

Article

Machine Learning-Based Stroke Patient Rehabilitation Stage Classification Using Kinect Data

Tasfia Tahsin ¹, Khondoker Mirazul Mumenin ², Humayra Akter ¹, Jun Jiat Tiang ^{3,*}  and Abdullah-Al Nahid ^{1,*} ¹ Electronics and Communication Engineering Discipline, Khulna University, Khulna 9208, Bangladesh; tasfiatahsin98@gmail.com (T.T.); humayraakternipa180938@gmail.com (H.A.)² Department of Computer Science, University of North Carolina at Charlotte, Charlotte, NC 28223, USA; k.mirazulmumenin@gmail.com³ Centre for Wireless Technology (CWT), Faculty of Engineering, Multimedia University, Cyberjaya 63100, Malaysia

* Correspondence: jjtiang@mmu.edu.my (J.J.T.); nahid.ece.ku@gmail.com or nahid@ece.ku.ac.bd (A.-A.N.)

Abstract: Everyone aspires to live a healthy life, but many will inevitably experience some form of disease, illness, or accident that results in disability at some point. Rehabilitation plays a crucial role in helping individuals recover from these disabilities and return to their daily activities. Traditional rehabilitation methods are often expensive, are inefficient, and lead to slow progress for patients. However, in this era of technology, various sensor-based automatic rehabilitation is also possible. A Kinect sensor is a skeletal tracking device that captures human motions and gestures. It can provide feedback to the users, allowing them to better understand their progress and adjust their movements accordingly. In this study, stroke-based rehabilitation is presented along with the Toronto Rehab Stroke Pose Dataset (TRSP). Pre-processing of the raw dataset was performed using various features, and several state-of-the-art classifiers were applied to evaluate the data provided by the Kinect sensor. Among the various classifiers, eXtreme Gradient Boosting (XGB) attained the maximum accuracy of 92% for the TRSP dataset. Furthermore, hyperparameters of the XGB have been optimized using a metaheuristic gray wolf optimizer for better performance.

Keywords: rehabilitation; stroke; kinect; machine learning



Citation: Tahsin, T.; Mumenin, K.M.; Akter, H.; Tiang, J.J.; Nahid, A.-A. Machine Learning-Based Stroke Patient Rehabilitation Stage Classification Using Kinect Data. *Appl. Sci.* **2024**, *14*, 6700. <https://doi.org/10.3390/app14156700>

Academic Editor: Luis Javier Garcia Villalba

Received: 30 June 2024

Revised: 29 July 2024

Accepted: 29 July 2024

Published: 31 July 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

Stroke is one of the principal causes of death nowadays. During the early stages of stroke rehabilitation, which can be traced back to the mid-20th century, the primary focus was on rest and immobilization. However, it was later acknowledged that prolonged periods of immobility could result in muscle weakness, joint stiffness, and a decline in overall function. In the 1970s, there was a notable shift in stroke rehabilitation toward more active approaches. Physical therapy emerged as a fundamental element of rehabilitation to enhance motor function and mobility. The advancements in stroke rehabilitation during the 1980s and 1990s brought about the popularity of task-specific training, which emphasizes the repetitive practice of functional tasks.

Though stroke causes long-term disability in some cases, patients need intensive care and more time to regain their everyday life. Moreover, most post-stroke patients have post-stroke disabilities like vision and sensory impairments, sensory deficits, language swallowing, paralysis, or other long-term consequences [1]. The impairments due to stroke depend on the type of stroke as 85% patients face ischemic strokes and 15% of them face hemorrhagic strokes [2]. Though the stroke management field is achieving remarkable development, most post-stroke patients rely on the rehabilitation process instead of any supervised or widely accepted treatment. Rehabilitation ensures the patients' lost skills return to their everyday lives. Some of the post-stroke rehabilitation processes include the following:

- Exercising the affected muscles to recover muscle strength and body coordination;
- Walking or standing with the help of walkers or wheelchairs to regain the lost functional abilities;
- Movement through active or passive ranges of motion helps to recover affected body joints;
- Forced used therapy ensures the regaining of limb functionalities by moving the affected one and keeping the other limb still at the same time [3,4].

Statistics show that 40% of post-stroke patients face moderate impairments and return to their normal life with special care and rehabilitation. In particular, postoperative rehabilitation and physical therapy are necessary for surgery patients to recover from the operation. Restoring the ability of the patient's physical, intellectual, sensory, and psychological conditions is the main motive of rehabilitation. This traditional rehabilitation is time-consuming and expensive to bear for the longer term. For that reason, therapists suggest in-home rehabilitation and continue the treatment according to the patient's self-report. Sometimes health services globally work together in a hospital under the supervision of renowned physicians to help patients with in-home rehabilitation. Patients continue their necessary rehabilitation while sitting in their homes and frequently visit their therapists to evaluate their progress [5]. However, these processes have limitations as the rehabilitation process should be executed with an expert to achieve better results. Moreover, experts may observe the patients physically and this can be very helpful for the patients. To overcome these limitations, various wearable or virtual sensors may help the patients execute their required exercise and movements of the affected body parts. Most importantly, using these sensors reduces the drain of transportation as well as saving time. One of the most common rehabilitation tools is virtual reality (VR), which compares rehabilitation in real and virtual worlds. In recent years, many therapists and clinical research groups have employed the use of virtual reality facilities to recover lost skills and disabled functionalities. In recent decades, the slight development of various sensors and ML methods has paved the way for a feasible technology-assisted rehabilitation monitoring system with a Kinect sensor. Microsoft Kinect is a line of motion-sensing input devices that identify persons through the recognition of face or voice which was launched in 2010. Kinect depth cameras transmit near-infrared rays to measure the movements of an object. This non-wearable device efficiently records data and can provide the precision of time series data accurately. Joint positions and angle trajectories are also detected using Kinect. Then, the joint rotational data are used to find out kinematic metrics such as range of motion, mean error, and so on. A wide range of rehabilitation robots help to move different body parts of patients to regain their everyday lives. Three rehabilitation robots assist clinical or elderly patients: entire body, upper limb, and lower limb rehabilitation robots [6]. Even in the case of stroke patients, rehabilitation robots can perform the exercises needed by applying a specific force to cause movement of the patient's body parts. According to the latest research, the number of elderly and disabled people is growing daily. Various algorithms are used in this rehabilitation field to ensure proper classification, prediction, and treatment strategies [7]. Support vector machines (SVMs) and random forest (RF) algorithms help by learning human behavior. Flexible sensors can identify various postures of the upper body with the help of different algorithms [5,8]. However, most of the ML models on rehabilitation assessment mainly work to develop a much more accurate and improved model by applying complex algorithms. ML models not only identify and classify patients' movements but also predict patients' rehabilitation as well as recovery status. Furthermore, most of the research has been conducted on upper limbs as well as vision-based gesture detection. However, in this study, we shed light on a Kinect sensor-based automated rehabilitation model.

Objectives

The main purposes of this study include the following:

- Implementing an automatic rehabilitation system that provides feedback to
- patients based on detected compensatory movements.

- Investigating the use of Kinect sensors as a substitute for therapist supervision in rehabilitation settings.
- Applying various state-of-the-art classifiers to identify the most effective and high-performing classifier for this TRSP dataset.

The Section 2 discusses the literature review as well as the research gaps of the stroke rehabilitation field. In the Section 3, the methodology is analyzed. Afterwards, the results are displayed in Sections 4 and 5 illustrates the conclusion and the future directions of stroke rehabilitation.

2. Literature Review

Rehabilitation has gained much attention in recent years, though it has been serving people since ancient times. Many people have conducted research in this field and some of their works are related to our objectives and motivations. A comprehensive summary of the previous works and a critical analysis of the information pave the way for a new research outcome by describing the previous quality of works on post-stroke rehabilitation conducted by prominent authors and comparing them to find out the research gap and scopes of new research. Rehabilitation is a vast field where ML is applied to stroke-based rehabilitation and a lot of work has been carried out by prolific scholars. Some of the renowned research regarding the TRSP dataset is described in the table below.

Table 1 shows that J. Khoramdel et al. [9] applied deep learning algorithms on the TRSP dataset in 2021 and evaluated the final result with previous works. RNN, GRU, transformer, and LSTM are applied to detect compensatory movements of upper limb rehabilitation.

Table 1. A comparison of the previous works using the TRSP dataset.

Ref.	Publication Year	Used Models	Objective and Contribution
[9]	2021	Transformer	Deep learning-based compensatory movements of stroke survivors and GRU gained better performance
[10]	2020	SVM	Explore class imbalance and identify compensatory movements of stroke patients
[11]	2017	–	Detect upper limb compensatory movements of stroke survivors to use proper positioning
[11]	2017	SVM, RNN	SVM-based computer vision system to detect robotic stroke rehabilitation therapy

Decreasing focal loss reduces the imbalanced data distribution of the Kinect sensor data. A Kinect camera along with a robotic arm ensures the detection of compensations as well as joint positions. A vector analysis and threshold techniques are applied to find out the compensations in terms of angles. Firstly, LSTM is applied to the dataset as it has a noteworthy impact on RNN architecture. LSTM is mainly different from the conventional recurrent unit in that LSTM is capable of handling information for a longer time basis. Another upgraded version of the recurrent neural network is GRU which uses cell memory to memorize sequence information. This architecture is simple as well as less time-consuming as it does not contain any memory cells. Last but not least, the type of deep learning algorithm used here is the transformer network which uses an encoder–decoder system dependent on attention layers with three inputs, query, key, and value [12]. As transformers do not process data on an ordered basis, they can pass the inputs parallelly. Based on the recurrent and attention mechanisms, some selected models are used. One recurrent layer of 20 units is used to activate the ReLU output in the model. Increasing the depth of the network increases the possibility of accuracy. For this reason, 10 units are set for two individual layers that are also connected to the classifier. Setting the starting learning rate to 0.01, the models are trained with a batch size of 128 for 50 epochs. It is seen that the training models have performed better in the case of per class and average

precision, recall, and F1 score in comparison with previous works. The precision is 90% in the NC class, whereas it is less than 30% in the other classes. In the case of LSTM and GRU, models that contain two layers have become more accurate. Though the GRU model performs better in this case, the transformer model is also close to the GRU model.

Another work by Sean Rich U. Uy et al. [10] was conducted in 2020 with the dataset where they found the class imbalance to be a probable cause of lower F1 scores while classifying with ML algorithms. The TRSP dataset contains 25 three-dimensional values which have 75 numerical values per second. The spine shoulder joint position is considered the origin point and the other points are normalized as well as translated. The stroke survivors do not perform all of the compensatory movements during the therapy. Data-level, ensemble, and algorithm-level methods are applied to find out the class imbalance. The next step is to train the outlier detection algorithms that have no compensation data. In the case of testing, compensatory movement is supposed to be an outlier. Finally, differentiating between the two approaches and their performance is the main goal of the mentioned work. LOPOCV is applied distinctly at the time of splitting and training the healthy as well as stroke patients. These separate data groups paved the way for selecting the easier one to train and addressing the class imbalance. Undersampling and oversampling are two data-level methods that are applied to the dataset. Undersampling causes information loss as it removes data or the majority class. On the contrary, oversampling duplicates samples of the minority class. SMOTE is used in the mentioned work as SVM SMOTE depends on the classifiers of the SVM to detect unfamiliar data. The next part of the methodology contains algorithm-level and ensemble methods like cost-sensitive learning, RF, etc. The first one is mainly used in cases of imbalance class distribution and RF uses bootstrap samples of the data. The next technique is the detection of outliers. An isolation forest is an ensemble of iTrees and the path length often determines if the sample is an anomaly, whereas the local outlier factor determines the isolation in terms of its neighbors. A linear SVM classifier is mainly used except for the RF variants for the healthy as well as stroke participants. After analyzing the result, we can see that some of the classifiers successfully differentiated the compensations. The result section has also three parts, namely imbalance learning, outlier detection, and comparing healthy and stroke results. In the case of addressing the class imbalance, oversampling performs the best compared to the other methods. The lower values of NC indicate that NC is misclassified as other compensatory movements. Outlier detection algorithms also work to find out the class imbalance. Due to some similar movements like NC, the outlier detection classifiers hardly differentiate the compensations. Moreover, the result of the isolation forest is much higher for the stroke survivor data. The t-distributed Stochastic Neighbor Embedding (t-SNE) method reveals that the compensations of the stroke patients are more clustered, containing no compensation samples. Future research based on the mentioned study should apply other methods to classify the compensations as well as image classification.

Another novel work of Elham Dolatabadi et al. [11] aimed to establish an automatic system that can identify compensatory movements of post-stroke survivors in 2017. Any kind of incorrect exercise postures of the upper limb can be detected with the help of this tool along with a feedback mechanism. This dataset contains both the healthy and impaired stroke patients' data while performing the same movements. Nine stroke survivors as well as ten healthy people were included. A two-degree-of-freedom haptic robot is employed to help with shoulder as well as elbow moving exercises. Stroke patients performed two movements to reveal the range of motion of the upper part of the body. On the other hand, the healthy contributors had to perform some additional movements including these. The homogeneous transformation matrix paved the way for converting the depth camera data to real-world coordinates. Ten certain points of the upper body are considered the representation of the skeleton. No compensation occurs when healthy participants operate the robot and simulate shoulder elevation, trunk rotation, and other compensations. The motion of the compensatory movements is much shorter than with no compensation activities. The sensitivity as well as specificity while categorizing various compensatory

activities are represented here by the receiver operating characteristic (ROC) curve and the area under the ROC curve in the individual cases. For the binary classification, α , β , and γ are the angles while performing good and poor compensatory postures. So, the mentioned study concludes that the TSP dataset is reasonable to use in place of using marker-based movement capture systems. In this baseline process, the algorithms identify the relation between compensatory activities and the angles between them. Thus, comparatively better sensitivity and specificity can be obtained from the ML algorithms while working with the compensatory postures.

The very first study regarding marker-free vision-based rehabilitation therapy was proposed by Y. Xuan Zhi [11] in 2017 which automatically detects compensatory movements of stroke survivors. The Toronto Rehabilitation Institute (TRI) developed a robotic prototype to pave the way for shoulder and elbow postures during rehabilitation therapy. This system identifies the accuracy of the performance of the compensatory activities. Moreover, this system detects abnormal movements due to muscle weakness, incorrect postural adjustments, and so on. Each patient sat in front of the robotic equipment and repeated the movements five times. Compensatory motions are categorized manually and multiclass classifiers are trained depending on the orientation of the three-dimensional segments. The Hidden Markov SVM classifier attained 86% accuracy. The main objective of the mentioned work is to detect common compensatory movements of healthy persons with a classifier. As the levels are classified into four classes, a multiclass classifier is required. A filter of a window size of 31 frames is used here to remove unwanted additional noise effects. Then, the Kinect-centric coordinates are converted to real-world coordinates through translation and rotation. The spine shoulder joint position is considered the reference position with a (0,0,0) 3D coordinate system. SVM as well as recurrent neural network (RNN) classifiers are used to train the various movements of the classifiers. Leave-one-participant-out cross-validation (LOPOCV) detects classifier performance as well as hyperparameter tuning. Though a multiclass RF classifier and Softmax classifier are applied in the mentioned study, they cannot provide superior outcomes. Moreover, the classifiers are trained and tested with the healthy participants first and then with the stroke survivors. The ROC curve and the F1 score represent the performance matrices. Micro-average, macro-average, and the area under the ROC curve are indicated in the plots to detect each type of movement. LF compensations are detected with an SVM that shows a pretty high accuracy, with a 98% AUC and 82% F1 score, whereas TR compensation indicates a 77% AUC and a 57% F1 score. The other two classifiers SE and RNN have quite similar accuracy values, but the moderate values of the SVM and RNN are much better. F1 score represents better the performance of classification. LF achieved outstanding performance, followed by TR and SE.

Among all of previous studies conducted with the TRSP dataset, most of them used SVM or RNN classifiers. Moreover, one of them used deep learning and the rest of them applied various ML classifiers. Upper limb compensatory movement detection and vision-based gesture detection have been carried out in previous works. Analyzing all of the previous works, we have focused on a more robust Kinect-based automatic rehabilitation system with a feedback mechanism. In our work, we are going to detect compensatory postures with real-time monitoring and a better fitted model with better performance.

3. Materials and Methods

Automatic Kinect-based rehabilitation using ML is our main aim in this study. To handle this supervised pattern recognition process, we have used a variety of pre-processing approaches and effectively extracted significant features to make these raw data fit into classification models. Then, we trained and tested the dataset into an acceptable ratio. The flowchart in Figure 1 below demonstrates the stepwise processes of the research.

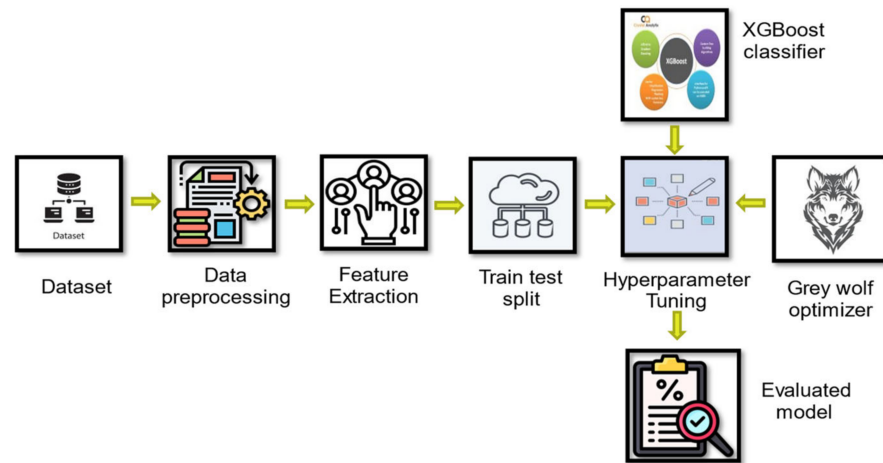


Figure 1. The overall methodology process.

3.1. Dataset

We have used the Toronto Rehab Stroke Pose Dataset (TRSP) which is available in kaggle [13]. The data contain some three-dimensional human poses reflecting stroke patients and normal people performing a set of tasks with the help of a rehabilitation robot. Kinect sensors capture various three-dimensional values of different poses or movements of body parts. Robotic rehabilitation is used to provide assisted and resisted therapy of the shoulder as well as the elbow. The K4W v2 sensor develops a recording application based on SDK 2.0 that tracks and captures motions. The application records a set of three-dimensional locations along the X, Y, and Z axes. It also records 10 upper body parts' orientations at a rate of 30 frames per second. The next part of this study requires the participants to perform several short scripted motions according to their comfort. They first perform the motions with their left hand and then their right. The repeated values of the workout are gathered on a sheet to obtain the required dataset. Table 2 represents that some movements are common for both stroke and normal patients, whereas normal patients have to perform some additional movements to simulate the common movements of post-stroke patients.

Table 2. The movements of the stroke survivors and the healthy participants.

Participants	Movements	
Stroke survivors	1.	Reach forward–backward
	2.	Reach side-to-side
Healthy participants	1.	Reach forward–backward
	2.	Reach side-to-side
	3.	Lean forward
	4.	Trunk rotation
	5.	Shoulder elevation

After that, the data rate is selected as one frame per second. The compensatory movements are labeled as four different stages such as label 1 for no compensation, 2 for lean-forward, 3 for shoulder elevation, and trunk rotation is represented by 4. Further labels are categorized as other if a person performs multiple movements at a time. The calibration data of the movements are recovered for the K4W v2 depth camera with the help of SDK 2.0. After that, the real-world points have to be retrieved from the camera coordination system using a homogeneous transformation matrix.

3.2. Data Pre-Processing

Raw data or initial data files are normally messy and imbalanced, which is not suitable for training models. So, the very first necessary step is to clean up the unnecessary

information from the dataset to avoid missing values as well as irrelevant features. The dataset was an imbalanced one so we have prepared it for further classification models. The initial data file has 3 rows and 11,525 columns with various postures of body parts which are represented with X, Y, and Z axes. As the Kinect frame has captured 25 joints of the subject along the X, Y, and Z coordinates according to the Figure 2, every 25 rows of the dataset represents different joint positions.

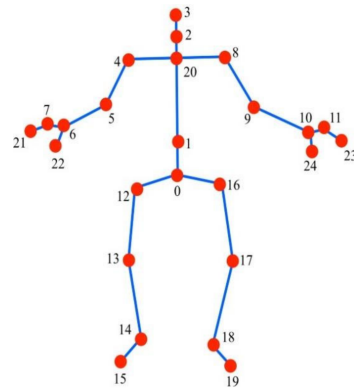


Figure 2. The 25 joints of a subject captured by Kinect.

The other rows are the repetitive captured data of the first 25 joint positions. We have conducted processing of the data file to make it suitable for further work. The columns of the individual CSV files are added in one row and then we reshaped the dataset by labeling the exercises from 1 to 6 according to the movements of the upper body.

3.3. Feature Extraction

ML models choose several data inputs to a target variable which are known as features. The aim of this feature extraction process is for the model to map a pattern between new target variables and the inputs. Extracted features ensure the accurate prediction of the target variable. The removal of redundant features refers to feature selection. Reductions in computational cost and time are another motive of feature selection. We have selected the seven most significant features for our work.

Median $|M_d|$: It helps in identifying abnormal deviations in joint positions that indicate compensatory movements. This feature also represents the particular joint positions during rehabilitation. The median refers to the separator point of the data if we order the data points in ascending order and we split the cross-section into upper- and lower- half portions. In case of an odd number of data counts, the middle value will be the median value. The location of the median can be expressed as m .

$$|M_d| = \left(\frac{N+1}{2} \right)^{th} \quad (1)$$

Otherwise, the average value of the two middle point values is the median for an even number of data counts.

The location of the median is the average of $\left(\frac{N}{2} \right)^{th}$ and $\left(\frac{N}{2} + 1 \right)^{th}$.

Variation: The variation changes can identify when patients do not follow the prescribed exercises. High variation represents inconsistent movements. Moreover, variation represents the changes that occur in the model while using different parts of the training dataset. It demonstrates the possibility of adjustment of the ML function on a given data point. Variation mainly manages comparatively large models with lots of features. The bias of the models has an inversely proportional relation with the variation.

Interquartile Range (IQR): This feature differentiates between smooth and erratic or compensatory actions. The IQR refers to the difference between the 3rd and the 1st quartile of a certain distribution. As this range indicates half of the points of that dataset, the IQR

represents the shape of the distribution. The IQR can identify the outliers of the system if it is less than the 25th percentile or greater than the 75th percentile.

Root Mean Square (RMS): This feature indicates the excessive movements of the actions that deviate from the normal exercises. RMS deviation or RMS error determines the prediction quality and it uses Euclidean distance to do so. The residual represents the difference between prediction and truth values. Calculating the residual for each data point requires calculating the norm and then the mean of residuals. So, RMS error is the square root of the mean and its scale invariant.

$$\text{RMS} = \sqrt{\frac{\sum_{k=1}^n ||Y(k) - Z(k)||^2}{n}} \quad (2)$$

Here, n is the number of data points, $Y(k)$ is the K th measurement, and $Z(k)$ is the corresponding prediction.

Mean $|M|$: The mean feature identifies deviations that suggest compensatory strategies employed by the patients to complete the exercises. The mean is considered the middle value as the total deviation is zero from this value and it is appropriate for all of the data.

$$|M| = \frac{\text{sum of all the signal values}}{\text{the total number of signal values}} \quad (3)$$

Standard Deviation $|\sigma|$: This feature aids in detecting unusual or extreme movements that deviate from normal rehabilitation exercises. It aids in distinguishing between patients who follow the exercise protocol consistently and those who exhibit compensatory movements. The standard deviation means calculating the variability of a sample or the spreading of values in any dataset measured according to the standard deviation. We can compare the ML model's accuracy with the real-world data through it. The square root of the variance represents the standard deviation where the variance is the average of the squared differences from the mean.

$$|\sigma| = \sqrt{E[Y^2] - (E[Y])^2} \quad (4)$$

Here, $E[Y^2]$ indicates the mean of the squared data and $(E[Y])^2$ is the square of the mean of the data.

Kurtosis $|K|$: This feature aids in detecting unusual or extreme movements that deviate from normal rehabilitation exercises. Kurtosis is a statistical measuring process that represents the degree of presence of the outliers in the distribution of our rehabilitation dataset. Kurtosis differentiates the light-tailed and heavy-tailed peaks or outliers at the mean values of the dataset.

$$|K| = n \times \frac{\sum_j^n (X_j - \mu)^4}{\left(\sum_j^n (X_j - \mu)^2\right)^2} \quad (5)$$

$X_j = j^{th}$ variable of the distribution;

μ = the mean of the distribution;

n = the number of variables in the distribution.

3.4. Classification Algorithm: XGB

One of the state-of-the-art algorithm-based classification networks is XGB which is more efficient compared to the traditional gradient boosting decision tree. This gradient-enhanced decision tree improves computational speed through parallel computing, performance, and scalability. We have used XGB in the dataset because of the ease of finding the optimal solution through the second-order Taylor expression [6]. The classification performance of the XGB model mainly depends on quite a few hyperparameters. Hyper-

parameters are parameters containing certain values to determine the learning process as well as evaluate the model parameters. Before training the model, hyperparameters are set, and then, we obtain the model parameters that were learned. Those several hyperparameters are mostly important for optimization in the case of our rehabilitation dataset [14]. In the case of the TRSP dataset, *learning_rate*, *gamma*, *colsample_bytree*, *max_depth*, *min_child_weight*, *subsample*, and *alpha* are the seven necessary hyperparameters which influence the training process and model architecture. The learning rate generally controls the model's learning speed. More precisely, it governs the pace at which an algorithm updates the values of a parameter estimation and controls the step size for a model to achieve the lowest loss function. Gamma is used to make an additional part of the decision tree [15]. *Colsample_bytree* determines the percentage of column or feature numbers to build each tree. The maximum depth of the tree is represented through *max_depth* which also prevents overfitting so that trees cannot grow so deep. It mainly measures the number of nodes defining the longest path from the root node to the most distant leaf node. *Min_child_weight* should be smaller as it is an extremely imbalanced class problem. *Subsample* reduces overfitting and it occurs once in the case of every iteration. Last but not least, *alpha* controls L1 regularization of the leaf weights. It also demonstrates the depth of each tree with a round of boosting. Table 3 represents the hyperparameters and their optimal values. Both the epoch and the pop size are 50 and the upper and lower bounds are set to a certain value. However, these seven hyperparameters have to be optimized to achieve proper results and performance [16,17].

Table 3. The hyperparameters and their values for the TRSP dataset.

Hyperparameter	Value (TRSP)	Value Range
<i>learning_rate</i>	8.64×10^{-2}	$(1.5 \times 10^{-15} - 0.9)$
<i>gamma</i>	3.13×10^{-2}	$(1 \times 10^{-9} - 1.0)$
<i>colsample_bytree</i>	1.06×10^{-1}	$(0.001 - 1.00)$
<i>max_depth</i>	8.31×10^1	$(1 - 200)$
<i>min_child_weight</i>	1.12×10^0	$(1 - 200)$
<i>subsample</i>	5.95×10^{-1}	$(1 - 200)$
<i>alpha</i>	5.78×10^{-6}	$(1 \times 10^{-6} - 1.0)$

We calculated the precision as well as recall curves using the equations below, and the curves are compared in the Section 4. In the case of various thresholds, precision and recall refers to the rate of false positive as well as false negative classes.

$$Precision = \frac{T_{pos}}{T_{pos} + F_{pos}} \quad (6)$$

$$Recall = \frac{T_{pos}}{T_{pos} + F_{neg}} \quad (7)$$

where T_{pos} is the true positive value, T_{neg} is the true negative value, F_{pos} is the false positive value, and F_{neg} is the false negative value.

3.5. Hyperparameter Optimization: GWO

Every model consists of various parameters which define the accuracy and performance of the model. These parameters can be varied to obtain better results for any particular model. There are a lot of optimization techniques that optimize the hyperparameters of the classifier. In our work, we have used a swarm intelligence-based gray wolf optimization (GWO) model to tune the hyperparameters. This meta-heuristic algorithm is used in our work to optimize the hyperparameters of XGB. GWO represents the social hierarchy of wolves and their approach to attacking and encircling their prey while maintaining

nature-inspired categorization. *Canis Lupus* is known as gray wolves who are the topmost hunters in the food chain. Depending on the hunting approach and decision-making power, wolves are categorized as alpha (α), beta (β), delta (δ), and omega (ω) [11]. The most dominant and leading wolves are the alpha ones and other wolves follow their instructions. Beta wolves are advisors who help the other wolves to support the commands as well as the feedback suggestions. Delta wolves are the predators or guards, and the omega wolves are from the lowest hierarchy [18]. In our TRSP dataset, this optimization technique is applied where the statistical structure of the model is based on the hunting behavior of the wolves. The hunting mechanism has three steps: tracking the prey, encircling it, and attacking. Alpha wolves are the best solutions, and beta as well as delta wolves are next. In our work, GWO is applied to find out the best hyperparameter values. In this study, we have used accuracy as the objective function for the optimization task. The mathematical derivation for the objective function is as follows:

$$\text{Objective Function, } F_{obj} = 1 - [\text{Accuracy_score}]$$

Here, the accuracy score (A) of the XGB classifier can be represented as

$$\text{Accuracy score} = 1 - \left(\frac{\text{Misclassified samples}}{\text{total number of samples}} \right) \quad (8)$$

- ROC Curve

The receiver operating characteristic (ROC) curve visually depicts the performance of a binary classification system as the decision threshold changes. It is created by plotting the true positive rate (TPR) against the false positive rate (FPR) for different threshold values. This curve illustrates the balance between the TPR and FPR for the classifier, allowing for an assessment of its performance. But here, we have used a multiclass ROC curve. A multiclass ROC curve is an extension of the traditional ROC curve that is used for binary classification problems to handle multiclass classification problems, where the target variable has more than two classes. In a multiclass classification problem, the ROC curve is computed for each class and the results are combined to obtain a single ROC curve.

- Sensitivity and Specificity

In ROC analysis, the sensitivity and specificity of a binary classifier are key metrics. Sensitivity, or the true positive rate (TPR), measures the proportion of correctly identified positive cases, while specificity, or the true negative rate, evaluates the proportion of correctly identified negative cases. The false positive rate (FPR) is calculated as one minus the specificity. The ROC curve is typically plotted with the TPR on the Y-axis and the FPR on the X-axis. When the decision threshold for classifying an instance as positive is lowered, both the TPR and FPR increase if the classifier's score indicates its confidence that an instance belongs in the positive class. By adjusting the decision threshold from its highest to lowest value, a piecewise linear curve from (0,0) to (1,1) is generated.

- FPR score

The false positive rate identifies the true negative instances that are classified inappropriately as positive ones by the model. A low FPR is expected to avoid false alarms.

$$\text{FPR} = \frac{F_{pos}}{F_{pos} + T_{neg}} \quad (9)$$

Here, F_{pos} is the false positive value and T_{neg} is the true negative value.

- Confusion Matrix

Confusion matrices are a beneficial tool for evaluating the performance of classifiers, as they provide a simple and intuitive way to picture the performance of the classifier and to identify areas where it is making incorrect predictions. Usually, a confusion matrix is

a table used to estimate the performance of a binary or multiclass classifier. It provides a summary of the number of correct and incorrect predictions made by the classifier and is used to calculate various metrics that provide a more detailed understanding of the performance of the classifier.

In a multiclass classification problem, a confusion matrix is a matrix with dimensions equivalent to the number of classes in the problem. The entries in the confusion matrix represent the number of instances from each class that are correctly or incorrectly classified as each class. The entries in the confusion matrix can be used to calculate various multiclass metrics, such as macro-average and micro-average precision, recall, and F1 score.

4. Results and Discussions

We have applied several classifiers to identify the best performer among them and compared various performance metrics. Extreme gradient boosting (XGB), random forest (RF), decision tree (DT), K-nearest neighbor (K-NN), and Gaussian Naïve Bayes (GNB) algorithms are applied in the TRSP dataset. Among the classifiers, XGB works best for handling the imbalanced dataset in spite of facing some computational complexities. RF is prone to overfitting in comparison with the rest of the classifiers but its ensemble nature sometimes makes it harder to interpret. On the other hand, the decision tree algorithm is well known for its simplicity and interpretability, but it faces some overfitting issues which result in lower accuracy. Though implementing KNN in this study is simple, it struggles with large and high-dimensional data. Lastly, GNB is computationally efficient but the assumption of feature independence leads the accuracy to a minimum value. Furthermore, the comparison of the results and analysis of the models is represented by confusion matrices, precision–recall curves, and ROC curves. We have adopted two approaches to evaluate the performances of our models. The first approach is with the raw dataset and the second one is with the feature-extracted dataset.

4.1. Performance Analysis with Raw Data

We have made a comparison between the classifiers' performances. Table 4 represents the various results of the classifiers on the raw dataset and we have made a comparison among the performance metrics. We can compare the values of accuracy, F1 scores, FPR scores, and so on from the table. Moreover, we have made a comparison between the confusion matrix and ROC curves for the following classifiers. We have also represented the multiclass curves to show the interrelations between the classes and evaluate them. From Table 4, representing performance metrics, we have obtained a maximum accuracy value of 92% with XGB, 88% with RF, and the rest follow behind these values. Meanwhile, the F1 score with XGB is 92%, with RF, it is 88%, and with KNN, it is 77%. XGB has achieved the best accuracy among them and that is why we have selected XGB for our work on the TRSP dataset.

Table 4. The performance metrics for the XGB, RF, DT, KNN, and GNB classifiers.

Classifiers	Acc (%)	F1 Score	FPR Score	Sensitivity	Specificity
XGB	92.45%	92.3555	1.50%	92.45%	98.49%
RF	88.82%	88.52%	2.24%	88.82%	97.75%
DT	63.68%	64.24%	7.23%	63.68%	92.76%
KNN	78.21%	77.56%	4.50%	78.21%	95.49%
GNB	51.95%	51.26%	9.68%	51.95%	90.31%

Acc = accuracy.

The multiclass ROC curves are represented in Figure 3, where the six different curves represent six classifiers. They are multiclass as they have six classes representing six different exercises. Each color indicates each class versus the rest of the classes. The classes or exercises are labeled as forward–backward (FB), side-to-side (SS), lean forward (LF),

shoulder elevation, forward–backward trunk rotation (FB_TR), and side-to-side trunk rotation (SS_TR).

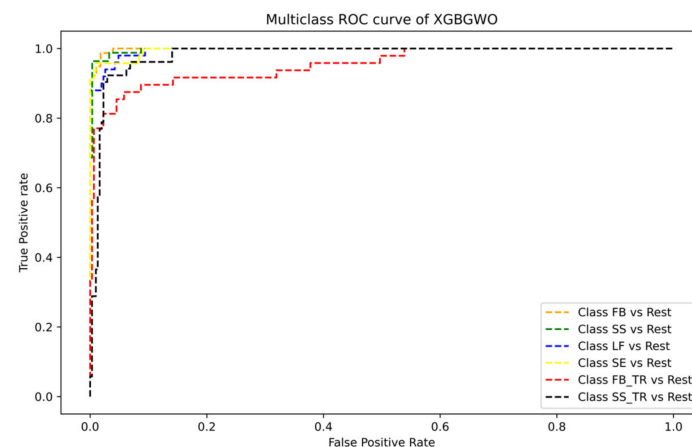


Figure 3. Multiclass ROC curve for XGB classifier for forward–backward (FB), side-to-side (SS), lean forward (LF), shoulder elevation, forward–backward trunk rotation (FB_TR), and side-to-side trunk rotation (SS_TR) exercises.

The area under the curve (AUC) shows the performance of the classification, so a larger AUC is better. In the case of XGB, FB and SS show a better performance compared to the others but the difference between them is not much, whereas LF shows better results with DT and GNB.

The confusion matrix represents the efficacy of a binary or categorical classifier and visualizes the possible errors that can be made by the classifiers. For better visualization of the recall, precision, and ROC curve, the confusion matrix helps a lot. The diagonal elements indicate the proper labels, whereas the off-diagonal elements show the misclassified elements. There may occur two types of errors, namely false positive and false negative. False positive refers to an actual negative value being predicted as positive and false negative means predicting the positive value as negative. In the two particular cases, both of them are wrong. Hence, Figure 4 represents the confusion matrices of the raw dataset:

Predicted	FB	78 21.79%	0 0.0%	1 0.28%	2 0.56%	0 0.0%	2 0.56%	83 93.98% 6.02%
	SS	0 0.0%	80 22.35%	1 0.28%	0 0.0%	0 0.0%	1 0.28%	82 97.56% 2.44%
	LF	0 0.0%	0 0.0%	46 12.85%	0 0.0%	6 1.68%	1 0.28%	53 86.79% 13.21%
	SE	0 0.0%	0 0.0%	2 0.56%	44 12.29%	0 0.0%	1 0.28%	47 93.62% 6.38%
	FB_TR	0 0.0%	3 0.84%	0 0.0%	0 0.0%	37 10.34%	1 0.28%	41 90.24% 9.76%
	SS_TR	0 0.0%	0 0.0%	0 0.0%	1 0.28%	5 1.40%	46 12.85%	52 88.46% 11.54%
	sum_col	78 100% 0.00%	83 96.39% 3.61%	50 92.00% 8.00%	47 93.62% 6.38%	48 77.08% 22.92%	52 88.46% 11.54%	358 92.46% 7.54%
		Actual						
		FB	SS	LF	SE	FB_TR	SS_TR	sum_row

Figure 4. Confusion matrix of the XGB classifier for the forward–backward (FB), side-to-side (SS), lean forward (LF), shoulder elevation, forward–backward trunk rotation (FB_TR), and side-to-side trunk rotation (SS_TR) exercises.

4.2. Performance Analysis with Feature-Extracted Data

As feature extraction ensures the accuracy of the target variables, we have also extracted seven significant features from the TRSP dataset. To compare the performance metrics of the feature-extracted data, we have plotted some of the confusion matrices, ROC curves, and figures here.

After extracting important features, we achieved 76% accuracy with XGB, whereas accuracy values of 71% and 63% were achieved with RF and KNN, as shown in Figure 5. Moreover, a 76% F1 score and a 76% sensitivity are provided by the XGB. GNB and DT have achieved the minimum scores in this case. In Figure 6 of the extracted data, we have presented several multiclass ROC curves of the five classifiers. In the case of XGB and GNB, class SE and LF show better results as the area under the curve is more than the others. FB indicates a higher area under the curve for DT as well as KNN. Finally Figure 7 represents the confusion matrix of XGB classifier.

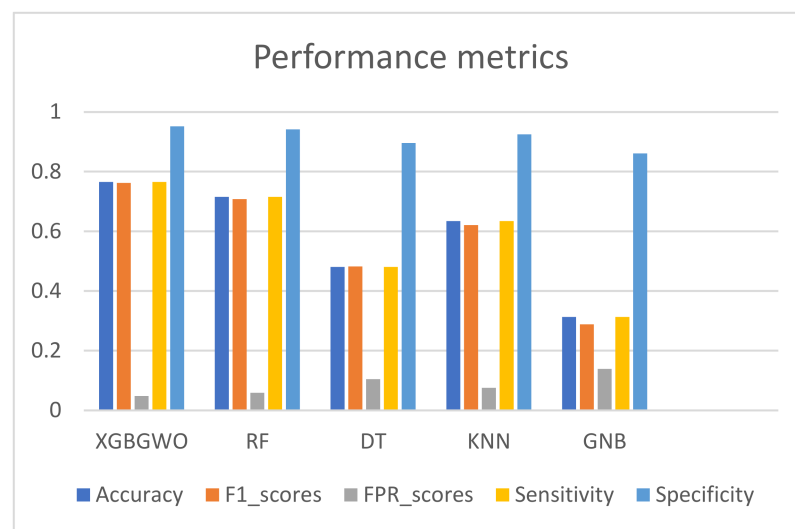


Figure 5. Performance metrics for XGB, RF, DT, KNN, and GNB classifiers.

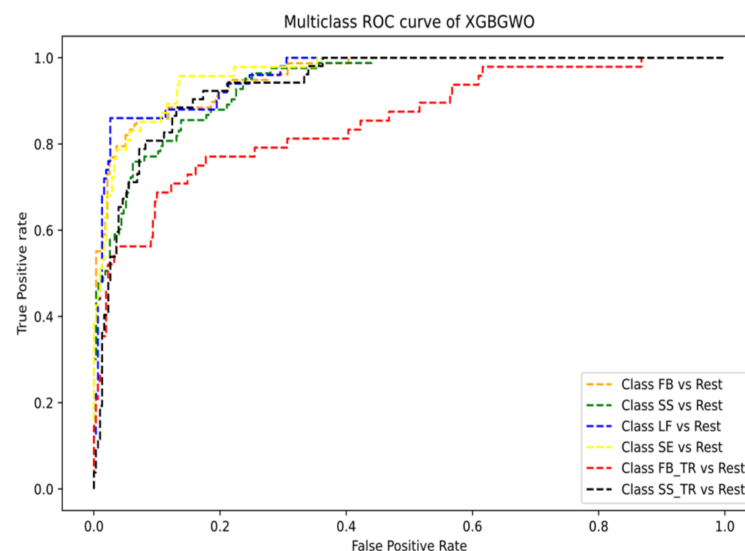


Figure 6. Multiclass ROC curve for XGB classifier for forward-backward (FB), side-to-side (SS), lean forward (LF), shoulder elevation, forward-backward trunk rotation (FB_TR), and side-to-side trunk rotation (SS_TR) exercises for feature-extracted data.

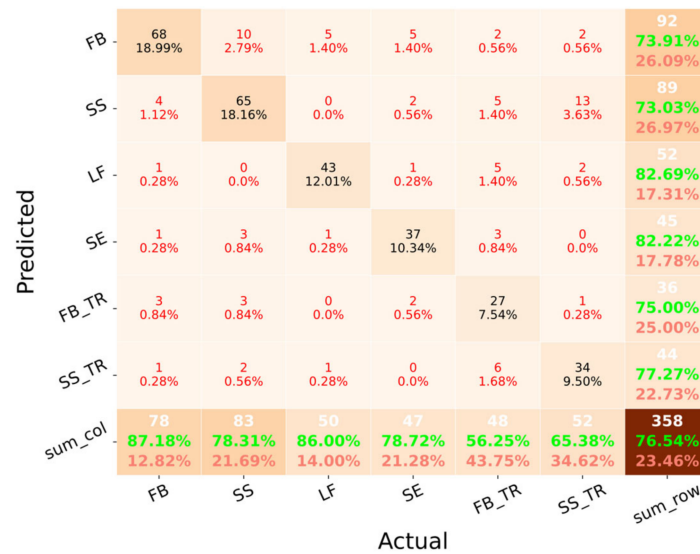


Figure 7. Confusion matrix of XGB classifier for forward-backward (FB), side-to-side (SS), lean forward (LF), shoulder elevation, forward-backward trunk rotation (FB_TR), and side-to-side trunk rotation (SS_TR) exercises for feature-extracted data.

In comparison with the previous works using the same dataset and some related datasets, the table shows the different values of the performance metrics of those studies as well as ours. We have compared the performance metrics of the related works with our research and found out the limitations of our work.

The TRSP dataset is imbalanced and we had some challenges while pre-processing the dataset. However, identification of the imbalanced data during the training period and working on the dataset by randomly selecting the amount of data for each element helps to avoid the possible issues of an imbalanced dataset.

These issues have become a great obstacle to obtaining higher accuracy as well as a higher F1 score in our research work. Moreover, we have extracted seven significant features and compared the results of the raw data and the feature-extracted data. Though the accuracy of the feature-extracted data has decreased a little bit, we intended to analyze the results with all of the classifiers.

Finally, we have compared our performance metrics with the previous works conducted with the TRSP dataset and evaluated our study. Table 5 shows the values of the different metrics of the research.

Table 5. A comparison of our work with the related works.

Work	Dataset	Used Model	Acc (%)	AUC (%)	F1 Score (%)	Precision (%)	Recall (%)
[10]	TRSP	SVM	—	—	94	—	—
[9]	TRSP	Transformer	—	—	87	86	88
[11]	TRSP	SVM	—	80	87	86	88
		RNN	—	62	88	87	90
[11]	TRSP	—	—	51	—	—	—
[19]	PDD	SVM	—	—	98	98	98
[20]	PDD	KNN	98	—	98	98	98
		SVM	98	—	98	98	98
Our study	TRSP	XGB	92	—	92	—	—

PDD = pressure distribution data. Acc = accuracy.

As the PDD dataset is a different dataset, 98% accuracy has been achieved with this. In the case of the TRSP dataset, the highest accuracy of 92% was obtained with the XGB classifier, whereas others used SVM, KNN, or RNN models. A maximum of 94% for the F1 score has been achieved with the SVM classifier in [10] and we have obtained 92% in our work. So, the model performances on the dataset are much better in this case to detect compensatory movements.

5. Conclusions

Kinect-assisted ML-based automatic compensatory posture detection and feedback systems can be a great help for a patient undergoing rehabilitation. Kinect sensors can be a great alternate solution to therapist supervision for post-stroke survivors. We intended to detect compensatory movements of stroke patients with the help of ML algorithms from the TRSP dataset.

Detecting compensations is the very first step of the automatic rehabilitation system. Several classifiers are applied here, among which XGB provides an outstanding classification performance of 92% with the TRSP dataset. The hyperparameters of XGB are tuned with a gray wolf optimizer for the dataset. This proposed framework can detect any kind of upper body posture impairments of post-stroke survivors or other patients with impairments. Kinect can provide detailed information on joint angles, trajectories, and muscle activity that can help to evaluate the effectiveness of rehabilitation exercises and to adjust them if needed.

This automatic rehabilitation technique can be a great help to clinical patients going through rehabilitation or personal recovery processes without the physical supervision of a therapist. Moreover, future works will focus on more improved models with higher accuracy involving advanced deep learning techniques. Implementing robust validation techniques will ensure the generalizability of the model. The results of our study can be a great help in developing a robust movement-detecting system for rehabilitation survivors to monitor real-time movements.

Author Contributions: Conceptualization, T.T., K.M.M., H.A. and A.-A.N.; data curation, T.T., H.A. and J.J.T.; formal analysis, T.T., K.M.M., H.A. and A.-A.N.; funding acquisition, J.J.T.; investigation, T.T., K.M.M., H.A. and A.-A.N.; methodology, T.T., K.M.M., H.A. and A.-A.N.; project administration, J.J.T. and A.-A.N.; resources, K.M.M. and H.A.; software, T.T., K.M.M., H.A. and A.-A.N.; supervision, J.J.T. and A.-A.N.; validation, K.M.M., H.A., J.J.T. and A.-A.N.; visualization, T.T., J.J.T. and A.-A.N.; writing—original draft, T.T. and H.A.; writing—review and editing, J.J.T. and A.-A.N. All authors have read and agreed to the published version of the manuscript.

Funding: This Research was supported by “Research Management Centre, Multimedia University”.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data presented in this study are available on request from the corresponding author.

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

References

- Bernhardt, J.; Langhorne, P.; Bernhardt, J.; Kwakkel, G. Stroke Care 2 Stroke rehabilitation. *Lancet* **2011**, *377*, 1693–1702.
- Fang, G.; Huang, Z.; Wang, Z. Predicting Ischemic Stroke Outcome Using Deep Learning Approaches. *Front. Genet.* **2022**, *12*, 827522. [\[CrossRef\]](#) [\[PubMed\]](#)
- Agbo, F.J.; Oyelere, S.S.; Suhonen, J.; Tukiainen, M. Scientific production and thematic breakthroughs in smart learning environments: A bibliometric analysis. *Smart Learn. Environ.* **2021**, *8*, 1. [\[CrossRef\]](#)
- Boukhenoufa, I.; Zhai, X.; Utti, V.; Jackson, J.; McDonald-Maier, K.D. Wearable sensors and machine learning in post-stroke rehabilitation assessment: A systematic review. *Biomed Signal Process Control* **2022**, *71*, 103197. [\[CrossRef\]](#)
- Xiao, Z.; Qin, Y.; Xu, Z.; Antucheviciene, J.; Zavadskas, E.K. The Journal Buildings: A Bibliometric Analysis (2011–2021). *Buildings* **2022**, *12*, 37. [\[CrossRef\]](#)
- Jia, Y.; Jin, S.; Savi, P.; Gao, Y.; Tang, J.; Chen, Y.; Li, W. GNSS-R soil moisture retrieval based on a XGboost machine learning aided method: Performance and validation. *Remote Sens.* **2019**, *11*, 1655. [\[CrossRef\]](#)
- Chaturvedi, S. Clinical prediction on ML based internet of things for E-health care system. *Int. J. Data Inform. Intell. Comput.* **2023**, *2*, 29–37. [\[CrossRef\]](#)

8. Fong, J.; Ocampo, R.; Gross, D.P.; Tavakoli, M. Intelligent Robotics Incorporating Machine Learning Algorithms for Improving Functional Capacity Evaluation and Occupational Rehabilitation. *J. Occup. Rehabil.* **2020**, *30*, 362–370. [[CrossRef](#)] [[PubMed](#)]
9. Khoramdel, J.; Moori, A.; Moghaddam, M.M.; Najafi, E. Compensatory Movement Detection in Upper Limb Rehabilitation with Deep Learning Methods. In Proceedings of the 9th RSI International Conference on Robotics and Mechatronics, ICRoM 2021, Tehran, Iran, 17–19 November 2021; pp. 465–471. [[CrossRef](#)]
10. Uy, S.R.U.; Abu, P.A. Analysis of Detecting Compensation for Robotic Stroke Rehabilitation Therapy using Imbalanced Learning and Outlier Detection. In Proceedings of the 2020 International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2020, Fukuoka, Japan, 19–21 February 2020; pp. 432–437. [[CrossRef](#)]
11. Zhi, Y.X.; Lukasik, M.; Li, M.H.; Dolatabadi, E.; Wang, R.H.; Taati, B. Automatic Detection of Compensation During Robotic Stroke Rehabilitation Therapy. *IEEE J. Transl. Eng. Health Med.* **2017**, *6*, 1–7. [[CrossRef](#)] [[PubMed](#)]
12. Nisha, C.M.; Thangarasu, N. Deep learning algorithms and their relevance: A review. *Int. J. Data Inform. Intell. Comput.* **2023**, *2*, 1–10.
13. Toronto Rehab Stroke Pose Dataset. Available online: <https://www.kaggle.com/datasets/derekdb/toronto-robot-stroke-posture-dataset> (accessed on 28 July 2024).
14. Tahsin, T.; Mumenin, K.M.; Pinki, F.T.; Tuli, A.B.; Sikder, S.; Rahman, M.A.; Bulbul, A.A.-M.; Awal, M.A. GWO-XGB: Grey Wolf Optimization-based eXtreme Gradient Boosting for Hypertension Prediction in Bangladesh. In Proceedings of the International Conference on Electronics, Communications and Information Technology, ICECIT 2021, Khulna, Bangladesh, 11–14 September 2021. [[CrossRef](#)]
15. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [[CrossRef](#)]
16. Ørebæk, O.E.; Geitle, M. Exploring the hyperparameters of XGBoost through 3D visualizations. In Proceedings of the AAAI Spring Symposium Combining Machine Learning with Knowledge Engineering, Stanford, CA, USA, 22–24 March 2021.
17. Li, J.; Zhang, R. Dynamic Weighting Multi Factor Stock Selection Strategy Based on XGboost Machine Learning Algorithm. In Proceedings of the 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018, Chongqing, China, 10–12 December 2018; pp. 868–872. [[CrossRef](#)]
18. Kusumo, B.H.; Singh, P. Asynchronous Federated Learning with Grey Wolf Optimization for the Heterogeneity IoT Devices. *Int. J. Data Inform. Intell. Comput.* **2024**, *3*, 16–26.
19. Cai, S.; Li, G.; Su, E.; Wei, X.; Huang, S.; Ma, K.; Zheng, H.; Xie, L. Real-Time Detection of Compensatory Patterns in Patients with Stroke to Reduce Compensation during Robotic Rehabilitation Therapy. *IEEE J. Biomed. Health Inform.* **2020**, *24*, 2630–2638. [[CrossRef](#)] [[PubMed](#)]
20. Cai, S.; Li, G.; Zhang, X.; Huang, S.; Zheng, H.; Ma, K.; Xie, L. Detecting compensatory movements of stroke survivors using pressure distribution data and machine learning algorithms. *J. Neuroeng. Rehabil.* **2019**, *16*, 131. [[CrossRef](#)] [[PubMed](#)]

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.