)

Long Short-Term Memory (LSTM) networks [38] are a specialized form of Recurrent Neural Networks (RNNs) designed to handle sequential data and capture long-term dependencies. This makes LSTMs highly suitable for HAR, where temporal patterns are crucial.

In this study, a multi-layer LSTM network was implemented to model the sequential patterns in sensor data, optimizing key parameters such as the number of LSTM units and learning rate for effective classification performance. The core of the LSTM consists of gates—forget, input, and output—that control information flow, allowing the network to manage both short-term and long-term dependencies in the data. These gates interact with the cell state and the network's memory, to retain relevant information and discard what's unnecessary over time.

As depicted in Figure 7, the forget gate discards irrelevant information, the input gate allows new data to be added, and the output gate determines what is passed to the next layer. These mechanisms enable LSTM to retain useful patterns for activity recognition, selectively managing both short-term and long-term dependencies.

The LSTM's core equations are expressed as:

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f), \tag{21}$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i), \tag{22}$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o), \tag{23}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c \cdot [h_{t-1}, x_t] + b_c), \tag{24}$$

$$h_t = o_t \odot \tanh(c_t), \tag{25}$$

where f_t, i_t, and o_t represent the activations of the forget, input, and output gates, respectively, while c_t represents the cell state update. These gates use the sigmoid function σ, while cell updates rely on element-wise operations $\odot$, and the previous hidden state h_{t-1} and input x_t influence the gating mechanisms.

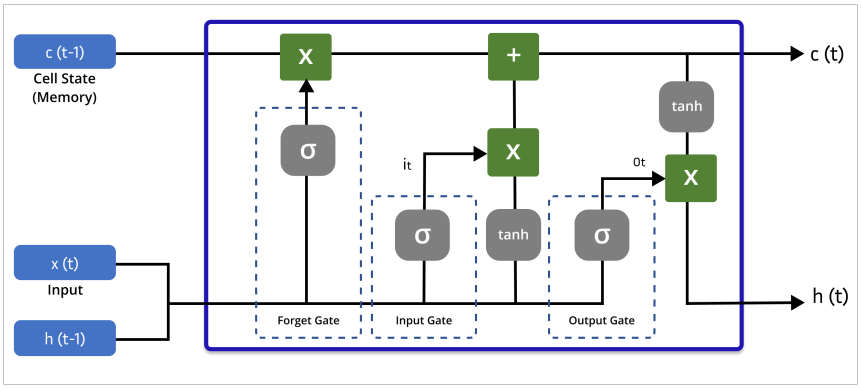


Figure 7. Architecture of a Long Short-Term Memory (LSTM) Network utilized in Activity Recognition. Adapted from [38], showing the flow of information through the forget, input, and output gates to manage long-term dependencies in sequential data.

The LSTM's hyperparameters were optimized using a nested cross-validation process, as shown in Table 13. For the two-sensor, non-overlapping configuration, two LSTM layers with 32 units and a dropout rate of 0.30 performed best, while the 50%-overlapping setup used one LSTM layer with 128 units. The four-sensor non-overlapping configuration similarly required one LSTM layer, whereas the 50%-overlapping setup needed a more complex architecture with three layers.

Table 13. Optimal Hyperparameters for LSTM in Different Sensor Configurations.

Sensor Configuration	LSTM Layers	LSTM Units	Dropout Rate	Optimizer/Learning Rate
2 Sensors—Non-overlapping	2	32	0.30	Adam/0.0086
2 Sensors—50% Overlapping	1	128	0.40	RMSProp/0.0014
4 Sensors—Non-overlapping	1	128	0.20	Adam/0.0088
4 Sensors—50% Overlapping	3	128	0.20	Adam/0.0014

The LSTM's ability to capture both short-term and long-term dependencies was key to improving classification accuracy across different sensor configurations. By fine-tuning the hyperparameters, the model was able to effectively differentiate between various human activities, achieving robust performance through careful management of sequential data.

3.7.8. Temporal Convolutional Networks (TCN)

Temporal Convolutional Networks (TCN) [28] are highly effective for sequence modeling, particularly in handling long-term dependencies through dilated causal convolutions. This enables TCNs to capture temporal patterns over extended periods without losing context. In this study, TCNs were employed to classify human activities based on multi-sensor data. Key parameters such as filter size, layers, and dilation factor were optimized for performance.

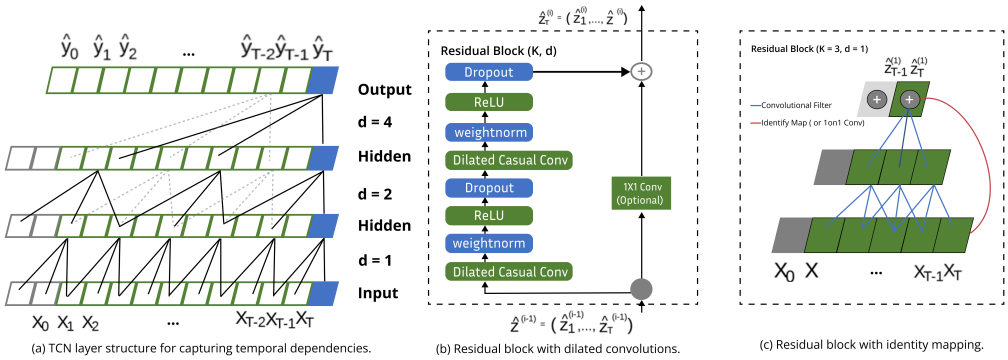
The core dilated convolution operation in a TCN is given by the following:

$$y(t) = \sum_{i=0}^{k-1} x(t - d \cdot i) \cdot w(i), \tag{26}$$

where $y(t)$ is the output at time t, $x(t)$ is the input signal, $w(i)$ denotes filter weights, and d is the dilation factor. This formulation enhances the model's ability to recognize temporal dependencies across varying time steps.

The TCN architecture, as shown in Figure 8, processes temporal sequences using convolutional layers that expand the receptive field efficiently. Dilations in increasing order capture long-range dependencies without a significant computational cost. Dropout layers

with a rate of 0.25 were applied to mitigate overfitting, and different optimizers (Adam, RMSProp) were used depending on sensor configurations.



(a) TCN layer structure for capturing temporal dependencies. (b) Residual block with dilated convolutions. (c) Residual block with identity mapping.

Figure 8. Architecture of a Temporal Convolutional Network (TCN) for Activity Recognition, adapted from [28]. Dilated causal convolutions capture long-term dependencies, with dropout layers to prevent overfitting.

The TCN demonstrated robustness in handling complex sensor configurations. Dilated convolutions enabled the model to efficiently capture long-range temporal relationships, crucial for differentiating subtle variations in human activities. In more complex configurations, such as the four-sensor setup, dropout layers further improved generalization by reducing overfitting, especially with high-dimensional data.

The optimal hyperparameters, identified through nested cross-validation, are summarized in Table 14. For the two-sensor 50%-overlapping setup, the best results were achieved using two layers with filter sizes of 96, 32, and 128, and an Adam optimizer with a learning rate of 0.0006. In contrast, the four-sensor non-overlapping setup favored filter sizes of 96, 64, and 32, using Adam with a learning rate of 0.0011.

Table 14. Optimal Hyperparameters for TCN in Different Sensor Configurations.

Sensor Configuration	Layers	Filter Sizes	Kernel Sizes	Optimizer/Learning Rate
2 Sensors—Non-overlapping	2	[96, 128, 96]	[2, 3, 4]	Adamax/0.0043
2 Sensors—50% Overlapping	2	[96, 32, 128]	[3, 2, 2]	Adam/0.0006
4 Sensors—Non-overlapping	2	[96, 64, 32]	[3, 3, 2]	Adam/0.0011
4 Sensors—50% Overlapping	1	[32, 32, 128]	[4, 3, 3]	RMSProp/0.0025

In the more complex four-sensor, 50%-overlapping configuration, a simplified architecture of one layer, with filter sizes of 32, 32, and 128, and RMSProp (learning rate 0.0025) was optimal. These configurations highlight the adaptability of TCNs in different sensor setups, capturing both local and long-range temporal dependencies effectively.

The nested cross-validation approach ensured unbiased tuning, contributing to the model's ability to generalize across various sensor setups. By capturing both short-term and long-term dependencies, the TCN model achieved robust performance in human activity recognition tasks across different sensor configurations.

3.7.9. Transformer

The Transformer model [29] is a groundbreaking architecture that relies solely on attention mechanisms to capture dependencies in sequential data, eliminating the need for recurrence. This self-attention mechanism allows the Transformer to focus on relevant parts of the input sequence, making it highly effective for handling long-term dependencies in HAR.

The core of the Transformer is the multi-head attention mechanism, which computes attention weights for each input token across multiple heads, allowing the model to focus on various aspects of the input sequence. The attention mechanism is defined as follows:

$$Attention(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}}\right)\mathbf{V}, \tag{27}$$

where the query $\mathbf{Q}$, key $\mathbf{K}$, and value $\mathbf{V}$ are matrices derived from the input embeddings, with d_k representing the dimension of the queries and keys. This formulation allows the model to simultaneously attend to various parts of the sequence in a parallel manner. As illustrated in Figure 9, the multi-head attention mechanism allows the model to attend to various parts of the sequence in parallel, supported by positional encoding to handle sequential dependencies effectively.

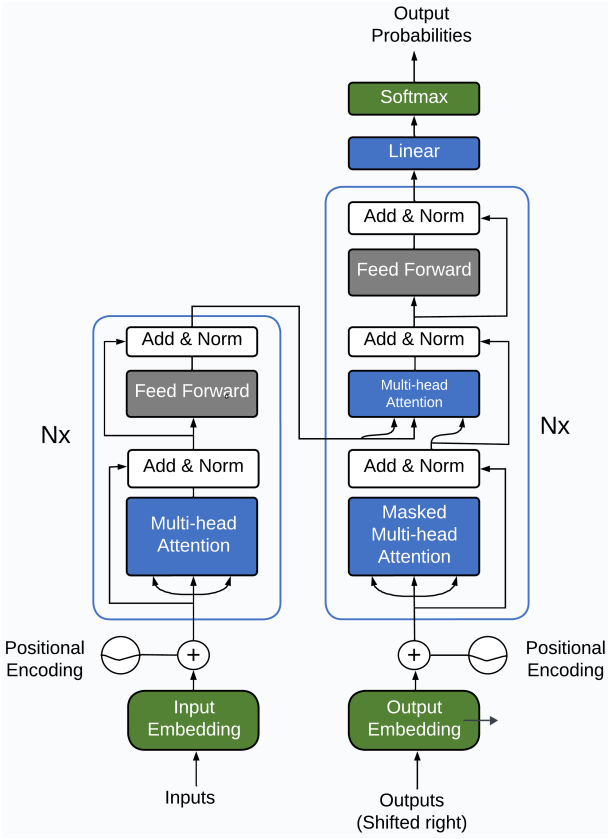


Figure 9. Architecture of the Transformer Model used in Activity Recognition, illustrating the multi-head attention and feed-forward layers, adapted from [29]. The positional encoding enables handling of sequential data without recurrence.

The hyperparameters for the Transformer model were optimized using nested cross-validation (outer = 5, inner = 3), as outlined in Table 15. Temporary values include four attention heads, two layers, and an Adam optimizer with a learning rate of 0.0008.

Table 15. Optimal Hyperparameters for Transformer in Different Sensor Configurations.

Sensor Configuration	Layers	Head Size/Numbers	MLP Layers/MLP Units	Optimizer/Learning Rate
2 Sensors—Non-overlapping	6	168/4	2/[224, 128, 224]	Adamax/0.0060
2 Sensors—50% Overlapping	4	40/8	1/[96, 224, 128]	AdaMax/0.0066
4 Sensors—Non-overlapping	4	8/16	2/[224, 256, 192]	Adamax/0.0055
4 Sensors—50% Overlapping	2	8/6	2/[96, 96, 64]	Adam/0.0015

The Transformer's self-attention mechanism, combined with its ability to process sequences in parallel, makes it highly efficient and powerful for HAR tasks, particularly in capturing complex temporal relationships within sensor data.

4. Results and Discussion

This section evaluates the performance of nine models—DT, KNN, RF, SVC, XGBoost, GRU, LSTM, TCN, and Transformer—for activity recognition within *Intelligent Transportation Systems (ITS)*. Effective activity recognition is crucial for enhancing safety and efficiency in ITS applications. The analysis compares model performance across two sensor configurations (two-sensor and four-sensor) and two data segmentation strategies (non-overlapping and 50% overlapping), focusing on key performance metrics such as accuracy, precision, and recall to assess their viability for real-time applications in ITS.

4.1. Model Performance Overview

Decision Trees (DT) were selected for their simplicity and interpretability, making them ideal for real-time ITS applications where model transparency is crucial. While DT maintains a balance between accuracy and efficiency, it is somewhat prone to overfitting, particularly with smaller datasets.

K-Nearest Neighbors (KNN) was included for its flexibility in handling diverse activity patterns. However, its performance can decline in high-dimensional spaces, which leads to computational challenges. Various values for k and distance metrics were tested to optimize KNN's sensitivity to the sensor data.

Random Forest (RF) was introduced as an ensemble model to assess its classification consistency and ability to mitigate overfitting through the aggregation of multiple decision trees. RF produced reliable results across sensor configurations, demonstrating particularly strong performance in non-overlapping datasets, where it maintained high accuracy across various activities.

Support Vector Classifier (SVC) with the RBF kernel was chosen for its capacity to manage non-linear data effectively, enhancing accuracy in activity recognition, especially with smaller datasets. Its strong generalization capabilities contributed to high classification performance.

XGBoost, a gradient-boosting algorithm, was selected for its computational efficiency and advanced feature-handling capabilities. It effectively captured complex activity patterns, excelling in overlapping data segmentation, and achieving accuracy comparable to deep learning models such as TCN. Its robustness and precision make it particularly suitable for ITS applications where both accuracy and efficiency are paramount.

Gated Recurrent Unit (GRU) was incorporated to evaluate its sequential learning capabilities while offering reduced computational requirements compared to LSTM. The gating mechanism in GRU enables effective learning of activity sequences, which is beneficial for real-time ITS applications. GRU performed well across sensor configurations, particularly in non-overlapping data, where its lower complexity supported faster inference.

Long Short-Term Memory (LSTM) networks are well-suited for capturing long-term dependencies, which are crucial for distinguishing activities such as walking and running. However, its performance was slightly diminished with non-overlapping data configurations, as LSTM relies on the continuity of sequential data.

Temporal Convolutional Networks (TCN), utilizing dilated convolutions to capture extended temporal dependencies, consistently achieved high accuracy, particularly in overlapping segmentation. Its ability to retain contextual information across sequences makes it an excellent choice for real-time ITS tasks.

Transformer models, which leverage self-attention mechanisms, were evaluated for their ability to manage complex dependencies in sequential data without the limitations of recurrent connections. They excelled in overlapping segmentation scenarios, effectively capturing long-term dependencies across activities, achieving competitive accuracy, and highlighting their scalability for ITS applications.

4.2. Sensor Configurations and Data Segmentation

The evaluation utilized two sensor configurations—two-sensor (accelerometer and gyroscope) and four-sensor (which includes a linear accelerometer and gravity sensor)—in

conjunction with two segmentation strategies: non-overlapping and 50% overlapping. The implementation of overlapping segmentation proved advantageous by preserving transitional information between activities, which is crucial for accurately capturing the dynamics of human motion. This strategy significantly enhanced model performance, leading to notable improvements in accuracy across all models, demonstrating the effectiveness of this approach in real-time activity recognition within ITS.

4.3. Two-Sensor Configurations

4.3.1. Two-Sensors: Model Performance with Non-Overlapping Data

In this configuration, data from two sensors—an accelerometer and a gyroscope—were analyzed using non-overlapping segments. The models evaluated include DT, KNN, RF, SVC, XGBoost, GRU, LSTM, TCN, and Transformer.

The evaluation results in Table 16 reveal that the **TCN** model achieves the highest performance across most metrics, with an F1 Score of **0.9757** on the test set. This model demonstrates strong consistency across training, validation, and test sets, underscoring its robustness and suitability for non-overlapping configurations in human activity recognition.

Table 16. Performance comparison of two sensors with non-overlapping data for various models, where the performance metrics are Accuracy, Precision, Recall, and F1 Score.

Models	Accuracy			Precision			Recall			F1 Score		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
DT	0.9726	0.9724	0.9725	0.9726	0.9724	0.9725	0.9726	0.9724	0.9725	0.9726	0.9724	0.9725
KNN	0.9693	0.9723	0.9703	0.9693	0.9723	0.9703	0.9693	0.9723	0.9703	0.9693	0.9723	0.9703
RF	0.9759	0.9792	0.9720	0.9759	0.9792	0.9720	0.9759	0.9792	0.9720	0.9759	0.9792	0.9720
SVC	0.9742	0.9752	0.9726	0.9742	0.9752	0.9726	0.9742	0.9752	0.9726	0.9742	0.9752	0.9726
XGBoost	0.9755	0.9767	0.9753	0.9755	0.9767	0.9753	0.9755	0.9767	0.9753	0.9755	0.9767	0.9753
GRU	0.9760	0.9765	0.9748	0.9760	0.9765	0.9748	0.9760	0.9765	0.9748	0.9760	0.9765	0.9748
LSTM	0.9761	0.9724	0.9738	0.9761	0.9724	0.9738	0.9761	0.9724	0.9738	0.9761	0.9724	0.9738
TCN	**0.9762**	**0.9768**	**0.9757**	**0.9762**	**0.9768**	**0.9757**	**0.9762**	**0.9768**	**0.9757**	**0.9762**	**0.9768**	**0.9757**
Transformer	0.9724	0.9745	0.9724	0.9724	0.9745	0.9724	0.9724	0.9745	0.9724	0.9724	0.9745	0.9724

DT and KNN models offer reliable baseline performance, with DT reaching an F1 Score of 0.9725 on the test set. However, these simpler models tend to lag behind more advanced models, particularly when handling high-dimensional sensor data [33,34].

Among the ensemble models, RF and XGBoost exhibit solid performance, leveraging ensemble learning to enhance generalization. RF achieves a test F1 Score of 0.9720, while XGBoost achieves a higher F1 Score of 0.9753, reflecting the benefits of boosting techniques for managing complex data interactions [26,27].

SVC also demonstrates adaptability to non-linear data with a test F1 Score of 0.9726, making it a strong candidate for activity recognition in configurations lacking temporal overlap [36]. The deep learning models GRU and LSTM offer consistent results, with F1 Scores of 0.9748 and 0.9738, respectively, on the test set, highlighting their effectiveness in sequential data processing even without overlapping segmentation [37,38].

Despite its strengths, the Transformer model performs slightly lower than TCN in this configuration, with an F1 Score of 0.9724 on the test set, suggesting that while its self-attention mechanism is beneficial for long-range dependencies, TCN may be more effective in handling structured, sequential data in non-overlapping setups.

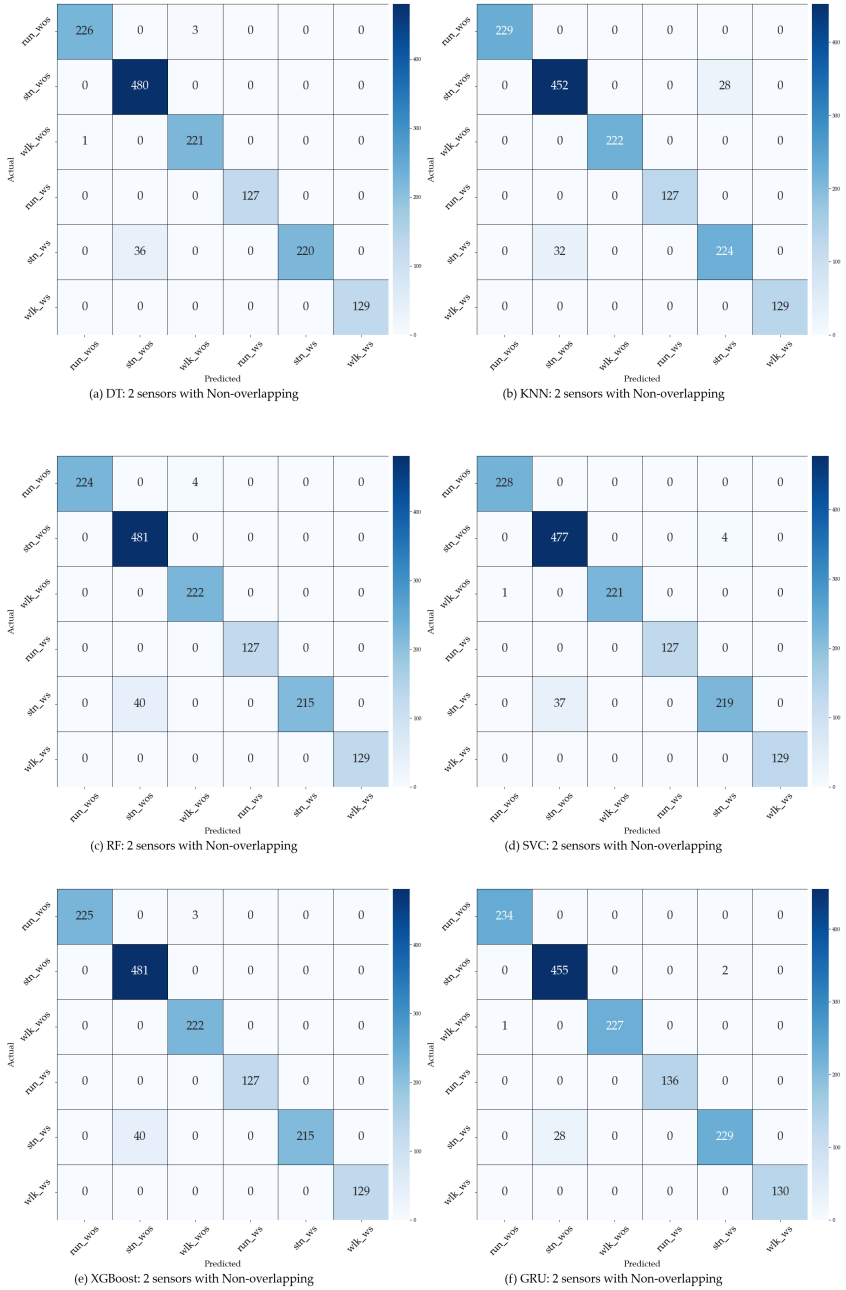
In conclusion, the **TCN** model emerges as the best performer in this configuration, demonstrating high accuracy, precision, recall, and F1 Score across all sets. Its superior

generalization capabilities highlight its robustness for human activity recognition in non-overlapping data settings.

4.3.2. Confusion Matrix Analysis for Two-Sensor Non-Overlapping Configuration

By examining the confusion matrices presented in Figure 10 we can gain deeper insights into how effectively each model classifies different activity categories, revealing specific strengths and weaknesses in handling certain tasks.

The confusion matrices provide insightful details about the performance of different models in the two-sensor non-overlapping configuration:



(a) DT: 2 sensors with Non-overlapping

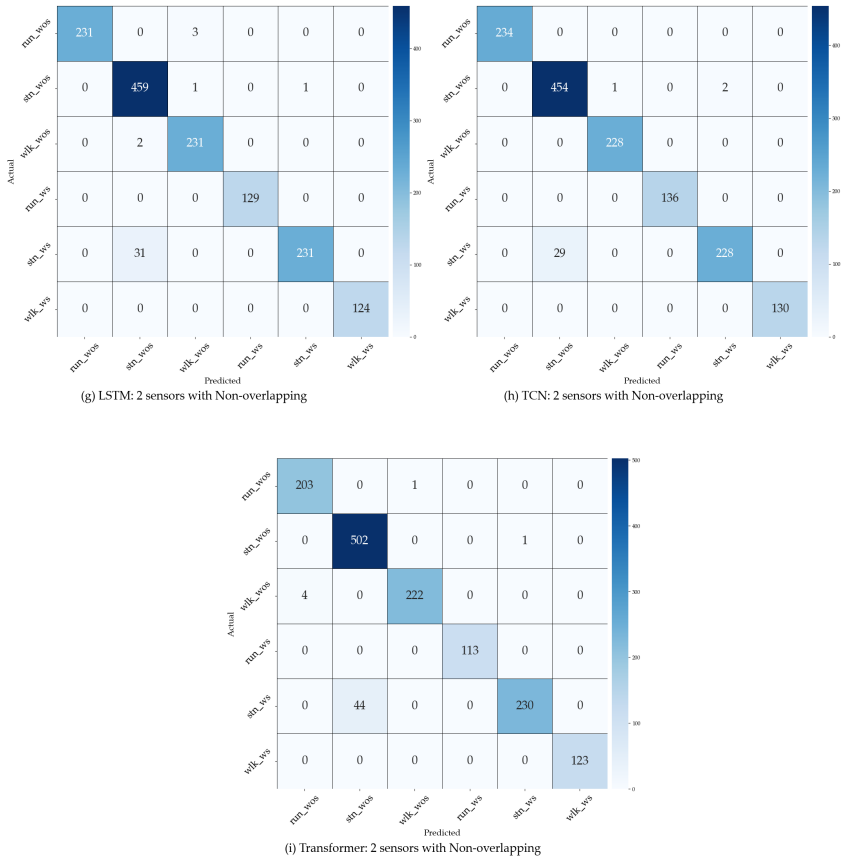
(b) KNN: 2 sensors with Non-overlapping

(c) RF: 2 sensors with Non-overlapping

(d) SVC: 2 sensors with Non-overlapping

(e) XGBoost: 2 sensors with Non-overlapping

(f) GRU: 2 sensors with Non-overlapping



(g) LSTM: 2 sensors with Non-overlapping

(h) TCN: 2 sensors with Non-overlapping

(i) Transformer: 2 sensors with Non-overlapping

Figure 10. Confusion Matrices for models using a two-sensor configuration with non-overlapping data segments: (**a**) DT, (**b**) KNN, (**c**) RF, (**d**) SVC, (**e**) XGBoost, (**f**) GRU, (**g**) LSTM, (**h**) TCN, (**i**) Transformer.

DT: As shown in Figure 10a, the DT model demonstrates relatively consistent performance across most activity classes, correctly classifying 221 instances of *wlk_wos*. However, the model struggles with 36 misclassifications of *stn_ws*, suggesting difficulties in distinguishing between subtle differences in stationary and non-stationary activities. This pattern highlights DT's known tendency to overfit, making confident but incorrect predictions, especially in cases with smaller datasets or less distinct class boundaries [26].

KNN: As depicted in Figure 10b, KNN shows sensitivity to closely spaced classes in the feature space, misclassifying 28 instances of *stn_wos*. KNN correctly classifies 222 instances of *wlk_wos*, but its instance-based approach struggles in high-dimensional sensor data without overlap, where the feature space becomes more complex [35].

RF: Figure 10c shows RF's consistent performance, especially in classifying stationary activities like *stn_wos* (481 correct classifications). However, it misclassifies 40 instances of *stn_ws*, indicating room for improvement in handling subtle distinctions in movement. RF's ensemble learning structure provides stability, which is reflected in its balanced performance across various activity classes [26].

SVC: The confusion matrix in Figure 10d illustrates the strong performance of SVC, with minimal misclassifications. SVC's capacity to maximize margins effectively separates the classes, correctly identifying 228 instances of *run_wos* and 477 instances of *stn_wos*, confirming its suitability for non-linear data in non-overlapping setups [39].

XGBoost: As seen in Figure 10e, XGBoost excels in distinguishing complex activity patterns, with only a few misclassifications, such as three instances of *run_wos* and 40 of *stn_ws*. Its gradient-boosting approach enhances its capacity for handling complex data structures, making it a strong contender in the non-overlapping configuration [27].

GRU: The GRU model, shown in Figure 10f, demonstrates efficient classification of sequential activities, correctly classifying 136 instances of *run_vs* and 229 of *stn_ws*. Although

it has a few misclassifications in *stn_wos*, GRU's simplicity and lower computational cost make it suitable for real-time ITS applications [37].

LSTM: As indicated in Figure 10g, LSTM misclassifies 31 instances of *stn_ws*, reflecting its limitation in non-overlapping data. However, it still performs well across other activities, showing flexibility in handling sequential data despite the lack of continuity [38].

TCN: As shown in Figure 10h, TCN provides the highest classification accuracy, especially for dynamic activities, with minimal misclassifications. It is particularly effective in correctly classifying activities like *run_wos* (234 correct classifications) and *wlk_wos* (228 correct classifications). Its convolutional structure allows efficient handling of non-overlapping segments, reinforcing TCN as the top-performing model in this setup [28].

Transformer: The Transformer model, illustrated in Figure 10i, performs well in classifying *stn_wos* with 502 correct classifications but shows slight limitations with dynamic activities, misclassifying 44 instances of *stn_ws*. Although its self-attention mechanism aids in long-range dependency handling, it may not fully leverage the sequential structure needed in non-overlapping setups [29].

In summary, both the table and confusion matrix analyses consistently highlight the **TCN** model as the best performer for this two-sensor, non-overlapping configuration. Its ability to handle non-overlapping temporal data without sacrificing accuracy or interpretability makes it ideal for human activity recognition tasks where data continuity is limited. TCN outperforms other models in classifying a range of activities accurately, proving its robustness and adaptability in this setup.

4.3.3. Two-Sensors: Model Performance with 50% Overlapping Data

In this configuration, the same two sensors—accelerometer and gyroscope—are utilized, but with a 50% overlapping data segmentation strategy. This method enhances the ability of models to capture contextual transitions between activities by retaining more temporal information, which is crucial for improved classification accuracy.

The evaluation results in Table 17 indicate that the **TCN** model achieves the highest performance across the majority of metrics, with an F1 Score of **0.9765** on the test set. This model exhibits strong consistency throughout training, validation, and test phases, emphasizing its robustness in addressing overlapping configurations for human activity recognition.

The Decision Tree (DT) and K-Nearest Neighbors (KNN) models serve as reliable baseline performance indicators, with DT attaining an F1 Score of 0.9707 on the test set. However, these simpler models typically fall short compared to their more advanced counterparts, particularly when processing high-dimensional sensor data [33,34].

Ensemble models such as Random Forest (RF) and XGBoost demonstrate commendable performance, effectively utilizing ensemble learning techniques to enhance generalization. RF achieves a test F1 Score of 0.9730, while XGBoost performs slightly better with an F1 Score of **0.9749**. This improvement underscores the advantages of boosting methods in managing intricate data interactions [26,27].

Support Vector Classifier (SVC) also shows strong adaptability to non-linear data, reaching a test F1 Score of 0.9745. This positions SVC as a strong candidate for activity recognition, particularly in configurations that require efficient handling of overlapping data. Additionally, the deep learning models GRU and LSTM yield consistent outcomes, with F1 Scores of 0.9750 and 0.9762, respectively, on the test set, demonstrating their effectiveness in processing sequential data [37,38].

The Transformer model exhibits robust capabilities, achieving an F1 Score of 0.9746 on the test set. Although it shows promise with its self-attention mechanism for capturing complex relationships, it falls slightly short of TCN's performance, indicating that while effective, TCN may be better suited for handling structured, sequential data in overlapping contexts.

In conclusion, the **TCN** model stands out as the best performer in this configuration, exhibiting high accuracy, precision, recall, and F1 Score across all evaluation sets. Its

superior generalization capabilities further reinforce its robustness for human activity recognition in settings utilizing overlapping data.

Table 17. Performance comparison of two sensors with 50% overlapping data for various models, where the performance metrics are Accuracy, Precision, Recall, and F1 Score.

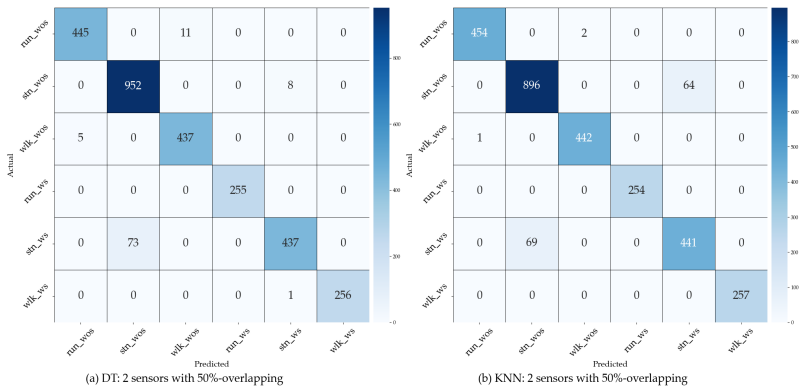
Models	Accuracy			Precision			Recall			F1 Score		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
DT	0.9749	0.9675	0.9707	0.9749	0.9675	0.9707	0.9749	0.9675	0.9707	0.9749	0.9675	0.9707
KNN	0.9705	0.9712	0.9672	0.9705	0.9712	0.9672	0.9705	0.9712	0.9672	0.9705	0.9712	0.9672
RF	0.9758	0.9679	0.9730	0.9758	0.9679	0.9730	0.9758	0.9679	0.9730	0.9758	0.9679	0.9730
SVC	0.9774	0.9715	0.9745	**0.9774**	0.9715	**0.9745**	**0.9774**	0.9715	**0.9745**	**0.9774**	0.9715	**0.9745**
XGBoost	0.9776	0.9706	0.9749	0.9776	0.9706	0.9749	0.9776	0.9706	0.9749	0.9776	0.9706	0.9749
GRU	0.9751	0.9749	0.9750	0.9751	0.9749	0.9750	0.9751	0.9749	0.9750	0.9751	0.9749	0.9750
LSTM	0.9764	0.9758	0.9762	**0.9764**	**0.9758**	**0.9762**	**0.9764**	**0.9758**	**0.9762**	**0.9764**	**0.9758**	**0.9762**
TCN	**0.9771**	**0.9764**	**0.9765**	**0.9771**	**0.9764**	**0.9765**	**0.9771**	**0.9764**	**0.9765**	**0.9771**	**0.9764**	**0.9765**
Transformer	0.9740	0.9763	0.9746	0.9740	0.9763	0.9746	0.9740	0.9763	0.9746	0.9740	0.9763	0.9746

4.3.4. Confusion Matrix Analysis for Two-Sensor 50%-Overlapping Configuration

The confusion matrices, displayed in Figure 11, provide detailed insights into the classification results for each model under the 50% overlapping data configurations. By presenting a comprehensive view of the predicted versus actual activity classes, these matrices enable a thorough assessment of model performance. They not only reveal the overall accuracy of each model but also illustrate specific strengths and weaknesses in handling different activities.

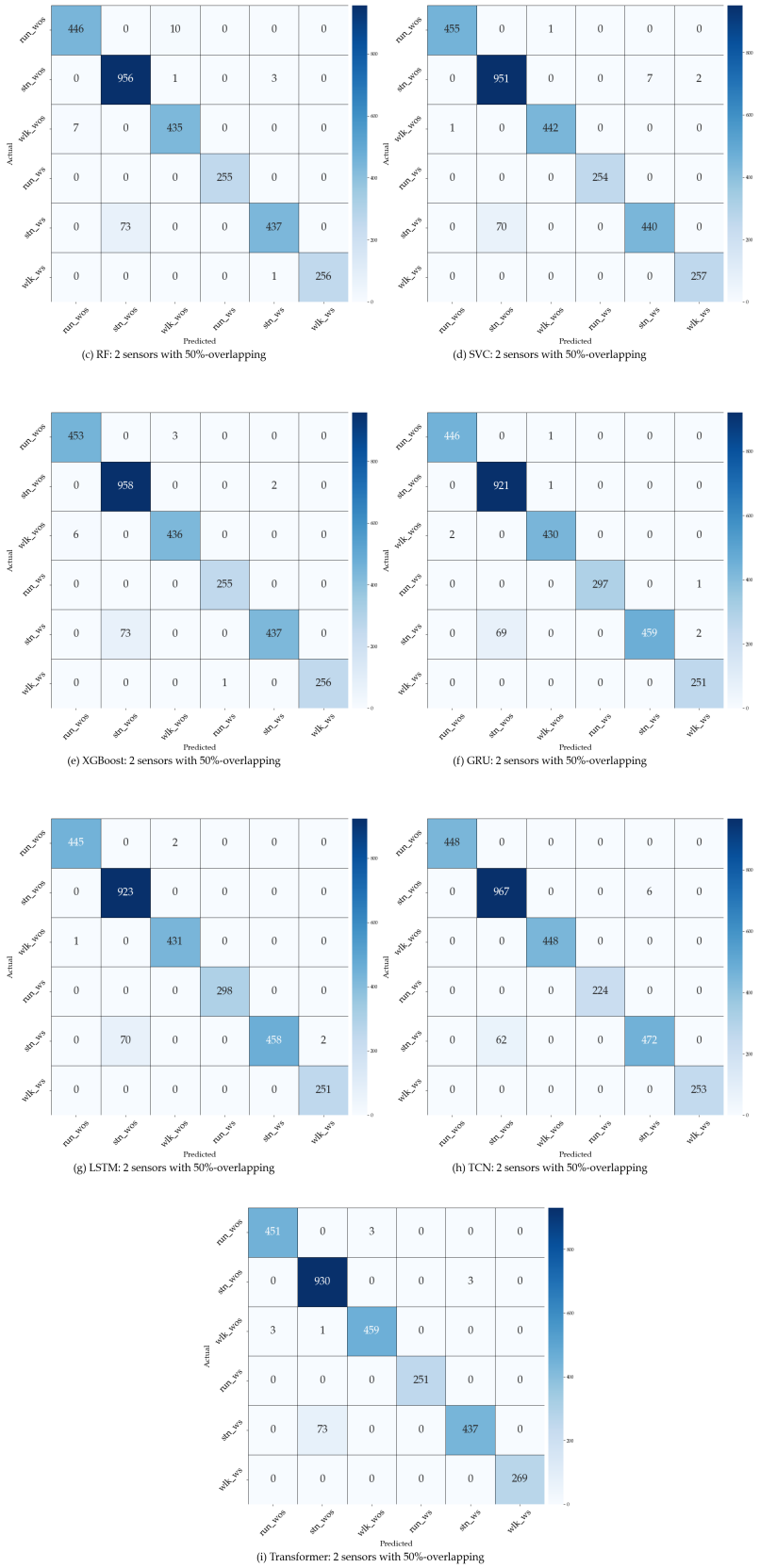
The confusion matrices provide insightful details about the performance of different models in the two-sensor with 50%-overlapping configurations, showcasing how well each model can capture temporal dependencies and transitions between activities.

DT: In Figure 11a, the DT model correctly classifies a majority of the activity classes. However, it struggles with distinguishing between similar stationary activities, as seen in the 73 misclassifications of the class *stn_ws*. This limitation highlights the tendency of the DT model to overfit in overlapping data scenarios, which can result from its hierarchical structure that may not generalize well to nuanced differences between similar classes [26]. Moreover, the reliance on predetermined splitting criteria might lead to a lack of adaptability when encountering the subtle variations present in stationary activities.



(a) DT: 2 sensors with 50%-overlapping (b) KNN: 2 sensors with 50%-overlapping

Figure 11. *Cont.*



(c) RF: 2 sensors with 50%-overlapping

(d) SVC: 2 sensors with 50%-overlapping

(e) XGBoost: 2 sensors with 50%-overlapping

(f) GRU: 2 sensors with 50%-overlapping

(g) LSTM: 2 sensors with 50%-overlapping

(h) TCN: 2 sensors with 50%-overlapping

(i) Transformer: 2 sensors with 50%-overlapping

Figure 11. Confusion Matrices for models using a two-sensor configuration with 50% overlapping data segments: (**a**) DT, (**b**) KNN, (**c**) RF, (**d**) SVC, (**e**) XGBoost, (**f**) GRU, (**g**) LSTM, (**h**) TCN, (**i**) Transformer.

KNN: The KNN confusion matrix in Figure 11b reflects its ability to accurately classify dynamic activities such as *run_wos*, where it achieves 454 correct classifications. However, it shows some difficulty with stationary activities, particularly in the class *stn_ws*, where 69 instances were misclassified. This behavior aligns with KNN's sensitivity to local neighbor choices and highlights its struggles in higher-dimensional feature spaces [35]. The model's reliance on proximity for classification can lead to misclassifications, particularly when different activity classes have similar features or when noise is present in the data.

RF: Figure 11c shows that the RF model effectively handles the majority of classes, particularly *stn_wos*, with 956 correctly classified instances. However, it faces some difficulty with closely related classes, such as *wlk_wos*, where it misclassifies seven instances. This performance indicates that while ensemble learning provides robustness, RF may still struggle with nuanced distinctions in overlapping configurations [26].

SVC: Figure 11d demonstrates the robustness of the SVC model. It correctly classifies 951 instances of *stn_wos* and 455 instances of *run_wos*, with only nine and one misclassifications, respectively. The SVC's performance here highlights its ability to manage overlapping data effectively, particularly in scenarios with non-linear decision boundaries [39]. Its effectiveness in capturing the relevant features of the data allows it to achieve high accuracy, even in the presence of overlapping segments. Additionally, SVC's capacity to use kernel functions can help in transforming the feature space to better separate classes, further enhancing its classification accuracy.

XGBoost: As shown in Figure 11e, XGBoost achieves strong classification performance, particularly in *stn_wos*, with 958 correct classifications. However, it encounters challenges in distinguishing *wlk_ws*, misclassifying only one instance as *wlk_wos*, which may result from the complexity of interactions in overlapping sequences. This outcome underlines the effectiveness of boosting in refining predictions across diverse activity classes [27].

GRU: Figure 11f reveals GRU's strengths in handling temporal data with overlapping segments, as it achieves 446 correct classifications for *run_wos* and 459 for *stn_ws*. Despite its sequential learning ability, GRU misclassifies 69 instances of *stn_ws*, suggesting that it may still benefit from additional temporal smoothing in stationary categories [37].

LSTM: As shown in Figure 11g, the LSTM model effectively classifies both stationary and dynamic activities, achieving 445 correct classifications for *run_wos* and 923 for *stn_wos*. However, it struggles slightly with *stn_ws*, misclassifying 70 instances. This performance highlights LSTM's strength in capturing long-term dependencies, leveraging overlapping temporal data to improve differentiation between similar activities. The model's architecture enables it to retain past information, making it well-suited for complex activity recognition tasks, especially where temporal overlap enhances classification accuracy [38].

TCN: Similarly, Figure 11h illustrates the high performance of the TCN, which correctly classifies 448 instances of *run_wos* and 967 instances of *stn_wos*, demonstrating comparable performance to LSTM. TCN benefits from its ability to model both short-term and long-term dependencies through its convolutional architecture, which allows it to process overlapping data more efficiently by capturing spatial and temporal relationships within the data [28]. TCN shows particular strength in activities such as *run_ws*, correctly classifying 224 instances. However, like LSTM, it misclassifies instances of *stn_ws*, suggesting that even with overlapping data, stationary activities remain a challenge for both deep learning models.

Transformer: Figure 11i demonstrates the Transformer's strong performance, correctly classifying 451 instances of *run_wos* and 930 of *stn_wos*. However, the model misclassifies 73 instances of *stn_ws*, suggesting that while its self-attention mechanism captures long-range dependencies effectively, it faces challenges distinguishing closely related stationary activities. This indicates that, despite its proficiency with sequential patterns, the Transformer may benefit from further tuning for overlapping configurations in stationary states [29].

In conclusion, overlapping data segmentation enhances model performance by preserving important temporal transitions. While LSTM, TCN, and Transformer exhibit strong

classification capabilities, TCN emerges as the most consistent and accurate model, further affirming the importance of deep learning architectures designed for sequential data in human activity recognition applications. The deep learning models' capability to leverage both short-term and long-term dependencies enables them to handle the complexity introduced by overlapping segments effectively, making them suitable choices for capturing activity transitions. Further analysis of the comparative performance of these models across configurations will be presented in Section 4.5, where a holistic evaluation of their strengths and limitations will be discussed.

4.4. Four-Sensor Configurations

4.4.1. Four-Sensors: Model Performance with Non-Overlapping Data

In this configuration, four sensors—accelerometer, gyroscope, linear accelerometer, and gravity—were utilized with non-overlapping data segments. The inclusion of additional sensors aims to capture a wider range of movement patterns, potentially enhancing classification accuracy across different models. This strategy leverages the unique strengths of each sensor, allowing for a more comprehensive analysis of monitored activities.

The four-sensor setup enriches the feature set and provides nuanced information about the subject's movements, aiding in distinguishing between similar activities. For instance, the gyroscope contributes to rotational movement detection, while the linear accelerometer and gravity sensor provide insights into linear acceleration and gravitational effects. This multidimensional approach is expected to improve the models' ability to recognize subtle variations in activity patterns.

The evaluation results in Table 18 reveal that the **TCN** model achieves the highest performance across most metrics, with an F1 Score of **0.9770** on the test set. This model demonstrates strong consistency across training, validation, and test sets, underscoring its robustness in handling non-overlapping configurations for human activity recognition.

DT and KNN models offer reliable baseline performance, with DT reaching an F1 Score of 0.9719 on the test set. However, these simpler models tend to lag behind more advanced models, particularly when handling high-dimensional sensor data [33,34].

Among the ensemble models, RF and XGBoost exhibit solid performance, leveraging ensemble learning to enhance generalization. RF achieves a test F1 Score of 0.9753, while XGBoost achieves a higher F1 Score of 0.9735, reflecting the benefits of boosting techniques for managing complex data interactions [26,27].

Table 18. Performance comparison of four sensors with Non-overlapping data for various models, where the performance metrics are Accuracy, Precision, Recall, and F1 Score.

Models	Accuracy			Precision			Recall			F1 Score		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
DT	0.9723	0.9709	0.9719	0.9723	0.9709	0.9719	0.9723	0.9709	0.9719	0.9723	0.9709	0.9719
KNN	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681	0.9681
RF	0.9755	0.9750	0.9753	0.9755	0.9750	0.9753	0.9755	0.9750	0.9753	0.9755	0.9750	0.9753
SVC	0.9739	0.9754	0.9735	0.9739	0.9754	0.9735	0.9739	0.9754	0.9735	0.9739	0.9754	0.9735
XGBoost	0.9774	0.9801	0.9735	0.9774	0.9801	0.9735	0.9774	0.9801	0.9735	0.9774	0.9801	0.9735
GRU	0.9735	0.9767	0.9738	0.9735	0.9767	0.9738	0.9735	0.9767	0.9738	0.9735	0.9767	0.9738
LSTM	**0.9756**	**0.9793**	0.9752	**0.9756**	**0.9793**	**0.9752**	**0.9756**	**0.9793**	**0.9752**	**0.9756**	**0.9793**	**0.9752**
TCN	**0.9775**	**0.9759**	**0.9770**	**0.9775**	**0.9759**	**0.9770**	**0.9775**	**0.9759**	**0.9770**	**0.9775**	**0.9759**	**0.9770**
Transformer	0.9763	0.9755	0.9760	0.9763	0.9755	0.9760	0.9763	0.9755	0.9760	0.9763	0.9755	0.9760

SVC also demonstrates adaptability to non-linear data with a test F1 Score of 0.9735, making it a strong candidate for activity recognition in configurations that require ef-

ficient handling of non-overlapping data. The deep learning models GRU and LSTM offer consistent results, with F1 Scores of 0.9738 and 0.9752, respectively, on the test set, highlighting their effectiveness in sequential data processing, even without overlapping segmentation [37,38].

Despite its strengths, the Transformer model performs slightly lower than TCN in this configuration, with an F1 Score of 0.9760 on the test set, suggesting that while its self-attention mechanism is beneficial for long-range dependencies, TCN may be more effective in handling structured, sequential data in non-overlapping setups [29].

In conclusion, the **TCN** model emerges as the best performer in this configuration, demonstrating high accuracy, precision, recall, and F1 Score across all sets. Its superior generalization capabilities highlight its robustness for human activity recognition in non-overlapping data settings. The ensemble models such as RF and XGBoost show commendable results, reinforcing the utility of ensemble learning approaches in enhancing classification performance. Meanwhile, GRU and Transformer also contribute valuable insights, particularly in capturing temporal dependencies, although they slightly trail behind TCN in this context.

4.4.2. Confusion Matrix Analysis for Four-Sensor Non-Overlapping Configuration

The confusion matrices in Figure 12 provide additional insights into the classification performance for each model.

The confusion matrices provide insightful details about the performance of different models in the four-sensor non-overlapping configuration:

DT: The DT confusion matrix in Figure 12a reveals that the model accurately classifies 226 instances of *run_wos* and 220 instances of *stn_ws*. It also shows strong performance in classifying *wlk_ws*, with 127 instances correctly identified. However, there are notable misclassifications, with 36 instances of *stn_ws* incorrectly labeled as *stn_wos*. This misclassification pattern suggests a tendency for the DT model to overfit, especially when dealing with similar stationary activities where sensor readings may overlap [26].

KNN: The KNN confusion matrix in Figure 12b demonstrates strong performance, particularly for the *run_wos* activity, where 229 instances are correctly classified. However, KNN struggles with distinguishing between *stn_ws* and *stn_wos*, as evidenced by the 33 misclassifications in the *stn_ws* class. This confusion likely arises from KNN's sensitivity to feature space proximity, particularly in high-dimensional data [35].

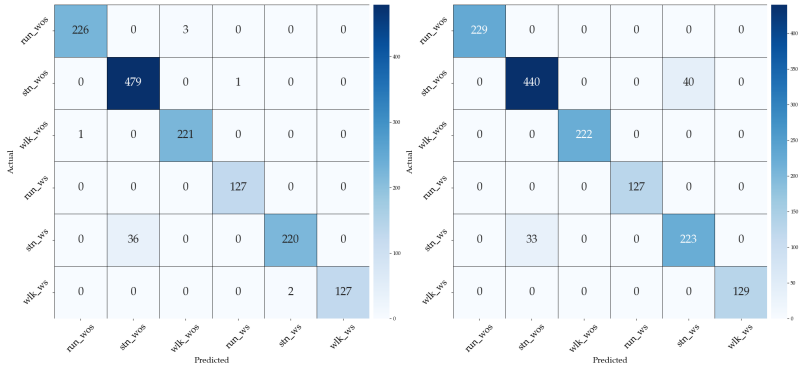
RF: The RF model in Figure 12c performs well, accurately classifying 228 instances of *run_wos* and 480 of *stn_wos*. However, it misclassifies 36 instances of *stn_ws* as *stn_wos*, suggesting limitations in distinguishing subtle temporal features in stationary activities [26]. While effective in managing feature complexity, RF is slightly outperformed by deep learning models in capturing nuanced activity differences.

SVC: The SVC confusion matrix in Figure 12c highlights SVC's high accuracy, particularly in *stn_wos*, where it correctly classifies 476 instances. However, the model struggles with distinguishing between *stn_ws* and *stn_wos*, with 42 instances of *stn_ws* misclassified as *stn_wos* and five instances of *stn_wos* misclassified as *stn_ws*. This confusion could be due to the overlapping nature of sensor readings between these stationary activities [39].

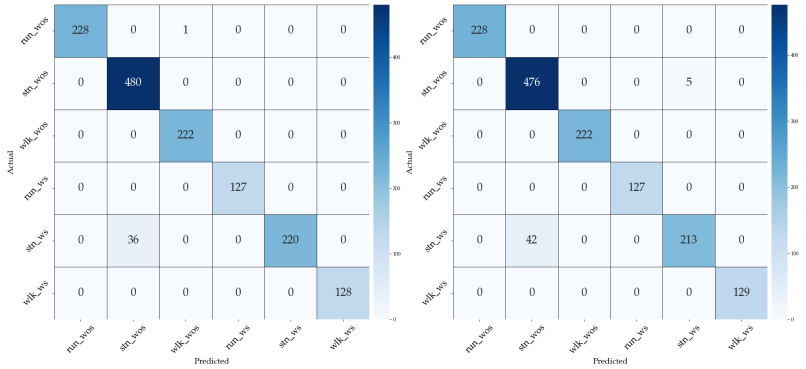
XGBoost: The XGBoost confusion matrix (Figure 12e) shows strong classification of *run_wos* (226 instances) and *stn_wos* (481 instances). However, it struggles with subtle distinctions, misclassifying 40 instances of *stn_ws* as *stn_wos*. While XGBoost's ensemble framework captures complex data patterns, distinguishing closely related stationary activities remains challenging without sequential processing, a strength of temporal models like LSTM and TCN [27].

GRU: The GRU model in Figure 12f shows solid performance, particularly in the classification of *run_wos* with 234 correctly identified instances and *stn_ws* with 211 correct classifications. However, it has minor challenges in distinguishing between *stn_ws* and *stn_wos*, with 31 instances of *stn_ws* misclassified as *stn_wos*. GRU's sequential processing

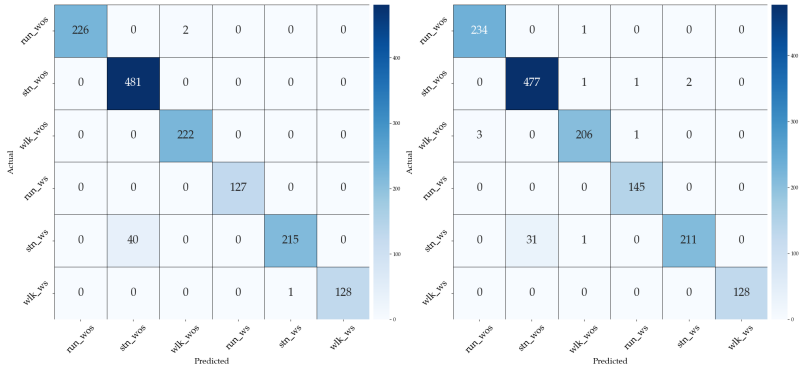
enables it to retain important context over time, aiding in recognizing both stationary and dynamic activities [37].



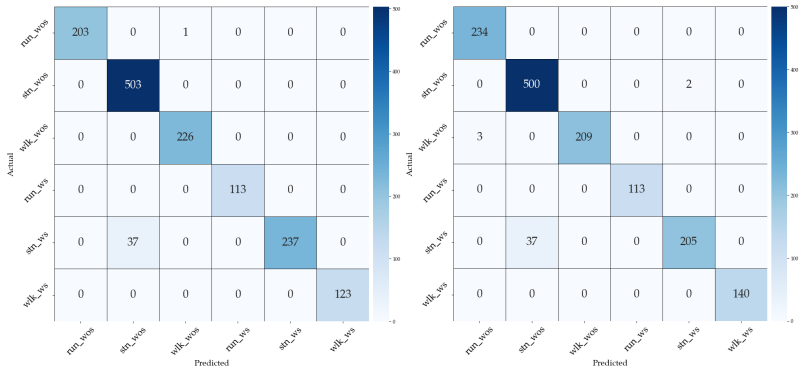
(a) DT: 4 sensors with Non-overlapping (b) KNN: 4 sensors with Non-overlapping



(c) RF: 4 sensors with Non-overlapping (d) SVC: 4 sensors with Non-overlapping



(e) XGBoost: 4 sensors with Non-overlapping (f) GRU: 4 sensors with Non-overlapping



(g) LSTM: 4 sensors with Non-overlapping (h) TCN: 4 sensors with Non-overlapping

Figure 12. *Cont.*



Figure 12. Confusion Matrices for models using a four-sensor configuration with Non-overlapping data segments: (**a**) DT, (**b**) KNN, (**c**) RF, (**d**) SVC, (**e**) XGBoost, (**f**) GRU, (**g**) LSTM, (**h**) TCN, (**i**) Transformer.

LSTM: The LSTM model in Figure 12g shows strong classification performance across various activities. Specifically, it accurately classifies 203 instances of *run_wos* and 503 instances of *stn_wos*. However, the model struggles slightly with *stn_ws*, where it misclassifies 37 instances as *stn_wos*. This pattern highlights LSTM's ability to capture both short-term and long-term dependencies in time-series data, which enhances its robustness across activities [38]. The model also performs well in distinguishing between the *run_ws* and *wlk_ws* classes, achieving high classification accuracy in these categories, further underlining its effectiveness in handling sequential data.

TCN: In Figure 12h, the TCN model shows excellent performance in dynamic activities, accurately classifying 234 instances of *run_wos* and 500 instances of *stn_wos*. TCN's use of dilated convolutions allows it to model both short-term and long-term dependencies, making it ideal for recognizing complex movement patterns in non-overlapping configurations [28]. The TCN model proves to be a strong competitor, performing comparably to LSTM, especially in dynamic classes, with minimal misclassifications.

Transformer: The Transformer model, shown in Figure 12i, demonstrates strong classification performance, leveraging self-attention to effectively capture complex, long-range dependencies [29]. Although it ranks just below TCN and LSTM for non-overlapping data, the model's attention mechanism excels at identifying extended sequences. However, subtle distinctions in stationary activities present a challenge, as seen in the 40 instances of *stn_ws* misclassified as *stn_wos*, likely due to limitations of self-attention in distinguishing closely related activities without temporal continuity [40].

In conclusion, the analysis of the four-sensor non-overlapping configuration underscores **TCN** as the best-performing model overall. The TCN model demonstrates superior accuracy in both dynamic and stationary activities, with minimal misclassifications, thanks to its ability to capture both short-term and long-term dependencies through its dilated convolutional architecture [28]. While the **LSTM** model remains highly competitive and excels in capturing long-term dependencies in stationary activities, TCN's efficiency and robustness across diverse activity types make it the top choice for this configuration. The **Transformer** model, though slightly behind TCN and LSTM, still shows substantial potential by leveraging self-attention to capture long-range dependencies, proving valuable for applications requiring extensive sequential data analysis.

Overall, TCN's strength in handling non-overlapping, dynamic activities with high accuracy and LSTM's capability in managing long-duration dependencies highlight the effectiveness of both models for human activity recognition in scenarios where accurate temporal pattern recognition is essential. A detailed comparative analysis of these models across configurations will be presented in Section 4.5, offering a comprehensive view of their relative advantages and trade-offs.

4.4.3. Four-Sensors: Model Performance with 50% Overlapping Data

In this configuration, four sensors—accelerometer, gyroscope, linear accelerometer, and gravity—are used with a 50% overlapping strategy applied to the data segments. The overlapping segments provide additional continuity and capture transitions between activities more effectively, enhancing the models' ability to recognize patterns across time and improving overall performance.

The evaluation results in Table 19 reveal that the **TCN** model achieves the highest performance across most metrics, with an F1 Score of **0.9762** on the test set. This model demonstrates strong consistency across training, validation, and test sets, underscoring its robustness in handling overlapping configurations for human activity recognition.

DT and KNN models provide reliable baseline performance, with DT achieving an F1 Score of 0.9714 on the test set. However, these simpler models tend to lag behind more advanced architectures, particularly when managing high-dimensional sensor data [33,34].

RF and XGBoost show solid performance. RF achieves a test F1 Score of 0.9741, while XGBoost attains a slightly higher score of 0.9760, reflecting the effectiveness of ensemble learning in capturing complex data interactions [26,27].

SVC also performs well, reaching a test F1 Score of 0.9745. Its ability to handle non-linear relationships makes it a strong contender for activity recognition in overlapping data configurations. The deep learning models GRU and LSTM offer consistent results, with F1 Scores of 0.9762 and 0.9760, respectively, demonstrating their capacity to manage sequential data processing effectively [37,38].

Table 19. Performance comparison of four sensors with 50%-overlapping data for various models, where the performance metrics are Accuracy, Precision, Recall, and F1 Score.

Models	Accuracy			Precision			Recall			F1 Score		
	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test	Train	Validation	Test
DT	0.9752	0.9671	0.9714	0.9752	0.9671	0.9714	0.9752	0.9671	0.9714	0.9752	0.9671	0.9714
KNN	0.9681	0.9725	0.9632	0.9681	0.9725	0.9632	0.9681	0.9725	0.9632	0.9681	0.9725	0.9632
RF	0.9768	0.9692	0.9741	0.9768	0.9692	0.9741	0.9768	0.9692	0.9741	0.9768	0.9692	0.9741
SVC	0.9772	0.9718	0.9745	0.9772	0.9718	0.9745	0.9772	0.9745	0.9718	0.9772	0.9718	0.9745
XGBoost	0.9785	0.9721	0.9760	0.9785	0.9721	0.9760	0.9785	0.9721	0.9760	0.9785	0.9721	0.9760
GRU	0.9756	0.9773	0.9762	0.9756	0.9773	0.9762	0.9756	0.9773	0.9762	0.9756	0.9773	0.9762
LSTM	0.9764	0.9776	0.9760	0.9764	0.9776	0.9760	0.9764	0.9776	0.9760	0.9764	0.9776	0.9760
TCN	**0.9776**	**0.9771**	**0.9762**	**0.9776**	**0.9771**	**0.9762**	**0.9776**	**0.9771**	**0.9762**	**0.9776**	**0.9771**	**0.9762**
Transformer	0.9763	0.9769	0.9761	0.9763	0.9769	0.9761	0.9763	0.9769	0.9761	0.9763	0.9769	0.9761

While the Transformer model shows robust capabilities, achieving an F1 Score of 0.9761, it falls just slightly below the TCN in this configuration. This outcome suggests that while the Transformer's self-attention mechanism is advantageous for long-range dependencies, the TCN may be better suited to handle structured, sequential data in overlapping setups [29].

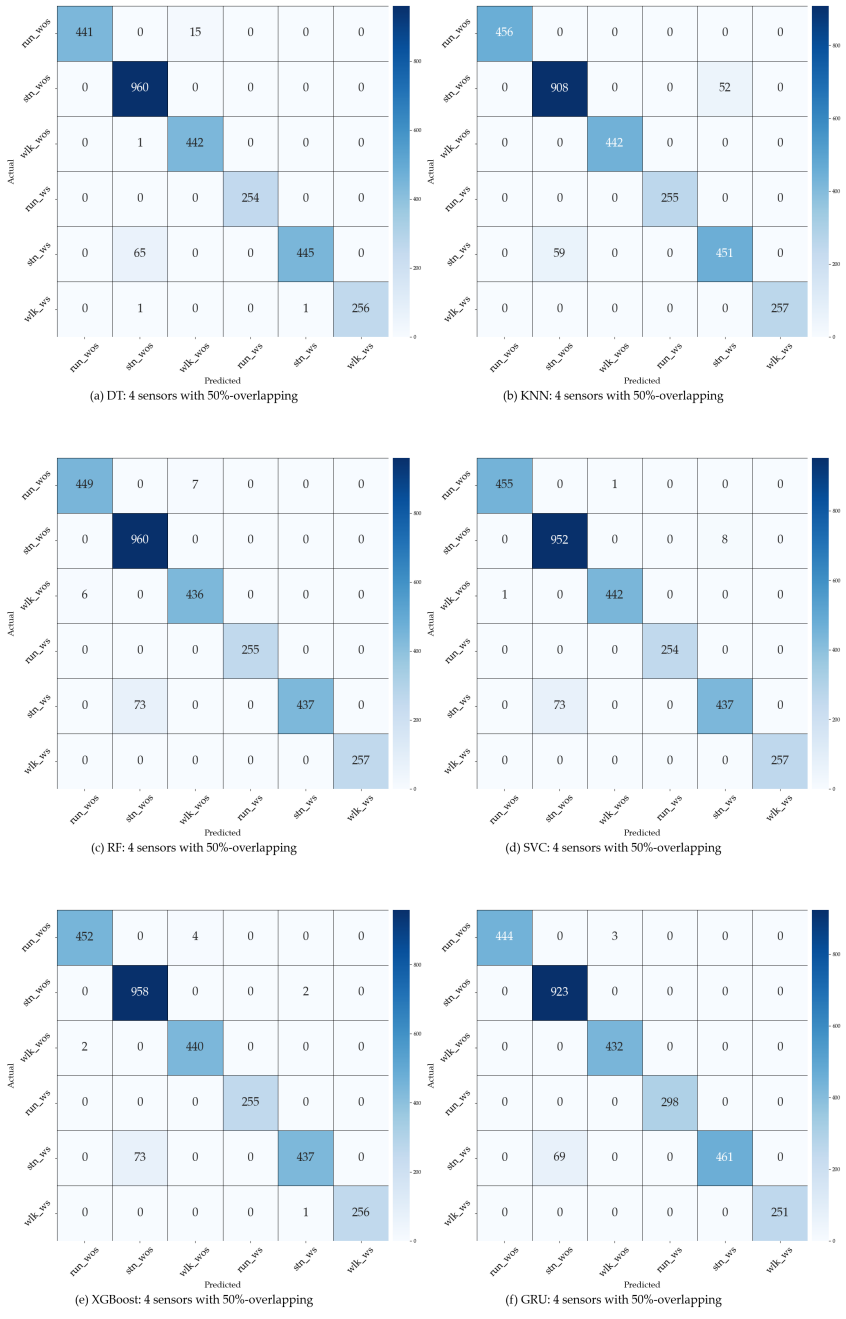
In conclusion, although the TCN and GRU models achieve identical high scores in overall metrics, the TCN model is superior due to its lower misclassification rates revealed in the confusion matrices. This means TCN more effectively distinguishes between similar activities in overlapping data, capturing temporal dependencies and subtle variations better than GRU. Consequently, TCN's enhanced class-wise performance and superior generalization make it the most robust choice for human activity recognition in overlapping data settings, meeting the demands for high accuracy and reliability in practical applications.

4.4.4. Confusion Matrix Analysis for Four-Sensor 50%-Overlapping Configuration

In this configuration, four sensors—accelerometer, gyroscope, linear accelerometer, and gravity—are used with a 50% overlapping strategy applied to the data segments. This method enhances the models' ability to capture contextual transitions between activities, improving overall performance and classification accuracy.

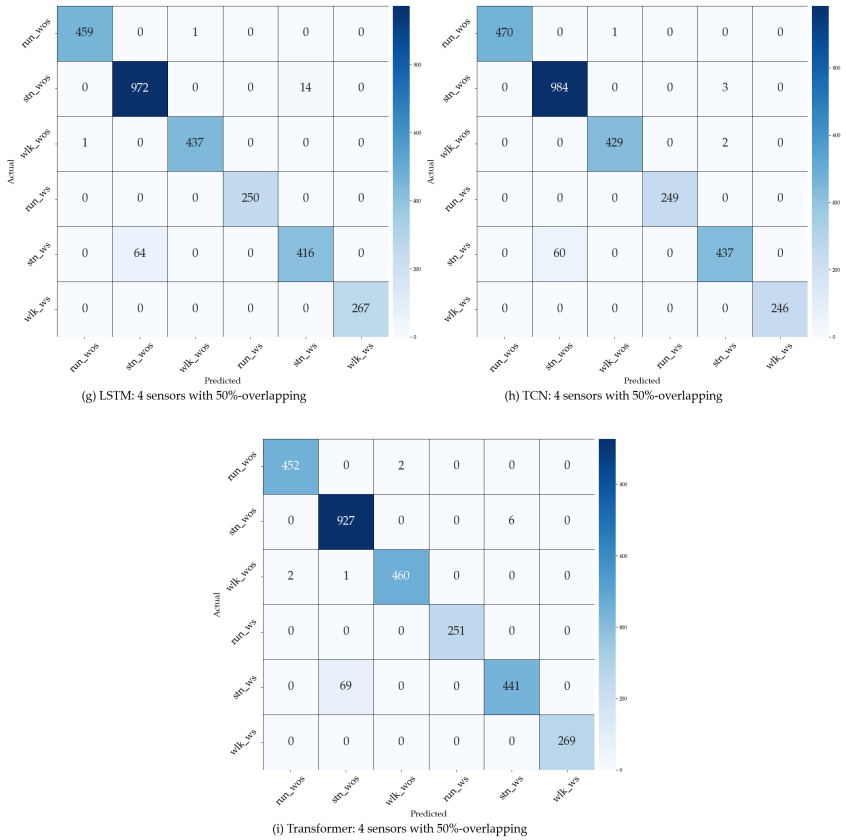
The confusion matrices reveal the following patterns:

DT: The DT confusion matrix in Figure 13a shows accurate classification of 441 instances of *run_wos* and 960 instances of *stn_ws*. However, it struggles with 15 misclassifications for the *run_wos* class and 1 for the *wlk_wos*. These misclassifications suggest that despite the additional overlapping data, DT still faces challenges in accurately distinguishing activities with similar sensor signals [26].



(a) DT: 4 sensors with 50%-overlapping
(b) KNN: 4 sensors with 50%-overlapping
(c) RF: 4 sensors with 50%-overlapping
(d) SVC: 4 sensors with 50%-overlapping
(e) XGBoost: 4 sensors with 50%-overlapping
(f) GRU: 4 sensors with 50%-overlapping

Figure 13. *Cont.*



(g) LSTM: 4 sensors with 50%-overlapping

(h) TCN: 4 sensors with 50%-overlapping

(i) Transformer: 4 sensors with 50%-overlapping

Figure 13. Confusion Matrices for models using a four-sensor configuration with 50% overlapping data segments: (**a**) DT, (**b**) KNN, (**c**) RF, (**d**) SVC, (**e**) XGBoost, (**f**) GRU, (**g**) LSTM, (**h**) TCN, (**i**) Transformer.

KNN: The KNN confusion matrix in Figure 13b indicates improved performance, particularly for the 456 correct classifications of *run_wos* and 442 correct classifications of *wlk_wos*. However, it misclassified 59 instances for *stn_ws*, reflecting KNN's sensitivity to overlapping data and potential difficulty in distinguishing between stationary activities [35].

RF: The RF confusion matrix in Figure 13c shows strong results, with 960 correct classifications for the *stn_ws* class and only seven misclassifications for the *run_wos*. RF's ensemble learning approach enhances its ability to generalize well across various sensor features, making it effective in this configuration, despite its occasional challenges in capturing complex dependencies among sensor inputs [26].

SVC: The SVC confusion matrix in Figure 13d demonstrates strong performance, correctly classifying 455 instances of *run_wos* and 952 instances of *stn_wos*. While generally robust, SVC faces some challenges with stationary classes, misclassifying 73 instances of *stn_ws* as *stn_wos*. This suggests that although SVC effectively handles overlapping sensor data, further refinement could enhance accuracy in distinguishing stationary states [39].

XGBoost: The XGBoost confusion matrix in Figure 13e demonstrates strong performance, with 958 correct classifications for *stn_wos* and 452 for *run_wos*, with only four misclassifications in the latter. However, it misclassifies 73 instances of *stn_ws* as *stn_wos*, highlighting a challenge in distinguishing closely related stationary classes. This indicates XGBoost's effectiveness with complex data interactions, albeit with slight limitations in handling subtle distinctions in stationary activities.

GRU: As shown in Figure 13f, the GRU model performs well, with 923 correct classifications for *stn_wos* and 461 for *stn_ws*. However, it misclassifies 69 instances of *stn_ws* as *stn_wos*, highlighting some difficulty in distinguishing similar stationary activities. Overall,

GRU's ability to capture temporal dependencies makes it effective for most classes, though additional tuning could improve its handling of overlapping sensor data [37].

LSTM: The LSTM confusion matrix in Figure 13g indicates strong performance, particularly with 972 correct classifications for the *stn_wos* class. However, there are some minor misclassifications, including 64 instances of *stn_ws* incorrectly classified as *stn_wos*. Despite these few errors, the LSTM model generally shows robustness across most activity categories, leveraging its ability to capture sequential dependencies effectively [38].

TCN: The TCN confusion matrix in Figure 13h demonstrates high classification accuracy across activities, with 984 correct classifications for the *stn_wos* class. However, there are 60 misclassifications for the *stn_ws* class, indicating a slight limitation in distinguishing between stationary activities. Overall, TCN effectively handles long-term dependencies, underscoring its strength in processing sequential data [28].

Transformer: The Transformer confusion matrix in Figure 13i showcases a strong ability to classify activities, achieving 927 correct classifications for the *stn_wos* class. Despite some misclassifications, its self-attention mechanism effectively captures complex relationships over longer sequences, demonstrating its applicability for activity recognition.

In summary, the 50% overlapping configuration enhances the performance of all models. The TCN stands out as the best performer, achieving the highest accuracy and lowest misclassifications, while the GRU and XGBoost also demonstrate commendable results. These findings underscore the importance of selecting appropriate models based on the specific requirements of human activity recognition tasks, particularly in leveraging overlapping data to enhance classification accuracy.

4.5. Comparative Analysis Across Configurations

This section provides a detailed comparative analysis of the performance metrics for various models across different sensor configurations and data segmentation strategies, focusing on the **test set** results to evaluate the models' generalization capabilities.

4.6. Performance Overview

The analysis reveals that the **Temporal Convolutional Network (TCN)** consistently outperforms other models across all configurations. As summarized in Table 20, TCN achieves the highest scores in accuracy, precision, recall, and F1 Score, reaching up to **0.9770** in the four Sensors Non-overlapping setup. This superior performance is attributed to TCN's ability to capture temporal dependencies effectively through dilated convolutions, making it particularly well-suited for handling multi-sensor data.

Table 20. Overall Performance Comparison of Models Based on Test Set Metrics.

Configuration	Best Model	Accuracy	Precision	Recall	F1 Score
two sensors Non-overlapping	TCN	0.9757	0.9757	0.9757	0.9757
two-sensor 50%-overlapping	TCN	0.9765	0.9765	0.9765	0.9765
four-sensor non-overlapping	TCN	0.9770	0.9770	0.9770	0.9770
4 Sensor 50%-overlapping	TCN	0.9762	0.9762	0.9762	0.9762

Recurrent models like **LSTM** and **GRU** also show strong performance on the test set, particularly in configurations with overlapping data and multiple sensors. For instance, LSTM achieves a test accuracy of **0.9760** in the four-sensor 50%-overlapping setup, closely following TCN and demonstrating its effectiveness in capturing long-term temporal patterns.

Traditional models such as **Random Forest (RF)** and **Support Vector Classifier (SVC)** perform adequately but do not match the performance of deep learning models. For example, RF achieves a test accuracy of **0.9753** in the four-sensor non-overlapping configuration,

while SVC reaches up to **0.9745** in the two-sensor 50%-overlapping setup. This highlights their limitations in handling complex temporal dependencies inherent in ITS applications.

4.7. Detailed Analysis of Model Performance

4.7.1. Two Sensors with Non-Overlapping Data

In this configuration, the TCN achieves a test accuracy of **0.9757**, outperforming other models. LSTM and GRU also perform well, with test accuracies of **0.9738** and **0.9748**, respectively. Traditional models like RF and SVC achieve lower accuracies of **0.9720** and **0.9726**, respectively, indicating challenges in capturing temporal patterns with limited sensor input.

4.7.2. Two Sensors with 50% Overlapping Data

With overlapping data, TCN's performance improves to a test accuracy of **0.9765**. LSTM and GRU also see performance gains, achieving test accuracies of **0.9762** and **0.9750**, respectively. The overlapping data provide additional temporal context, enhancing the models' ability to capture dependencies. Traditional models show modest improvements but remain behind deep learning models.

4.7.3. Four Sensors with Non-Overlapping Data

Increasing the sensor count to four, TCN achieves its highest test accuracy of **0.9770**. LSTM follows with a test accuracy of **0.9752**, and GRU achieves **0.9738**. Traditional models like RF and XGBoost show improved performance due to the additional sensor data, with test accuracies of **0.9753** and **0.9735**, respectively. KNN and DT lag behind, indicating limitations in scaling with more sensors.

4.7.4. Four Sensors with 50% Overlapping Data

In this configuration, both the TCN and GRU models achieve a high test accuracy of **0.9762**. However, analysis of their confusion matrices reveals that the TCN model outperforms GRU in terms of misclassification rates across all activity classes.

Confusion Matrix Insights:

- TCN Model: Fewer misclassifications overall, particularly in challenging stationary activities, indicating better discrimination of subtle patterns.
- GRU Model: More misclassifications in certain classes, suggesting difficulties in distinguishing similar activities with overlapping data.

Why TCN is the Best Model:

- Superior Class-wise Accuracy: TCN minimizes misclassifications across all activities, enhancing robustness and reliability critical for ITS applications.
- Effective with Overlapping Data: TCN's dilated causal convolutions capture long-term dependencies more effectively than GRU.
- Better at Distinguishing Similar Activities: TCN excels in identifying subtle differences, reducing misinterpretation risks in real-world scenarios.

Conclusion: Despite identical overall metrics, the **TCN model is the superior choice** due to its lower misclassification rates and better class-wise performance. This underscores the importance of evaluating models beyond aggregate metrics to ensure reliable and effective HAR systems in ITS applications.

The **LSTM** model also performs well with a test accuracy of **0.9760** but does not surpass TCN in class-wise accuracy. Traditional models show improvement but still do not match the performance of deep learning models.

4.8. Impact of Overlapping Data and Sensor Count

The inclusion of overlapping data and additional sensors generally enhances model performance on the test set:

- **Overlapping Data**: Provides additional temporal context, improving the models' ability to capture dependencies. Deep learning models like TCN and LSTM benefit significantly, as evidenced by their higher test set metrics in overlapping configurations.
- **Increased Sensor Count**: Offers more comprehensive data, allowing models to learn from richer features and improving classification capabilities on unseen data. TCN and LSTM benefit greatly from additional sensors, capturing complex spatiotemporal patterns more effectively.

However, in configurations with four sensors and overlapping data, the performance of TCN slightly decreases, suggesting that excessive redundancy may not always contribute to better generalization and might introduce overfitting.

4.9. Comparison Based on Performance Metrics

- **Accuracy**: TCN consistently achieves the highest test set accuracy across most configurations, indicating robust overall performance and generalization capability.
- **Precision**: High precision scores by TCN (up to **0.9770**) reflect its effectiveness in minimizing false positives, crucial in ITS to avoid incorrect detections.
- **Recall**: TCN's high recall values demonstrate its ability to identify true positives reliably, ensuring critical events are not missed.
- **F1 Score**: The balanced F1 Scores suggest that TCN maintains a good trade-off between precision and recall on unseen data, essential for reliable decision-making in ITS applications.

4.10. Recommendation

Based on the test set analysis, the **TCN in the four-sensor non-overlapping configuration** emerges as the most suitable model for ITS applications, achieving the highest test set performance metrics. The advantages of this configuration include:

- **Enhanced Temporal Modeling**: TCN's architecture effectively captures both short- and long-term dependencies in the data, even on unseen data.
- **Improved Contextual Understanding**: The use of multiple sensors provides richer context, enabling better interpretation of dynamic environments in real-world scenarios.
- **Optimal Data Utilization**: Non-overlapping data with multiple sensors avoids redundancy, enhancing generalization and preventing potential overfitting.
- **Robustness and Scalability**: High test set performance indicates that the model is robust and can generalize well, increasing adaptability to different ITS challenges.

5. Conclusions and Future Works

5.1. Conclusions

This study makes significant contributions to the field of Human Activity Recognition (HAR) within Intelligent Transportation Systems (ITS) by effectively classifying passenger activities into six distinct categories: walking, running, and stationary (both while using and not using the smartphone). These insights hold substantial implications for optimizing smart transportation systems, as they can enhance decision-making processes for drivers and improve the overall commuter experience. By leveraging these findings, the study lays a foundation for the future development of smart transportation systems aimed at making public transit more efficient and user-friendly, thereby encouraging greater adoption of public transportation.

In this investigation, a comprehensive comparison was conducted between traditional machine learning models and advanced deep learning techniques. The models evaluated included Decision Tree (DT), K-Nearest Neighbors (KNN), Random Forest (RF), Support Vector Classifier (SVC), XGBoost, Gated Recurrent Unit (GRU), Long Short-Term Memory (LSTM) networks, Temporal Convolutional Networks (TCN), and Transformer models. The evaluation encompassed various data segmentation strategies, including both non-overlapping and 50% overlapping configurations, utilizing smartphone sensor data. The notable performance enhancements observed with overlapping data segments,

particularly for deep learning models, underscore the advantages of integrating temporal continuity into activity recognition tasks.

The results reveal that employing four sensors significantly elevates model performance compared to two-sensor configurations. The implementation of 50% overlapping data segments further augmented the models' generalization capabilities. Among the models evaluated, deep learning models—specifically LSTM, TCN, GRU, and Transformer—demonstrated superior outcomes. The TCN model exhibited exceptional performance in the four-sensor non-overlapping configuration, achieving the highest accuracy of 97.70%. In the same configuration, the LSTM and GRU models also achieved high accuracies of 97.52% and 97.38%, respectively, reinforcing their strong capacity for capturing both short-term and long-term dependencies in time-series data. The Transformer model also showed robust performance, highlighting the effectiveness of attention mechanisms in capturing temporal patterns.

Traditional machine learning models such as KNN, DT, RF, and SVC showed reasonable performance but consistently underperformed compared to deep learning models, especially in complex scenarios characterized by multi-sensor inputs and overlapping data segments. Ensemble models like RF and XGBoost improved upon simpler models by leveraging multiple decision trees and gradient boosting techniques, respectively. However, they still fell short in capturing the intricate temporal dependencies inherent in sequential sensor data, which deep learning models managed more effectively.

Moreover, the TCN model displayed notable performance across various configurations, particularly in handling the intricacies of overlapping sequences. Its architecture, leveraging dilated convolutions, facilitated the capture of temporal patterns over extended periods, making it an effective alternative to recurrent models like LSTM and GRU in certain scenarios. These findings accentuate the increasing significance of deep learning models, particularly TCN and LSTM, in real-time, context-aware ITS applications, such as driver monitoring and traffic condition prediction. This research not only highlights the strengths of deep learning methodologies in HAR tasks but also advocates for their broader application in advancing smart transportation systems [28,41].

5.2. Future Work

Future research directions include exploring real-time implementation of the proposed models within ITS environments to evaluate their practical performance and computational efficiency. Additionally, incorporating a broader range of activities and more diverse user populations can enhance the generalizability of the models. Investigating sensor fusion techniques to optimally combine data from multiple sensors and exploring energy-efficient model architectures suitable for deployment on resource-constrained devices are also promising avenues. Finally, integrating explainable AI techniques can enhance the transparency of deep learning models, facilitating their acceptance in safety-critical ITS applications.

The findings of this study open several promising avenues for future research aimed at enhancing the role of machine learning and deep learning in Intelligent Transportation Systems (ITS):

1. Hybrid Model Development: Exploring hybrid models that integrate traditional machine learning techniques with advanced deep learning architectures such as LSTM, TCN, GRU, and Transformer. Such models could leverage the interpretability and efficiency of traditional methods alongside the sophisticated pattern recognition capabilities of deep learning, potentially yielding improved performance in dynamic ITS environments, particularly in real-time applications [42,43].
2. Customized Preprocessing Techniques: Investigating the customization of preprocessing strategies tailored to specific ITS applications. Traffic monitoring and driver behavior analysis may benefit from domain-specific data preprocessing that accounts for unique noise characteristics and data variability inherent in various sensor inputs.

These refinements are likely to enhance model accuracy and adaptability in real-time conditions, enabling better detection of critical activities [44,45].

3. Personalized Models through Transfer Learning: Developing personalized models that adapt to individual driving patterns and environmental contexts using techniques like transfer learning. This approach could fine-tune models to cater to individual users, enhancing the precision of activity recognition and making systems more context-aware. Personalized models would significantly improve the relevance and accuracy of predictions in tailored transportation applications, addressing diverse user needs effectively [46].
4. Integration of Multimodal Data Sources: Extending the research by combining smartphone sensor data with inputs from vehicle sensors, environmental monitoring systems, and other relevant technologies. Integrating multimodal data can lead to richer datasets and a deeper understanding of transportation behaviors and conditions, ultimately fostering more intelligent and responsive ITS solutions [28,41].
5. Real-Time Implementation and Optimization: Exploring the real-time implementation of the proposed models within ITS environments to evaluate their practical performance and computational efficiency. Future research should focus on addressing computational constraints and latency issues associated with real-time ITS applications. This includes optimizing models for scalability and ensuring efficient performance in live environments, where response times and computational resources are often limited [39].
6. Energy-Efficient Architectures: Investigating energy-efficient model architectures suitable for deployment on resource-constrained devices like smartphones and embedded systems. This is crucial for practical ITS applications where power consumption can be a limiting factor.
7. Explainable AI Techniques: Integrating explainable AI methods to enhance the transparency and interpretability of deep learning models. This can facilitate their acceptance in safety-critical ITS applications by providing insights into model decision-making processes.
8. Expanded Activity Sets and Diverse Populations: Incorporating a broader range of activities and more diverse user populations to enhance the generalizability of the models. This can improve the robustness of HAR systems across different demographic groups and usage scenarios.

5.3. Recommendation

Based on the comprehensive analysis and performance metrics evaluated across different configurations, the Temporal Convolutional Network (TCN) in the four-sensor non-overlapping configuration emerges as the most suitable model for ITS applications. This recommendation is grounded in its superior accuracy, precision, recall, and F1 Score, as detailed in Table 20.

The advantages of this configuration include:

Enhanced Temporal Modeling: TCN's architecture effectively captures both short- and long-term dependencies in the data, even on unseen data, due to its use of dilated causal convolutions [28]. Improved Contextual Understanding: The use of multiple sensors provides richer context, enabling better interpretation of dynamic environments in real-world scenarios. Optimal Data Utilization: Non-overlapping data with multiple sensors avoids redundancy, enhancing generalization and preventing potential overfitting. Robustness and Scalability: High test set performance indicates that the model is robust and can generalize well, increasing adaptability to different ITS challenges. This recommendation aligns with the objectives of ITS, which require reliable and accurate real-time analysis of multi-sensor data to improve transportation efficiency and safety.

5.4. Summary of Findings

In conclusion, the Temporal Convolutional Network stands out as the optimal model for Intelligent Transportation Systems, particularly in the four-sensor non-overlapping configuration. Its superior performance across all key metrics on the test set makes it highly recommended for applications that require dependable and precise real-time analysis of multi-sensor data.

Author Contributions: Conceptualization was carried out by J.B.L.B., A.T. and H.M.; data collection, source code development, and data analysis were performed by J.B.L.B., B.M.P. and U.T.; J.B.L.B. led the visualization of the results and the writing of the original draft of the manuscript; all authors contributed to data analysis and manuscript preparation, with A.T. critically reviewing the manuscript and providing valuable feedback for improvement. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ranasinghe, S.; Al Machot, F.; Mayr, H.C. A review on applications of activity recognition systems with regard to performance and evaluation. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 1550147716665520. [CrossRef]
2. Golestani, N.; Moghaddam, M. Human activity recognition using magnetic induction-based motion signals and deep recurrent neural networks. *Nat. Commun.* **2020**, *11*, 1551. [CrossRef] [PubMed]
3. Ahad, M.A.R.; Antar, A.D.; Shahid, O. Vision-based Action Understanding for Assistive Healthcare: A Short Review. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops 2019, Long Beach, CA, USA, 15–21 June 2019; pp. 1–11.
4. Shao, L.; Han, J.; Xu, D.; Shotton, J. Computer vision for RGB-D sensors: Kinect and its applications [special issue intro.]. *IEEE Trans. Cybern.* **2013**, *43*, 1314–1317. [CrossRef] [PubMed]
5. Dubois, A.; Charpillet, F. Human activities recognition with RGB-Depth camera using HMM. In Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS, Osaka, Japan, 3–7 July 2013; pp. 4666–4669. [CrossRef]
6. Martínez-Ballesté, A.; Pérez-Martínez, P.A.; Solanas, A. The pursuit of citizens' privacy: A privacy-aware smart city is possible. *IEEE Commun. Mag.* **2013**, *51*, 136–141. [CrossRef]
7. Palimote, J.; Georgewill, O.; Atu, L. A Deep Learning Framework for Human Activity Recognition Using Smartphone Data. *Int. J. Adv. Res. Comput. Commun. Eng.* **2021**, *10*, 1–7. [CrossRef]
8. Köping, L.; Shirahama, K.; Grzegorzek, M. A general framework for sensor-based human activity recognition. *Comput. Biol. Med.* **2018**, *95*, 248–260. [CrossRef]
9. Jahangiri, A.; Rakha, H.A. Applying machine learning techniques to transportation mode recognition using mobile phone sensor data. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 2406–2417. [CrossRef]
10. Lopez, A.J.; Semanjski, I.; Gautama, S.; Ochoa, D. Assessment of smartphone positioning data quality in the scope of citizen science contributions. *Mob. Inf. Syst.* **2017**, *2017*, 4043237. [CrossRef]
11. Garcia-Gonzalez, D.; Rivero, D.; Fernandez-Blanco, E.; Luaces, M.R. A public domain dataset for real-life human activity recognition using smartphone sensors. *Sensors* **2020**, *20*, 2200. [CrossRef]
12. Singh, D.; Merdivan, E.; Psychoula, I.; Kropf, J.; Hanke, S.; Geist, M.; Holzinger, A. Human Activity Recognition Using Recurrent Neural Networks. In *Machine Learning and Knowledge Extraction: Proceedings of the IFIP TC 5, WG. 8.4, 8.9, 12.9 International Cross-Domain Conference, CD-MAKE 2017, Reggio, Italy, 29 August–1 September 2017*; Springer International Publishing: Cham, Switzerland, 2017; pp. 267–274. [CrossRef]
13. Mekruksavanich, S.; Jitpattanakul, A. Lstm networks using smartphone data for sensor-based human activity recognition in smart homes. *Sensors* **2021**, *21*, 1636. [CrossRef]
14. Kumar, P.; Chauhan, S. Human activity recognition with deep learning: Overview, challenges & possibilities. *CCF Trans. Pervasive Comput. Interact.* **2021**, *3*, 339.
15. Sousa Lima, W.; Souto, E.; El-Khatib, K.; Jalali, R.; Gama, J. Human activity recognition using inertial sensors in a smartphone: An overview. *Sensors* **2019**, *19*, 3213. [CrossRef] [PubMed]

16. Ferrari, A.; Micucci, D.; Mobilio, M.; Napoletano, P. Trends in human activity recognition using smartphones. *J. Reliab. Intell. Environ.* **2021**, *7*, 189–213. [CrossRef]
17. Nweke, H.F.; Teh, Y.W.; Mujtaba, G.; Al-Garadi, M.A. Data fusion and multiple classifier systems for human activity detection and health monitoring: Review and open research directions. *Inf. Fusion* **2019**, *46*, 147–170. [CrossRef]
18. Ouyang, X.; Shuai, X.; Zhou, J.; Shi, I.W.; Xie, Z.; Xing, G.; Huang, J. Cosmo: Contrastive Fusion Learning with Small Data for Multimodal Human Activity Recognition. In Proceedings of the 28th Annual International Conference on Mobile Computing and Networking (MobiCom), Sydney, Australia, 17–21 October 2022.
19. Siargkas, C.; Papapanagiotou, V.; Delopoulos, A. Transportation Mode Recognition Based on Low-Rate Acceleration and Location Signals with an Attention-Based Multiple-Instance Learning Network. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 14376–14388. [CrossRef]
20. Yuan, G.; Wang, Z.; Meng, F.; Yan, Q.; Xia, S. An overview of human activity recognition based on smartphone. *Sens. Rev.* **2019**, *39*, 288–306. [CrossRef]
21. Yu, T.; Jin, H.; Nahrstedt, K. ShoesLoc: In-Shoe Force Sensor-Based Indoor Walking Path Tracking. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)* **2019**, *3*, 1–22. [CrossRef]
22. Hong, Z.; Li, Z.; Zhong, S.; Lyu, W.; Wang, H.; Ding, Y.; He, T.; Zhang, D. CrossHAR: Generalizing Cross-dataset Human Activity Recognition via Hierarchical Self-Supervised Pretraining. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol. (IMWUT)* **2024**, *8*, 1–34. [CrossRef]
23. Rahn, V.X.; Zhou, L.; Klieme, E.; Arnrich, B. Optimal Sensor Placement for Human Activity Recognition with a Minimal Smartphone-IMU Setup. In Proceedings of the 10th International Conference on Sensor Networks—Volume 1: SENSORNETS, Online, 9–10 February 2021; pp. 37–48.
24. Barua, A.; Jiang, X.; Fuller, D. The effectiveness of simple heuristic features in sensor orientation and placement problems in human activity recognition using a single smartphone accelerometer. *BioMedical Eng. OnLine* **2024**, *23*, 21. [CrossRef]
25. Ho, T.K. Random decision forests. In Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, Canada, 14–16 August 1995; Volume 1, pp. 278–282.
26. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
27. Chen, T.; Guestrin, C. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
28. Bai, S.; Kolter, J.Z.; Koltun, V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv* **2018**, arXiv:1803.01271.
29. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*, 1–11.
30. BenAbdelkader, C.; Cutler, R.; Davis, L. View-invariant estimation of height and stride for gait recognition. In *Biometric Authentication: Proceedings of the International ECCV 2002 Workshop, Copenhagen, Denmark, 1 June 2002*; Proceedings 1; Springer: Berlin/Heidelberg, Germany, 2002; pp. 155–167.
31. Zheng, Y.; Liu, Q.; Chen, E.; Ge, Y.; Zhao, J.L. Time series classification using multi-channels deep convolutional neural networks. *Int. J. Data Warehous. Min.* **2014**, *10*, 13–24.
32. Varma, M.; Simon, H. Bias-variance trade-off for zero-error and bounded-error learning models. In Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 4–7 December 2006; pp. 1217–1224.
33. Breiman, L. *Classification and Regression Trees*; Routledge: Abingdon-on-Thames, UK, 1984.
34. Cover, T.; Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **1967**, *13*, 21–27. [CrossRef]
35. Altman, N.S. An introduction to kernel and nearest-neighbor nonparametric regression. *Am. Stat.* **1992**, *46*, 175–185. [CrossRef]
36. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: New York, NY, USA, 1995.
37. Cho, K.; Van Merrienboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1724–1734.
38. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
39. Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297.
40. Lin, S.; Wang, Y.; Liu, Y.; Qiu, X.; Huang, X. A Survey of Transformers. *arXiv* **2021**, arXiv:2106.04554. [CrossRef]
41. Lea, C.; Flynn, M.D.; Vidal, R.; Reiter, A.; Hager, G.D. Temporal convolutional networks for action segmentation and detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 156–165.
42. Hammerla, N.Y.; Halloran, S.; Plötz, T. Deep, convolutional, and recurrent models for human activity recognition using wearables. *arXiv* **2016**, arXiv:1604.08880.
43. Ordóñez, F.J.; Roggen, D. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* **2016**, *16*, 115. [CrossRef]
44. Ronao, C.A.; Cho, S.B. Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Syst. Appl.* **2016**, *59*, 235–244. [CrossRef]

45. Ravi, D.; Wong, C.; Lo, B.; Yang, G.Z. Deep learning for human activity recognition: A resource efficient implementation on low-power devices. In Proceedings of the 2016 IEEE 13th International Conference on Wearable and Implantable Body Sensor Networks (BSN), San Francisco, CA, USA, 14–17 June 2016; pp. 71–76.
46. van den Oord, A.; Dieleman, S.; Zen, H.; Simonyan, K.; Vinyals, O.; Graves, A.; Kalchbrenner, N.; Senior, A.; Kavukcuoglu, K. WaveNet: A Generative Model for Raw Audio. *arXiv* **2016**, arXiv:1609.03499.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.