Engineered Science
DOI: https://dx.doi.org/10.30919/es1207

Performance Analysis of Machine Learning Models for Human Activity Classification

Anshul Sheoran,[1, 2] Ritu Boora,[1,*] Manisha Jangra[1] and Camilo E. Valderrama[2]

Abstract

One of the ultimate goals of machine learning is to design an application that can accurately understand 'humans' actions and intentions to serve us better. Hence, this research paper aims to explore the application of machine learning algorithms in human activity classification. It focuses on evaluating the performance of some popular machine learning methods, namely *K*-NN, Support Vector Machine (SVM), Logistic Regression and XGBoost algorithm, for classifying human activities. Hyperparameters of algorithms are tuned to find the optimized result and performance. The performance of *K*-NN is evaluated over a range of *K*, while the logistic regression is implemented with various solvers and penalty functions. Further, SVM is tested for multiple linear and non-linear kernel functions while the XGBoost algorithm is implemented with different learning rates and trees. On performance comparison, it is observed that all four Machine Learning Models have encountered ambiguity in identifying similar activities. Further, when implemented with a polynomial kernel, the SVM method outperformed other state-of-the-art techniques, achieving an accuracy of 96.9% and obtaining peak values of precision and F1-score. At the same time, Logistic Regression failed to classify several activities, leading to the lowest accuracy.

Keywords: Machine learning models; Human activity recognition; Sensor data; *K*-NN; Support vector machine (SVM); XG Boost; Logistic regression; Data pre-processing.
Received: 02 December 2023; Revised: 30 June 2024; Accepted: 11 July 2024.
Article type: Research article.

1. Introduction

Human Activity Recognition (HAR) is a research field that focuses on developing computational methods and algorithms to automatically identify human actions and behaviours based on sensor data. It has the potential to revolutionize human life. In HAR, the data collected from sensors is interpreted and analyzed accurately to identify and interpret human activities. The sensors may be wearable devices, cameras, or environmental sensing equipment. HAR has diverse application areas, including healthcare, sports, surveillance, human-computer interaction, gaming, smart environments, and smart homes.

In healthcare, it can be used for patient monitoring and detecting abnormal activities in individuals with mental disorders. HAR aids athletes by enabling a data-driven training and performance approach, leading to improved outcomes and reduced injuries in sports. The smart environment facilitates seamless automation and personalized control of various homedevices and systems, enhancing convenience, comfort, and energy efficiency. By employingsophisticated machine learning algorithms, these systems can identify abnormal behaviors orpotential threats in real time, enabling proactive security measures and timely intervention.

[1] Department of Electrical & Electronics Engineering, Guru Jambheswar University of Science and Technology, Hisar 125001, India.
[2] Department of Applied Computer Science, University of Winnipeg, Winnipeg, R3B 2E9, MB, Canada.
*Email: rituboora@gmail.com (R. Boora)

Hayat *et al*.[1] conducted a comprehensive study on various machine learning and deep learning techniques for activity recognition. The Long Short-Term Memory Network achieved the highest accuracy at 95.04%, while the Support Vector Machine performed well with 89.07% accuracy and minimal computational time, making it a practical choice for elderly activity monitoring. In a paper,[2] the author presents a novel approach for human activity recognition using Convolutional Neural Networks (C-NN) and Bidirectional Gated Recurrent Units (Bi-GRU). A Bi-GRU model is introduced to capture temporal motion patterns in the video sequences by processing the selected deep features in both forward and backward directions. The proposed technique is evaluated on real-world human activity recognition datasets and demonstrates superior performance compared to existing methods.

In the author utilizes built-in sensors in smart mobile devices for HAR[3]. It explores the performance of two algorithms, Random Forest (RF) and Modified Random Forest (MRF), in an online activity recognition system running on Android platforms. The Modified Random Forest approach reduces the computational complexity of the base version by creating smaller training units for each activity, improving classification efficiency. The study evaluates the classifiers' performance in recognizing human movements in real time, considering limited training data and memory constraints on smart devices. In literature,[4] wearable sensor-based human activity recognition (HAR) in pervasive computing is surveyed. Its significance is emphasized, a system architecture is shown, a two-level taxonomy is introduced, important issues are addressed, and 28 systems are evaluated. Whereas a detailed analysis of HAR from 2011 to 2021 is given in paper,[5] with an emphasis on its three pillars: devices, AI, and applications. The importance of HAR in the healthcare industry, the demand for better hybrid AI models, and the paucity of research in abnormality detection and action forecasting are all highlighted. According to the study, these three pillars will advance together as the HAR sector changes, with AI being essential to its continued development.

From the literature, activity recognition has seen tremendous growth over the last decade. However, despite significant research on human activity recognition, existing techniques quite often misclassify human actions due to several challenges, such as:

Ambiguous Interpretability: A single body movement can be associated with multiple activities. For instance, distinguishing between dancing, jumping, and playing based on similar movements is difficult. Similarly, activities like walking and jogging can have similar patterns, making differentiation difficult. Developing advanced algorithms that analyze subtle differences and contextual cues is crucial for accurate classification.[6]

Diverse Patterns: Activity diversity refers to variations in how different users perform the same activity, and even within the same user, activities can differ due to factors such as stress, fatigue, emotional state, and environmental conditions. Additionally, multiple activities often exhibit similarities, leading to confusion in recognition.[7] These factors significantly impact recognition accuracy but are often neglected in research.

Simultaneous activities: A common challenge in activity recognition is detecting concurrent or coincidental activities, like when a person is having dinner and watchingtelevision simultaneously.[8] These activities need special algorithms for HAR classification.

Interleaved Actions: In real-life situations, it is feasible for certain actions to be interleaved. These cases are problematic and increase the complexity of the system.[9]

For the real-world application of HAR, there is a scope for highly optimized and computationally efficient algorithms. This gives a scope for further development of these algorithms for seamless integration of HAR into diverse domains. This research delves into using machine learning to classify human activities, testing popular algorithms like *K*-NN, SVM, Logistic Regression, and XGBoost, by fine-tuning these methods to optimize accuracy. Our study addresses gaps by focusing on interpreting subtle differences in actions, handling diverse activity patterns, recognizing simultaneous activities, and dealing with interleaved actions.

The structure of this study is as follows: The second section outlines the experimental setup and modeling approach. The presentation of results, which follows this and discussione presentation of results and discussion. Finally, the paper concludes with key insights and implications drawn from the study.

2. Experimental and modelling

In this paper, machine learning-based techniques were implemented and their performance is analyzed in the context of human activity recognition. Based on results the best performing machine learning model is proposed. The machine learning-based HAR system is implemented in the following steps: data collection, data pre-processing, feature extraction & and reduction, and classification model (Fig. 1). Here, each step plays a pivotal role in achieving precise activity recognition. Fig. 1 provides the outlook of the experimental methodology adopted for the proposed work. The following subsections provide a detailed theoretical background of the

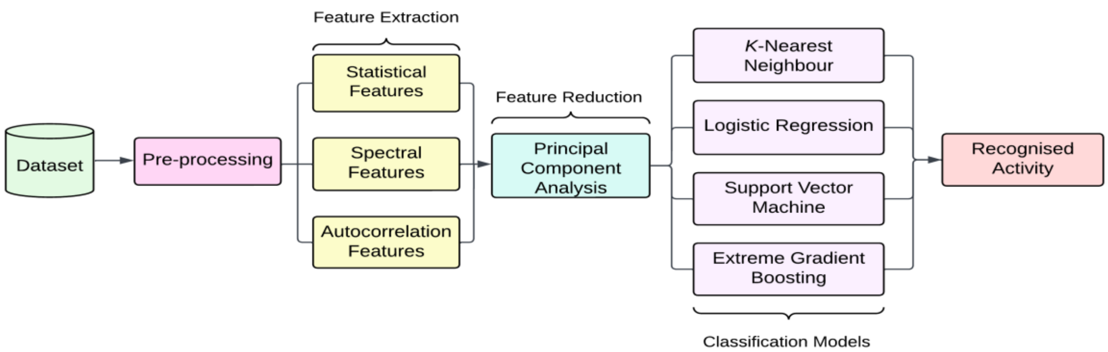

Fig. 1 Flow diagram of HAR.

dataset, pre-processing, feature extraction and selection techniques and classifier models. Further, the performance measures are also discussed in this section. This section provides details of the experimental methodology adopted and a discussion on the implementation of these steps for the proposed work.

2.1 Data collection

This study utilizes a benchmark dataset[10–12] for Human Activity Recognition. The dataset consists of sensor data captured by five MTx 3-DOF (Degree of Freedom) orientation trackers. Each MTx unit incorporates a magnetometer, gyroscope, and tri-axial accelerometerto measure changes in body orientation in three-dimensional space. The sensor units are connected to an Xbus Master device that transmits the collected data to a receiver via Bluetooth™ connection.

The data was collected from eight persons, including four females and four males; each subject performed each activity for five minutes. The data was collected at a sampling frequency of 25 Hz using five MTx sensor units placed on each person's right arm, left arm, right leg, torso, and left leg. It was recorded for a total of nine signals in the following order: acceleration along the x, y, and z axes; rate of turn alongthe x, y, and z axes; and Earth's magnetic field along the x, y, and z axes.

The participants performed 19 activities namely sitting(Z1), standing(Z2), lying on their back(Z3) and lying on the right side(Z4), ascending(Z5) and descending stairs(Z6), standing in a still(Z7) and moving elevator(Z8), walking in a parking lot(Z9), walking on a treadmill with flat(Z10) and inclined positions(Z11), running on a treadmill(Z12), exercising on a stepper(Z13), exercising on a cross trainer(Z14), cycling on an exercise bike in horizontal(Z15) and vertical positions(Z16), rowing(Z17), jumping(Z18) , and playing basketball(Z19). Hence, in this dataset, we have collected 45 recordings for each subject and activity. These recordings are obtained from 5 sensor units, each equipped with three tri-axial sensing devices. The collected data was organized and structured according to the sensor units and activities performed, providing a comprehensive dataset for further analysis and evaluationin the field of Human Activity Recognition.

2.2 Pre-processing

In data pre-processing, the raw data is transformed and organized to make it suitable for modeling and training purposes. Since each recording has a duration of 5 minutes, it resultsin 7500 (5 ×60 × 25) samples at a 25Hz sampling rate and a raw dataset of size 7500 × 45. Anotherimportant step in pre-processing is segmentation, which involves dividing the data into smallerframes or segments.

For this study, the entire signal is segmented into frames of 5 seconds, resulting in eachsegment with a discrete sequence of N_S samples, where N_S is 125. The resulting segmented dataset comprises 45 signals for each subject performing a specific activity. Therefore, the dimension of each segment per subject per activity is represented as 125 × 45.

2.3 Feature extraction and reduction

Feature extraction means selecting and transforming organized sensor data into meaningful and representative features. The goal is to create a more compact and informative representation of the data, subsequently improving efficiency and accuracy in activity recognition. Different feature extraction techniques are implemented on each segmented data to extract the respective segment's statistical, spectral, and autocorrelation features.

Let $x_i = [x_{1i}, x_{2i}, x_{3i}, x_{4i}, \ldots \ldots . . x_{Nsi}]^T$ be the i^{th} segment of the signal $\boldsymbol{x}$.

2.4 Statistical feature extraction

Statistical Features are used to extract the numerical features of the dataset; they are used to capture complex patterns and relationships in data.[13] In this work, we have obtainedfive statistical features: variance, mean value, skewness, kurtosis, and minimum and maximum values. These are represented by Equations (1-4) where x_{ji} is the j^{th} term of signal $\boldsymbol{x_i}$ while $E\{\cdot\}$ is the expectation operator. These five features were extracted from each of the 45 signals per segment per activity (45 × 5=225 features/activity/segment) and cascaded to form a column vector of size 225×1 per segment.

$$Mean(\boldsymbol{x_i}) = E\{\boldsymbol{s}\} = \mu_{\boldsymbol{x}} = \frac{1}{N_s}\sum_{j=1}^{N_s} x_{ji} \tag{1}$$

$$Variance(\boldsymbol{x_i}) = \sigma^2 = E\{(\boldsymbol{s} - \mu_{\boldsymbol{x}})^2\} = \frac{1}{N_s}\sum_{j=1}^{N_s}\left(x_{ji} - \mu_{\boldsymbol{x}}\right)^2 \tag{2}$$

$$skewness(\boldsymbol{x_i}) = \frac{E\{(\boldsymbol{s}-\mu_{\boldsymbol{x}})^3\}}{\sigma^3} = \frac{1}{N_s\sigma^3}\sum_{j=1}^{N_s}\left(x_{ji} - \mu_{\boldsymbol{x}}\right)^3 \tag{3}$$

$$Kurtosis(\boldsymbol{x_i}) = \frac{E\{(\boldsymbol{s}-\mu_{\boldsymbol{x}})^4\}}{\sigma^4} = \frac{1}{N_s\sigma^4}\sum_{j=1}^{N_s}(x_{ji} - \mu_{\boldsymbol{x}})^4 \tag{4}$$

2.5 Spectral feature extraction

Discrete Fourier Transformation is a mathematical tool that converts discrete-time signals from time to frequency domain. DFT feature extraction is a technique that enables efficient analysis and manipulation of signals in the frequency domain, which can be calculated usingequation 5.

$$X_{DFT}(k) = \sum_{i=0}^{N_s} x_{ji}\, e^{-j2\pi ki/N_s} \tag{5}$$

$k = 0, 1, 2, \ldots \ldots, N_s - 1$

In this work, five Discrete Fourier Transform (DFT) peaks of the $\boldsymbol{x_i}$ were obtained withthe corresponding frequencies. Therefore 225 (5 x 45) Fourier peaks and 225 frequencies of corresponding peaks were extracted from each segment, generating a feature vector of dimension 450×1 from each segment.

Autocorrelation feature extraction is a technique used in signal processing and time series analysis to extract information about the correlation between a signal and its past values.

It is a powerful tool for identifying repeating patterns and trends in time series data. It can be calculated with Equation 6, where $R_{xx}(\Delta)$ denotes the unbiased autocorrelation sequence of $\boldsymbol{x_i}$.

$$R_{xx}(\Delta) = \frac{1}{N_s-\Delta}\sum_{i=0}^{N_s-\Delta-1}(x_{ji} - \mu_x)(-\mu_x) \tag{6}$$

$\Delta = 0, 1, 2, \ldots . . N_s - 1$

In this work, 11 autocorrelation samples, including the first sample and every 5th sample up to the 50th sample were extracted from each of 45 signals in a segment, leading to 495 features per segment.

Conclusively, in this work, we employed the three types of feature extraction techniques to extract a total of 1170 features from each 5-second signal segment, resulting in a dataset dimension of 9120 × 1170. The dataset consisted of 9120 instances, representing eight subjects, 19 activities, and 60 segments per subject per activity.

Feature reduction in machine learning refers to the process of reducing the number of input features or variables in a dataset while preserving relevant information. In this work, theprincipal component analysis[14] method is employed to eliminate redundant and noisy features to improve 'models' performance and reduce computational complexity so, the size offeature vectors is reduced to 30 from 1170.

2.6 Classification methods

The literature demonstrates that various classification methods can be applied to address diverse and complex classification problems. In this study, we have analyzed the performance of these four machine learning models: *K*-Nearest Neighbor (*K*-NN), Logistic Regression, Support Vector Machines (SVM), and XGBoost Algorithm.

2.6.1 K-Nearest neighbor

The K-Nearest Neighbor is one of the most efficient and straightforward machine learning methods in literature.[15] It operates under the tenet that labels for similar data points are likely to be similar. In this, *K* denotes the number of neighbors that were considered when forecasting a new data point. The classifier's performance is highly dependent on the value of the hyperparameter *K*. Euclidean distance is utilized to calculate similarity metrics with other data points. [16] The *K*-NN algorithm can handlechallenging classification tasks by replicating its non-linear decision boundaries. It also applies to other tasks, such as regression and visual recognition.

2.6.2 Logistic regression

Logistic regression[17] is a statistical algorithm for binary classification that predicts the probability of a dependent variable belonging to one of two categories using a logistic function. The model can be trained using a labeled dataset and can predict the class membership of new data points.[18] It can be trained on labeled data and used to predict newdata points. Various parameters such as regularization, penalty function, and solver can be used for the optimization of the algorithm.

Regularization prevents overfitting, while penalty functions help reduce the complexity of the model by encouraging the model to have smaller weight values.[19] Further, the solvers are optimization algorithms that help estimate the model's coefficients. Inthis study, solver (*solver*) and penalty (*penalty*) functions are tuned for performance evaluation.

2.6.3 Support vector machines

SVM-trained models are significantly used in machine learning because of their abilityto handle both linearly and non-linearly separable classification problems. While dealing with non-linearly separable problems, SVM uses a special function known as the kernel to map the data to a higher dimensional feature space where it becomes linearly separable.[20] Some of the SVM kernel functions, such as linear, radial basis function, sigmoid and polynomial, used to perform Support Vector Classification, are defined in Table 1.

Table 1. Different Types of kernel functions and their equations.

Name of kernel functions		Equation
1.	Linear Kernel	$p^T \cdot q$
2.	Radial Basis Kernel	$e^{-\gamma\|p-q\|^2}$
3.	Sigmoid Kernel	$\tanh(\gamma(p^T q) + c)$
4.	Polynomial Kernel	$(\gamma(p^T q) + c)^d$

In Table 1, p and q denote the input feature vectors, $|p - q|$ is the Euclidean distance between p and q, whereas γ is a hyperparameter for width control of kernel. The variable c is a non-negative constant that is used to minimize the disparity between the higher and lower degreesof the polynomial and defines the degree of the polynomial.

2.6.4 Extreme gradient boosting algorithm

XGBoost is an ensemble learning algorithm that combines predictions of multiple weak models to create a stronger optimized model.[21,22] It follows a gradient-boosting framework where errors in existing models are minimized by sequentially adding new models to the ensemble. During training, the loss function, which measures the difference between predictedand actual values, is minimized.

The XGBoost algorithm is known for its effectiveness in handling complex and large datasets. Its key feature is the tunability of hyperparameters, allowing users to fine-tune the model for optimal performance. Parameters such as *gamma* (width control), maximum depth of node (*maximum_depth*), number of trees (*n_estimators*), and learning rate (*learning_rate*), can be adjusted to get enhanced performance.

2.7 Performance assessment metrics

This study evaluates different models' performance by computing the classification algorithm's precision, recall, accuracy, and F1-score from its confusion matrix (refer. eq 7-10). Precision is the measure of the proportion of true positives among all positive predictions, and recall is the measure of the proportion of true positives among all actual positives, F1-

score is the harmonic mean of precision and recall, while accuracy is the measure of the proportion of correct predictions among all predictions.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (7)$$

$$Precision = \frac{TP}{TP+FP} \quad (8)$$

$$Recall = \frac{TP}{TP+FN} \quad (9)$$

$$F1-score = \frac{Recall+Precision}{Recall\times Precision} \quad (10)$$

3. Results and discussion

The entire process of classification is carried out on the same computer with the specification: MD Ryzen 5 3500U with Radeon Vega Mobile Gfx (2.10 GHz), RAM 8.00 GB, AMD Radeon Vega 8 and 64-bit operating system, x64-based processor. Moreover, the algorithmic and computational process is done by using Python 3.9.5. For the validation of the output, a *p*-foldcross-validation technique, where *p* is 10, has been used in the proposed work. It involves dividing a dataset into *p* subsets, using *p*-1 subsets for training a model and the remaining subset for testing, and repeating the process *p* times to obtain a more reliable estimate of the model's performance. This section presents the performance evaluation of the above-discussedclassification algorithms.

3.1 K-Nearest neighbor

The *K*-NN algorithm is implemented with different values of *K*, where *K* varies from 3 to 19. Table 2 provides the ' 'model's mean accuracy, precision, F1-score, and sensitivity/recall for different values of *K*. The results indicate that the model achieves its highest mean accuracy, specifically 91.92% when *K* is set to 5. While *K* = 3 shows higher F1-score and recall values than *K* = 5, increasing *K* significantly drops mean accuracy. This decrease is primarilyattributed to the overfitting of the data and an increase in the number of false negatives (refer to Table 3). Notably, the study is conducted on a balanced dataset. Therefore, among the various *K*-NNmodels evaluated, the choice is made based on the mean accuracy value, which is optimal at *K* equal to 5.

Table 2. Performance of *K*-NN algorithm for different *K* values.

K	Precision	F1-score	Recall	Mean Accuracy %
3	0.95	0.95	0.95	90.1
5	0.95	0.94	0.94	91.92
7	0.94	0.94	0.94	91.3
9	0.93	0.93	0.93	91.3
11	0.93	0.93	0.93	91.1
15	0.93	0.92	0.92	90.6
19	0.92	0.91	0.91	89.9

Table 3. Confusion Matrix of *K*-NN Algorithm when *K* value is 5.

True	Predicted Label																		
Label	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
Z1	446	3	16	1	0	0	7	7	0	0	0	0	0	0	0	0	0	0	0
Z2	1	461	0	0	0	0	13	5	0	0	0	0	0	0	0	0	0	0	0
Z3	21	7	438	1	0	0	6	6	0	0	0	0	0	0	0	0	0	0	1
Z4	12	3	1	460	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0
Z5	0	0	0	0	464	6	0	4	3	0	0	0	1	1	0	0	1	0	0
Z6	0	0	0	0	24	447	0	5	3	0	1	0	0	0	0	0	0	0	0
Z7	7	85	0	2	0	0	375	11	0	0	0	0	0	0	0	0	0	0	0
Z8	2	25	1	2	1	0	46	396	3	0	0	0	0	0	0	0	1	0	3
Z9	0	0	0	0	10	7	0	13	396	53	1	0	0	0	0	0	0	0	0
Z10	0	0	0	0	0	0	0	5	27	437	11	0	0	0	0	0	0	0	0
Z11	0	0	0	0	13	3	0	1	9	30	423	0	1	0	0	0	0	0	0
Z12	0	0	0	0	0	2	0	0	13	12	5	429	0	10	4	1	0	2	2
Z13	0	0	0	0	8	0	0	0	0	1	2	0	465	1	0	2	1	0	0
Z14	0	0	0	0	1	2	0	4	0	0	1	0	5	467	0	0	0	0	0
Z15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	463	15	1	0	0
Z16	0	0	0	0	0	0	0	0	0	1	1	0	0	0	21	457	0	0	0
Z17	0	0	0	0	1	1	0	3	0	0	0	0	0	0	0	0	475	0	0
Z18	0	0	0	0	0	4	0	1	0	1	3	14	9	25	0	0	1	422	0
Z19	0	0	0	0	3	5	0	29	3	0	1	0	2	0	0	0	0	0	437

Table 4. Performance of logistic regression algorithm with different types of solver andpenalty functions

Solver	Penalty	Precision	Recall	F1-score	Accuracy %
saga	L1	0.80	0.78	0.77	77.8
Liblinear	L1	0.86	0.85	0.85	85.0
Liblinear	L2	0.79	0.76	0.77	77.1
sgd	L2	0.62	0.61	0.59	61.2
lbfgs	L2	0.84	0.84	0.83	84.2
Newton-cn	L2	0.83	0.82	0.82	81.8

3.2 Logistic regression

In the context of logistic regression, a comprehensive analysis was conducted involving the utilization of five distinct solver algorithms (Saga, Liblinear, Stochastic Gradient Descent (SGD), Limited-memory Broyden-Fletcher-Goldfarb-Shanno (lbfgs), Newton Conjugate Gradient and in conjunction with the application of two penalty functions, specifically L1 and L2. The outcomes of this empirical investigation, as presented in Table 4, clearly demonstrate that the classification model employing the liblinear solver in conjunction with the L1 penalty consistently exhibited superior performance across various evaluation metrics. Specifically, this model outperformed its counterparts in terms of peak accuracy, F1 score, recall, and precision, thereby establishing its prominence as the optimal choice within the dataset studied.

Furthermore, in Table 5, the intricate insight emerges through the presentation of the confusion matrix for this best-performing model. This matrix affirms the model's performance and highlights its impressive ability to classify activity numbers 18 and 19 accurately. This level of detail serves to underline the model's robustness in addressing the nuances within the dataset, ultimately showcasing its efficacy in delivering precise classifications for specific activities of interest. Indeed, among the various solver algorithms and penalty functions explored, the combination of L1 with the Liblinear solver stands out as the clear choice for further in-depth investigation and study.

3.3 Support Vector Machines (SVM)

In this study, SVM models are trained with different SVM kernel functions, including linear, radial basis function (RBF), sigmoid and polynomial. The outcomes of these experiments are organized in Table 6. The results unmistakably highlight the superiority of the polynomial kernel within the SVM model, achieving the highest accuracy of 96.9%. Notably, the F1 score, precision, and recall values also stand at an impressive 0.97, further reinforcing the exceptional performance of this configuration. These outcomes underscore the robustness and reliability of the polynomial kernel, making it a standout choicefor continued investigation and analysis. Further, the sigmoid kernel exhibits the poorest performance in terms of classification accuracy and other parameters. s are in the order Linear > RBF > Sigmoid. Delving deeper into the performance of the polynomial kernel, as evident in Table 7, it is worth noting that it consistently achieved the highest accuracy when compared to the other kernels. This best-performing SVM model showcased its proficiency by accurately classifyingall 19 activities, apart from Activity 5, which includes A1, A2, A3, A4, A7, and A8. This levelof precision and detailed classification underscores the strength of the polynomial kernel withinthe SVM model, making it a noteworthy candidate for further investigation and analysis.

Table 5. Confusion matrix of logistic regression with liblinear solver and L1 PenaltyFunction.

True	Predicted Label																		
Label	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
Z1	226	1	201	44	0	0	3	5	0	0	0	0	0	0	0	0	0	0	0
Z2	1	444	1	9	0	0	20	5	0	0	0	0	0	0	0	0	0	0	0
Z3	25	3	433	7	0	0	5	7	0	0	0	0	0	0	0	0	0	0	0
Z4	13	0	45	411	0	0	8	2	0	0	0	0	0	0	0	0	1	0	0
Z5	0	0	0	0	407	37	0	1	5	0	15	0	12	1	0	0	2	0	0
Z6	0	1	0	0	29	438	0	1	3	0	4	0	0	0	0	0	4	0	0
Z7	13	151	19	4	0	0	278	15	0	0	0	0	0	0	0	0	0	0	0
Z8	9	37	4	2	1	8	66	330	9	4	0	0	0	3	0	0	3	0	4
Z9	0	2	0	0	16	31	0	1	336	75	18	0	0	1	0	0	0	0	0
Z10	0	0	0	0	0	4	0	0	71	392	8	0	1	2	0	2	0	0	0
Z11	0	0	0	0	14	15	1	0	19	40	360	0	18	6	0	7	0	0	0
Z12	0	0	0	0	0	0	0	0	1	0	0	479	0	0	0	0	0	0	0
Z13	0	0	0	0	13	0	0	0	0	1	15	0	447	2	0	1	0	0	1
Z14	0	0	0	0	7	4	0	0	0	0	0	0	4	464	0	0	0	0	1
Z15	0	0	0	0	0	1	0	0	1	0	0	0	0	1	460	16	1	0	0
Z16	0	0	0	0	5	6	0	0	0	1	0	0	0	4	20	444	0	0	0
Z17	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	469	0	0
Z18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
Z19	0	0	0	0	4	0	0	14	0	0	0	0	0	0	0	0	2	0	460

Table 6. Performance of support vector machine with different types of kernel function.

Kernel Function	Precision	F1-score	Recall	Accuracy %
Linear	0.94	0.94	0.94	89.9
Rbf	0.92	0.88	0.87	88.2
Sigmoid	0.64	0.62	0.60	62.2
Polynomial	0.97	0.97	0.97	96.9

3.4 XGBoost

The model is trained with different values for the three different learning rates (*r*) whilekeeping the maximum depth (*m*) at 3. The results, thoughtfully presented in Table 8, reveal an interesting trend. Although precision, F1-score, and recall values show similarities across n_estimators ranging from 1000 to 5000, the optimal model choice materializes when n_estimators is set at 1000 with a learning rate of 0.1. However, it is essential to note that a compelling alternative emerges when n_estimators is 3000, coupled with a learning rate of 0.01, achieving an accuracy score of 92.4%, closely trailing the highest accuracy score. The decision to select the model hinges on the balance between computational requirements and performance. Opting for n_estimators at 1000 ensures relatively lower computational intensity,all while delivering impressive accuracy. Indeed, the model configuration with n estimators set at 1000 and a learning rate of 0.1 was selected as the best-performing choice for further in-depth analysis. It strikes a balance between computational efficiency and accuracy.

Table 9 presents the comprehensive confusion matrix for this top-performing model. This selection underlines the importance of optimizing both computational resources and model effectiveness in pursuing the most suitable configuration.

A performance analysis of best-performing models from each Machine Learning Algorithm is carried out, namely *K*-NN with *K* value 5, Polynomial SVM of degree 2, Logistic Regression having *liblinear solver* and *L1 Penalty* function, and XGBoost with 0.1 *Learning_rate* and 1000 total number of trees. Based on the findings depicted in Fig. 2, notable observations can be made regarding the precision values associated with different activities. The XGBoost model demonstrates the highest precision for the first and third activities, namely sitting and lying on the back. In contrast, the precision for the second activity, which involves standing, shows a significant decline across all four models. This decline is primarily attributed to an increase in False Positive values across the models, as evident from the data presented in Tables 3, 5, 7 and 9).

Among the models, the SVM model performs relatively better in terms of precision forthe second activity. For the fourth activity, the precision values are comparable among three ofthe four algorithms, with values nearing 1, except for Logistic Regression. Moving on to the fifth and sixth activities, similar patterns are observed in Fig. 2. However, in the case of thesixth activity, where the subject is descending stairs, *K*-NN outperforms XGBoost. Further, a sharp decrease in the line graphs indicates poor precision across all models in accurately classifying a standing subject in a still elevator in the seventh activity. This decline can be attributed to ambiguous interpretability between similar activities such as standing, standing still in a moving elevator, and standing in a moving elevator (Refer to Tables 3, 4, 5, 7 and 8). Acrossactivities 12 to 19, all models exhibited comparable performance with precision values ranging from 0.8 to 0.99. Notably, SVM

Table 7. Confusion matrix of SVM with polynomial kernel.

True	Predicted Label																		
label	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
Z1	448	0	30	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Z2	0	43	0	0	0	0	23	4	0	0	0	0	0	0	0	0	0	0	0
Z3	31	0	444	3	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0
Z4	1	1	3	475	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z5	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Z6	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0	0	0	0
Z7	13	72	3	1	0	0	380	11	0	0	0	0	0	0	0	0	0	0	0
Z8	4	20	2	2	0	0	36	416	0	0	0	0	0	0	0	0	0	0	0
Z9	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0	0
Z10	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0	0
Z11	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0	0
Z12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
Z13	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0
Z14	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0
Z15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0
Z16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0
Z17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0	0
Z18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
Z19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480

Table 8. Performance of XGBoost algorithm with different number trees and learning rates.

r	n	Precision	F1-score	Recall	Best Accuracy %
0.001	100	0.79	0.78	0.78	78.49
0.001	200	0.80	0.80	0.79	79.6
0.01	250	0.89	0.89	0.89	88.6
0.01	3000	0.93	0.92	0.92	92.4
0.1	1000	0.93	0.93	0.93	92.9
0.1	2000	0.93	0.93	0.93	92.8
0.1	4000	0.93	0.93	0.93	92.7
0.1	5000	0.93	0.93	0.93	92.8

Table 9. Confusion Matrix of XGBoost algorithm when the total numbers of trees are 1000and the learning rate of the algorithm is 0.1.

True	Predicted Label																		
Label	Z1	Z2	Z3	Z4	Z5	Z6	Z7	Z8	Z9	Z10	Z11	Z12	Z13	Z14	Z15	Z16	Z17	Z18	Z19
Z1	462	1	5	1	0	0	3	8	0	0	0	0	0	0	0	0	0	0	0
Z2	0	466	0	0	0	0	5	9	0	0	0	0	0	0	0	0	0	0	9
Z3	9	2	456	1	0	0	5	7	0	0	0	0	0	0	0	0	0	0	0
Z4	5	0	0	472	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0
Z5	0	0	0	0	473	0	0	3	3	0	0	0	0	1	0	0	0	0	0
Z6	0	0	0	0	6	472	0	1	1	0	0	0	0	0	0	0	0	0	0
Z7	5	49	0	0	0	0	408	18	0	0	0	0	0	0	0	0	0	0	0
Z8	1	15	0	1	0	1	40	413	9	0	0	0	0	0	0	0	0	0	0
Z9	0	0	0	0	2	3	0	1	445	29	0	0	0	0	0	0	0	0	0
Z10	0	0	0	0	0	0	0	1	15	462	2	0	0	0	0	0	0	0	0
Z11	0	0	0	0	0	1	0	1	4	12	461	0	1	0	0	0	0	0	0
Z12	0	0	0	0	0	0	0	0	0	0	0	480	0	0	0	0	0	0	0
Z13	0	0	0	1	0	0	0	0	0	0	1	0	478	0	0	0	0	0	0
Z14	0	0	0	0	1	0	0	0	0	0	0	0	0	479	0	0	0	0	0
Z15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	474	0	6	0	0
Z16	0	0	0	0	0	0	0	0	0	1	0	0	0	0	9	470	0	0	0
Z17	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	478	0	0
Z18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	480	0
Z19	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	1	0	475

classifiers consistently demonstrated superior results in this range of activities. In terms of overall precision, SVM outperformed the other models, except for activities 1 and 3, where the XGBoost model showed higher precision values. However, whenconsidering the entirety of the classification task, SVM emerges as the best-performing model.

Figure 3 illustrates the comparative patterns of the F1-score, which encompasses both precision and recall, as defined in Equation 10. In this analysis, XGBoost demonstrates the highest accuracy for the 1st and 3rd activities, while SVM outperforms the other models for the remaining activities. However, all four models exhibit poor F1-score values when classifying activity number 7, which involves standing still in an elevator. This can be attributed to an increase in the number of false negatives and false positives, as indicated in Tables 3, 5, 7 and 9). For most activities, SVM consistently achieves the highest F1 score, while Logistic Regression consistently yields the lowest score.

Accuracy is an intuitive measure of overall performance, as mentioned in Equation 7. In Fig. 4, out of the best-performing four models SVM with polynomial kernel achieved the highest accuracy of 96.9%, whereas the accuracy score of others is in the order XGBoost>*K*-NN>Logistic Regression.

4. Conclusion

This study provides a comparative study of four human activity classification algorithms. These algorithms are based on machine learning and were tested on a recorded dataset. The algorithms were tuned with different hyperparameters to find the best performance and optimum classification results. The study's findings reveal that the SVM method employing a polynomial kernel exhibited the most impressive performance in classifying events, achieving a remarkable accuracy rate of 96.9%. Additionally, it attained peak values in precision and F1-score metrics. Comparatively, while XGBoost's event

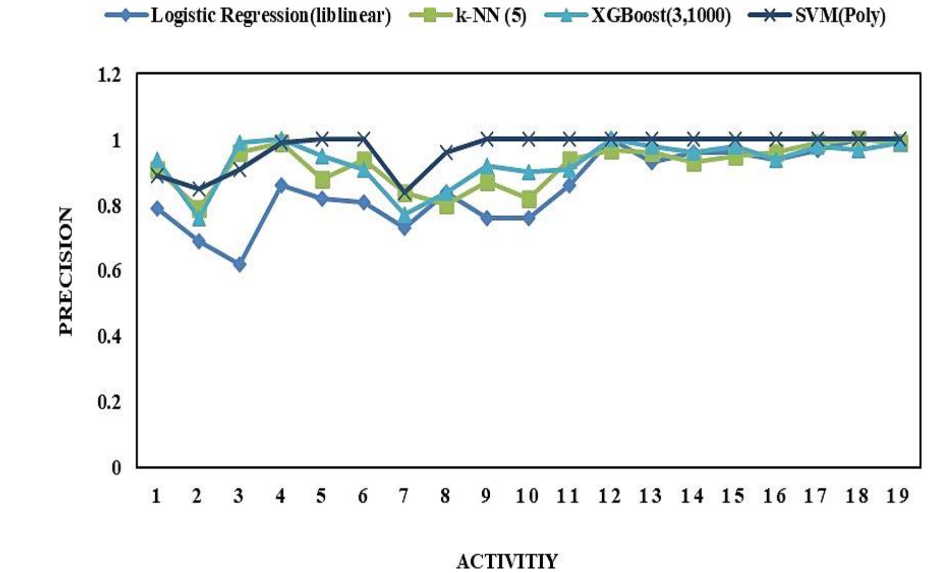

Fig. 2 Precision Comparison of best-performing models from each classification algorithm.

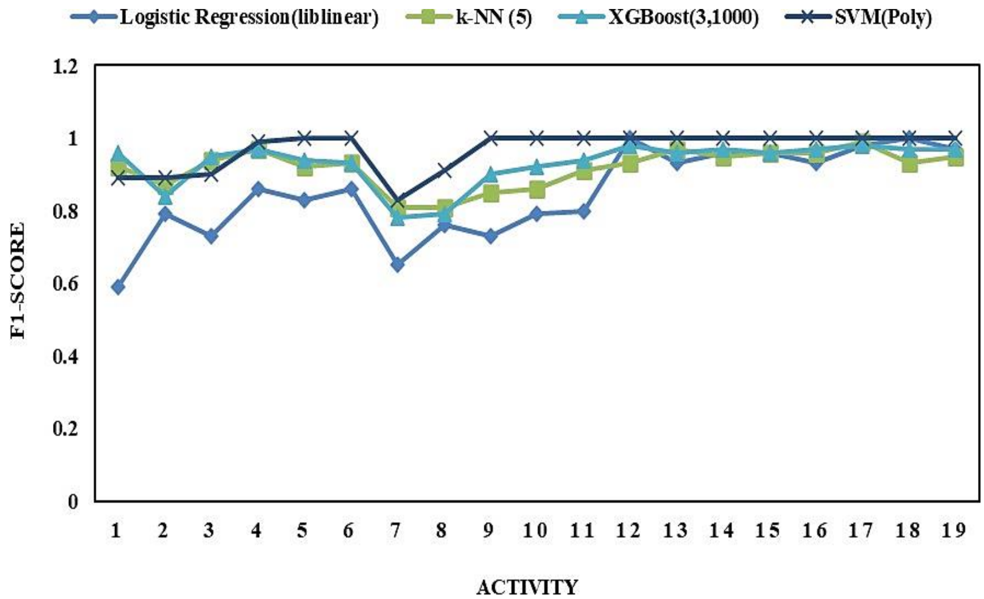

Fig. 3 F1-score Comparison of best performing models from each classification algorithm.

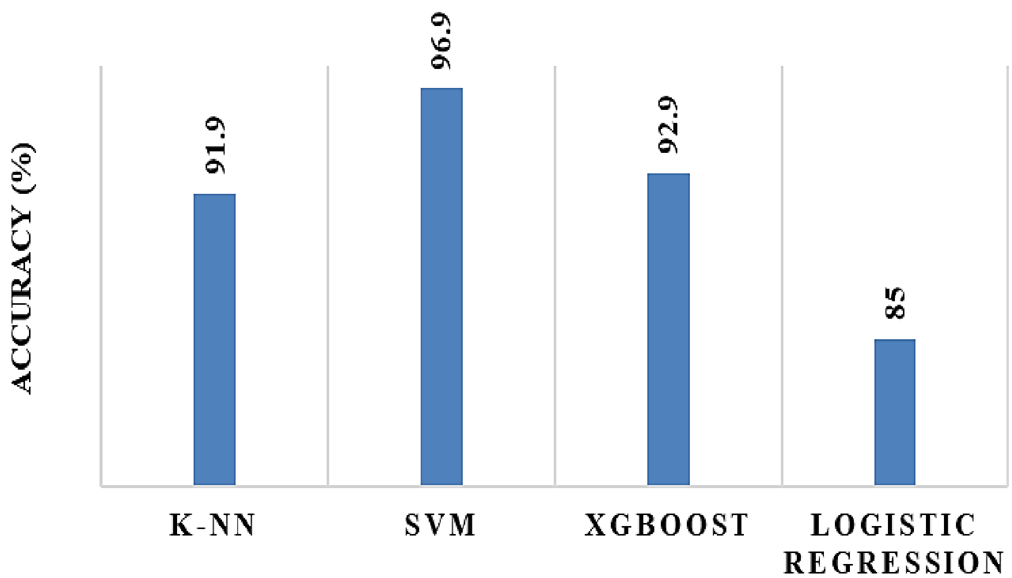

Fig. 4 Accuracy Comparison of best-performing models from each classification algorithm.

classification results were only marginally better than those of *K*-NN by approximately 1%, they still fell significantly short of the performance demonstrated by SVM. Consequently, the SVM model with a polynomial kernel outperformed other state-of-the-art methods. An important insight is related to the unsatisfactory performance of all four algorithms in recognizing similar activities. This observation underscores a potential area for future researchand development.

An intriguing revelation from this study pertains to the challenges faced by all four algorithms in recognizing similar activities. This observation highlights a potential avenue for future research and development. Further investigations could delve into enhancing the algorithms' capabilities in distinguishing similar activities, contributing to more robust and accurate human activity classification systems.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] A. Hayat, F. Morgado-Dias, B. Bhuyan, R. Tomar, Human activity recognition for elderly people using machine and deep learning approaches, *Information*, 2022, **13**, 275, doi: 10.3390/info13060275.

[2] T. Ahmad, J. Wu, H. S. Alwageed, F. Khan, J. Khan, Y. Lee, Human activity recognition based on deep-temporal learning using convolution neural networks features and bidirectional gated recurrent unit with features selection, *IEEE Access*, 2023, **11**, 33148-33159, doi: 10.1109/ACCESS.2023.3263155.

[3] S. K. Polu, Human activity recognition on smartphones using machine learning algorithms, *International Journal for Innovative Research in Science & Technology*, 2018, **5**, 31-37.

[4] O. D. Lara, M. A. Labrador, A Survey on Human Activity Recognition using Wearable Sensors, *IEEE Communications Surveys & Tutorials*, 2013, **15**, 1192-1209, doi: 10.1109/SURV.2012.110112.00192.

[5] N. Gupta, S. K. Gupta, R. K. Pathak, V. Jain, P. Rashidi, J. S. Suri, Human activity recognition in artificial intelligence framework: a narrative review, *Artificial Intelligence Review*, 2022, **55**, 4755-4808, doi: 10.1007/s10462-021-10116-x.

[6] M. Vrigkas, C. Nikou, I. A. Kakadiaris, A review of human activity recognition methods, *Frontiers in Robotics and AI*, 2015, **2**, 28, doi: 10.3389/frobt.2015.00028.

[7] J. Y. L. Chan, S. M. H. Leow, K. T. Bea, W. K. Cheng, S. W. Phoong, Z. W. Hong, Y. L. Chen, Mitigating the multicollinearity problem and its machine learning approach: a review, *Mathematics*, 2022, **10**, 1283, doi: 10.3390/math10081283.

[8] E. Kim, S. Helal, D. Cook, Human activity recognition and pattern discovery, *IEEE Pervasive Computing*, 2010, **9**, 48-53, doi: 10.1109/MPRV.2010.7.

[9] F. Gu, M.-H. Chung, M. Chignell, S. Valaee, B. Zhou, X. Liu, A survey on deep learning for human activity recognition, *ACM Computing Surveys*, 2022, **54**, 1-34, doi: 10.1145/3472290.

[10] B. Barshan, M. C. Yüksek, Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units, *The Computer Journal*, 2014, **57**, 1649-1667, doi: 10.1093/comjnl/bxt075.

[11] K. Altun, B. Barshan, Human activity recognition using inertial/magnetic sensor units, In: HBU 2010 Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, 2010.

[12] K. Altun, B. Barshan, O. Tunçel, Comparative study on classifying human activities with miniature inertial and magnetic sensors, *Pattern Recognition*, 2010, **43**, 3605-3620, doi: 10.1016/j.patcog.2010.04.019.

[13] M. Elangovan, N. R. Sakthivel, S. Saravanamurugan, B. B. Nair, V. Sugumaran, Machine learning approach to the prediction of surface roughness using statistical features of vibration signal acquired in turning, *Procedia Computer Science*, 2015, **50**, 282-288, doi: 10.1016/j.procs.2015.04.047.

[14] W. Dong, M. Woźniak, J. Wu, W. Li, Z. Bai, Denoising aggregation of graph neural networks by using principal component analysis, *IEEE Transactions on Industrial Informatics*, 2023, **19**, 2385-2394, doi: 10.1109/TII.2022.3156658.

[15] M. Bansal, A. Goyal, A. Choudhary, A comparative analysis of K-Nearest Neighbor, Genetic, Support Vector Machine, Decision Tree, and Long Short Term Memory algorithms in machine learning, *Decision Analytics Journal*, 2022, **3**, 100071, doi: 10.1016/j.dajour.2022.100071.

[16] S. Zhang, D. Cheng, Z. Deng, M. Zong, X. Deng, A novel k NN algorithm with data-driven k parameter computation, *Pattern Recognition Letters*, 2018, **109**, 44-54, doi: 10.1016/j.patrec.2017.09.036.

[17] T. Rymarczyk, E. Kozłowski, G. Kłosowski, K. Niderla, Logistic regression for machine learning in process tomography, *Sensors*, 2019, **19**, 3400, doi: 10.3390/s19153400.

[18] A. Basu, R. Roy, N. Savitha, Performance analysis of regression and classification models in the prediction of breast cancer, *Indian Journal of Science and Technology*, 2018, **11**, 1-6, doi: 10.17485/ijst/2018/v11i3/119179.

[19] M. Z. Khan, Salahuddin, M. Arif, B. Q. B. Pakistan, I. U. Islamabad, Artificial neural network versus binary logistic regression for determination of risk factors of myocardial infarction, *Indian Journal of Science and Technology*, 2019, **12**, 1-7, doi: 10.17485/ijst/2019/v12i2/140252.

[20] D. Jahed Armaghani, P. G. Asteris, B. Askarian, M. Hasanipanah, R. Tarinejad, V. Van Huynh, Examining hybrid and single SVM models with different kernels to predict rock brittleness, *Sustainability*, 2020, **12**, 2229, doi: 10.3390/su12062229.

[21] M. Li, X. Fu, D. Li, Diabetes prediction based on XGBoost algorithm, *IOP Conference Series: Materials Science and Engineering*, 2020, **768**, 072093, doi:

10.1088/1757-899x/768/7/072093.
[22] N. S. Chandrahas, B. S. Choudhary, M. V. Teja, M. S. Venkataramayya, N. S. R. K. Prasad, XG boost algorithm to simultaneous prediction of rock fragmentation and induced ground vibration using unique blast data, *Applied Sciences*, 2022, **12**, 5269, doi: 10.3390/app12105269.

