technologies

MDPI

Article

Technology for Improving the Accuracy of Predicting the Position and Speed of Human Movement Based on Machine Learning Models

Artem Obukhov *, Denis Dedov, Andrey Volkov and Maksim Rybachok

Laboratory of VR Simulators, Tambov State Technical University, Tambov 392000, Russia; hammer68@mail.ru (D.D.); didim@eclabs.ru (A.V.); rybachok.maksim@yandex.ru (M.R.)
* Correspondence: obuhov.art@gmail.com

Abstract: The solution to the problem of insufficient accuracy in determining the position and speed of human movement during interaction with a treadmill-based training complex is considered. Control command generation based on the training complex user's actions may be performed with a delay, may not take into account the specificity of movements, or be inaccurate due to the error of the initial data. The article introduces a technology for improving the accuracy of predicting a person's position and speed on a running platform using machine learning and computer vision methods. The proposed technology includes analysing and processing data from the tracking system, developing machine learning models to improve the quality of the raw data, predicting the position and speed of human movement, and implementing and integrating neural network methods into the running platform control system. Experimental results demonstrate that the decision tree (DT) model provides better accuracy and performance in solving the problem of positioning key points of a human model in complex conditions with overlapping limbs. For speed prediction, the linear regression (LR) model showed the best results when the analysed window length was 10 frames. Prediction of the person's position (based on 10 previous frames) is performed using the DT model, which is optimal in terms of accuracy and computation time relative to other options. The comparison of the control methods of the running platform based on machine learning models showed the advantage of the combined method (linear control function combined with the speed prediction model), which provides an average absolute error value of 0.116 m/s. The results of the research confirmed the achievement of the primary objective (increasing the accuracy of human position and speed prediction), making the proposed technology promising for application in human-machine systems.

Keywords: machine learning; neural networks; computer vision; speed prediction; human movement analysis; running platforms

check for updates

Academic Editor: Juan Gabriel Avina-Cervantes

Received: 3 February 2025
Revised: 27 February 2025
Accepted: 1 March 2025
Published: 3 March 2025

Citation: Obukhov, A.; Dedov, D.; Volkov, A.; Rybachok, M. Technology for Improving the Accuracy of Predicting the Position and Speed of Human Movement Based on Machine Learning Models. *Technologies* **2025**, *13*, 101. https://doi.org/10.3390/technologies13030101

Copyright: © 2025 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

In human-machine systems, tracking user actions to form a relevant system response is fundamental [1]. One of the modern approaches to user tracking is computer vision technology, which has rapidly developed in recent years due to advancements in computing power, camera quality, and software tools [2,3]. Another approach is to use different kinds of sensors to recognise human movements [4]. Each approach has both strengths and weaknesses; for example, cameras, and therefore computer vision algorithms, are sensitive to low light, while sensors are sensitive to electromagnetic noise or overlap [5–7].

Computer vision systems show promising results in solving a wide variety of object recognition and segmentation tasks. This is especially important in the context of analysing the motion activity of living organisms, such as humans. Determining the current position of the human body, its movement vector, and patterns of interaction with the environment is relevant for various areas, from medicine and industry to sports and professional training. Computer vision technology makes it possible to solve such problems in real time, not only in laboratory settings but also under actual conditions due to the widespread use of video surveillance systems, photographic and video equipment, and mobile cameras. It provides new opportunities for the development of decision-making systems that analyse a person's current activity, which can be used during rehabilitation exercises [4]. This approach also finds its application in the analysis of motor activity in sports to accurately determine the quality of human movements, allowing the automation of assessment, monitoring, and control processes with high performance and a degree of objectivity [8]. Since computer vision technology can be used in a wide variety of environments (unprepared rooms, large areas, and open spaces [9]), this allows it to be applied in situations where other tracking systems are ineffective.

However, the use of computer vision-based human action tracking systems (or other means of motion capture) within any human-machine system inevitably introduces latency effects. There are several factors that reduce system performance [10,11]: the time to receive data from a source (e.g., cameras or sensors); the time to transmit it to equipment with the necessary software; data processing (object recognition using intelligent algorithms and machine learning models); transmission of the result to the control system; generation of a control command; time for hardware or software tools to respond to the command. As a result, the latency can last up to several seconds. For human-machine systems, such performance can lead to delayed responses to user actions. For instance, if a person is already in a different position and performing a subsequent operation, or if they are unable to react to an emergency promptly, the system might send a signal for actions that are no longer relevant. The basic approach to reduce latency is to predict the user's movements, which can be performed by applying various regression methods (ranging from ordinary linear regression to more advanced machine learning methods) [12,13].

This article considers a human-machine system based on physical workload imitation in the form of an active treadmill [14]. Such devices are successfully used for musculoskeletal rehabilitation, professional training, and the formation of a certain level of physical activity. The specificity and quality of the functioning of active running platforms directly depend on the control algorithms used. In their realisation, the determining role is played by the user, who continuously creates hard-to-predict impacts on the control system, which must adapt in time to human actions, changing parameters, or modes of its functioning.

The process of forming the control command for the running platform is based on the receiving and analysing of user movement data, which is carried out in real time while the platform is working. The resulting delay between the user's action and the running platform's response makes it difficult to control smoothly and leads to various negative effects (jarring, rapid acceleration, or deceleration), which leads to a significant stress effect for the user. In combination with the running platform with virtual reality systems, the overall negative effect is significantly increased [15]. However, there is a problem of mismatch between the speed of the treadmill and the human speed, which also leads to increased discomfort and user dissatisfaction in the process of operation.

Interaction with the running platform causes a certain stress in the user, in which positive (eustress) and negative (distress) forms can be distinguished [16,17]. Eustress is expressed by the need for the person to adapt to the physical activities that occur, resulting in the formation of the required level of physical conditioning. Conversely, the presence

of latency and the mismatch between the speed of the track and the user's movement can lead to additional forms of negative stress that distract users from engaging with virtual reality and complicate the fulfilment of tasks.

Therefore, the aim of this research is to develop a technology based on the application of machine learning models to predict the position and speed of a person on a running platform. This will improve the accuracy of recognising motion parameter data, reduce the effect of latency on human actions, reduce the discrepancy between platform speed and human speed, and ultimately reduce stress levels. The authors' contributions toward achieving this objective are as follows:

- describing a technology for improving the accuracy of predicting the position and speed of a person on a running platform, comprising the stages of data preprocessing, training a set of machine learning models that perform transformations of the original dataset to increase its informational value by improving the accuracy of positioning or prediction data to obtain more relevant information, integrating them into various neural network methods;
- describing neural network methods for running platform control aimed at compensating for latency effects and analysing human activity during movement to generate predictions;
- conducting comparative experiments to evaluate different architectures of machine learning models in solving problems of improving the accuracy of positioning, predicting the position and speed of human movement, and evaluating their performance and applicability in real control systems.

The article has the following structure. The Section 1 analyses the subject area and related works while defining the specifics of the problem to be solved. The Section 2 presents the methodology of the study, including a description of the proposed technology and neural network methods. In the Section 3, four experiments are conducted to compare different machine learning models and neural network methods in solving the given objectives. The Section 4 presents a discussion and evaluation of the obtained results. The article concludes with a summary and conclusions.

Related Works

Analyses of existing research in the field of running platform and track control revealed the following. Many control algorithms for such devices demonstrate a significant level of latency [18]. The latency varies from 0.43 to 2 s [19]. Thus, with a lag of 0.43–0.57 s at a speed of 1.2 m/s, the user's displacement can reach 0.64 m, which is a significant value for the common length of such tracks (from 1 to 2 m). If average performance is assessed (for example, for systems from [17,20]), latency ranges between 1.5 and 2 s, and displacement ranges between 1 and 5 m, which will inevitably lead to a fall off the track, health risks, or emergency activation of stop algorithms or safety systems [21].

Following this, it is important to elaborate on the data collection process. Collecting data on user activities to make timely decisions may require the use of additional equipment that restricts freedom of movement. Therefore, the actual problem in scientific and practical terms is the development of algorithms that allow predicting the actions of a trained person and provide compensation for the delayed response of the running platform.

Various devices (mechanical and ultrasonic sensors, 3D positioning devices, video cameras) are used to collect information about the user's movement. As a result of analysing their features, the following requirements were formulated:

- absence of equipment restricting the user's freedom of movement;
- immunity to background electromagnetic radiation from the running platform;
- range $\geq$ 1.5 m (typical running platform length).

Analysis of existing motion tracking systems [4,22] has shown that only video cameras meet these requirements if the user's silhouette is clearly different from the background or recognised in the frame. Video stream manipulation can be organised using modern machine learning models such as MediaPipe, MoveNet, PoseNet, and newer, and more advanced algorithms [23]. Compared to, for example, trackers or inertial navigation systems, computer vision systems provide a more complete and detailed set of data on the position of the human body in space and are not sensitive to electromagnetic interference. Meeting the requirements outlined above allows for the selection of computer vision as the main motion capture technology within this research.

On the other hand, as previously mentioned, when employing human action tracking systems that rely on computer vision (or other forms of motion capture), an inherent latency effect is unavoidable. The primary factors that influence system performance have been discussed above [10,11].

Previous studies [14] have demonstrated the highly promising and effective implementation of control systems based on the analysis of the current human position, including the use of computer vision technologies. On the other hand, previously developed human body model recognition algorithms (for example, MediaPipe Objectron for leg segmentation) did not solve the latency problem, did not predict the user's state, and faced difficulties in detecting a human at high-speed movements. In developing a new control algorithm, it is proposed to analyse and predict not only the person's position on the running platform but also to estimate their current speed. This additional information can have a positive effect on the implementation of the control algorithm, as the control speed of the track is directly related to the speed of human movement. In ideal conditions, the values of human and track speeds should be comparable.

The use of machine learning methods as the main tool of the prediction tasks is justified by the lack of need to establish an analytical dependence between the initial data (normalised coordinates of human body points) and its current speed in metric units, which is not a trivial task. This is because the initial normalised data are affected by the scale and position of the body model in the frame, and the direction and angle of movement in relation to the camera. The second reason for using this group of methods is the ability to collect a large amount of data on a person moving at different speeds by recording the person's movement on a camera and then capturing key points on the body; the person's speed is determined by adjusting the speed value on the treadmill by the user, as well as using manual start/stop to record changes in walking.

Furthermore, for data collection, it is possible to simulate the process of moving a human model in a virtual environment, saving the animation in the form of a video sequence with the subsequent capture of the positions of all necessary points and the current speed of movement. This allows for varying speed of movement by changing the speed of the animation [24]. Different animations (slow or normal walking, running) can be set for different model speeds; moreover, animations can be different for the same speed to simulate various gait. Preliminary tests have shown that simulations may not include specific movement patterns of humans and their limbs. Therefore, this study will collect data directly from the actual running platform to ensure a better match between real-world activities and the data used to train the neural network models.

The number of points of the human body model is determined by the recognition algorithms in each frame. Previous studies have shown that 18 points are sufficient for complete reconstruction of a human body model [22]. To capture human body points, it is possible to use various machine learning models (MediaPipe, MoveNet), as well as modern solutions such as TokenPose and its modifications.

Therefore, based on the previous research conducted in the field of human motion capture and the implementation of control algorithms, it is essential to develop technology for creating neural network methods for controlling a running platform based by solving the problems of determining a person's current speed by analysing the dynamics of their movements over multiple frames, as well as accurately predicting their future speed and position. Such an approach will help to reduce latency and ensure a timelier response from the system. The use of machine learning methods is proposed as the main tool for solving the problems of speed determination and prediction.

2. Materials and Methods

To achieve the objective of enhancing the accuracy of speed prediction for a person using a treadmill, a methodology is proposed, the overall structure of which is illustrated in Figure 1. This methodology implements the proposed technology for improving the accuracy of positioning and speed prediction through the application of various types of machine learning models. Its main stages will be discussed below.

The first stage involves preparing data for model training. Since the specific task of determining the speed of a person on a treadmill is to be solved, it is necessary to carefully prepare the experimental base, including the designed treadmill with the ability to precisely control its speed in real time, as well as hardware for data collection. A 1920 $\times$ 1080 pixel resolution camera covering the entire treadmill working area is used for motion capture. The treadmill hardware allows not only setting the current treadmill belt speed but also monitoring it at a high frequency (30–50 times per second), which will be used in the future as output data for machine learning algorithms.

The video stream captured from the camera is processed using the OpenCV and MediaPipe (Pose) libraries, which extract 33 key points on the human body. As a result of the first stage, a dataset including a human body model (33 points with X, Y, Z coordinates, along with the degree of visibility of each point; all values are normalised between 0 and 1) and a corresponding treadmill speed value (in m/s) is recorded.

To enhance the stability of the input data, the coordinates of the body keypoint model are normalised relative to a reference object (e.g., the position of the head), and the body model is also scaled. These operations improve the generalisability of the collected dataset under real-world conditions by ensuring data invariance with respect to the subject's position in the frame or their height. A detailed description of this procedure is provided in Section 2.3.

The second stage involves evaluating and training various models, including both common machine learning algorithms and neural network architectures. In total, three main tasks are identified, for each of which the selection of the appropriate optimal model in terms of accuracy (lowest error) is carried out:

ML1: improving human positioning accuracy by processing key points that have low visibility value; the learning process is based on introducing noise into the raw data at the input and recovering the original correct coordinates at the output;

ML2: estimating human movement speed by analysing the positions of key points over a short time window (10–20 frames), thereby capturing the dynamics of limb movements to predict speed, which allows evaluation of the dynamics of changing limb positions and predicting the movement speed;

ML3: Predicting human position by using a sequence of previous key point positions as input and predicting the next positions of key points, which compensates for processing latency in the overall system.

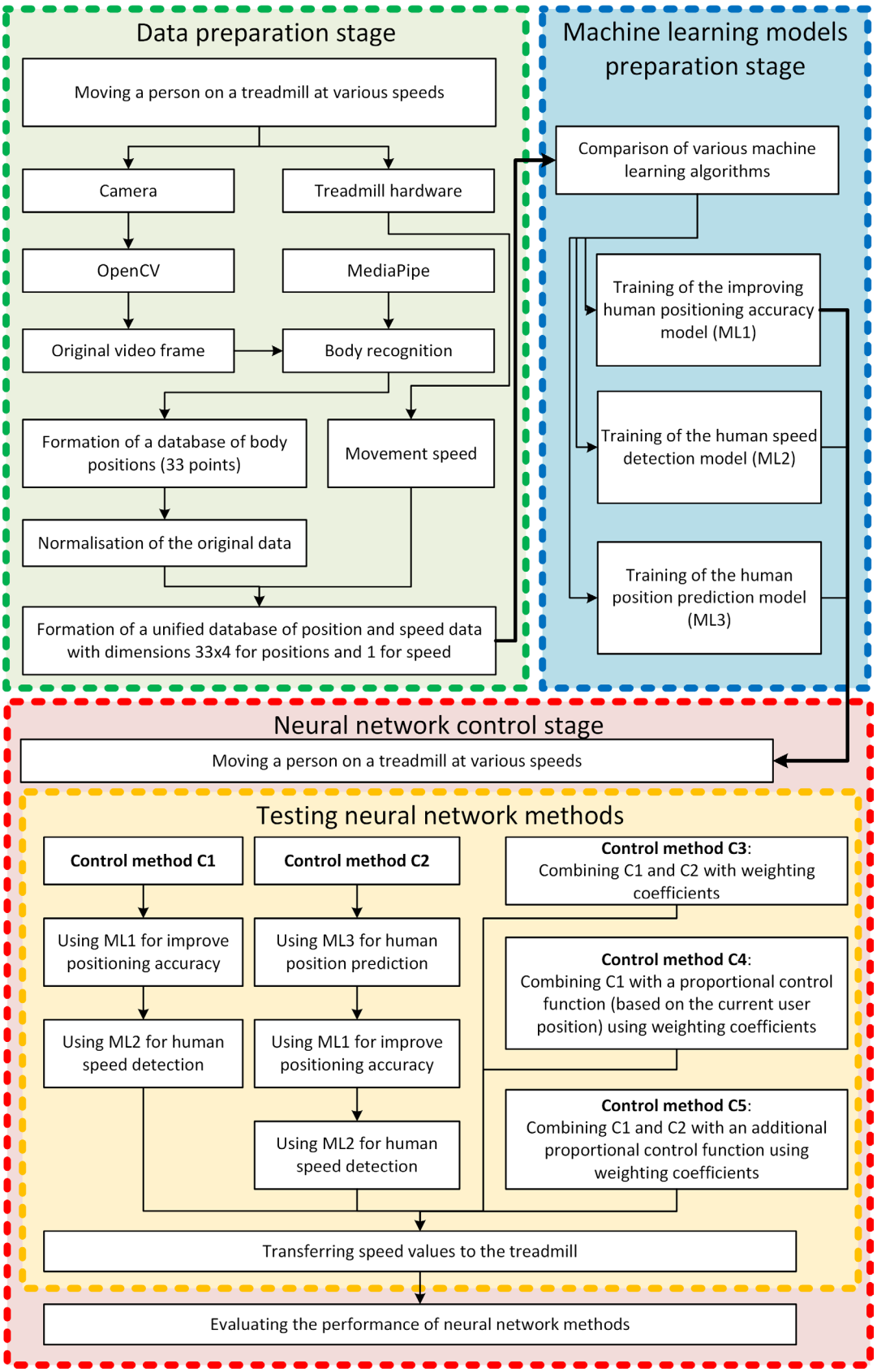


Figure 1. Schematic Diagram of the Research Methodology. The diagram illustrates the entire process—from video acquisition and key point extraction, through the stages of preprocessing, model design, and training of ML1 (positional correction), ML2 (speed detection), and ML3 (position prediction), to their subsequent integration into five neural network-based control methods (C1–C5). Different stages of the methodology are highlighted in distinct colours.

In this stage, training and testing datasets are generated based on the pre-processed data. These datasets may include both "clean" data and data that have been artificially distorted to simulate real-world conditions (for example, due to occlusions of body parts or low illumination levels) for the purpose of conducting specific tests and evaluations.

To select the optimal machine learning model for each task, a comparative analysis is conducted between classical machine learning algorithms and neural networks. The architectures of the employed models, the hyperparameter tuning procedure, and the evaluation metrics (mean absolute error and computation time) are presented in Section 2.2.

In the third stage, the trained machine learning models are integrated into the treadmill control system. Several variants of this integration are proposed, which are designated as neural network control methods (C1–C5):

C1: the raw data from the camera is processed by the ML1 model to correct positional errors caused by overlapping body parts, and then the ML2 model estimates the speed;

C2: the ML3 model is used to predict the person's position over a prediction horizon of W frames (10–20, to be determined in the second stage of the experiment), followed by operations similar to method C1, but using the predicted position values as inputs to reduce latency

C3: methods C1 and C2 run in parallel and then their calculated values are then combined using weighting coefficients (e.g., 0.5) to obtain a final averaged value.

C4: a simple linear control function, proportional to the deviation of the person's position from the starting position, is implemented, after which the resulting speed estimate is combined with a value from method C1 using predetermined weighting coefficients;

C5: similarly to C4, a simple linear control function is implemented, after which the obtained speed value is combined with the values from methods C1 and C2 using predetermined coefficients.

Thus, the experimental comparison will evaluate the effectiveness of control methods C1 and C2 individually, in combination, and when integrated with more classical control strategies.

Testing the neural network methods under various conditions—using pre-recorded video data for evaluation and comparison in simulated settings, as well as during real-world operation on the treadmill—will allow for the identification of the strengths and weaknesses of each method (C1–C5). Moreover, the experimental studies will assess the stability of the control system, which is based on computer vision technologies, under different environmental conditions and in the presence of adverse factors. The implementation of the stages outlined in this methodology will enhance the accuracy of predicting the position and speed of human movement on the treadmill through the integration of machine learning models, determine the effectiveness of the neural network methods based on the combination of these models, and ultimately establish the efficacy of this approach in real-world applications under various conditions.

2.1. Description of Neural Network Control Methods

Neural network control methods will be considered in more detail. By X_t it will be denoted the key point data corresponding to the time moment t, where each element contains three values of the point's coordinate and its corresponding visibility score: $X_t = \{\langle x_{it}, y_{it}, z_{it}, v_{it}\rangle | i = 0..32\}$. The treadmill speed Y_t is also set at each time point.

Then the neural network model ML1 realises the following mapping: $X_t \rightarrow X_t^*$, where X_t^* represents the corrected key points of the human body model. The ML2 model provides the mapping: $X_t \rightarrow Y_t$, and the ML3 model is defined as: $\{X_{t-Q+1}, \ldots X_{t-1}, X_t\} \rightarrow \{X_{t+1}, \ldots X_{t-1+W}, X_{t+W}\}$, where Q is the quantity of analysed previous frames and W is the forecast length (in frames). The lengths of Q and W can be equal or varied, considering

that a larger length of Q can simplify the prediction due to the increased amount of data, while a larger length of W can complicate and reduce the accuracy of the prediction. On the other hand, an increase in Q leads to the need for more data collection.

The neural network control method involves a series of mandatory preparatory operations for working with the running platform. This includes initializing all necessary software modules, libraries, and connections to the platform's control driver, which will serve as an intermediary between the control system software and the microcontroller integrated into the platform. The main cycle of the algorithm consists of launching the platform, preparing in a separate thread the execution of libraries for working with the camera (OpenCV), neural networks (Keras), or machine learning algorithms (scikit-learn). Further, a video stream is received, and a human body model with a set of key points is recognised on it.

The implementation of method C1 uses the ML1 model to correct the position of the points, after which ML2 predicts the current velocity: $Y_t^* = ML_2(ML_1(X_t))$. Accordingly, the smaller the deviation of the found value of Y_t^* from the real Y_t^{real} ($\left|Y_t^* - Y_t^{real}\right| \to \min$), the more accurately the method works.

The C2 method differs in that it requires the prediction of the position of a human body model $X_{t+W} = ML_3(X_t)$, after which $Y_{t+W}^* = ML_2(ML_1(X_{t+W}))$ can be determined. Considering the latency arising during calculations and data processing, Y_{t+W}^* can be used as the current value ($\left|Y_{t+W}^* - Y_t^{real}\right| \to \min$).

The C3 method involves combining two values with selected coefficients: $Y_{C3}^* = k_1 ML_2(ML_1(X_t)) + k_2 ML_2(ML_1(X_{t+W}))$, then $\left|Y_{C3}^* - Y_t^{real}\right| \to \min$ must be provided.

Within method C4, it is necessary to introduce a new speed s_0, which is determined by the relative position of a person on it based on a linear function: $s_0 = s_{\max}(p_t - p_0)/l$, where $s_{\max}$ is the maximum speed of the platform; p_t is the current position of the person on the platform, defined based on X_t; p_0 is the initial position of the person, where their speed is zero; and l is the total length of the platform. Due to the high degree of unpredictability and randomness inherent in human movement, as well as the acceleration and deceleration of leg movement during walking, the algorithm proposes a combined calculation of platform speed that takes into account correction coefficients (k_0, k_1), multiplied by s_0 and the current human speed Y_t^*:

$$Y_{C4}^* = k_0 s_0 + k_1 ML_2(ML_1(X_t)). \quad (1)$$

In the C4 method, it is necessary to provide $\left|Y_{C4}^* - Y_t^{real}\right| \to \min$.

Finally, method C5 considers the prediction of a person's position and their corresponding speed. Taking into account the correction coefficients (k_0, k_1, k_2), which are multiplied by s_0, the current human speed Y_t^* and the speed prediction Y_{t+W}^*, it is possible to consider the following situations: when the user is preparing to brake, stopping suddenly, or accelerating:

$$Y_{C5}^* = k_0 s_0 + k_1 ML_2(ML_1(X_t)) + k_2 ML_2(ML_1(X_{t+W})). \quad (2)$$

Therefore, in the C5 method, it is necessary to ensure that $\left|Y_{C5}^* - Y_t^{real}\right| \to \min$.

2.2. Preparation and Evaluation of Machine Learning Models

In order to realise the models, it is necessary to implement and compare different algorithms of machine learning and neural networks. The conducted analysis revealed [22,25,26] that the following options are justified in this field:

Linear regression (LR)—standard linear regression model.

Decision tree regressor (DT)—regression based on decision trees with a depth of 10.

Random forest regressor (RF)—regression based on a random forest of 20 decision trees with a depth of 5.

XGB regressor (XGB)—regression based on trees with depth of 5 with gradient effort.

CNN—convolutional neural network consisting of 4 Conv1D layers with varying numbers of filters (32 to 256) and batch normalisation, followed by 2 dense layers with 100 and 200 neurones, respectively.

NN—conventional multilayer neural network with 5 dense neurones, having between 100 and 200 each.

For the linear regression model, the default settings were maintained. The hyperparameters listed above for the other models were selected using the GridSearchCV method in the following ranges: tree depth, from 3 to 15; number of trees, from 5 to 30; number of neurones in the dense layer, from 50 to 400; and in the Conv1D layer, from 16 to 512. The general architecture and the list of models are chosen on the basis of their application in previous studies in analysing human motor activity [4].

Since all three models are focused on solving regression problems, the mean absolute error (MAE) is used as a measure of quality, as it clearly assesses the magnitude of the deviation between the forecasted values y_i and the actual values y_i^{real}:

$$MAE = \frac{\sum_{i=1}^{n} \left| y_i - y_i^{real} \right|}{n}. \quad (3)$$

Additionally, the metric for evaluating the prediction time $\bar{t}$ of the machine learning model is of great importance. This metric measures the computational efficiency of the model by calculating the average time required to predict values over a specified number of runs (e.g., $n = 100$). It is critical in real-time control systems because a lower prediction time is essential; increased prediction latency can impair the system's responsiveness and overall performance. The average prediction time $\bar{t}$ is given by:

$$\bar{t} = \frac{\sum_{i=1}^{n} t_i}{n}, \quad (4)$$

where n is the number of runs.

2.3. Pre-processing of Human Body Model Data

At the data pre-processing stage, it was determined that it was necessary to normalise key points of the human body relative to the outermost points. This is because during data acquisition, a person may move in different areas of the image, so using only initial normalised coordinates relative to the frame size is insufficient.

Subsequently, an additional transformation is applied to all points of the human body model relative to the position of the user's head (in the MediaPipe model, this zone corresponds to index 1), as it best allows normalising the points relative to the frame on both axes. Then the following transformations are obtained:

$$\{\langle x_{t,i}, y_{t,i}, z_{t,i}\rangle | i = 0..32\} \rightarrow \{\langle x_{t,i} - x_{t,1}, y_{t,i} - y_{t,1}, z_{t,i} - z_{t,1}\rangle | i = 0..32\}. \quad (5)$$

Within the presented method, it is also important to consider the situation when the size of the objects of observation may change during movement, such as when a person approaches or moves away from the camera, which may lead to significant differences in the input data values. Therefore, in such cases, it is necessary to normalise the human body model based on the person's height. This transformation allows the system to disregard the

absolute size of the human body model detected in the frame. The following definitions will be used:

$x_{\min} = \min(\{x_{t,i} | x_{t,i} \in U_t\})$—minimum value along the *X*-axis among all points of the human body;

$x_{\max} = \max(\{x_{t,i} | x_{t,i} \in U_t\})$—maximum value along the *X*-axis among all points of the human body;

$y_{\min} = \min(\{y_{t,i} | y_{t,i} \in U_t\})$—minimum value along the *Y*-axis among all points of the human body;

$y_{\max} = \max(\{y_{t,i} | y_{t,i} \in U_t\})$—maximum value along the *Y*-axis among all points of the human body.

Then:

$$\begin{gathered} U_t = \{\langle x_{t,i}, y_{t,i} \rangle\} \to \left\{ \left\langle \frac{x_{t,i} - x_{\min}}{m_{xt}}, \frac{y_{t,i} - y_{\min}}{m_{yt}} \right\rangle \right\}, \\ m_{xt} = x_{\max} - x_{\min}, m_{yt} = y_{\max} - y_{\min}, \\ i = 0.32. \end{gathered} \tag{6}$$

Therefore, these additional transformations improve the accuracy of the machine learning models by minimising the influence of the person's position in the frame and their current dimensions.

3. Results

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results and their interpretation, as well as the experimental conclusions that can be drawn.

3.1. Experimental Scheme and Data Preparation

The experimental research consists of four main stages.

In the first experiment, the ML1 model was compared and selected to improve the positioning accuracy of key points of the human body model. The effectiveness of the methods was analysed in complex conditions involving limb overlap, such as using handrails. To do this, the original video dataset recorded while walking on a track was programmatically modified to add overlaps that made it difficult to recognise the user's legs and body (Figure 2). The ML1 model trained on noisy data by reconstructing the original data in this way. After training, the models were further compared using the MAE metric and validated on new video data featuring different overlapping areas.

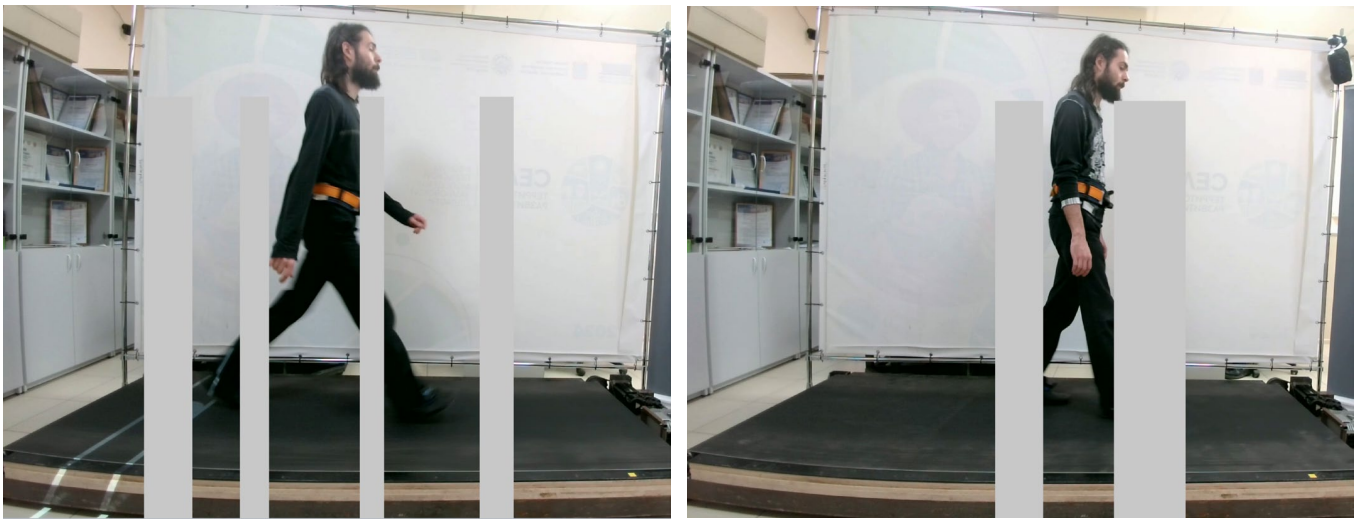
The second experiment evaluates the ML2 model, which estimates human speed under various conditions, by comparing its speed prediction accuracy for different lengths of the analysed frame window.

The third experiment compares different models for predicting body position. It also evaluates how the length of the analysed frame set and the prediction interval influence the accuracy of the solution.

In the fourth experiment, the selected optimal machine learning models are used in the implementation of neural network control methods; the comparison of these methods is carried out.

In summary, the performance of computer vision technology under different conditions (illumination, user's clothing, background) has been tested.

The experiment uses a 2.54 m long treadmill developed by the author's group (the length of the area on which a person can walk is 2.4 m).



(**a**) (**b**)

Figure 2. Example of inserting noise into the original video data: (**a**) during model training; (**b**) during testing. Artificial noise in the form of grey rectangles is used to complicate the operation of the human body model recognition system.

To successfully perform the experiments described above, video data were collected and processed using the MediaPipe Pose model to extract key points of the human body. The raw data consist of a set of videos with a resolution of 1920 by 1080 pixels, 30 frames per second, and a quality of approximately 13 Mbps. Video duration was 33 min for training samples and 9 min for testing samples. After the MediaPipe transformation, key point values were normalised between 0 and 1. Tests showed that 1 real meter corresponds to an approximate normalised value of 0.35 with the camera model used. This can be used to convert normalised values into metric values. The value of 0.35 depends on the distance between the treadmill and the camera, as well as the angle of view of the camera. This is determined at the calibration stage by projecting a 1 m line onto the treadmill and then determining the position of the person in the frame at both ends of the line. By measuring the difference between the positions of the person's waist points at the two ends of the 1 m line, it is possible to determine the corresponding normalised value in the frame.

3.2. The Experiment to Improve Positioning Accuracy

In the first experiment, training data were processed using the scheme shown in Figure 2 above. Four artificial rectangles were used to create the effect of overlapping the user's legs and body, complicating the operation of the MediaPipe Pose model. Data were saved in parallel, with key point values from the frames containing noise saved to the input data array, while key points from the original frames were saved to the output data array.

Processing of the training sample results in two arrays of sizes: the first one depends on the size of Q and is equal to (75,610, Q, 33, 4), and the second one is equal to (75,610, 33, 4) for all cases. This verifies how many frames it makes sense to analyse for key point correction. The results of evaluating the models discussed in Section 2.1 on the MAE metric are summarised in Table 1.

Table 1 presents a comparative analysis of different machine learning models on the MAE metric in the task of determining the real position of key body points in the presence of partial overlap. The values Q = 1, 2, and 3 denote the number of previous frames fed to the model input to predict the position. As the comparison has shown, introducing additional frames for analysis has not had a positive impact for the majority of models; therefore,

ML1 will analyse only the current frame. The LR and XGB were the best performers in this task among all models.

Table 1. Results of the first experiment (MAE).

Model	Q = 1	Q = 2	Q = 3
LR	**0.0018**	**0.0022**	**0.0022**
DT	0.0058	0.0057	0.0059
RF	0.0044	0.0045	0.0046
XGB	**0.0015**	**0.0019**	**0.0019**
CNN	0.0159	0.0249	0.0312
NN	0.0068	0.0066	0.0068

The best results for each model are highlighted in bold font.

The performance of the trained models was subsequently analysed. A total of 100 runs of each model were performed, and the average speed prediction time was recorded. The performance results are presented in Table 2. Model performance evaluation is important at every stage, as a slow model cannot be employed in the control system due to increased latency in the data processing.

Table 2. Performance comparison of ML1, in seconds.

Model	Q = 1	Q = 2	Q = 3
LR	0.0672	0.0533	0.0528
DT	4.0×10^{-5}	4.0×10^{-5}	4.0×10^{-5}
RF	0.0080	0.0092	0.0092
XGB	0.0011	0.0010	0.0014
CNN	0.0714	0.0378	0.0383
NN	0.0329	0.0350	0.0372

The most effective model is DT, followed by XGB. The chosen model, LR does not show the best results. Based on the results of the comparative analysis, the models have a clear advantage in terms of performance—LR and XGB. However, in order to validate the obtained performance results, it is necessary to conduct a subjective visual comparison using real data. This is because a smaller deviation may not always provide the best result, as reference key points can also deviate from their actual position of the legs due errors from MediaPipe. In addition, due to the poor performance of the LR model, the DT model was added to the comparison. The fragments of this model comparison are shown in Figure 3.

White lines and orange-blue dots in Figure 3 indicate the initial positions of segments and key points of the human body model obtained from MediaPipe Pose. The green colour corresponds to segments and point positions obtained after reconstruction by machine learning models. Grey rectangles represent the interferences introduced during testing. In Figure 3, the names of the used models are indicated in the header of each frame; the X and Y axes indicate pixel values.

Based on the analysis of the results presented in Figure 3, it can be concluded that the lowest error does not guarantee the best result under real-world conditions. It is worth noting that the stability of the model results varies, with some instances demonstrating incorrect predictions of the actual position of the limbs (in the first example, this is the case for LR and XGB; in the second, for DT; in the third, for LR and partly XGB; and in the fourth, for XGB and partly LR). In most situations, reconstruction after DT proves to be more accurate. Therefore, considering the performance results of the first experiment, and accounting for the high performance, it can be concluded that the effectiveness of using the

ML1 model in its DT variant to improve the quality of positioning accuracy in scenarios where the human body overlaps with external objects.

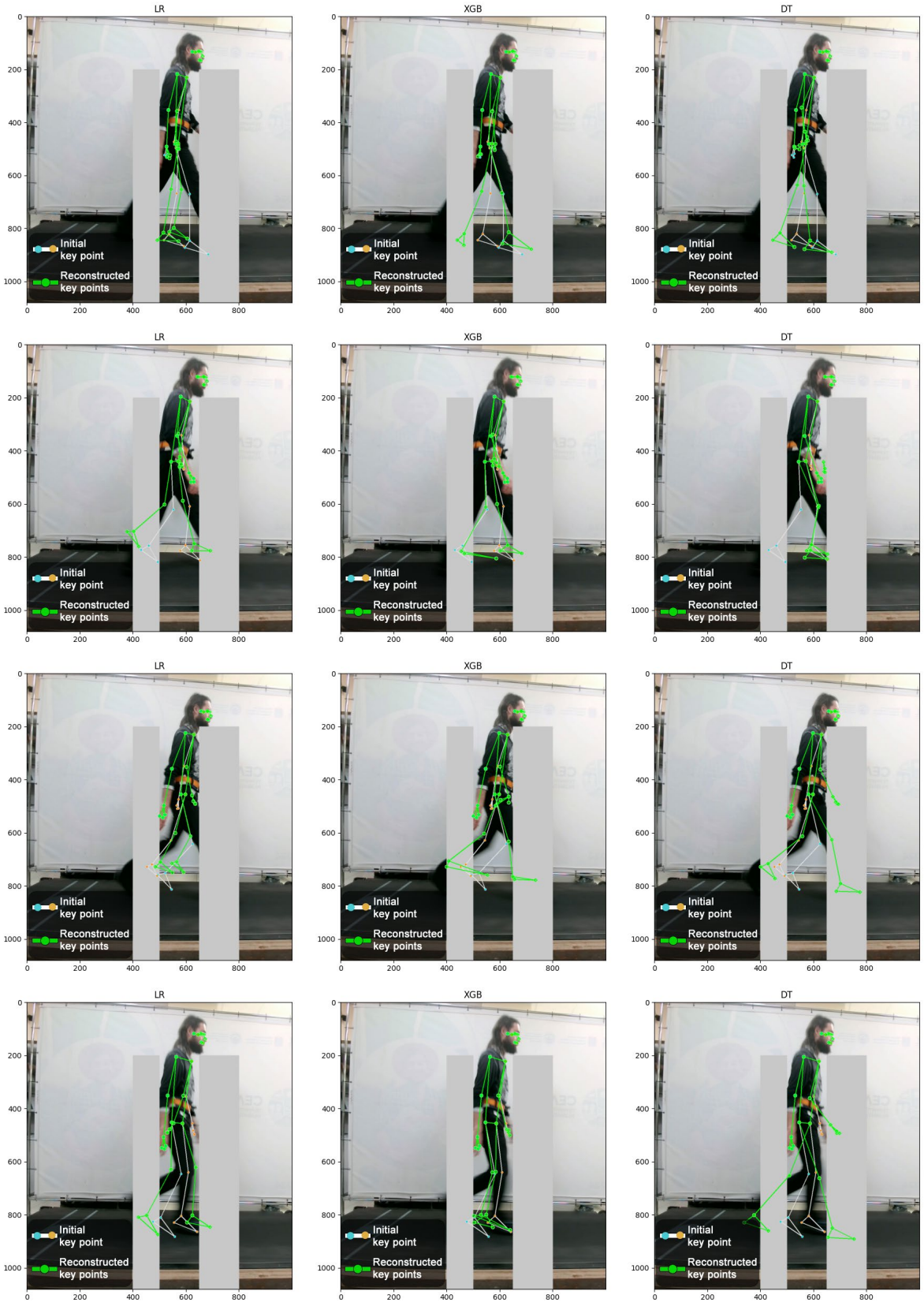


Figure 3. Comparison of LR, XGB, and DT models under conditions of artificial interference (grey rectangles) for reconstructing the correct positions of body segments (green dots and lines) during treadmill movement.

3.3. Experiment to Determine the Speed of Human Movement

In the second experiment, information about the position of the human body during Q frames is used as input data, on the basis of which the speed of movement can be measured (output data). Since it is impossible to predict the value of Q in advance, the second experiment compares the accuracy of all models for 3 values of Q: 10, 15, and 20. The amount of training data is 75,625 records, and the number of test data is 17,620 for each value of Q. The results of the second experiment are presented in Table 3.

Table 3. Results of the second experiment (MAE).

Model	Q = 10	Q = 15	Q = 20
LR	**0.2739**	0.3358	0.4631
DT	1.1146	1.1247	1.1241
RF	1.1660	1.1717	1.1734
XGB	1.2521	1.3455	1.1954
CNN	1.0057	1.6515	1.2142
NN	1.1963	1.1871	1.2372

The best results for each model are highlighted in bold font.

Therefore, Table 3 shows the influence of the length of the analysed set of frames (Q) on the accuracy of human speed prediction. As shown by the experimental results, increasing Q leads to an increase in the error. It should be noted that the larger the value of Q, the longer it takes the model to determine the initial velocity. Therefore, increasing this parameter may not be justified. Among all models, LR at Q = 10 provided the best results.

Then, similarly to the previous experiment, the performance of the trained models was analysed. One hundred runs of each model were performed, and the average speed prediction time was recorded. The performance results are presented in Table 4.

Table 4. Performance comparison of ML2, in seconds.

Model	Q = 10	Q = 15	Q = 20
LR	0.0013	0.0013	0.0015
DT	4.0×10^{-5}	4.0×10^{-5}	6.0×10^{-5}
RF	0.0091	0.0091	0.0090
XGB	0.0005	0.0007	0.0008
CNN	0.0363	0.0361	0.0366
NN	0.0350	0.0365	0.0370

The performance evaluation of the models shows that the prioritised model (LR) has a sufficiently high performance, which does not prevent its further usage.

The testing of the models on the video from the test samples is shown in Figure 4. Three fragments of a common speed experiment (at low, medium, and high track speeds) are presented. The top right corner of each frame displays the current track speed, while the left corner shows the prediction results of each machine learning model at three values of Q, allowing for a simultaneous visual comparison under real conditions.

The results of the second experiment show that the optimal value of Q is 10, and LR should be used as the ML2 model. The other models showed significant variation in values and low stability of performance. From the experiment we can conclude that a longer buffer reduces the recognition efficiency (if at Q = 15 LR shows acceptable results, then at Q = 20 the error increases more severely).

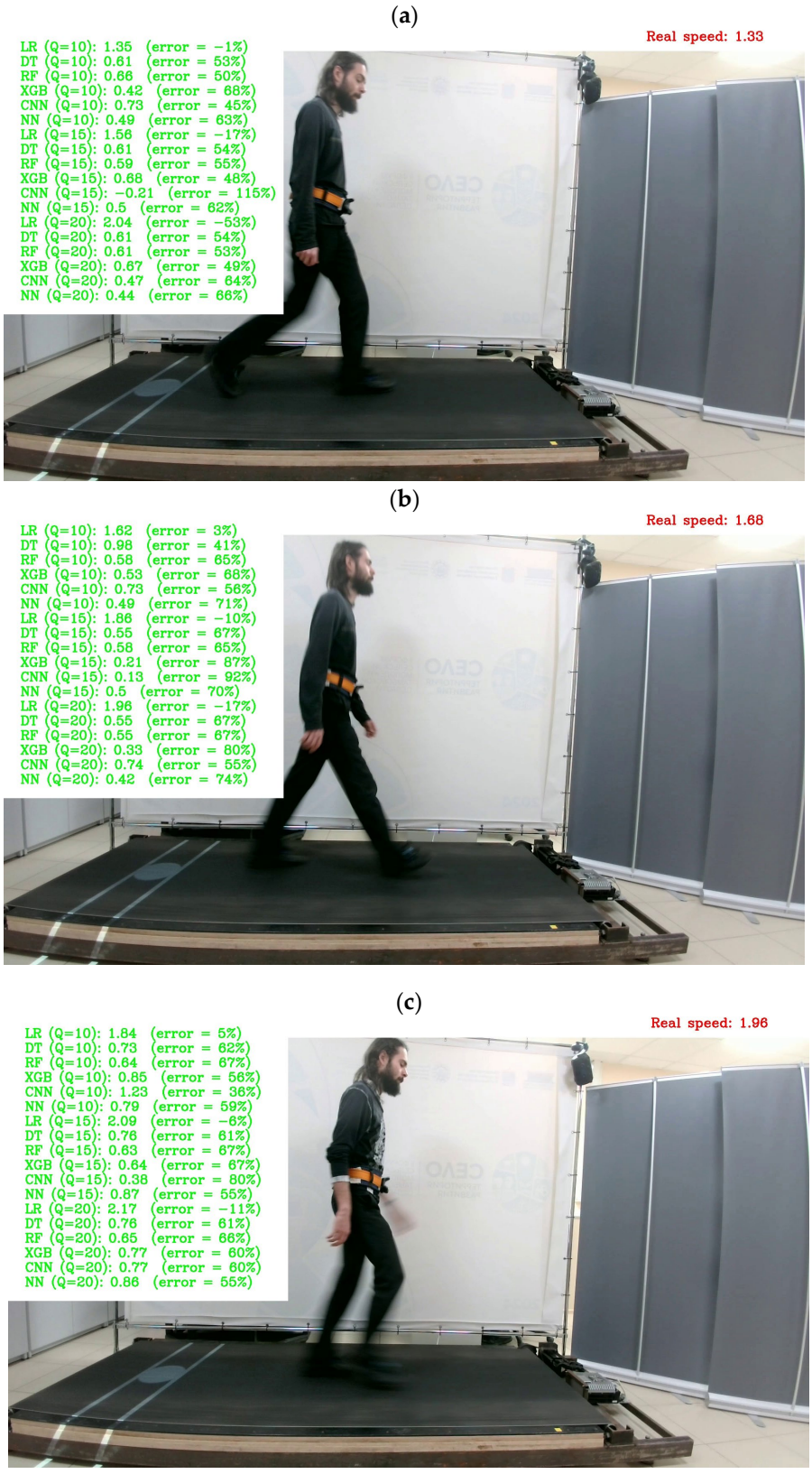


Figure 4. Comparison of models for velocity determination: (**a**) at low speed; (**b**) at medium speed; and (**c**) at high speed. The caption at the top right displays the current treadmill speed reference, while the left side shows the speed predictions for different numbers of analysed frames (10, 15, and 20) across various machine learning models applied to the video frames.

3.4. Experiment on Predicting Human Positions

Further, a comparison of different machine learning algorithms was conducted in solving the problem of predicting human body position. In this comparison, two variables were chosen: Q and W with values of 10, 15, and 20. The training dataset consisted of 75,600 records, while the test dataset contained 24,770 records for each combination of Q and W. This resulted in obtaining 9 different parameter combinations for the previously mentioned model architectures, for each of which the comparison of machine learning algorithms on a control dataset of 1761 values was carried out (Table 5). It should be noted that the visibility variable for each point was ignored when calculating the prediction accuracy, as it was not related to the position data and might affect MAE calculations. The most accurate model for each Q group was highlighted in bold.

Table 5. Results of the third experiment (MAE).

Model	Q = 10			Q = 15			Q = 20		
	W = 10	W = 15	W = 20	W = 10	W = 15	W = 20			
LR	0.0121	0.0133	0.0139	0.0129	0.0140	0.0145			
DT	0.0091	0.0103	0.0101	0.0098	0.0103	0.0102			
RF	0.0099	0.0099	0.0100	0.0098	0.0100	0.0100			
XGB	**0.0071**	0.0075	0.0078	**0.0071**	0.0075	0.0076			
CNN	0.0704	0.0697	0.0618	0.0946	0.0813	0.0980			
NN	0.0113	0.0111	0.0111	0.0111	0.0111	0.0110			

The best results for each model are highlighted in bold font.

Since the results of the second experiment revealed that the most accurate model does not always work correctly in real test conditions, an additional experiment was conducted that includes the use of models within the framework of the C2 method operation. A validation of ML2 model's performance after applying the position prediction (ML3) is accomplished. The ML2 model (LR) and a test video with a recorded track speed that is satisfying to the user are applied for evaluation. The accuracy results of speed detection by the MAE metric from real values considering the use of ML3 models are presented in Table 6.

Table 6. Testing ML3 models in speed detection (MAE, m/s).

Model	Q = 10			Q = 15			Q = 20		
	W = 10	W = 15	W = 20	W = 10	W = 15	W = 20			
LR	**0.4393**	0.5696	0.6359	0.5819	0.6986	0.7325			
DT	1.1806	1.2090	1.2117	1.2057	1.2093	1.2115			
RF	1.2021	1.2017	1.2038	1.2005	1.2010	1.2023			
XGB	3.6×10^7	3.2×10^8	1.6×10^8	3.6×10^7	3.2×10^8	1.6×10^8			
CNN	5.3×10^{10}	4.8×10^{10}	3.0×10^{10}	2.4×10^{10}	3.9×10^{10}	3.6×10^{10}			
NN	1.1×10^8	2.4×10^8	9.0×10^6	6.3×10^7	1.1×10^8	3.8×10^7			

The best results for each model are highlighted in bold font.

The results of Table 6 show that after predicting the position by XGB, CNN, and NN models, certain changes occur in the position of the points, making it difficult for the ML2 speed detection function to work. Therefore, these models are further excluded from consideration. This is an important point, showing that data processed by a machine learning model may not be applicable when sent to another model, despite visual correctness, low error, and so forth. The two-stage testing carried out (first, the models are compared in terms of MAE value when solving the prediction problem, then the error when determining

the speed on the predicted data is compared) is closer to the actual operating conditions of the models.

The performance of ML3 models (average time of 100 model calls) was also evaluated. The results are presented in Table 7.

Table 7. Performance comparison of ML3, in seconds.

Model	Q = 10			Q = 15		
	W = 10	W = 15	W = 20	W = 10	W = 15	W = 20
LR	0.2757	0.3478	0.5083	0.3525	0.4307	0.6383
DT	4.0×10^{-5}	6.0×10^{-5}	8.1×10^{-5}	4.1×10^{-5}	6.0×10^{-5}	6.0×10^{-5}
RF	0.0099	0.0099	0.0098	0.0100	0.0101	0.0102
XGB	0.0027	0.0060	0.0096	0.0031	0.0065	0.0097
CNN	0.0518	0.0395	0.0661	0.0393	0.0709	0.0445
NN	0.0349	0.0340	0.0352	0.0335	0.0352	0.0347

Based on the performance evaluation results, it has been concluded that despite its accuracy, the LR model is not suitable for real-world applications. The calculation time of 0.27 s for predicted points would result in a whole control system being able to process no more than 3–4 frames per second, which is unacceptable for a real-time system. Therefore, the fastest DT model with parameter values of Q = 10 and W = 10 will be moving forward. According to the results of Table 6, this model provides the second-best accuracy. Additionally, according to the results in Table 5, this model is also insignificantly worse than alternative variants.

3.5. Experiment on the Implementation of Neural Network Treadmill Control Methods

The final experiment aims to test the trained models under real-world usage conditions in the implementation of different treadmill control methods. Therefore, software implementation and integration of methods C1–C5 into the existing treadmill control system were carried out, and each method was tested.

The fourth experiment includes several steps based on conducting computer simulations of the methods on a pre-recorded video to assess what speed will be exhibited at each particular time point. This strategy allows evaluating the performance of each neural network method in a safe environment before conducting real tests. The reference speed is the speed set by the user while walking comfortably on the platform in manual mode. In the first stage, preliminary experiments are conducted to determine the coefficient values for methods C3–C5, which allows us to assess the contribution of each component to the final performance of the methods. A small safe zone (24% of the left edge of the frame) is also defined for each method, which corresponds to the zero-speed zone used in the existing treadmill control system.

To briefly reiterate the specifics of how each method works, as detailed in Figure 1 and Section 2:

C1: ML1 (DT selected) improves the positioning accuracy of body points, after which the ML2 (LR selected) model determines the movement speed;

C2: ML3 (DT selected) predicts the person's position through 10 frames, after which ML1 and ML2 are applied to the predicted position values;

C3: methods C1 and C2 run in parallel with coefficients k1 and k2;

C4: the linear control function with coefficient k0 is combined with method C1 using coefficient k1;

C5: linear control function with coefficient k0 is combined with method C1 using coefficient k1 and C2 with coefficient k2.

The results are presented in Tables 8–10. The best result is highlighted in bold.

Table 8. Determination of k1, k2 values for method C3.

k1	k2	MAE (m/s)
0.10	0.90	0.9821
0.20	0.80	0.8334
0.30	0.70	0.6856
0.40	0.60	0.5381
0.50	0.50	0.3913
0.60	0.40	0.2607
0.70	0.30	**0.1956**
0.80	0.20	0.2157
0.90	0.10	0.3020

The best results are highlighted in bold font.

Table 9. Determination of k0, k1 values for method C4.

k0	k1	MAE (m/s)
0.10	0.90	0.3508
0.20	0.80	0.2899
0.30	0.70	0.2326
0.40	0.60	0.1810
0.50	0.50	0.1397
0.60	0.40	**0.1161**
0.70	0.30	0.1230
0.80	0.20	0.1636

The best results are highlighted in bold font.

Table 10. Determination of k0, k1, and k2 values for method C5.

k0	k1	k2	MAE (m/s)
0.10	0.10	0.80	0.8983
0.10	0.15	0.75	0.8237
0.10	0.20	0.70	0.7495
0.20	0.10	0.70	0.8144
0.20	0.15	0.65	0.7399
0.20	0.20	0.60	0.6656
0.30	0.10	0.60	0.7306
0.30	0.15	0.55	0.6560
0.30	0.20	0.50	0.5817
0.40	0.10	0.50	0.6468
0.40	0.15	0.45	0.5721
0.40	0.20	0.40	0.4978
0.50	0.10	0.40	0.5629
0.50	0.15	0.35	0.4882
0.50	0.20	0.30	0.4139
0.60	0.10	0.30	0.4791
0.60	0.15	0.25	0.4044
0.60	0.20	0.20	0.3300
0.70	0.10	0.20	0.3953
0.70	0.15	0.15	0.3205
0.70	0.20	0.10	**0.2461**
0.80	0.10	0.10	0.3116

The best results are highlighted in bold font.

During the selection of coefficients k1 and k2 for method C3, values of 0.70 and 0.30 were chosen, respectively (Table 8). Thus, in method C3, a higher priority is given to the first component (of method C1) when it is combined with C2.

The best results were obtained for method C4 when combining k0 = 0.6 and k1 = 0.4 (Table 9). This shows that the linear component, at a certain proportion, has a positive effect on the performance of the neural network method.

Close results to the reference ones were obtained by combining k0 = 0.7, k1 = 0.2, and k2 = 0.1 for method C5 (Table 10). The positive influence of the linear component can also be observed here.

At the next step, the comparison of methods C1–C5 was carried out with regard to the performed coefficient settings for C3–C5 (Table 11), as well as the visualisation of speed control graphs on the test video (Figure 5).

Table 11. Results of comparison of neural network methods.

Method	MAE (m/s)
C1	0.4131
C2	1.0702
C3 (k1 = 0.7, k2 = 0.3)	0.1956
C4 (k0 = 0.6, k1 = 0.4)	**0.1160**
C5 (k0 = 0.7, k1 = 0.2, k2 = 0.1)	0.2460

The best results are highlighted in bold font.

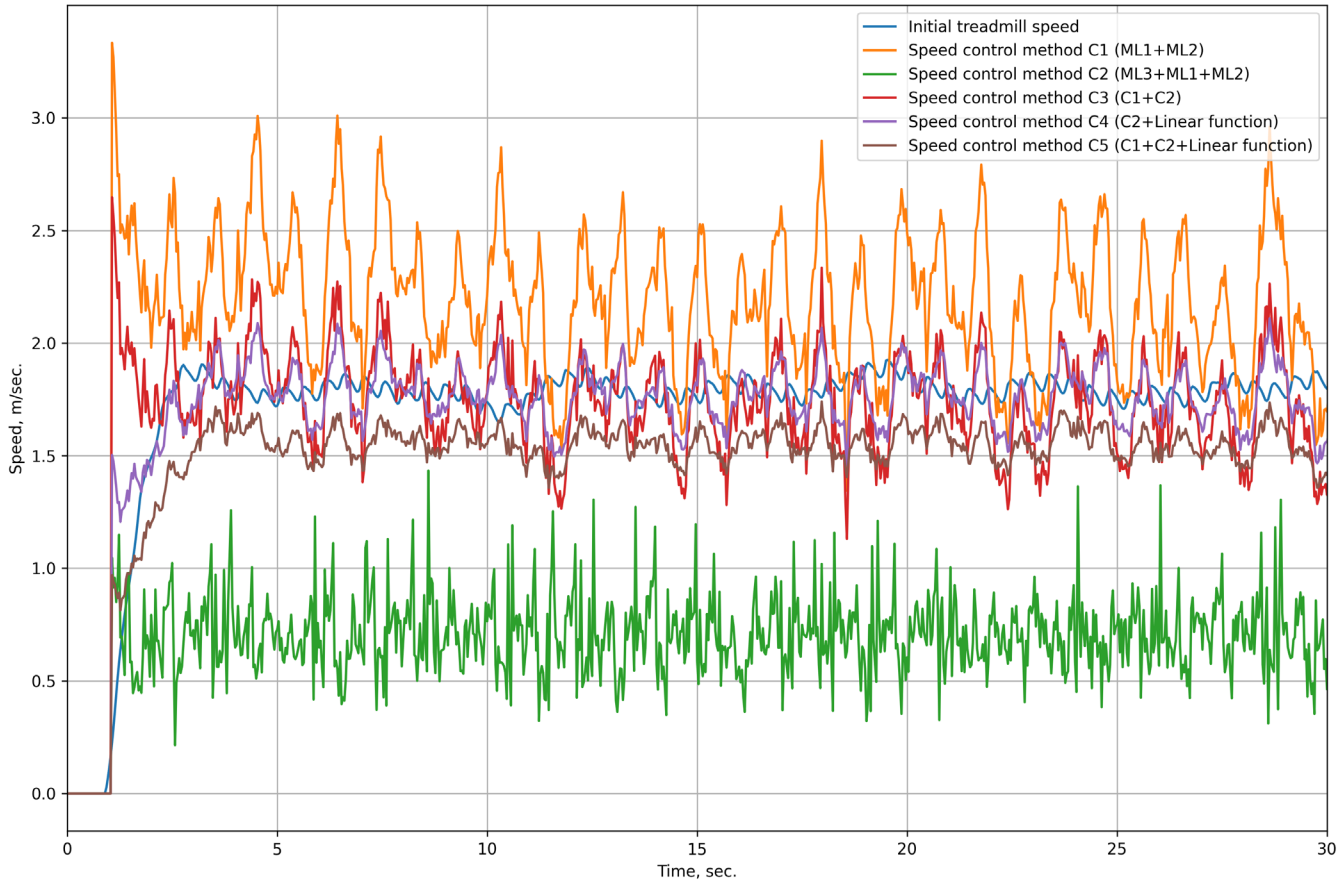


Figure 5. Performance visualisation of different neural network methods. Comparison graphs of treadmill control methods C1–C5 are presented, illustrating the predicted speed values relative to the reference speed set by the user.

According to the results presented in Table 11, the C4 method shows the smallest error in selecting speed values. The error values presented in Tables 8–11 should be perceived as a deviation of the speed calculated by the methods from the speed that was comfortable for a person when recording the test video. Consequently, at this stage the C4 method is

preferred, however, it is necessary to verify the methods under conditions where they are controlling a real treadmill.

According to the preliminary test results, methods C1 and C2 separately did not show high efficiency. The performance of the methods presented in Figure 5 shows that method C1 regulates speed with very large amplitude, while method C2 does not achieve high speed values. The combined methods show results closer to the actual speed control trajectory. Thus, more detailed testing of methods C3-C5 will be conducted in the following phases.

The final stage of the experiment is testing the methods in real conditions. Test trials of methods C4 and C5 with given parameters in real conditions were performed. The trajectory (position of a person in the treadmill) and the treadmill speed are presented in Figure 6 for method C3, in Figure 7—for C4, in Figure 8—for C5.

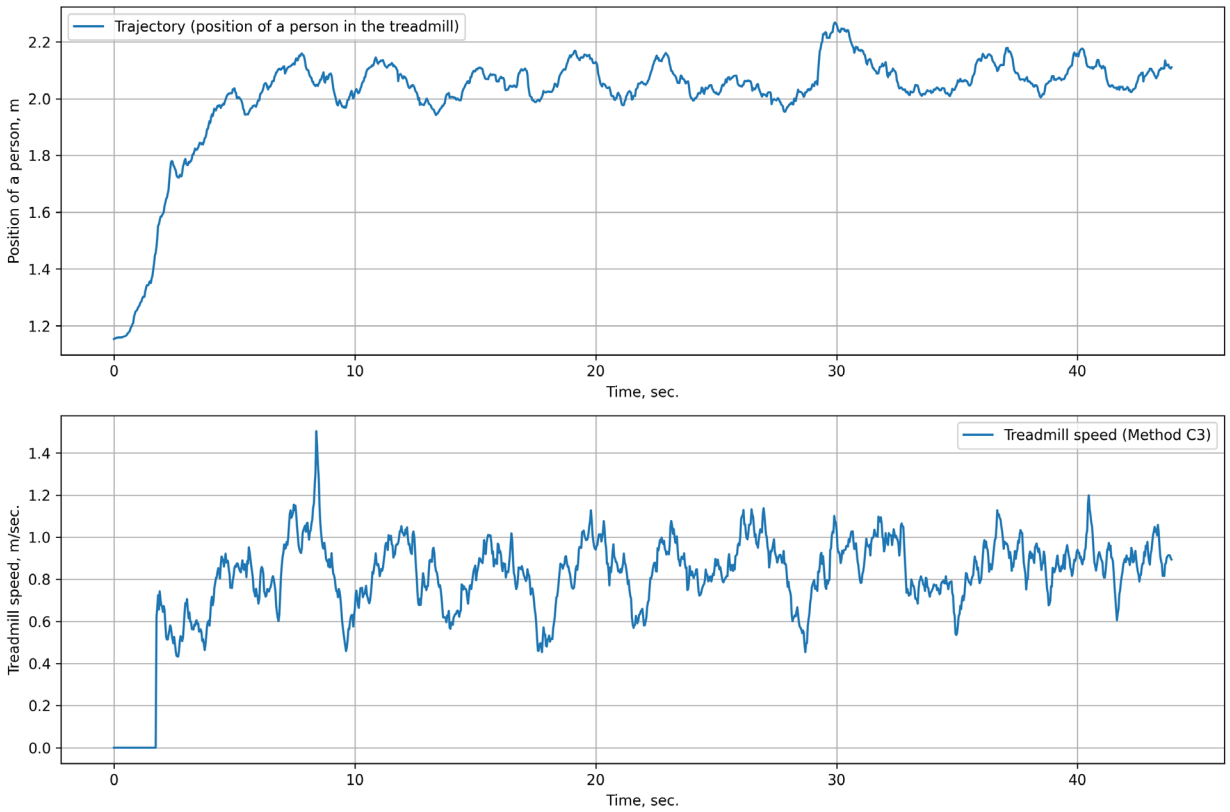


Figure 6. Visualisation of C3 method operation (the position of the person and the speed of the treadmill in this position). The absence of a linear component in method C3 leads to abrupt changes in speed.

Among the three combined methods, C3 was the most ineffective (Figure 6), as its maximum speed was also limited, and the lack of a linear component resulted in a rather abrupt start and stop, which appeared uncomfortable for the user. Furthermore, the speed in it is adjusted with a large amplitude, leading to abrupt changes, which is also uncomfortable for the user.

Method C4 (Figure 7) and C5 (Figure 8) set higher values when adjusting the speed, allowing the person to walk more comfortably without restraining their stride length or their speed. Given that the length of the treadmill's working area is 2.4 m, in method C3, the user comes too close to the edge of the treadmill, which is a negative aspect. Analysing the position of the person for methods C4 and C5 (upper graphs of Figures 7 and 8), it can be seen that with C4 the person can move at a more comfortable speed, staying far enough from the edge of the track, without sudden leaps in speed values. More detailed analyses of neural network methods are presented below in the Section 4.

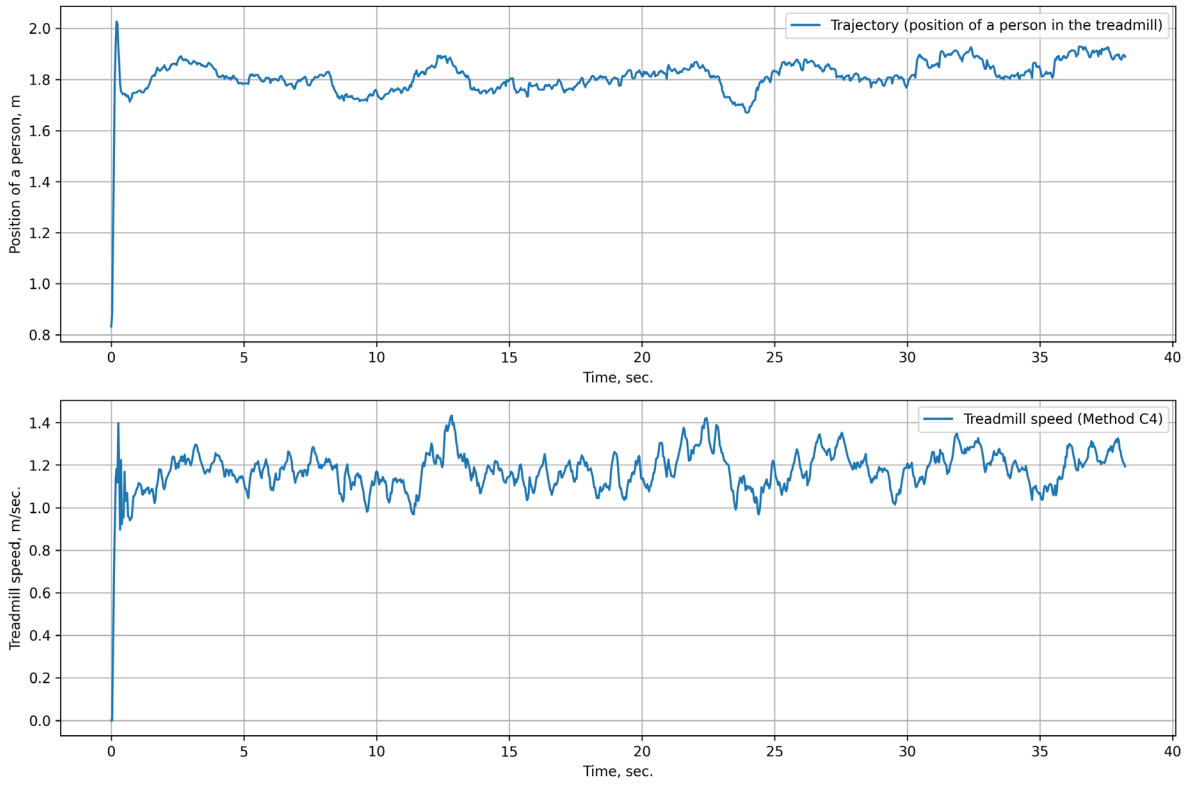


Figure 7. Visualisation of C4 method operation (the position of the person and the speed of the treadmill in this position). The combined approach employed in method C4 maintains a comfortable, smooth speed trajectory throughout the entire time interval.

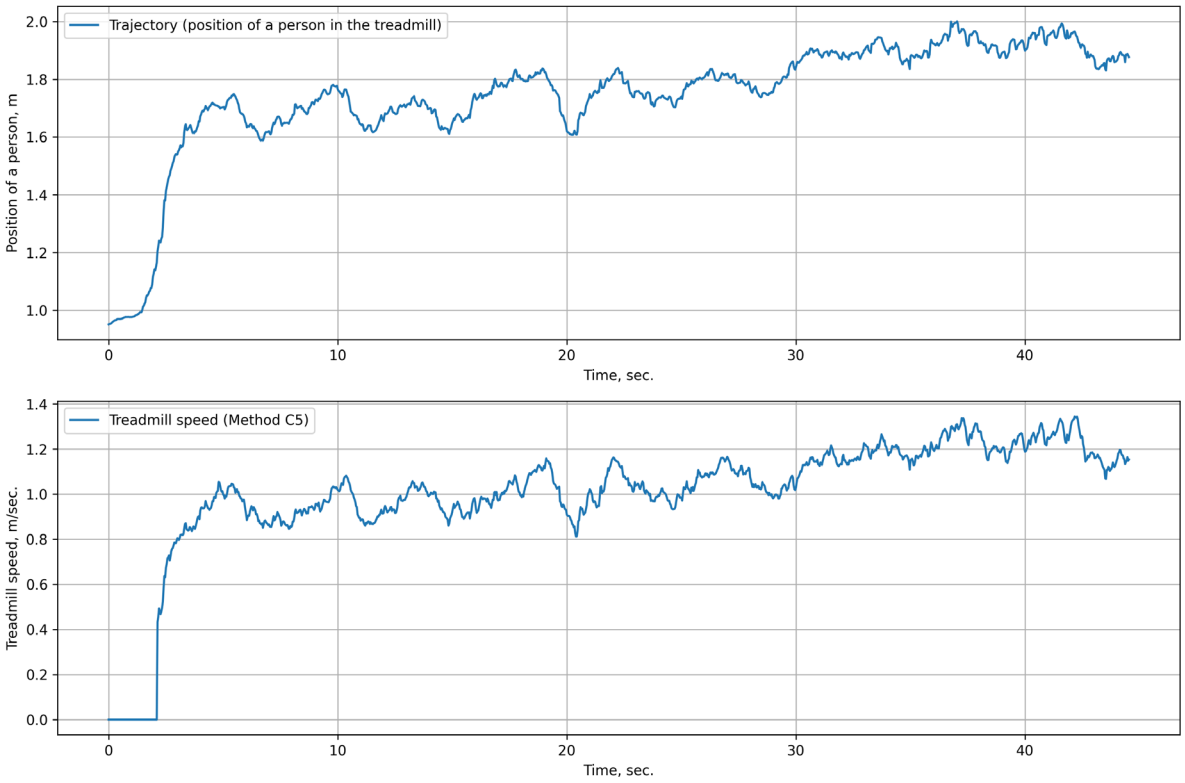


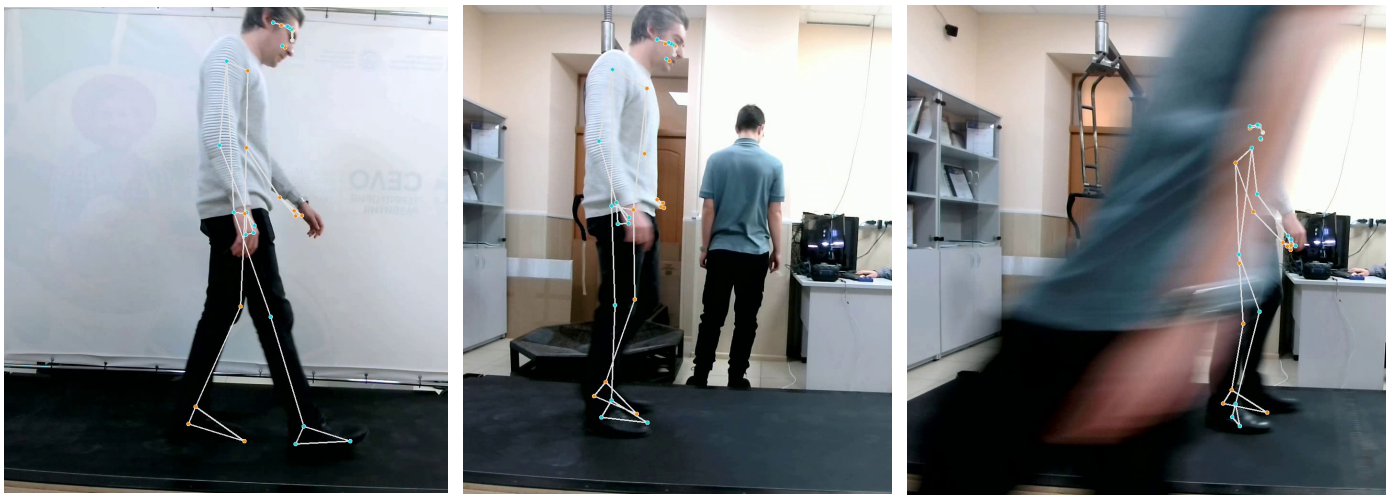
Figure 8. Visualisation of C5 method operation (the position of the person and the speed of the treadmill in that position). Method C5 is characterised by a very smooth start; however, despite incorporating three components, it does not demonstrate any advantages over method C4.

Therefore, the machine learning models obtained in the course of the research have been successfully applied to improve the accuracy of human position and speed prediction performance. The developed models were integrated into the control system of the running platform within the neural network methods.

3.6. Testing the Stability of Computer Vision Technology Under Various External Conditions

As the final stage of the research, the stability of the computer vision technology is tested using MediaPipe Pose as an example under different environmental conditions. This is of great importance, as conditions such as low light, lack of contrasting clothing, or poor backgrounds can have a significant negative impact on the performance of human body model recognition. Consequently, the performance of all machine learning models and neural network methods will also be negatively affected. The results of the conducted testing of computer vision technologies under different conditions are presented below.

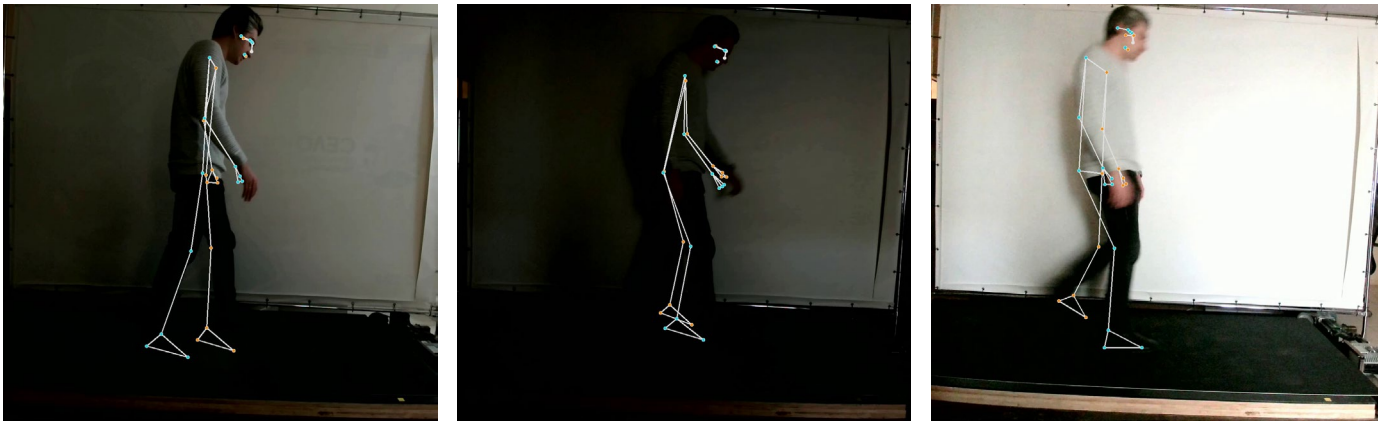
In Figure 9, two scenarios are considered: in the first case, the user's clothes are close in colour to the background and the track canvas, which can make it difficult to recognise the body and legs (Figure 9a); in the second case, there are additional people in the frame, and the white background that could prevent them from entering the frame is removed. For the last situation, two situations are conducted—a person walks in the background (Figure 9b) and directly between the camera and the user (Figure 9c). In all three cases, the performance of the machine learning technique can be assessed positively; there was no loss of the user, and the extraneous interference, although it introduced distortions, was not drastic, which additional machine learning models could not remove. Nevertheless, the presence of bystanders in the frame is undesirable because if the MediaPipe Pose model switches to recognising an additional person rather than a treadmill user, it will lead to significant errors. In the conducted testing, this situation did not occur due to the fact that the MediaPipe Pose was tracking the initially recognised user. In the event that the user was not recognised or lost due to sudden movement and the model started tracking an additional person, this could have been ruled out by additional software checks for sudden position changes and using OpenCV (version 4.11.0.86) software methods to track the user.



(**a**) (**b**) (**c**)

Figure 9. Test fragments of computer vision technology under different conditions: (**a**) non-contrasting user's clothing; (**b**) no white background and an additional person in the background; (**c**) an additional person in front of the camera. The results demonstrate the viability of the computer vision technology (for body model recognition) under real-world conditions in the presence of external interference.

Further testing of the technology's performance at different light levels is then performed in Figure 10: at half the normal light level (Figure 10a), at the minimum light level when the camera exposure has not yet adjusted to the changed light (Figure 10b), and after such an adjustment has been made 1–2 s later (Figure 10c).



(**a**) (**b**) (**c**)

Figure 10. Fragments of low-light computer vision technology tests: (**a**) half of normal; (**b**) minimum level; (**c**) minimum level after exposure correction. The results indicate that the computer vision technology (for body model recognition) remains functional under challenging lighting conditions.

It should be noted that from a technical point of view, low illumination leads to a degradation in the video stream quality and may increase the exposure and shutter speed, resulting in a drop in the number of frames processed per second. In addition, if the luminance changes during system operation, the camera does not immediately adapt to the new light level, which may disrupt or degrade the recognition model's performance (Figure 10b). After increasing the exposure and shutter speed length, there is also an effect of smearing and fuzziness, along with an increase in noise (Figure 10c). The human body model is still recognised, but the quality of such recognition decreases.

Consequently, the considered negative factors can negatively affect the performance of the whole system. Therefore, in case of using computer vision technologies in human-machine systems, it is recommended to provide lighter conditions: absence of extraneous objects, presence of clothing with minimal contrast relative to the background, and acceptable illumination for the room. This will maximise the efficiency of computer vision technologies and the methods based on them.

4. Discussion

The results obtained from both the theoretical and experimental investigations were comprehensively analysed to assess the performance and limitations of the proposed methodology.

During the development of the methodology, the main stages of the study were formulated, and a set of necessary machine learning models was determined: ML1—to improve the accuracy of positioning in overlapping conditions; ML2—to determine the current speed of a person; ML3—to predict their position. These models are combined into various treadmill control methods, ranging from the simpler C1 to C5, which includes all 3 models as well as a linear control function.

The first experiment revealed discrepancies between the model's error estimation during training on the collected dataset and the real-world testing. Although linear regression and XGB models yielded the lowest MAE values on the collected dataset, real-world tests demonstrated that the decision tree model more reliably reconstructed the positions of

key points (e.g., the legs) when occluded by overlapping objects. This suggests that error metrics calculated in controlled training conditions may not fully capture the robustness required for practical applications. A promising solution could be the use of different types of ensemble models, for example, Stacking Regressor by analogy with the study [26], but their use could lead to a significant increase in computational complexity and, consequently, computation time, which is unacceptable within the framework of the considered system. Nevertheless, it is possible to consider such architectures in the future.

The second experiment indicated that the short time window of 10 frames (Q = 10) is optimal for predicting human speed. Extending the buffer to 15 or 20 frames significantly reduced prediction accuracy, likely because additional frames introduced unnecessary noise and delay, thereby diminishing the model's responsiveness. The LR model was chosen as the most accurate and sufficiently productive among the considered models.

The third experiment on human position prediction showed that the accuracy of the models depends on the number of the analysed frames (Q) and the prediction horizon (W). Although LR achieved the lowest MAE in position prediction, its high computational complexity rendered it impractical for real-time control (with processing rates limited to 3–4 frames per second). Moreover, experiments evaluating the sequential application of ML3 results in ML2 for speed prediction revealed that models such as XGB, CNN, and NN could not be effectively chained due to incompatibilities between their output and the input requirements of subsequent models. Therefore, the DT model was selected for integration, as it supports the necessary processing frequency and provides acceptable accuracy when using parameters Q = 10 and W = 10.

Analysing various studies on the application of machine learning to analyse and process data on human movement and motion, it is impossible to make an unambiguous conclusion about the effectiveness of one or another architecture without direct experimental research. On each specific dataset, different models exhibit varying effectiveness, which is consistent with existing studies in this area [27–30].

In the final stage of the experiment, different methods of running platform control (C1–C5), in which the developed machine learning models were integrated, were tested and compared. The most effective method proved to be C4, which realises a combined speed calculation taking into account the relative position of the person and the predicted speed. The determined coefficients (k0 = 0.6, k1 = 0.4) allowed obtaining the minimum MAE value (0.116 m/s) relative to the other methods on the test dataset. Method C5, despite its more complex structure, showed slightly worse results. Subsequently, the methods C3–C5 were tested in real conditions, and their performance was verified. Based on the results of these tests, it can be concluded that using only neural network methods within the C3 framework was not effective enough at this stage of realisation. On the other hand, combining neural network methods with the classical control function showed that both methods (C4 and C5) enable the user to move along the track with sufficient comfort. The differences in the methods in real-world conditions consist of the smoother start of the C5 method.

During the neural network methods testing, the influence of each component (linear and from methods C1 and C2) was examined as part of the performance of methods C3–C5. Detailed analyses (as shown in Tables 9 and 10) demonstrated that the inclusion of a linear component is critical for rapid system response. Neural network models, while powerful, tend to exhibit high fluctuation ranges in real-time, which can lead to unstable control. Thus, integrating a classical control function helps to dampen these fluctuations, ensuring more reliable and responsive behaviour.

As a result, the inclusion of an additional component in the C5 method (speed based on the predicted position) did not significantly improve the result, but it significantly increased the computational complexity of this method. The C3 method showed low maximum

speed and the presence of sudden spurts, which gave negative feedback from users. Thus, the C4 method was preferred among neural network methods.

In addition, during the experiments, the stability of the computer vision technology operation in different conditions (low illumination, low contrast of clothing relative to the background, and absence of a homogeneous background) was tested, and recommendations were made to ensure maximum performance of this technology. Experiments have confirmed that despite the performance degradation under adverse conditions, the system remains functional. These tests underscore the importance of maintaining optimal environmental conditions (adequate lighting, sufficient contrast, and homogeneous background) to maximise the efficiency of the overall system.

The experimental results have revealed an inherent trade-off between predictive accuracy and computational complexity in real-time control systems. On the one hand, high-accuracy models such as linear regression or complex neural network architectures require longer computation times despite exhibiting very low error metrics under controlled conditions (though not always under real-world circumstances). On the other hand, low processing speeds restrict the system's ability to promptly respond to dynamic changes in user behaviour. This limitation hinders the practical deployment of state-of-the-art solutions, such as multi-layer Transformers or LSTM networks, whose performance must be rigorously validated. It is also important to note that this study employed the efficient MediaPipe Pose model, which is optimised for real-time operation. Other models, for instance, YOLO, might provide greater accuracy in segmenting body parts within a frame; however, they often incur higher processing times, potentially several times longer, which is particularly critical in treadmill-based training or rehabilitation systems where system lag can compromise user safety and comfort.

Conversely, models with lower computational complexity, such as decision trees, may exhibit slightly higher prediction errors but deliver significantly faster processing speeds. In real-time scenarios, small losses in accuracy may be acceptable if they enable the system to achieve higher frame rates and more timely responses, ultimately enhancing both safety and user comfort. Therefore, resolving this trade-off requires identifying a balance where reductions in computation time compensate for marginal decreases in predictive accuracy, thereby ensuring the robustness and responsiveness of the system across varying conditions.

The study also revealed several limitations. The first problem is the high computational complexity of some models, such as LR, which limits their use in real-time control systems. Generally, the task of predicting position using machine learning models can lead to artefacts and data distortion, making it difficult to use them later in other models (e.g., ML2 for speed detection). This leads to the conclusion that in solving such a class of problems, it is necessary not only to evaluate the accuracy of models using generally recognised metrics but also to test them in real-world conditions, as models operating independently may be incompatible in practical use.

Another limitation is the need for more extensive testing of neural network methods, including varying the weighting coefficients, increasing the dataset, replacing the linear component with a non-linear one, and using additional machine learning models. A promising direction, which will be explored in further research, is also the development of human tracking modules based on trained machine learning models for recognising emergency scenarios and regulating individual components of the control system [30,31].

Nevertheless, the obtained results show the promising potential of the proposed technology, the positive effect in the form of solving the problems of predicting the position and speed of a person, as well as the possibility of its further adaptation for use in various

human-machine systems, including rehabilitation complexes, simulators, tracking, and monitoring systems.

5. Conclusions

In the course of this research, the technology for improving the accuracy of predicting the position and speed of a person interacting with a running platform using machine learning and computer vision methods was developed and implemented. The proposed methodology included data collection, training and testing of neural network models, and development of five treadmill platform control methods (C1–C5).

Three main tasks have been successfully solved in this research. The first task, aimed at improving positioning accuracy, was solved using a decision tree (DT) model, which in real-world conditions showed the most stable results (at the training stage the MAE was 0.0058 of the frame size) in conditions of overlapping user body parts. The second task, determining the speed of human movement, confirmed the effectiveness of using a short data sequence (10 frames) and the use of a linear regression (LR) model. The third task, related to human pose prediction, demonstrated the importance of an integrated approach to model selection, which led to the selection of a balanced solution in the form of a DT model.

The trained models are further used at the final stage of the experiment during comparative tests of neural network control methods. The selection of weighting coefficients and the comparison of calculated speed values with the reference speed set by the user showed that the most effective approach is the combined method (C4), which integrates the linear function with the speed prediction model in a 0.6:0.4 ratio. Its application allowed the advantages of both linear control and machine learning models to be combined. Nevertheless, the results confirmed the need for further development of this area for more accurate adjustment of the combined approaches' coefficients.

The main results achieved by the study include:

- Application of the decision tree (DT) model to improve positional accuracy in overlapping body parts.
- Determination of the time window optimal length (Q = 10) for predicting the movement speed, and the use of the linear regression model to solve this task.
- Accurate prediction of human position using DT at Q = 10 and W = 10 (MAE < 0.009 of human body size).
- Realisation of a set of combined control methods and testing these combined methods in real-world conditions showed that methods C4 and C5 are applicable.

In summary, the developed technology has shown its efficiency and perspective, which can be the basis for further improvement of control algorithms in human-machine systems.

Author Contributions: Conceptualisation, A.O.; methodology, A.O.; software, A.V.; validation, M.R.; formal analysis, A.O. and D.D.; investigation, A.O. and D.D.; resources, M.R.; data curation, A.V. and M.R.; writing—original draft preparation, A.O.; writing—review and editing, A.O.; visualisation, A.V. and M.R.; supervision, A.O.; project administration, D.D.; funding acquisition, D.D. All authors have read and agreed to the published version of the manuscript.

Funding: The research was carried out at the expense of the grant of the Russian Science Foundation No. 22-71-10057, https://rscf.ru/en/project/22-71-10057/, (accessed on 1 March 2025).

Institutional Review Board Statement: This study was conducted in accordance with the Declaration of Helsinki and approved by the Scientific and Technical Board of Tambov State Technical University (protocol 3 of 15 December 2024, project 22-71-10057). All participants of the study agreed to the informed consent document. All data were anonymous to protect the participants.

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

Data Availability Statement: The datasets are available on request from the corresponding author only, as the data are sensitive and participants may be potentially identifiable.

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

References

1. Mourtzis, D.; Angelopoulos, J.; Panopoulos, N. The Future of the Human–Machine Interface (HMI) in Society 5.0. *Future Internet* **2023**, *15*, 162. [CrossRef]
2. Manakitsa, N.; Maraslidis, G.S.; Moysis, L.; Fragulis, G.F. A Review of Machine Learning and Deep Learning for Object Detection, Semantic Segmentation, and Human Action Recognition in Machine and Robotic Vision. *Technologies* **2024**, *12*, 15. [CrossRef]
3. Kim, J.-W.; Choi, J.-Y.; Ha, E.-J.; Choi, J.-H. Human Pose Estimation Using MediaPipe Pose and Optimization Method Based on a Humanoid Model. *Appl. Sci.* **2023**, *13*, 2700. [CrossRef]
4. Obukhov, A.; Volkov, A.; Pchelintsev, A.; Nazarova, A.; Teselkin, D.; Surkova, E.; Fedorchuk, I. Examination of the Accuracy of Movement Tracking Systems for Monitoring Exercise for Musculoskeletal Rehabilitation. *Sensors* **2023**, *23*, 8058. [CrossRef]
5. Šlajpah, S.; Čebašek, E.; Munih, M.; Mihelj, M. Time-Based and Path-Based Analysis of Upper-Limb Movements during Activities of Daily Living. *Sensors* **2023**, *23*, 1289. [CrossRef]
6. Filippeschi, A.; Schmitz, N.; Miezal, M.; Bleser, G.; Ruffaldi, E.; Stricker, D. Survey of motion tracking methods based on inertial sensors: A focus on upper limb human motion. *Sensors* **2017**, *17*, 1257. [CrossRef]
7. Rapczyński, M.; Werner, P.; Handrich, S.; Al-Hamadi, A. A baseline for cross-database 3d human pose estimation. *Sensors* **2021**, *21*, 3769. [CrossRef]
8. Goh, G.L.; Goh, G.D.; Pan, J.W.; Teng, P.S.P.; Kong, P.W. Automated Service Height Fault Detection Using Computer Vision and Machine Learning for Badminton Matches. *Sensors* **2023**, *23*, 9759. [CrossRef]
9. Thomas, G.; Gade, R.; Moeslund, T.B.; Carr, P.; Hilton, A. Computer vision for sports: Current applications and research topics. *Comput. Vis. Image Underst.* **2017**, *159*, 3–18. [CrossRef]
10. Yu, H.; Du, S.; Kurien, A.; van Wyk, B.J.; Liu, Q. The Sense of Agency in Human–Machine Interaction Systems. *Appl. Sci.* **2024**, *14*, 7327. [CrossRef]
11. Wang, T.; Zhan, W. Design and control a hybrid human–machine collaborative manufacturing system in operational management technology to enhance human–machine collaboration. *Int. J. Adv. Manuf. Technol.* **2024**, 1–11. [CrossRef]
12. Yang, S.; Garg, N.P.; Gao, R.; Yuan, M.; Noronha, B.; Ang, W.T.; Accoto, D. Learning-Based Motion-Intention Prediction for End-Point Control of Upper-Limb-Assistive Robots. *Sensors* **2023**, *23*, 2998. [CrossRef] [PubMed]
13. Vangi, M.; Brogi, C.; Topini, A.; Secciani, N.; Ridolfi, A. Enhancing sEMG-Based Finger Motion Prediction with CNN-LSTM Regressors for Controlling a Hand Exoskeleton. *Machines* **2023**, *11*, 747. [CrossRef]
14. Obukhov, A.; Krasnyanskiy, M.; Dedov, D.; Vostrikova, V.; Teselkin, D.V.; Surkova, E.O. Control of adaptive running platform based on machine vision technologies and neural networks. *Neural Comput. Appl.* **2022**, *34*, 12919–12946. [CrossRef]
15. Wehden, L.O.; Reer, F.; Janzik, R.; Tang, W.Y.; Quandt, T. The slippery path to total presence: How omnidirectional virtual reality treadmills influence the gaming experience. *Media Commun.* **2021**, *9*, 5–16. [CrossRef]
16. Lytaev, S. Psychological and Neurophysiological Screening Investigation of the Collective and Personal Stress Resilience. *Behav. Sci.* **2023**, *13*, 258. [CrossRef]
17. Albort-Morant, G.; Ariza-Montes, A.; Leal-Rodríguez, A.; Giorgi, G. How does Positive Work-Related Stress Affect the Degree of Innovation Development? *Int. J. Environ. Res. Public Health* **2020**, *17*, 520. [CrossRef]
18. Pyo, S.; Lee, H.; Yoon, J. Development of a Novel Omnidirectional Treadmill-Based Locomotion Interface Device with Running Capability. *Appl. Sci.* **2021**, *11*, 4223. [CrossRef]
19. Obukhov, A.D.; Dedov, D.L.; Teselkin, D.V.; Volkov, A.A.; Nazarova, A.O. Development of a stress-free algorithm for control of running platforms based on neural network technologies. *Inform. Autom.* **2024**, *23*, 909–935. [CrossRef]
20. De Luca, A.; Mattone, R.; Giordano, P.R.; Ulbrich, H.; Schwaiger, M.; Van den Bergh, M.; Koller-Meier, E.; Van Gool, L. Motion control of the cybercarpet platform. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 410–427. [CrossRef]
21. Pyo, S.; Lee, H.; Yoon, J. A Sensitive and Accurate Walking Speed Prediction Method Using Ankle Torque Estimation for a User-Driven Treadmill Interface. *IEEE Access* **2022**, *10*, 102440–102450. [CrossRef]
22. Obukhov, A.; Dedov, D.; Volkov, A.; Teselkin, D. Modeling of Nonlinear Dynamic Processes of Human Movement in Virtual Reality Based on Digital Shadows. *Computation* **2023**, *11*, 85. [CrossRef]
23. Chung, J.-L.; Ong, L.-Y.; Leow, M.-C. Comparative Analysis of Skeleton-Based Human Pose Estimation. *Future Internet* **2022**, *14*, 380. [CrossRef]

24. Degen, R.; Tauber, A.; Nüßgen, A.; Irmer, M.; Klein, F.; Schyr, C.; Leijon, M.; Ruschitzka, M. Methodical Approach to Integrate Human Movement Diversity in Real-Time into a Virtual Test Field for Highly Automated Vehicle Systems. *J. Transp. Technol.* **2022**, *12*, 296–309. [CrossRef]
25. Halilaj, E.; Rajagopal, A.; Fiterau, M.; Hicks, J.L.; Hastie, T.J.; Delp, S.L. Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. *J. Biomech.* **2018**, *81*, 1–11. [CrossRef]
26. Bartol, K.; Bojanić, D.; Petković, T.; Peharec, S.; Pribanić, T. Linear regression vs. deep learning: A simple yet effective baseline for human body measurement. *Sensors* **2022**, *22*, 1885. [CrossRef]
27. Meng, D.; Guo, H.; Liang, S.; Tian, Z.; Wang, R.; Yang, G.; Wang, Z. Effectiveness of a Hybrid Exercise Program on the Physical Abilities of Frail Elderly and Explainable Artificial-Intelligence-Based Clinical Assistance. *Int. J. Environ. Res. Public Health* **2022**, *19*, 6988. [CrossRef]
28. Saboor, A.; Kask, T.; Kuusik, A.; Alam, M.M.; Le Moullec, Y.; Niazi, I.K.; Zoha, A.; Ahmad, R. Latest Research Trends in Gait Analysis Using Wearable Sensors and Machine Learning: A Systematic Review. *IEEE Access* **2020**, *8*, 167830–167864. [CrossRef]
29. Trabassi, D.; Serrao, M.; Varrecchia, T.; Ranavolo, A.; Coppola, G.; De Icco, R.; Tassorelli, C.; Castiglia, S.F. Machine Learning Approach to Support the Detection of Parkinson's Disease in IMU-Based Gait Analysis. *Sensors* **2022**, *22*, 3700. [CrossRef]
30. Usmani, S.; Saboor, A.; Haris, M.; Khan, M.A.; Park, H. Latest Research Trends in Fall Detection and Prevention Using Machine Learning: A Systematic Review. *Sensors* **2021**, *21*, 5134. [CrossRef]
31. Patel, A.N.; Murugan, R.; Maddikunta, P.K.R.; Yenduri, G.; Jhaveri, R.H.; Zhu, Y.; Gadekallu, T.R. AI-powered trustable and explainable fall detection system using transfer learning. *Image Vis. Comput.* **2024**, *149*, 105164. [CrossRef]

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.