ktu
1922

CENKER CANBULUT

VIRTUAL REALITY CONTROL METHODS FOR PERIPHERAL DEVICE INTEGRATION AND HUMAN POSTURE ANALYSIS

DOCTORAL DISSERTATION

Kaunas
2026

KAUNAS UNIVERSITY OF TECHNOLOGY

CENKER CANBULUT

VIRTUAL REALITY CONTROL METHODS FOR PERIPHERAL DEVICE INTEGRATION AND HUMAN POSTURE ANALYSIS

Doctoral dissertation
Technological Sciences, Informatics Engineering (T 007)

2026, Kaunas

The dissertation has been prepared at the Department of Software Engineering of the Faculty of Informatics of Kaunas University of Technology in 2017–2025.

The doctoral right has been granted to Kaunas University of Technology together with Vilnius Gediminas Technical University.

Research supervisor:
Prof. Dr. Tomas BLAŽAUSKAS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007).

Edited by: English language editor Dr. Brigita Pantelejeva (KTU Centre of Foreign Languages), Lithuanian language editor Virginija Stankevičienė (KTU Centre of Foreign Languages).

Dissertation Defence Board of Informatics Engineering Science Field:
Prof. Dr. Vidas RAUDONIS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007) – **chairperson**;
Prof. Dr. Rimantas BUTLERIS (Kaunas University of Technology, Technological Sciences, Informatics Engineering, T 007);
Prof. Dr. Ibrahim A. HAMEED (Norwegian University of Science and Technology, Norway, Technological Sciences, Informatics Engineering, T 007);
Prof. Dr. Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Technological Sciences, Informatics Engineering, T 007);
Prof. Dr. Tomas KRILAVIČIUS (Vytautas Magnus University, Technological Sciences, Informatics Engineering, T 007).

The dissertation defence will be held on 10 February 2026, at 10 a.m. at the Rectorate Hall of Kaunas University of Technology in the meeting of the Dissertation Defence Board of the Informatics Engineering Science Field.

Address: K. Donelaičio 73-402, LT-44249 Kaunas, Lithuania.
Phone: +370 608 28 527; email doktorantura@ktu.lt

The dissertation was sent out on 9 January 2026.
The dissertation is available on the website http://ktu.edu, at the library of Kaunas University of Technology (Gedimino 50, Kaunas) and Vilnius Gediminas Technical University (Saulėtekio 14, Vilnius).

© C. Canbulut, 2026

KAUNO TECHNOLOGIJOS UNIVERSITETAS

CENKER CANBULUT

VIRTUALIOS REALYBĖS VALDYMO METODAI PERIFERINIŲ ĮRENGINIŲ INTEGRACIJAI IR ŽMOGAUS LAIKYSENOS ANALIZEI

Daktaro disertacija
Technologijos mokslai, informatikos inžinerija (T 007)

Kaunas, 2026

Disertacija rengta 2017–2025 metais Kauno technologijos universiteto Informatikos fakultete, Programų inžinerijos katedroje.

Doktorantūros teisė Kauno technologijos universitetui suteikta kartu su Vilniaus Gedimino technikos universitetu.

Mokslinis vadovas:
prof. dr. Tomas BLAŽAUSKAS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007).

Redagavo: anglų kalbos redaktorė dr. Brigita Pantelejeva (KTU Užsienio kalbų centras), lietuvių kalbos redaktorė Virginija Stankevičienė (KTU Užsienio kalbų centras).

Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:
prof. dr. Vidas RAUDONIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007) – **pirmininkas**;
prof. dr. Rimantas BUTLERIS (Kauno technologijos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Ibrahim A. HAMEED (Norvegijos mokslo ir technologijų universitetas, Norvegija, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, technologijos mokslai, informatikos inžinerija, T 007);
prof. dr. Tomas KRILAVIČIUS (Vytauto Didžiojo universitetas, technologijos mokslai, informatikos inžinerija, T 007).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje 2026 m. vasario 10 d. 10 val. Kauno technologijos universiteto Rektorato salėje.

Adresas: K. Donelaičio g. 73-402, LT-44249 Kaunas, Lietuva.
Tel: +370 608 28 527; el. paštas doktorantura@ktu.lt

Disertacija išsiųsta 2026 m. sausio 9 d.
Su disertacija galima susipažinti interneto svetainėje http://ktu.edu, Kauno technologijos universiteto bibliotekoje (Gedimino g. 50, Kaunas) ir Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, Vilnius).

© C. Canbulut, 2026

TABLE OF CONTENTS

LIST OF TABLES

LENTELIŲ SĄRAŠAS

LIST OF FIGURES

PAVEIKSLŲ SĄRAŠAS

INTRODUCTION

The Relevance of the Work

Virtual Reality (VR) is emerging as a promising field in the current technological landscape, driven by ongoing development and support [1]. Consequently, research related to VR is showing significant growth and increased attention from the academic community [2].

Input latency, synchronisation problems, and motion sickness significantly impact the development and usability of VR applications. These challenges often arise from limitations in VR control technology, creating bottlenecks in immersive experiences and scientific applications. The immersive VR experience is a topic that the scientific community continues to explore and refine the methods for evaluating what makes a VR experience immersive, considering various sensory inputs and user interactions [3, 4]. However, many studies aim to improve this experience in different aspects, notably the simulations integrated into Virtual Environments (VE). Creating a VR-supported simulation poses a significant challenge because it requires a convenient way to set up a suitable environment. Existing VR setups primarily rely on built-in control methods, such as HTC Vive or Steam Controllers, to enhance user interaction within virtual environments [5]. While effective in specific contexts, these systems often fail to provide a simulation-like experience when interfacing with real-world equipment. Conversely, integrating external peripheral controls, such as a Concept-II rowing machine, introduces unique challenges due to incompatible data transfer rates between components. These discrepancies often lead to motion inaccuracies, visual stutter, and user discomfort during the VR session. Addressing these integration issues is crucial to unlock VR's potential across broader domains. This dissertation introduces innovative solutions for overcoming such barriers. The research mitigates visual stuttering and data inaccuracy issues caused by data latency by proposing four data prediction techniques aiming to provide smooth, real-time data transmission between the external peripheral control and the VR system. Extrapolation by speed using the correction method yields the most appealing results for the developed system.

Despite its proven benefits in therapy and rehabilitation, VR adoption in clinical settings remains limited due to concerns over cost, accessibility, and clinician training [1]. Many research studies investigate off-the-shelf VR input devices in biomechanical exercises to determine if they could be low-cost alternatives to traditional gold-standard solutions [6, 7]. Common findings from these studies indicate significant deviations in accuracy compared to high-cost solutions, particularly during complex movements or when the tracker is at an off-sight angle. Based on these findings, this dissertation showcases the potential of cost-effective VR-based systems for capturing human motion using CNN and RF machine learning models to improve biomechanical analysis. The proposed system employs HTC Vive trackers and deep learning-based motion identification models to detect human posture disorders. The proposed approach relies on predefined scenarios that classify movements as correct or incorrect, providing a convenient tool for decision-making and evaluating the effectiveness of physical training.

The contribution of this dissertation lies in its innovative approach to resolving integration challenges, enhancing the immersive quality of VR systems, and expanding the scope of VR in human posture analysis and training environments. While the proposed methods offer a foundation for addressing issues in integrating peripheral controls into VR environments, the additional contribution of identifying the accuracy of physical training assessments is particularly suitable for clinical use.

Object of the Work

The object of the dissertation is the application of virtual reality technologies for the identification and categorisation of posture disorders using prediction methods, as well as the integration of peripheral devices into VR systems.

Aim of the Work

This research aims to improve the identification and categorisation of human postural motion using off-the-shelf VR input devices and to facilitate the seamless integration of peripheral input devices into virtual reality systems through data-prediction methods.

Tasks of the Work

1. To analyse the current state of integrating peripheral input devices into virtual reality.
2. To analyse the current state of full-body tracking methods for human posture analysis.
3. To propose methods to mitigate integration challenges of peripheral input devices related to input latency, data accuracy, and visual stuttering encountered in virtual reality.
4. To propose methods for utilising virtual reality technology and sensors for exercise tracking and classification.
5. To evaluate the efficiency of the proposed methods for integrating peripheral input devices in virtual reality.
6. To evaluate the efficiency of the proposed exercise tracking and classification methods that employ virtual reality technology and tracking sensors.

Scientific Novelty

1. The proposed correction method for speed-based extrapolation solves challenges related to accuracy, synchronisation, and immersion by eliminating micro-stuttering in VR systems while maintaining high accuracy with integrated peripheral controls.

2. The proposed rehabilitation exercise classification system allows accurate classification of eight exercises, detecting the performed exercise, and assessing the postural health status of a person. The method employs VR trackers placed at key anatomical points for postural tracking, identified nodes for each exercise to calculate joint angles, a CNN-based architecture for exercise data classification, and a detection approach for identifying the performed exercise.

Practical Value

The practical approach combines speed parameters with constant correction, ensuring data accuracy and a stutter-free VR experience for the user. The techniques explored establish a methodological foundation, especially for those integrating peripheral controls in VR systems. The capability to provide comprehensive measurements of the entire body, from simple postures to complex evaluations using VR and depth sensors, offers a practical way to analyse and understand patients' improper movements within a developed framework. This process involves creating a three-dimensional model that accurately reflects the patient's movements, including detailed angle information, frame by frame. Additionally, CNN and RF methods are applied to enhance the examination and analysis process, thereby improving the overall justification of the decision-making process for

1. Prediction using linear interpolation based on position provides sufficiently accurate results but cannot eliminate the stutter issue caused by differences in data recording and transmission rates between peripheral and VR devices.

2. Prediction using extrapolation by speed solves the stutter problem but introduces inaccurate data acquisition with a significant margin between the peripheral device's finished rowing session and the VR's finished rowing session in all sessions, preventing it from being an ideal solution.

3. A constant modifier introduced in prediction using extrapolation by speed, along with a correction method, yields accurate data acquisition in terms of distance and timing. At the same time, it provides stutter-free sessions overall for all exercise sessions. It is the most viable solution for addressing signal discrepancies when integrating peripheral controls into VR.

4. VR tracking systems combined with machine learning methods enable accurate detection of posture disorders after physical exercises. The proposed full-body tracking framework demonstrated that CNN models achieved the highest accuracy in classifying correct and incorrect movement patterns.

Scientific Approval

The experimental results were presented and discussed at three national and two international scientific conferences:

National Conferences

1. [P1b; DE; OA] Augustauskas, Rytis; Kudarauskas, Aurimas; Canbulut, Cenker. Implementation of artificial intelligence methods for virtual reality solutions: a review of the literature // CEUR workshop proceedings: IVUS 2018: proceedings of the international conference on information technologies, Kaunas, Lithuania, April 27, 2018 / edited by G. Capizzi, R. Damaševičius, A. Lopata, T. Krilavičius, Ch. Napoli, M. Woźniak. Aachen: CEUR-WS. ISSN 1613-0073. 2018, vol. 2145, p. 68-74. [Scopus] [FOR: T 007] [Input: 0,333]
2. [P1c; LT; OA] Drasute, Vida; Dzindzeletaite, Gintare; Kelpsaite, Neringa; Drasutis, Sigitas; Canbulut, Cenker. Videogames in education: analysis of videogames and APPS // ALTA'18: Pažangios mokymosi technologijos ir aplikacijos. Žaidybinimas švietime: konferencijos pranešimų medžiaga, 2018 m. gruodžio 5 d. = Advanced learning technologies and applications. Games for education: conference proceedings, 5th of December, 2018 / edited by D. Rutkauskiene and R. Bartkute. Kaunas: Kauno technologijos universitetas. ISSN 2335-2140. 2018, p. 25-29. [Index Copernicus] [FOR: T 007] [Input: 0,200]
3. [T2; LT] Paulauskas, Andrius; Valatkevičius, Tomas; Canbulut, Cenker. Investigation of interpolation methods for virtual rowing simulator // 9th International workshop on data analysis methods for software systems, DAMSS: Druskininkai, Lithuania, November 30 - December 2, 2017 / Lithuanian Computer Society, Vilnius University, Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius: Vilnius University, 2017. ISBN 9789986680642. p. 37. DOI: 10.15388/DAMSS.2017. [FOR: T 007] [Input: 0,333]

International Conferences

1. Presented poster “Tracking Rehabilitation Exercises Using Virtual Reality System” in 4th International Conference on Intelligent Technologies and Applications INTAP 2021 - Norway, Grimstad 13 October 2021
2. [P1d; UA; OA] Canbulut, Cenker; Blažauskas, Tomas. Using virtual reality technologies for full-body tracking in ice hockey training. // Immersive Technologies in Education: Proceedings of the 4th International Scientific and Practical Conference, Kyiv, April 30, 2024 / edited by Nosenko Yu.G. Kyiv: Institute of Digitalization of Education, National Academy of Pedagogical Sciences of Ukraine, 2024. ISBN 9786178330361. p. 4-8. [FOR: T 007] [Input: 0,500]

Articles in Journals

1. [S1; CH; OA] Maskeliūnas, Rytis; Damaševičius, Robertas; Blažauskas, Tomas; Canbulut, Cenker; Adomavičienė, Aušra; Griškevičius, Julius. BiomacVR: a virtual reality-based system for precise human posture and motion analysis in rehabilitation exercises using depth sensors // Electronics. Basel: MDPI. ISSN 2079-9292. 2023, vol. 12, iss. 2, art. no. 339, p. 1-31. DOI: 10.3390/electronics12020339. [Science Citation Index Expanded (Web of Science); Scopus; INSPEC] [IF: 2,600; AIF: 4,266; IF/AIF: 0,609; Q2 (2023, InCites JCR SCIE)] [FOR: T 007, T 009, M 001] [Input: 0,166]
2. [S1; RO; OA] Canbulut, C.; Paulauskas, A.; Blazauskas, T. Prediction of pending data using interpolation and extrapolation techniques for virtual rowing // International journal of computers communications & control. Oradea: Agora University. ISSN 1841-9836. eISSN 1841-9844. 2020, vol. 15, iss. 2, art. no. 3778, p. 1-15. DOI: 10.15837/ijccc.2020.2.3778. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,293; AIF: 4,101; IF/AIF: 0,559; Q3 (2020, InCites JCR SCIE)] [FOR: T 007] [Input: 0,334]

Thesis Organization

This dissertation comprises 166 pages and is structured into four chapters. It includes 132 figures, 27 tables, and references to 108 scientific sources.

The introduction describes the relevance of the research, aim, tasks, objectives, and scientific novelty and practical significance. This chapter also outlines the thesis statements and the scientific approval of the work.

Chapter 1 reviews existing virtual reality control technologies, focusing on both native and non-native integration approaches, and evaluates the effectiveness of these methods in immersive VR environments to identify their limitations and strengths for human posture analysis. The chapter also analyses data-processing methods for input devices used in virtual reality, including both off-the-shelf and peripheral devices, while discussing various techniques such as regression (interpolation and extrapolation), filtering, and machine learning for processing input data, especially for posture monitoring and movement analysis tasks.

Chapter 2 presents experimental evaluations of the HTC Vive Tracker sensor by comparing its positional tracking performance with that of a professional optical motion capture system, Qualisys.

Chapter 3 describes the experimental system, including its setup, data management methods, and predictive modelling techniques, such as interpolation and extrapolation. The chapter concludes with a visualisation of the results and key findings regarding prediction accuracy, smoothness (stutter-free), and system performance.

Chapter 4 focuses on system architecture and its deployment for posture analysis. It details machine learning methods (Random Forest and Convolutional Neural Network) applied for motion classification and anomaly detection. Results are presented for various exercises, showcasing the system's potential in motion tracking.

Conclusion summarizes the key findings, addresses the original objectives outlined in Chapter 1, and discusses the implications of the proposed methodologies for VR-based human posture analysis. The conclusions are structured to reflect the experimental validation of the work, its technical contributions, and its practical applications.

Summarisation (Santrauka) provides a summary of the dissertation. It outlines the research objectives and presents the key findings from the previous chapters, providing a concise overview of the work completed for this dissertation.

1. ANALYSIS DATA PROCESSING METHODS FOR INPUT DEVICES USED IN VIRTUAL REALITY

1.1. Off-the-Shelf VR Input Devices

The degree of immersion is related to realism rather than just the inherent quality of the VR product. Immersive experience depends on perceptual authenticity and sensory engagement provided by the VR environment, influenced by the interaction between realistic sensory simulation and the overall quality of the virtual experience. This interaction also affects users' ease of learning. As VR experiences become more realistic, they offer experiences that closely align with real-world scenarios. For instance, VR control methods have shown significant efficacy in training applications. In manufacturing assembly training, VR enables more efficient task performance with fewer errors, highlighting the impact of immersive control methods on skill acquisition [8]. Integrating controls specifically designed for VR technology is a direct solution that delivers practical, applicable results. Advancements in VR technology have led to diverse interaction techniques, each influencing user experience differently. Traditional VR systems often utilise handheld controllers with buttons, joysticks, triggers, and trackers. These controllers provide precise input mechanisms, making them ideal for tasks that require accuracy, such as object manipulation or navigation within complex environments. However, the abstraction introduced by using off-the-shelf VR tracking equipment, such as HTC Vive trackers, can sometimes diminish immersion, as users must adapt to a control scheme that does not directly replicate real-world interactions. Therefore, the strategic combination of VR control device modalities is necessary to enhance user interaction and simulation realism [9]. This strategic approach is viable if the user does not make sudden or impactful moves during the VR session [10]. Another identified issue in the literature is accuracy, which is significantly affected by the movement of the tracked object, leading to positional drift over time [11]. An alternative approach to addressing such issues is to integrate peripheral input devices into VR. Integrating a control that is not specifically designed for VR is highly likely to yield more stable, accurate VR sessions. Still, it also comes with its own set of challenges.

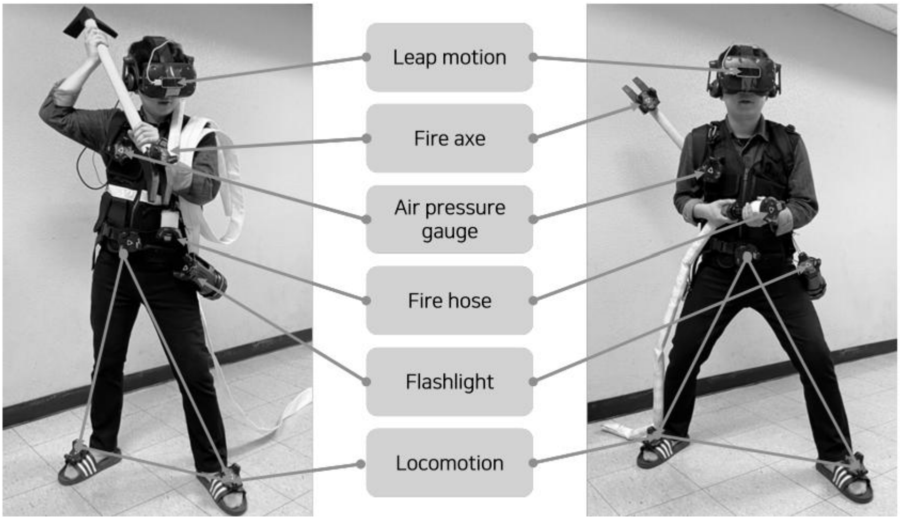


Fig. 1. Integration of VR Controllers for Fire Fighter Simulation [9]

The strategic approach in Fig. 1 demonstrates the integration of HTC VIVE trackers (controllers) into the firefighter training simulation. Due to device incompatibility, these solutions often cannot provide a simulation-like experience because input on an actual device does not match the behaviour of the same input on a VR tracker (controller). To address this impairment, a specialised controller could be implemented. This way, the inputs from the external peripheral device would give the intended device behaviour a natural feel within VR.

The design and functionality of the VR input device are essential for the user's immersive experience in a virtual environment. The analysis of input devices in VR examines the different input types, their impacts, challenges, and suggestions and proposals that could contribute to the field of study. In recent years, research exploring VR input methods has frequently employed devices specifically designed for VR, such as the HTC Vive tracker and various commercially available VR input devices. For instance, the research compared Oculus Touch controllers with hand tracking in a VR medical simulation, noting that while hand tracking offered more natural interactions, the controllers delivered better precision and slightly higher usability scores, highlighting a recurring trade-off between immersion and control accuracy in off-the-shelf systems [12].

The literature on VR input devices often focuses on adapting these tools for specific purposes rather than developing solutions that require extensive customization or data adaptation. An approach where a VR tracker is tied to a user's foot to calculate pedalling speed, and another tracker on the bike's handlebars to determine its turning direction in biking simulations [13]. This approach enables a functional approximation of real-world cycling dynamics. However, achieving a simulation-like experience in a Virtual Environment would require additional parameters, especially if the bike includes a shifting option. The absence of a shifting mechanism would require a different speed calculation method, as torque would become an additional factor, and the rotation of the user's foot would no longer be sufficient to accurately calculate the bike’s speed. Consequently, these hardware and software adaptations often reveal additional challenges that must be addressed to ensure the solutions work as intended. VR controls can further be seen in the adaption of the teleoperation interface of robots [14].

In the presented research example, off-the-shelf HTC Vive trackers designed for VR are securely mounted on the handle of the rowing machine, allowing for precise tracking and capture of the rowing shell's movements [5]. One of the key findings of this research is the technical issues encountered when calibrating the rowing machine with the VR system. While the integration was largely successful, the setup required precise calibration, and concerns arose about potential errors or faults during operation. The need for precise calibration arises from the tracker’s rapid movements, which are tightened onto the rowing machine’s handle. The system adjusts the rowing shell's speed and direction based on the user’s input data via the HTC Vive trackers. A robust and fast pull on the cord produces a more powerful stroke, propelling the shell forward more quickly in the virtual environment. Conversely, a weaker pull results in slower movement. However, this approach could be tricked by positioning the tracker so that the rapid motion could be imitated, leading one to believe that the rowing could reach fast speeds with less effort.

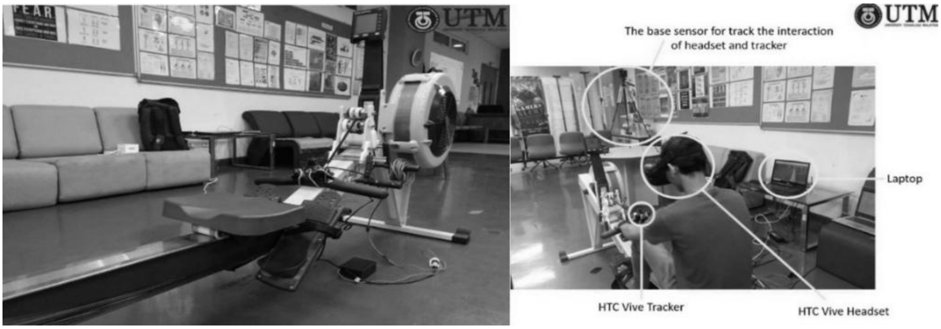

UTM
UTM
The base sensor for track the interaction of headset and tracker
Laptop
HTC Vive Tracker
HTC Vive Headset

Fig. 2. Left: The HTC-Vive Yracker, a key component in the setup, is securely attached to the Concept-II rowing machine’s handle; right: The overall setup, which includes the rowing machine, the VR system, and the user [5]

Additionally, the rowing technique depends not only on how fast the rower pulls the oars, but also on how synchronised the rowing is, as this plays a crucial role in the canoe’s speed. This concern is highly likely to affect the simulation-like experience during rowing sessions. However, the proposed solution in this dissertation utilises the device’s raw data, which involves gathering data exclusively from the Concept-II rowing machine and applying advanced prediction methods, presenting a higher level of integration complexity. This complexity is justified as it ensures the system’s integrity by relying on the inherent characteristics of the Concept-II rowing machine rather than the position and behaviour of the off-the-shelf HTC Vive tracker sensor input data. By basing the system’s responses on the mechanical properties and feedback from the rowing machine, the risk of manipulation through artificial or imitated motions is reduced. This approach enhances the accuracy of the virtual rowing experience, ensuring that the system faithfully reflects genuine physical effort and technique.

An example supporting the development of more adaptable VR control solutions is a study that explores two virtual reality interface control methods, trajectory and positional control for teleoperating the Baxter robot during both gross and fine motor tasks. Their research highlights the importance of intuitive and user-friendly interfaces in VR-based robot teleoperation, particularly for novice users [14]. The study indicates that the positional control interface was significantly more accurate, faster, and precise across tasks such as cup stacking and keyboard pressing, with statistical significance reported in multiple performance metrics (e.g., $p < 0.05$). This finding is particularly relevant for designing adaptive VR control systems tailored to the complexity of different manipulation tasks. Furthermore, while subjective usability scores did not show statistic differences between interface types, the cognitive workload was consistently higher for fine motor tasks across both conditions which underscore the need for dynamic interface designs that adjust to task demands, supporting the dissertation’s argument that adaptable and task-specific VR input paradigms can enhance the precision and usability of VR-based interaction systems, especially when integrating non-native input devices.

Off-the-shelf VR input technologies were used in a recent study that assessed the suitability of various handheld tools for neurosurgical training within a virtual craniotomy environment [15]. The experiment involved three distinct input modalities: a standard HTC Vive controller, a lightweight stylus (VR Ink), and a real craniotome fitted with an HTC Vive tracker for positional tracking. All devices operated within the HTC Vive Pro ecosystem, utilizing lighthouse base stations to maintain spatial registration. Each participant traced anatomical contours on a 3D skull model in VR. The system measured drawing precision as the proportion of points aligned with a predefined template within a 5 mm tolerance. Among the devices, the VR Ink demonstrated the highest mean precision at 63.00%, followed by the standard controller at 60.34%. The craniotome performed the worst, achieving only 53.60%. Spatial accuracy was further evaluated using the Hausdorff distance metric, in which the craniotome again performed least favourably, with a median distance of 6.63 mm, compared to 5.33 mm for VR Ink and 6.29 mm for the standard controller.

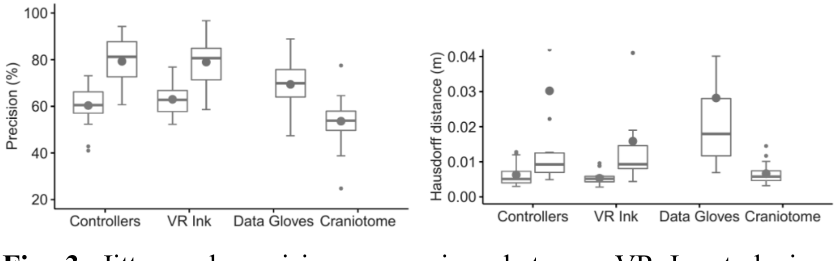


Fig. 3. Jitter and precision comparison between VR Input devices during neurosurgical tasks [15]Fig**. 3** Notably, the study identified substantial jitter in the

craniotome's tracked motion, attributed to occlusion between the HTC Vive tracker and base stations, which compromised tracking stability. This instability was evident even among experienced users, manifesting in visual feedback under degraded performance. These findings, summarized in Notably, the study identified substantial jitter in the craniotome's tracked motion, attributed to occlusion between the HTC Vive tracker and base stations, which compromised tracking stability. This instability was evident even among experienced users, manifesting in visual feedback under degraded performance. These findings, summarized in Notably, the study identified substantial jitter in the craniotome's tracked motion, attributed to occlusion between the HTC Vive tracker and base stations, which compromised tracking stability. This instability was evident even among experienced users, manifesting in visual feedback under degraded performance. These findings, summarized in Fig. 3, illustrate the technical challenges associated with adapting consumer-grade VR trackers to physical surgical instruments, particularly in precision-oriented contexts. The original article highlights the technical challenges of adapting consumer-grade VR trackers to physical surgical instruments, particularly in precision-oriented contexts.

A review examines the use of off-the-shelf VR input devices in robotic teleoperation and interaction tasks, highlighting the distinct roles and limitations of commercial controllers [16]. Among the devices surveyed, the HTC Vive controllers were the most frequently adopted, valued for their precise spatial tracking, wide availability, and robust software development support. These controllers were commonly used in training, telemanipulation, and human-robot collaboration applications, particularly where accurate movement reproduction was essential. Oculus Touch controllers were also well-represented in the literature. They were generally preferred for their ergonomic design and ease of interaction, though some studies reported reduced precision in tasks requiring fine-motor control. The review also identified the Valve Index controllers for their enhanced finger-tracking capabilities, though their adoption was limited. Additionally, Leap Motion was cited in hybrid configurations to enable gesture recognition through optical hand tracking, offering a controller-free alternative that increased interaction realism in specific applications. Despite their strengths, the study identified common challenges across these devices, including limited haptic feedback, inconsistent latency under dynamic conditions, and user fatigue during extended sessions. These insights underscore the necessity for application-specific selection and optimisation of VR controllers, particularly in domains where the balance between precision, usability, and physical strain is crucial to performance and realism.

Another research has developed an experimental VR-based training system to prepare manufacturing operators for industrial robotic tasks without enabling real-time control linkage between the VR environment and the physical robot [17]. Although not explicitly stated, this omission likely reflects the technical and safety challenges of enabling direct VR-to-robot interactions. These requirements may indicate the need for precise communication protocols, access to robot APIs, real-time synchronisation mechanisms, and strict adherence to safety standards to prevent hardware damage or personal injury during live operation. Instead, the training emphasized procedural understanding and operator confidence within a controlled virtual environment, deferring real-world execution to later stages. This design choice underscores the complexities of adapting VR inputs for industrial robotics and highlights the caution required to bridge virtual actions with physical systems in high-risk domains [17].

Reliance on off-the-shelf VR controls as the sole criterion for evaluating the immersive experience of VR applications is limited. Various interactive and perceptual elements influence immersive experience in VR. However, it is crucial to note that factors such as the long-term wearability of VR devices, natural interactions with virtual worlds, and, most importantly, realistic sensory feedback (e.g., haptic, visual, auditory, and thermal) play an essential role in enhancing VR’s immersive experience [18]. Similarly, VR controls significantly improve work zone inspection training effectiveness by providing realistic training environments [8]. While these studies offer valuable insights, their findings are still under discussion. Thus, these results should not be seen as definitive evidence of VR's ability to eliminate human errors in training environments, as learning effectiveness depends on multiple factors rather than a single VR application.

The reviewed literature and case studies demonstrate that off-the-shelf VR input devices enable immersive, interactive experiences, but come with trade-offs among usability, precision, and adaptability. However, recurring issues such as limited fidelity in fine-motor tasks, susceptibility to occlusion and jitter, and lack of real-world haptic feedback constrain their realism and effectiveness in high-precision domains. Furthermore, integrating these devices with physical tools such as craniotomes or industrial robots exposes technical challenges that demand careful calibration, predictive correction, and domain-specific customization. These limitations underscore the need for more adaptable controls. Consequently, this dissertation advances the field by proposing a predictive data integration method for non-native VR input devices, designed to overcome the limitations of off-the-shelf systems and enable more reliable, task-specific virtual experiences and solutions.

The following section explores VR input devices, categorised into three types, motion-tracking, peripheral, and off-the-shelf, and their respective use cases, limitations, and applicable use scenarios.

1.2. Motion Tracking Input Devices

Motion tracking systems are designed to monitor body posture, joint rotation, and movement dynamics, contributing to a diverse range of applications, including rehabilitation, biomechanics, and training simulations. One commonly utilised device is the HTC Vive Tracker, known for its relatively high positional fidelity and low-latency tracking performance under controlled conditions. It effectively supports body tracking tasks in three degrees of freedom (3 DoF), enabling lifelike avatar movement in VR applications such as physical therapy and immersive education [19]. Despite its strengths, notable limitations persist. Vive-based systems often encounter positional drift of the input device over time, particularly during fast or prolonged movements. For example, deviations of 10.4 mm at the knee and 11.3 mm at the ankle were observed during movements at 80 rpm, with error margins increasing as movement intensity rose [19]. These findings underscore the importance of assessing the long-term stability of positional tracking, particularly in scenarios where subtle deviations can impact outcome fidelity, such as joint-angle analysis during rehabilitation sessions.

Latency and network synchronisation further complicate motion tracking in collaborative or distributed VR systems. The study investigated a scenario where two users jointly manipulate a shared virtual object. To reduce latency-related desynchronisation, they proposed a unidirectional physics simulation model managed by a single device, with state data sent to other clients via the Mercury Messaging toolkit. This structure ensured minimal input delay. However, the study also questioned the robustness of such systems when state updates are irregular or incomplete, advocating the use of predictive modelling techniques to accurately interpolate user actions.

Head tracking systems utilise sensors to track the user's head position and orientation, converting these movements into the virtual environment to ensure a consistent relationship between real-world actions and virtual perspectives. This alignment reduces disorientation and increases immersion. A technique using high-speed cameras to measure motion-to-photon latency has revealed that sudden accelerations can impair spatial accuracy. It is essential to minimize latency to sustain VR immersion, as higher latency can cause motion sickness and degrade the user experience. This method records the time lag between a user's movement and the subsequent display update, generating vital data to enhance the responsiveness of VR systems. The research assessed motion-to-photon latency in well-known VR headsets and found that at the onset of sudden motion, average latencies ranged from 21 to 42 ms across devices. With the implementation of motion prediction algorithms, these latencies decreased to between 2 and 13 ms, occurring roughly 25 to 58 ms after the movement began [20]. In scenarios where real-time responsiveness is crucial, latency and data interruptions can cause perceptual inconsistencies, known as "breaks in presence" (BIP). These kinds of inconsistencies are addressed by using deep Echo State Networks (ESNs) embedded in a federated learning framework to anticipate user motion based on historical behaviour [21]. Their solution greatly enhanced tracking reliability in wireless VR environments, particularly under varying bandwidth conditions, offering practical implications for posture-sensitive training settings.

Marker-based motion capture systems, such as Vicon, are widely regarded as gold standards in precision; however, their high cost often limits their applicability. Markerless systems offer greater accessibility and ease of use, but they introduce trade-offs in terms of performance and accuracy. A comparative study on user embodiment found that marker-based systems provided superior avatar fidelity and a stronger sense of body ownership, although their restrictive suits and calibration requirements reduced comfort and hindered ease of adoption [22]. In contrast, markerless systems afforded natural freedom of movement but yielded lower embodiment scores due to visible tracking artefacts and motion jitter. The experiments support the following findings: the HTC Vive Tracker's accuracy and clinical feasibility were evaluated against a gold-standard Vicon system, yielding promising yet nuanced outcomes for VR-based posture-tracking applications [23]. Regarding positional tracking, the HTC Vive exhibited an RMSE (Root Mean Square Error) of 0.58 ± 0.89 cm, which closely aligns with Vicon's reference values and supports its reliability in static positioning. Rotational tracking also demonstrated acceptable performance, with an RMSE of 1.46 ± 0.62° for spinal movement analysis. Notably, no significant angular deviations were observed at 30° and 60° flexion angles, confirming the system's capacity for accurate postural measurement in those ranges. Consistent performance across robotic and human trials further underscored the device's robustness under varying execution conditions. These findings reinforce the HTC Vive Tracker's potential in rehabilitation, motion feedback, and immersive VR-based spinal alignment monitoring. However, several limitations were identified. A statistically significant discrepancy ($p = 0.034$) emerged at 45° flexion, likely due to sensor misalignment or inaccuracies in rotational estimation at intermediate angles. Furthermore, the device showed a systematic overestimation of sacral rotation, suggesting the need for correction algorithms in precision-critical scenarios. Minor tracking jitter and latency were observed by researchers during faster movements, though not extensively quantified, and are potentially linked to wireless transmission delays or IMU limitations. Lastly, the Vive's 90 Hz sampling rate, while sufficient for static or slow movements, falls short of the higher temporal resolution offered by Vicon (100–200 Hz), limiting its utility in high-speed motion capture tasks. These results collectively illustrate the trade-offs inherent in adapting consumer-grade motion tracking systems for posture-specific VR applications, highlighting areas where technical refinement remains essential for clinical-grade integration. Additional experimental results from this dissertation (**see Chapter 2**) further support the reliability of the HTC Vive Tracker in positional tracking tasks. Specifically, our tests demonstrated consistent RMSE values across multiple sessions using static and dynamic distance references, corroborating its potential for clinical and immersive posture-sensitive VR applications.

1.3. Peripheral Input Devices

In the context of this research, peripheral external input devices are physical input devices that are not initially designed for virtual reality but have been modified to operate within VR systems. Unlike off-the-shelf VR input devices, which are inherently designed for VR, external input devices require more extensive technical adjustments to function as intended in VR applications. Latency is a crucial challenge in integrating non-native external input devices. Historical adaptations also highlight the integration challenges of non-native VR devices. The literature has addressed the integration of different frequency domains across various research fields, including computer and electrical engineering, by utilizing interpolation and extrapolation techniques. The implementation of techniques often varies based on the specific problem at hand. In signal processing, for example, interpolation and extrapolation are commonly used to predict missing signals from existing signal samples [24]. The generally expected issue of using the integration of external control for VR can lead to a decrease in the quality of experience (QoE) due to the latency between integrated control vs the VR system [25]. Latency refers to the time it takes for a message to travel through a system [26]. It is often highlighted as a significant limitation in VR applications [26, 27]. The acceptable threshold for perceivable latency for VR is typically under 20 milliseconds [27]. The study found that display latency (the delay between the user's action and the visual feedback) and control latency (the delay between the user's action and the system's response) significantly impact the QoE in a VR environment. Latencies, especially in the control input, caused delays in the display update ranging from 0 to 30 ms. In comparison, delays in the hand controller ranged from 0 to 800 ms, resulting in slower task completion times and lower task accuracy.

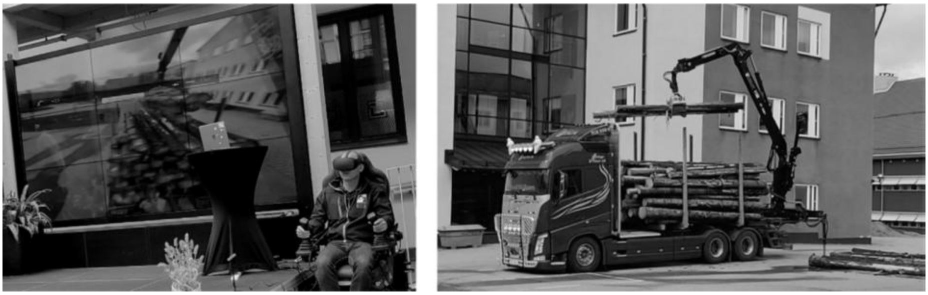


Fig. 4. Left: Operator with a VR Headset and two Non-native Controllers, with the Operator's view shown. Right: The remotely operated crane [25]

This highlights the critical role of low-latency control systems in maintaining effective, efficient operations in VR simulations. The findings suggest that minimising latency should be a priority in the design of VR systems, particularly those that involve remote control or high-precision tasks. This is especially important when integrating external control devices into VR, where even minor delays can significantly disrupt the user experience.

Another field where VR is emerging as a powerful tool is healthcare, where it offers innovative applications for patient treatment and medical training. Its immersive and interactive capabilities provide healthcare professionals with realistic simulations that enhance the precision and effectiveness of various medical procedures. For instance, VR has been employed in physical rehabilitation, making therapeutic exercises more engaging and personalized, improving patient adherence and outcomes [28]. Many studies have been conducted to assess the feasibility and effectiveness of new information technology tools and their design aimed at aiding rehabilitation at home following a stroke or trauma [29, 30], as well as in the treatment of musculoskeletal disorders [31]. In recent years, deep learning methods have gained prominence, with neural networks offering more abstract and comprehensive descriptions of behavioural characteristics [32]. Both machine learning and deep learning techniques deliver outstanding performance in tasks that previously demanded extensive knowledge and considerable time to model. These techniques enable efficient handling and processing of sensor-collected data, leading to more accurate and rapid assessments of human health conditions [33]. New technologies are recognized globally as drivers of change, offering high-reach and low-cost solutions that are user-friendly and widely accepted, particularly for individuals who need continuous progress monitoring, consultation, and training [32, 34].

The Polhemus Isotrak system, a popular choice for early VR applications, suffered from synchronisation issues due to its 30Hz transmission frequency. Researchers at Cambridge mitigated this problem by implementing a Kalman filter to align the Polhemus data with the VR application's frame rate, significantly improving positional and orientation accuracy [35]. However, latency in VR can manifest in various forms, including sensor latency (the time it takes to capture user input), processing latency (the time required to compute and simulate user interactions), and display latency (the delay in rendering and displaying the updated scene). These delays can lead to perceptual inconsistencies, known as "breaks in presence" (BIPs), which disrupt user immersion and hinder skill acquisition.

A notable exploration of these challenges is presented by Iskandar et al. [27], who focus on reducing latency in VR sports training environments, specifically within a virtual squash simulation. The authors identify multiple latency sources, such as delays in user input detection, graphical rendering, and motion simulation. To address these, they categorise latency mitigation strategies into four key approaches: bounding latency (limiting the maximum delay), reducing latency (optimizing software and hardware performance), compensating for latency (predicting user actions to counteract delay), and achieving registration despite latency (ensuring that user actions align with VR updates even under delayed conditions). A significant aspect of their work involves a detailed analysis of predictive algorithms designed to counteract latency effects. These algorithms are categorised into five types: interpolation, extrapolation, filter-based methods, neural network-based methods, and others. Interpolation techniques estimate values between known data points, providing a continuous representation of motion. Extrapolation extends this concept by predicting values beyond the range of existing data. Filter-based methods, such as the Kalman Filter, Extended Kalman Filter, Unscented Kalman Filter, and Wiener Filter, utilise statistical models to reduce noise and stabilize positional data. Neural network approaches, including Perceptron Neural Networks and Back-Propagation Networks, adaptively learn from user input patterns to improve prediction accuracy. This dissertation aligns with the predictive strategies highlighted in Iskandar et al.'s work [27], particularly in its focus on interpolation and extrapolation techniques. These methods are directly applied to maintain consistent motion tracking and mitigate latency effects in a VR environment. The research contributes to the latency management strategy, described as "achieving registration despite latency," ensuring that user actions remain accurately synchronised with VR updates, even under delayed conditions. By leveraging predictive models, this work directly addresses the challenge of maintaining smooth, accurate VR interactions despite inherent delays in data transmission.

Another study developed a VR cycling simulator with two setups to explore peripheral input devices in virtual environments: the Bluefruit BLE-based system and the Elite Axiom trainer [36]. Both systems aimed to provide an immersive cycling experience in VR, translating users' pedal and brake inputs into corresponding virtual interactions. However, the two setups exhibited significant differences in performance, particularly in response delay and data transmission frequency. The Bluefruit BLE setup demonstrated a relatively low response delay, with a typical value of 80 ms, ranging from 65 ms to 100 ms. Its data transmission frequency was recorded at approximately 68 Hz (between 63 Hz and 74 Hz), ensuring consistent and responsive user interaction. In contrast, the Elite Axiom system exhibited significantly longer and more variable response delays, reaching up to 2000 ms in some scenarios, despite a similar data transmission frequency of around 60 Hz. This substantial delay was primarily due to the system's braking mechanism, which struggled to respond quickly to user input, resulting in inconsistent and delayed virtual feedback. The study highlights that maintaining a high data transmission frequency alone is insufficient for achieving smooth VR interaction; the overall system must ensure rapid processing of user input and real-time synchronisation with the virtual environment. These findings underscore the importance of optimizing data transmission and response latency when integrating non-native input devices into VR systems, providing valuable insights into the challenges of ensuring a seamless user experience. Another cycling simulator input device for a VR system was developed, combining a physical mountain bicycle with a virtual landscape using a modular mechatronic architecture [37]. This setup comprises two main components: the Physical System (PS) and the Virtual Reality System, both of which are managed by a Control Behaviour System (CBS). The PS consists of a mountain bicycle mounted on a fixed stand, equipped with an electric DC motor for dynamic braking and a haptic feedback-enabled handlebar that simulates terrain effects. The braking system is controlled by a proportional control algorithm, providing variable resistance, while the handlebar provides tactile feedback via a rotational mechanism. Pedal and speed sensors capture user input, translating these actions directly into the VR environment. The VRS, developed in Unity 3D, mirrors the user's bicycle actions within a 3D virtual landscape, offering two modes: direct user control and automated VR-guided navigation with haptic feedback. The CBS ensures synchronisation between user actions and VR representation, maintaining an average frame rate of 80 FPS with a latency of 25-30 ms. Notably, the system avoids common latency issues associated with non-native VR input devices by optimizing the transmission rate between the physical sensors and the VR application, ensuring real-time responsiveness.

The Hex-Core-MK1 omnidirectional treadmill system enables users to walk freely in any direction while maintaining positional consistency within a VR environment. This system employs a motorized spiral roller mechanism and a proportional control algorithm to adjust roller speed in response to user input. In their study, the authors reported an average latency of 36 ms between user movement and VR response, which remained stable even during rapid direction changes. User tests showed a high sense of presence (4.5/5) and low discomfort (1.2/5) on a Likert scale, indicating that the system effectively replicates natural walking behaviour in VR [38]. The treadmill's robust control system minimizes slippage and adjusts speed smoothly, supporting a wide range of locomotion speeds from slow walking to jogging. Despite these strengths, the study also acknowledged that latency could vary depending on specific user behaviours, such as rapid acceleration or sudden direction changes, which might temporarily exceed this average latency threshold. The research did not address potential latency factors beyond the primary metric. Specifically, the influence of mechanical response delays within the treadmill's motorized rollers, IMU data transmission lag, and control signal processing has not been comprehensively analysed. These factors could introduce additional latency or inaccuracies, especially during rapid or erratic user movements. The study did not provide detailed error metrics for lateral and angular deviations, which could be critical for specific VR applications that require precise body-orientation tracking. Nonetheless, the Hex-Core-MK1 remains a noteworthy example of how peripheral input devices can be effectively adapted for immersive VR experiences. Its robust control system, accurate user tracking, and smooth locomotion integration provide valuable insights for future developments in VR input device integration.

In another instance, integrating the treadmill system can achieve seamless integration without significant latency issues, primarily due to high transmission frequencies between both devices [39]. A treadmill system for VR walking simulations maintained low latency and high responsiveness through optimised data processing and transmission. The system employed a high-frequency data transmission rate, enabling smooth real-time user motion tracking without perceptible delays. This means that systems with well-optimised data capture, transmission, and processing chains can maintain smooth, consistent VR interactions even at high frame rates. This observation underscores the importance of balancing transmission frequency with efficient data handling. As seen in other systems, increasing data frequency without optimizing processing can still lead to latency.

In another study, the Myo armband, a peripheral input device initially developed for general gesture control, was adapted for VR applications [40]. The study highlighted the challenge of maintaining consistent gesture recognition accuracy in VR, where user gestures were occasionally misinterpreted. Although the study did not provide detailed latency metrics, it demonstrated the feasibility of using a peripheral device for gesture control in VR. It revealed the importance of maintaining reliable gesture recognition for seamless interaction.

1.4. Virtual Reality Input Data Processing Methods

In VR applications, data processing is not a technical procedure but a step to ensure that user actions are accurately reflected in the virtual environment. This aims to prevent latency effects and maintain an immersive experience. The nature of VR input data, ranging from motion tracking and gesture recognition to interaction-based devices, involves a data processing phase that converts information into a computer environment. Predictive methods, such as interpolation and extrapolation, enable the estimation of user actions even when direct input data is temporarily unavailable, ensuring continuous movement and interaction during the Virtual experience. Filtering techniques are employed where sensor noise or signal degradation can occur. In addition, machine learning-based approaches aim to leverage neural network adaptability to predict user actions, recognize gestures, and refine sensor data. Hybrid strategies often complement these methods, combining multiple techniques to optimise performance across various scenarios. A fundamental distinction between real and virtual worlds lies in the temporal nature of interactions: natural interactions in the physical environment occur in real time, in synchronous synchrony. In contrast, interactions in virtual environments invariably involve latency due to the time required to simulate cause-and-effect responses [41]. VR experiences are significantly influenced by visual immersion and the quality and precision of input devices used to capture user interactions. These devices range from off-the-shelf VR input devices to sophisticated motion-capture systems and real-world equipment adapted for use. Depending on their origin and degree of integration, VR input devices are widely utilised across various fields, including healthcare, military training, education, sports, and more [42, 43]. Most research fields use these adaptive solutions to discover alternative, effective methods for achieving specific outcomes. This study utilises the off-the-shelf HTC Vive Tracker 2.0 system to detect human anomalies and classify patients into different anomaly categories. Additionally, this dissertation introduces the integration of peripheral input devices within VR environments. Each solution presents its complexities, particularly regarding system compatibility, latency, calibration, and data interpretation. The importance of latency management in VR cannot be overstated, as even minor delays between user actions and visual updates can significantly disrupt immersion, leading to discomfort or motion sickness from visual jitter. Various latency management strategies, including bounding, compensating, and predictive synchronisation, have been explored to address this challenge. The selection of an appropriate method depends on the specific application context, the nature of the input devices, and the desired user experience.

In the following section, we investigate various data processing methods applied in VR, exploring their principles, implementation strategies, and the outcomes.

1.4.1. Interpolation and Extrapolation Methods

Predictive data processing methods, such as interpolation and extrapolation, are not reliable for providing accurate, latency-sensitive results. These techniques are designed to estimate unknown data points from known information. The diagram below (Fig. 5) illustrates a generalized data processing flow for integrating physical input into systems.

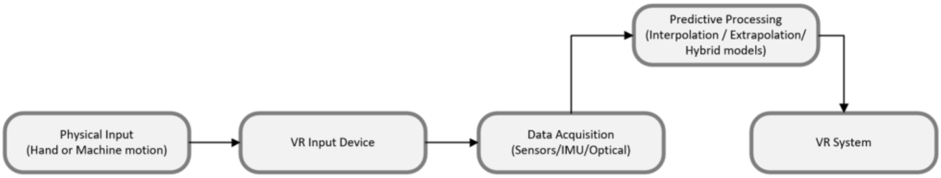


Fig. 5. Conceptual Pipeline for Predictive Processing of Input Signals in VR Systems

1.4.1.1. Existing Regression-Based Prediction Models

Multiple studies address the challenge of increased latency in VR environments using interpolation or extrapolation methods [26, 44]. One of the research aims is to mitigate latency effects by extrapolating future client states [26]. This approach allows the server to incorporate predicted client actions into its updates, thereby synchronizing client-side interactions with server responses. To implement predictive simulation, the study explores various regression techniques, including linear and polynomial regression methods, to enhance motion prediction accuracy. The study initially employed three regression functions: linear, quadratic (second-order), and cubic (third-order) polynomials, each designed to predict user movement using recent data points. The equation represents the linear regression model:

$$y = ax + b; \qquad 1.4.1—1$$

here y is the predicted value, x is the independent variable (e.g., time or frame index), a is the slope of the line, representing the rate of change, and b is the y-intercept, indicating the initial value of the function. Research indicates that while the linear model is computationally efficient, it struggles to accurately capture dynamic or curved motion patterns, particularly when an excessive number of samples is used. To improve prediction accuracy, the authors introduced the second-order polynomial regression (quadratic model), defined as

$$y = ax^2 + bx + c; \qquad 1.4.1—2$$

here a, b and c are coefficients that determine the curvature of the polynomial, and x^2 introduces a quadratic term, allowing the model to capture non-linear changes in movement. The research indicates that the quadratic model performed better than the linear model, offering a more adaptable approach to predicting user movements, particularly in scenarios with moderate curvature. The study also tested a third-order polynomial regression (cubic model):

$$y = ax^3 + bx^2 + cx + d; \qquad 1.4.1—3$$

here a, b, c and d are coefficients of the third-order polynomial providing the cubic model of the movement, it proves overly sensitive to minor fluctuations in the input data, leading to excessive prediction errors. Consequently, the cubic model was discarded in favour of the quadratic approach.

As can be seen, the functions used here are generic and are commonly used in regression analysis. However, their straightforward application demonstrates the foundational effectiveness of predictive modelling in VR [26, 44]. The simplicity of these models also highlights a potential area for further optimisation. In advanced VR applications, regression functions can and should be customized by incorporating weighted factors and multivariable inputs. In this dissertation, regression functions are applied and specifically adapted to the characteristics of VR input data, with experimental validation on a functional system.

According to our analysis of regression techniques used to address incompatibility issues in VR, the foundational techniques remain valuable for motion prediction. Recent literature increasingly explores hybrid and learning-augmented models. However, the practical adoption of such methods in VR remains contingent on their real-time feasibility, where latency, computational load, and frame-rate consistency are essential. A prior study examined predictive data estimation in a virtual rowing scenario, where asynchronous input signals from a physical rowing machine resulted in distance inaccuracies and visual stutters due to temporal desynchronisation [45]. The work implemented and tested three regression-based techniques: interpolation by position, extrapolation by speed, and extrapolation with correction, demonstrating their respective limitations and strengths when applied in a real-time VR context. Interpolation by position offered positional accuracy but caused noticeable stuttering. Extrapolation by speed introduced significant positional drift; however, the corrected extrapolation method, which included a constant factor, reduced timing errors and improved motion continuity. Although focused on a specific application domain, the study helped clarify how simple predictive models can be adapted for use in a VR system.

1.4.1.2. Hybrid Regression-Kinematic Prediction

Unlike existing approaches that emphasize endpoint prediction, a study focuses on a continuous-trajectory prediction model, enabling real-time estimation of hand paths in VR [46]. This approach addresses the challenge of maintaining accurate, smooth hand tracking for effective user interaction in VR environments. The authors identify that rapid and unpredictable hand movements can lead to mismatches between real and virtual environments, resulting in noticeable delays and reduced interaction accuracy. To overcome this, they propose a hybrid model that combines classical kinematic interpolation with regression-based adjustments, enabling dynamic adaptation to user movements while maintaining a physics-based foundation.

The model's classical component is derived from kinematic equations that predict hand position as a function of time, incorporating higher-order derivatives such as velocity, acceleration, jerk, snap, and crackle. The fundamental kinematic equation used is:

$$S(t_0 + t) = S(t_0) + vt + \frac{1}{2}at^2 + \frac{1}{6}jt^3 + \frac{1}{24}st^4 + \frac{1}{120}ct^5; \qquad 1.4.1—4$$

here, $S(t_0 + t)$ denotes the predicted hand position at time $t_0 + t$, where v is velocity, a is acceleration, j is jerk, s is snap, and c is crackle. These terms are carefully selected to capture the complexity of human hand motion, as each derivative addresses a specific aspect of movement dynamics. Velocity accounts for the hand's current speed, acceleration captures the change in speed, jerk represents sudden shifts in acceleration, snap addresses rapid changes in jerk, and crackle accounts for even finer variations in motion. Including these higher-order derivatives is essential for modelling the abrupt, ballistic movements often observed in VR environments. However, since real hand movements rarely follow smooth or constant acceleration, the model incorporates regression-based coefficients $\alpha_n(t)$ for each term, allowing the model to adjust dynamically based on user data:

$$S(t_0 + t) = S(t_0) + \alpha_1(t)vt + \alpha_2(t)at^2 + \alpha_3(t)jt^3 + \alpha_4(t)st^4 + \alpha_5(t)ct^5; \qquad 1.4.1—5$$

here the coefficients $\alpha_n(t)$ are time-dependent and optimised through regression, allowing the model to dynamically adjust to the changing nature of human movements without requiring manual calibration. This hybrid approach effectively balances the predictability of classical physics with the adaptability needed for real-world VR applications.

The research was validated through a user study involving 20 participants performing two types of tasks: structured (3D pointing) and unstructured (VR gaming). Data collection utilised an OptiTrack motion capture system (100Hz) alongside an Oculus Quest headset, providing precise spatial tracking. Prediction intervals of 100ms, 200ms, and 300ms were tested, with model accuracy evaluated using RMSE and Mean Absolute Error (MAE). Results showed the model achieving RMSE values of 0.80 cm, 0.85 cm, and 3.15 cm for the respective intervals, significantly outperforming both naive and classical baseline models, particularly at longer prediction intervals. The model maintained consistent accuracy across different users and activities without requiring individualized calibration, demonstrating its generalizability. Additionally, it could reconstruct lost hand trajectories during temporary tracking failures, a critical feature for maintaining interaction continuity. A follow-up study with three new participants and two new tasks confirmed the model's applicability, achieving similar prediction accuracy with previously unseen motion patterns.

However, the model’s performance declines for prediction intervals exceeding 340ms, where error rates increase. It is also primarily validated for ballistic (fast, directed) hand movements, which limits its applicability to slower, steering-type motions. Despite these limitations, the study demonstrates a practical approach to continuous hand trajectory prediction in VR, striking a balance between the predictability of classical kinematics and the adaptability of regression.

The examined studies demonstrate the use of interpolation and extrapolation to reduce latency and enhance responsiveness in VR applications. Repeatedly, it is essential to note that the scope of existing research remains relatively narrow when focusing on predictive integration of peripheral input devices. Most prior work applies predictive models to standard motion-tracked body parts or off-the-shelf input data, with limited emphasis on systems that involve external physical equipment or custom sensor platforms. Furthermore, despite several practical applications of regression-based and hybrid kinematic models, the broader methodological landscape lacks comprehensive experimental evaluations of non-traditional motion signals across varying interaction conditions. This dissertation addresses this gap by experimentally validating interpolation- and extrapolation-based predictive functions specifically targeting peripheral input devices.

1.4.2. Filter-Based Data Correction Methods

Filter-based data correction techniques integrate input from motion platforms, exercise equipment, or custom sensors to enhance accuracy. Unlike interpolation and extrapolation methods, which predict missing or future data points, filtering techniques aim to refine existing data streams by removing noise, reducing jitter, and correcting the sensor drift or latency. These techniques are fundamental when the input hardware operates asynchronously with the VR system's internal update loop, or when sensor quality introduces instability into the virtual scene for case-specific reasons. By applying filtering techniques, systems can achieve a more stable and accurate representation of user movement or environmental feedback. These filtering methods are often used in conjunction with sensor fusion strategies, enabling VR systems to adapt to multiple noisy or delayed input data sources. In latency-sensitive systems that rely on embedded rotary sensors or IMUs, techniques such as Moving Median Smoothing (MMS) have been applied to suppress jitter and outlier noise in real-time data to stabilize angular signals in a haptic open glove designed for immersive motion capture [47]. The diagram shown in Fig. 6 illustrates the bidirectional relationship between virtual reality systems and physical devices. On one hand, data originating from the virtual environment, such as the motion of a virtual object or avatar, is transmitted to physical output systems, like motion platforms, which must interpret and reproduce that motion with real-world accuracy. On the other hand, signals from physical input devices, such as rowing machines, IMUs, treadmills, or industrial tools, must be captured, filtered, and synchronised with the VR system's update cycle to ensure responsiveness. In both directions, filtering techniques aim to stabilize sensor readings, reduce latency and jitter, and correct drift or signal irregularities.

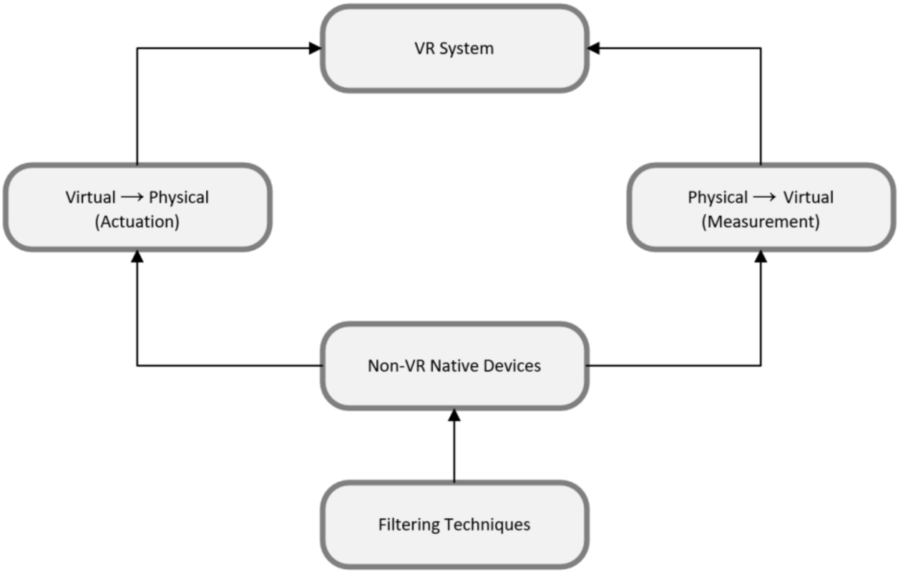


Fig. 6. Data Flow between VR systems and Physical Devices

The following sections highlight research that has implemented filters to enhance the performance and reliability of VR experiences that involve physical hardware inputs.

1.4.2.1. Exponential and Simple Smoothing Filtering Methods

Simple smoothing techniques, such as the Exponential Moving Average (EMA) and low-pass filters, are used in virtual reality systems to attenuate signal noise, suppress jitter, and stabilize motion data from input devices. These methods operate directly on the incoming data stream, eliminating the need for complex modelling or sensor fusion, and thus making them computationally efficient and well-suited for real-time applications.

The research that simulates surfing dynamics using a 6-DoF motion platform using VR, aims to replicate aquatic motion and user-initiated paddling forces in a virtual environment with high physical realism [48]. One of the key technical contributions is the use of a filter-based motion-mapping algorithm that synchronises the VR-rendered surfboard dynamics with the motion platform in real time. To ensure high-fidelity reproduction of surfboard acceleration without the limitations of traditional washout filters (which dampen signals and introduce delay), the authors implemented an Exponential Moving Average (EMA) filter. This filter was selected for its responsiveness to high-frequency motion input, enabling smooth and immediate feedback on minor wave-induced disturbances and user-driven forces. The filter is applied to linear and angular acceleration vectors derived from the surfboard's physics-based movement in Unity. The smoothed signals are then scaled and mapped to motion platform commands, ensuring the platform responds consistently within its physical constraints while closely tracking the virtual surfboard's motion profile. The system supports multi-axis translation (surge, sway, heave) and rotation (pitch, roll, yaw) by deriving platform commands from filtered kinematic parameters such as:

$$Surge = SF_1 \cdot a_f^z, \quad Sway = SF_2 \cdot a_f^x, \ \phi = SF_3 \cdot \alpha_f^y; \qquad 1.4.2—1$$

here, a_f^z, a_f^x, and α_f^y are the filtered components of the acceleration and angular acceleration vectors along the Z, X, and Y axes, respectively, while SF_i denote manually calibrated scaling factors that align the magnitudes of virtual motion with the platform's actuation range. This mapping enables the platform's surge (forward and backward), sway (side-to-side), and yaw (rotational) movements.

Further, vertical heave was driven by the filtered vertical velocity component v_f^ypitch and roll were derived using a separate transformation pipeline based on the surfboard's orientation vectors. The system computed the normalized forward vector $\vec{f}_{norm}$ of the surfboard and generated a corresponding flattened right vector $\vec{r}_{norm}$ via a cross product with the global up vector. These were then projected into the local coordinate system and converted into pitch and roll angles using inverse tangent operations:

$$Pitch = tan^{-1}\left(\frac{f_{local}^y}{f_{local}^z}\right), \quad Roll = tan^{-1}\left(\frac{r_{local}^y}{r_{local}^x}\right); \qquad 1.4.2—2$$

These formulas enabled the platform to simulate subtle surfboard tilts induced by both environmental wave disturbances and user-driven interactions.

To evaluate this system's perceptual and technical effectiveness, the authors conducted a user study involving 17 participants. The study assessed perceptions of acceleration magnitude across different virtual wave intensities (0.5 m/s^2, 1.5 m/s^2, and 3.0 m/s^2) and the user's sense of control during a paddling-based navigation task. The findings indicated that participants reliably distinguished between acceleration levels and that subtle "ocean ripple" cues preserved through EMA filtering significantly enhanced their perception of movement realism ($p < 0.001$). Moreover, participants rated the experience highly for realism, harmony, and involvement, while reporting minimal symptoms of simulator sickness. These results empirically validate the EMA filter's effectiveness in smoothing virtual motion signals without dampening their temporal structure, allowing for precise and naturalistic physical feedback.

The system developed in this research demonstrates filtering to stabilize physical feedback in VR; its application context differs fundamentally from the one addressed in this dissertation. Their approach focuses on translating virtual surfboard motion into physical actuation via a motion platform, with filtering applied to smooth the signals before they are sent to the hardware. In contrast, this dissertation addresses the integration of physical input from a real-world rowing machine into a virtual environment. Because the signal originates from a physical device and is intended for virtual representation (not physical output), there is no need to smooth or stabilize motion for hardware execution. Furthermore, the physical signals from the rowing machine are already structured and rhythmical, reflecting intentional, user-driven actions. Applying filters in such a case would risk weakening critical signal transitions, such as the beginning of a rowing stroke, which are crucial for accurate synchronisation. Therefore, while both studies aim to align data across physical and virtual layers, filtering was not applicable in this work due to the nature and direction of the input stream.

1.4.2.2. Kalman Filtering Methods

The Kalman filter is a recursive algorithm commonly used to reduce noise, correct drift, and estimate the actual state of a system from uncertain or delayed sensor data. It is especially effective when processing signals from hardware components that operate asynchronously with a VR system's internal timing, producing unstable output. The filter refines its estimate using prior values and new measurements through alternating prediction and correction steps. This makes it particularly suitable for applications involving inertial sensors, pressure-based inputs, or other physical measurement systems requiring reliable motion tracking and signal refinement. One relevant implementation involves a wearable leg-motion sensor system that enables seated users to control a VR avatar using lower-limb gestures. The system combines a MEMS-based IMU with a pressure sensor and communicates wirelessly with the VR application via an HC-06 Bluetooth module operating at 9600 bps. To mitigate the noise and drift in raw accelerometer and gyroscope signals, the authors implemented a Kalman filter that fuses both sources to estimate the knee joint's angular state in real time.

The filtering process is defined by the classical two-step Kalman algorithm. The prediction step estimates the current angular state based on previous estimates and control input:

$$\hat{x}_{k\,|\,k-1} = A \cdot \hat{x}_{k-1|\,k-1} + B\ \cdot u_k; \qquad 1.4.2—3$$

here $\hat{x}_{k\,|\,k-1}$ is the predicted state estimate, A is the state transition matrix, $\hat{x}_{k-1|\,k-1}$ is the previous corrected state estimate, B is the control-input model matrix and u_k is the control vector at time step k. The correction step adjusts this prediction using new sensor measurements z_k, weighted by the Kalman gain K_k:

$$\hat{x}_{k\,|\,k-1} = A \cdot \hat{x}_{k|\,k-1} + K_k\ \cdot \left(z_k - H \cdot \hat{x}_{k|\,k-1}\right); \qquad 1.4.2—4$$

here K_k is the Kalman gain.

This fused estimate is used to detect stepping motions, where angular velocity ω is compared against empirically defined thresholds to classify the user's movement as walking or running. For example:

$$speed\ state\ = \begin{cases} walking, if\ \omega < \omega_{treshold} \\ running, if\ \omega > \omega_{treshold} \end{cases} \qquad 1.4.2—5$$

here ω is the angular velocity, and $\omega_{\text{threshold}}$ is the classification threshold.

Five seated participants performed forward, backward, and lateral leg motions in their experimental validation. The system achieved an average recognition rate of approximately 96.7% for both forward and backward steps, while lateral motions were recognized with lower accuracy, at 81.7% for left steps and 85.3% for proper steps. While the authors attribute this to ambiguity in seated lateral leg movement, they do not report system latency or sampling performance metrics. The system focuses on real-time estimation and classification, without relying on predictive modelling or trajectory forecasting.

Although the system demonstrates effective signal stabilization using a Kalman filter, its recognition performance varied across movement directions. The relatively lower accuracy for lateral steps may also relate to the system's modest update frequency. Based on the HC-06 Bluetooth module operating at 9600 bps and typical IMU packet sizes, the effective data rate is approximately 20–30 Hz. While this update frequency was likely sufficient for the authors' intended goal, basic directional motion classification, it falls below the standard VR tracking rate, typically below 60 Hz. As such, while the system performed well within its intended scope, its lower temporal resolution may pose limitations.

Another study uses Kalman filtering on a virtual reality micro-manipulation system developed for surgical simulation, demonstrating how it can enhance spatial accuracy when commercial VR hardware is paired with an input device [49]. The system combines an Oculus Touch controller with a 9-DOF IMU attached to a microsurgical forceps, enabling control within a confined 12 mm × 12 mm workspace. Native tracking from the Oculus controller alone exhibited excessive jitter and lacked the spatial precision required for microsurgical tasks. To address this, a Kalman filter was used to fuse coarse Oculus position data with high-frequency IMU acceleration data, enabling noise suppression and predictive estimation. The filter uses a six-dimensional state vector that represents 3D position and velocity:

$$x = [p_x, v_x, p_y, v_y, p_z, v_z]^T; \tag{1.4.2—6}$$

here p_x, p_y, p_z are the 3D positional components, and v_x, v_y, v_z are their corresponding velocity components.

The prediction step is formulated as:

$$\hat{x}_{k|k-1} = A \cdot \hat{x}_{k-1\,|\,k-1} + B \cdot u_{k-1}, P_{k|k-1} = A \cdot P_{k-1\,|\,k-1} + A^T + Q; \tag{1.4.2—7}$$

here u_{k-1} is the IMU-derived acceleration input and A, B, and Q represent the transition, control, and process noise matrices, respectively. The correction step incorporates positional updates from the Oculus controller:

$$K_k = P_{k|k-1} \cdot H^T (H \cdot P_{k|k-1} \cdot H^T + R)^{-1} \tag{1.4.2—8}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \cdot \left(z_k - H \cdot \hat{x}_{k|k-1}\right), P_{k\,|\,k} = (I - K_k \cdot H) \cdot P_{k|k-1} \tag{1.4.2—9}$$

Fusing frequent but noisy IMU signals with less frequent but more stable positional input allowed for high-precision state estimation. In an experimental study with 15 participants performing both printed and virtual circle-tracing tasks, Kalman filtering significantly reduced RMSE. In printed tracing, Group A's RMSE decreased from 1.13 mm to 0.90 mm ($p = 0.001$), and Group B's from 1.00 mm to 0.74 mm ($p < 0.001$), reflecting 20.69% and 28.36% average improvements, respectively. Similar accuracy gains were seen in dynamic virtual tracing tasks. These results confirm the Kalman filter's effectiveness for reducing jitter and improving manipulation fidelity, even in precision-critical environments [49].

1.4.2.3. Hybrid Filtering Methods

In VR systems that integrate external hardware components, a single filtering technique may not be sufficient to address multiple types of signal imperfections, such as high-frequency noise. Hybrid filtering strategies combine filters into a structured pipeline that enables short-term smoothing and longer-term correction. These architectures often incorporate simple filters, such as exponential moving averages, to suppress rapid fluctuations, alongside more adaptive or model-based filters for trajectory refinement. In motion output systems, such as compact motion platforms, hybrid filters may incorporate dead-zone logic and workspace-aware restoration to prevent saturation and maintain realism. The following implementations demonstrate how combined filtering stages have improved stability, continuity, and perceptual fidelity in systems that convert interaction or motion data into physical responses.

For example, hybrid washout filtering architecture was developed to address the motion fidelity limitations of compact 6-DoF motion platforms, which are often integrated with VR systems as external physical actuators. Traditional filters such as the Classical Linear Washout Filter (CLWF) are insufficient in constrained workspaces, where abrupt or saturated motions can result in unrealistic physical responses, actuator limit violations, or degraded perceptual realism. The architecture combines multiple filtering stages, including a baseline CLWF for signal decomposition, a Yawing Washout Filter (YWF) for managing rotational constraints, an Adjustable Scaling Filter (ASF) to adapt angular velocity to platform limits, and a Dead Zone Washout Filter (DZWF) that suppresses accelerations falling below vestibular detection thresholds. Additionally, an Adaptive Washout Filter (AWF) incorporates inverse kinematics and workspace-aware position restoration based on cubic polynomial interpolation, ensuring smooth re-centring of the platform without introducing perceptible motion artifacts ([Fig. 3], [Sec. III] in [50]. To model imperceptible repositioning, restoration acceleration is computed using the equation (see [Eq. 5], [50]). The restoration trajectory is generated using piecewise cubic polynomial functions, ensuring continuity of position and velocity during dead-zone-controlled returns ([Eq. 6–7], [Sec. III-D], [50]). Experimental validation was performed on an SP-120 motion simulator using automotive VR scenarios. Multiple filter configurations were tested, and system fidelity was assessed using a custom performance index (PI) that combines the RMS error of acceleration and angular velocity cues with W_a and W_ω reflecting the differential sensitivity of human perception to translational and rotational motion ([Eq. 11], [50]). The configuration using complete hybrid filtering with restoration scaling $(\lambda_a, \lambda_\omega) = (1,0)$ achieved the lowest PI values, demonstrating significant improvement in motion realism compared to the baseline CLWF. The system maintained smooth, continuous motion output, preventing saturation and abrupt transitions, thereby confirming the effectiveness of hybrid filtering in producing accurate, perceptually valid motion cues in virtual environments.

Another hybrid signal-processing strategy was implemented in a wearable open-haptic glove designed to enhance real-time finger tracking and tactile interaction in VR environments using input components. The system replaces traditional flex or inertial sensors with rotary position sensors installed at each finger's metacarpophalangeal (MCP) joint, combined with a lightweight Min-Max Scaling (MMS) filter to normalize raw angular data across users with varying hand sizes. Unlike Kalman or low-pass filters, which introduce computational overhead and response lag, the MMS filter performs linear rescaling within a calibrated range defined by the minimum and maximum angular positions:

$$Scaled\,Value = \frac{X_t\ - min}{\max - min}; \qquad 1.4.2—10$$

here X_t is the current sensor reading, min, and max are the calibration limits for each user. This approach ensures consistent signal output with minimal processing latency. The glove's microcontroller (Arduino Nano 33 BLE) achieved a per-finger signal delay of approximately 145 μs, with a total hand data latency of under 4 ms, enabling seamless integration with Unity-based VR applications. Experimental validation with 10 participants demonstrated high angular tracking accuracy, with mean absolute errors (MAEs) of 3.091° for flexion/extension and 2.068° for adduction/abduction, and superior noise suppression during both fast and slow finger movements compared to baseline setups using flex sensors with low-pass filters or IMUs with Kalman filtering. These results indicate that hybrid configurations incorporating hardware-specific design and tailored lightweight filtering can significantly improve signal stability and responsiveness, integrating physical input into VR systems [51].

1.4.3. Machine Learning and Neural Network Techniques for Data Prediction

Machine learning models have been widely adopted for motion prediction tasks in virtual reality due to their ability to capture temporal dependencies and sequential patterns in sensor data. These models often anticipate user movement for real-time adaptation, path planning, or locomotion control. One study that exemplifies this application is the short-term prediction of walking direction in immersive virtual environments using LSTM networks trained on orientation-augmented motion data [52]. A structured review of machine learning techniques developed for 3D motion synthesis and musculoskeletal dynamics estimation is provided in [53]. The paper examines various data-driven methods that estimate joint positions, torques, and muscle forces from visual or sensor-based motion inputs. The approaches are categorised by motion representation (e.g., skeletal joint coordinates, body mesh models, or SMPL parameters), model architecture (including convolutional networks, recurrent networks, graph neural networks, and Transformers), and training strategy (supervised learning and physics-informed optimisation). These methods estimate physical properties, such as joint torque and muscle activation, from data acquired via monocular video, inertial sensors, or motion capture systems. The techniques and their characteristics are summarized in [Table 1] in [53]. It is reported that machine learning models trained on labelled datasets, including CMU MoCap, AMASS, and biomechanics-focused data, can achieve accurate pose estimation and dynamics inference using limited input modalities. Physics-informed learning models, which incorporate biomechanical constraints or simulation-based supervision during training, reduce structural errors such as unrealistic joint rotations or inconsistent limb movements. Several reviewed architectures combine kinematic consistency constraints with standard loss functions, allowing the model to output motion that conforms to anatomical structure. Examples of such architectures are shown in [Fig. 3] in [53]. These characteristics make the techniques suitable for posture evaluation, providing physical feedback, or ergonomic analysis in VR applications that utilise reduced sensor setups. The limitations identified include high variability in model performance across different types of motion and subject-specific anatomy. Models often exhibit reduced accuracy when applied to actions not represented in the training data or when tested on subjects with varying body proportions. Physics-informed training processes typically require higher computational resources and longer convergence times. Evaluation protocols are inconsistent across publications, and benchmark discrepancies are evident in cross-dataset results, as illustrated in [Fig. 6] in [53]. As a result, comparison between systems is limited to specific experimental setups.

Mayor, Bremer, and Lappe examine the application of LSTM networks for predicting short-term walking direction in virtual reality [52]. The motivation for this work lies in anticipatory locomotion systems, such as redirected walking, which require advance information about user movement to initiate subtle environmental manipulations. These systems benefit from accurate predictions of the user's heading direction up to 2 seconds into the future, enabling them to maintain immersion and prevent collisions or boundary violations in constrained physical spaces. The authors design an LSTM-based sequence model trained on motion capture data from 44 participants walking through a virtual corridor. The input features tested include Cartesian position, velocity, and quaternion-based rotation, either individually or in combination. The experiments compare four input configurations across multiple prediction horizons ranging from 0.5 to 2.0 seconds. The best results are obtained with a combined input of position and quaternion orientation, achieving mean angular errors of 9.6° at 0.5 seconds, 18.7° at 1.0 seconds, and 33.3° at 2.0 seconds. In contrast, models using position-only input show higher error, reaching 42.3° at the two-second mark. The authors attribute this difference to the ability of quaternion features to encode head-body orientation more effectively, improving directional intent modelling. The results are consistent across participants and motion conditions, suggesting good generalization within the tested scenario. Although the model does not involve explicit filtering or multi-stage processing, it falls within the domain of neural predictive systems, enhancing real-time interaction and motion planning in immersive environments.

In a related line of research, a separate study investigates walking trajectory prediction using LSTM models trained on headset-derived motion data collected from 10 participants during natural locomotion [54]. The dataset consists of head position, velocity, acceleration, and rotation sampled at 100 Hz using HTC Vive hardware. Forecast intervals range from 100 milliseconds to 3 seconds. The model achieves a mean position error of approximately 4.5 cm at 100 milliseconds, 12.5 cm at 500 milliseconds, and over 22 cm at 1 second. The LSTM network consistently outperforms a constant-velocity baseline across all intervals. However, prediction accuracy deteriorates rapidly beyond 1.5 seconds, indicating a limit to the forecasting utility of head motion alone. The study also identifies directional error tendencies, with predictions being more accurate along the primary walking axis than for lateral deviations. These findings underscore the need to enhance features, such as incorporating gaze or full-body pose, to improve generalizability in free locomotion. The work reflects typical VR neural predictive modelling characteristics: short-term utility, sensitivity to movement context, and the trade-off between complexity and real-time applicability.

The third study investigates low-latency head pose prediction in AR and VR using neural networks trained to learn temporal coherence in user motion sequences [55]. The system's goal is to anticipate head movement up to 96 milliseconds in advance, to support latency-sensitive applications such as predictive rendering, visual stabilization, and interaction alignment in immersive environments. To simulate realistic input dynamics, the authors generate time-series data from synthetic datasets, including Microsoft AirSim and the Cambridge Landmarks dataset, which reflect common head motion trajectories in virtual navigation. Input sequences incorporate position, orientation, linear velocity, and acceleration to capture instantaneous state and underlying motion trends. The predictive task is formulated as a supervised sequence-to-sequence mapping problem, and several network configurations are explored. These include classical LSTMs, gated recurrent units (GRUs), bidirectional LSTMs, and a CNN-LSTM hybrid model that combines temporal convolutional layers with recurrent processing. Prediction performance is evaluated at multiple horizons, ranging from 12 to 96 milliseconds. The CNN-LSTM model achieves the highest accuracy, producing an average positional error of 4.4 mm and an angular error of 1.9° at the maximum prediction interval. These outcomes are presented in Table 4 and Figures 5–6 of [55], and indicate that the hybrid model consistently outperforms Kalman filtering and linear extrapolation baselines, particularly during non-linear or dynamic movement phases. The convolutional component of the model extracts localized temporal features, such as directional shifts or deceleration patterns, while the LSTM layers model sequence-wide dependencies and maintain temporal continuity. Importantly, the architecture is designed to operate independently of specific hardware characteristics, ensuring compatibility across various HMD platforms. The system is lightweight enough to support real-time inference, making it suitable for deployment within low latency rendering pipelines. The study demonstrates that combining spatial-temporal feature extraction with sequence modelling provides measurable benefits for motion forecasting and latency compensation in AR/VR systems.

A comparative investigation evaluates the influence of output representation on human motion forecasting performance using deep learning techniques [56]. The study addresses the challenge of accurately predicting physically plausible 3D joint trajectories over both short- and long-term horizons, an essential requirement for immersive virtual reality applications, including full-body avatar control, anticipatory interaction systems, and locomotion planning. Rather than proposing a new model architecture, the paper aims to isolate and assess the effect of prediction representation format on performance. Two classes of methods are contrasted: direct coordinate prediction, in which future joint positions are estimated directly in Cartesian space, and geometry-aware approaches, in which intermediate kinematic or parametric representations such as joint rotations, bone directions, or mesh-based encodings are predicted and later reconstructed into joint positions via forward kinematics. The experiments utilise the Human3.6M dataset, which includes annotated full-body motion capture data. The researchers implement multiple deep learning backbones, RNNs, GCNs, and Transformer-based encoders while keeping the training and evaluation protocols constant across all output representations. Prediction horizons range from 80 ms to 1000 ms to simulate both low-latency and extended forecasting scenarios. Evaluation metrics include mean per-joint position error (MPJPE) to measure accuracy, bone length deviation to assess structural plausibility, and acceleration error to reflect kinematic smoothness. This multilayered evaluation framework enables the authors to examine how various representations impact output quality and physical consistency. The results show that direct prediction methods perform better at short-term horizons (≤160 ms), making them more suitable for tasks such as predictive rendering or near-future interaction alignment in VR. These methods bypass kinematic decoding stages, offering simpler inference and avoiding error propagation from intermediate representations. Conversely, geometry-aware approaches provide better long-term prediction stability by preserving physical constraints, such as fixed bone lengths, which are often violated by direct methods over extended periods. However, geometry-aware models involve more complex data transformations and decoding, which increases their computational overhead. Figures 3 and 4 in [56] compare MPJPE and bone consistency across different horizons, demonstrating these trade-offs. Importantly, the study does not evaluate or claim real-time system deployment; all findings are limited to offline training and testing on the Human3.6M benchmark.

1.4.4. Conclusion of VR Input Data Prediction Methodologies

The evaluation of VR input data processing methods presented in this section highlights the challenges and solutions for peripheral or off-the-shelf input devices, grouped into three categories. Predictive approaches (interpolation and extrapolation methods) offer viable solutions, especially for real-time applications, to estimate data during signal dropouts or latency-related lags. The interesting finding of the analysis was the lack of literary use of simple mathematical models in VR adaptive solutions, despite numerous studies on filtering, hybrid, or predictive machine learning approaches. Relatively simple regression models, including linear and polynomial formulations, remain practical for low-complexity tasks but tend to degrade under non-linear or ballistic movement dynamics, a gap that hybrid kinematic-regression models address by incorporating higher-order motion derivatives and time-dependent coefficients.

Filtering-based correction techniques serve a fundamentally different purpose: rather than estimating future data, they stabilize existing input by removing noise, drift, or latency artifacts. The application of EMA methods to Kalman filtering demonstrates success in both input stabilization (e.g., IMU-based leg tracking) and output fidelity (e.g., motion platform synchronisation). In accordance, hybrid filtering strategies that combine model-driven and threshold-based correction layers prove critical in tightly constrained systems with real-time physical actuation requirements. However, the applicability of these filters must be carefully assessed in terms of signal direction, specifically whether physical-to-virtual or virtual-to-physical, as indiscriminate filtering may weaken intentional motion patterns, particularly in rhythmic equipment such as rowing machines, as demonstrated in this dissertation.

Furthermore, integrating machine learning techniques extends the capabilities of VR systems beyond static rule-based modelling, enabling context-aware, temporally sensitive posture recognition. Methods based on CNNs, LSTMs, and GRUs have demonstrated strong potential for recognizing sequential motion patterns and compensatory behaviours, especially under unconstrained, real-time movement conditions. RF models remain valuable for static or low-dimensional posture classification tasks where interpretability, generalization, and robustness are more important than sequence modelling. The reviewed literature indicates that each machine learning approach has a distinct operational strength, shaped by the underlying data structure (e.g., joint angles, pressure matrices, visual pose), dataset size, and the complexity of user movements.

Despite the diversity of methods, a common limitation across the state of the art is limited validation on peripheral input devices. Most models are developed using standard motion capture setups, pressure mats, or embedded wearables, with minimal attention paid to integrating domain-specific equipment, such as rowing machines, treadmills, or specialized tools. This dissertation fills this gap by offering a technically validated, experimentally grounded research result that adapts peripheral input devices, transforming physical interaction into virtual action with minimal latency and high time-to-VR reaction accuracy.

1.5. Posture Monitoring, Analysis, and Recognition Techniques in VR

Neural network techniques are used to predict data, enabling autonomous behaviours in vehicles, devices, and software or hardware components. In these cases, accuracy is a critical factor in determining the quality of the applied technique. For example, the study "Autopilot Design for Unmanned Surface Vehicle based on CNN and ACO" employed a predictive control method to navigate an autonomous vessel to its destination, accounting for sea conditions. The study utilised an NPMC controller based on a CNN and an ant colony optimiser (ACO) to predict disturbances and ensure a smoother journey [57].

In the healthcare field, physiotherapy exercises, including active range-of-motion (ROM) movements such as extension, elevation, flexion, rotation, muscle strength, and endurance training, are crucial for patients recovering from stroke (PSR) [58]. Physiotherapists employ various techniques to help individuals regain their daily mobility, including task-based training, muscle strengthening, and the use of assistive devices. However, guiding patients through physiotherapy exercises can be time-consuming, labour-intensive, and costly [59]. Numerous other studies have explored the effectiveness of computer-assisted treatments and VR in rehabilitating and improving upper limb motor skills, balance, gait, posture, and walking [60]. Additionally, research has examined the therapeutic benefits of telerehabilitation, which enables patients to receive therapy from therapists remotely via telecommunication technologies, often in the comfort of their own homes. This approach has been widely applied in motor and cognitive recovery [61]. At the same time, there has been a rise in studies examining the use of information technology to support patients in home-based rehabilitation. Home-based recovery [62], especially within the scope of telehealth [63], is increasingly employed to reduce healthcare costs. However, there is a risk of poorer clinical outcomes due to a lack of patient motivation and the challenge of maintaining close medical supervision. This risk primarily concerns patients who must rigorously follow physicians' rehabilitation plans. Developing methods to monitor functional recovery, such as after a stroke [63], is essential to ensure that patients complete the necessary exercises and remain motivated to do so. One recognized method for assessing the accuracy of rehabilitation exercises involves monitoring posture in real time throughout the range of motion. The quality of these exercises is typically evaluated by analysing thoracic orientation, hip and knee joint rotations, and leg length. One of the primary advantages of motion monitoring technology is its affordability and compact size, making it suitable for widespread use in rehabilitation clinics, gyms, and even home settings. Moreover, posture monitoring is equally essential for healthy individuals, especially those at risk due to poor work habits and ergonomics, such as office workers who sit for prolonged periods [64, 65]. A rehabilitation system that integrates a gamified virtual environment with upper-limb rehabilitation technology offers an interactive, engaging approach that boosts patient motivation, making rehabilitation more efficient and effective [66]. Various methods are employed to collect raw sensor data to monitor human activities. Wearable sensors are explored in studies such as [67, 68], which often combine them with clinical tests and outcome measurements. Similarly, Capecci et al. [69] presented a dataset of rehabilitation activities for low back pain (LBP) collected using an RGB-D sensor, including RGB and depth videos and skeletal joint locations. This data is used to calculate performance scores for rehabilitation exercises. Wang et al. [70] developed a virtual rehabilitation system using Kinect to analyse upper-limb exercises for stroke patients with hemiplegic dyskinesia, leveraging Kinect's bone-tracking capabilities. Arsfield et al. [71] conducted qualitative and quantitative analyses of the pose estimation algorithms for the Xbox One Kinect to evaluate their suitability for rehabilitation scenarios, particularly for upper-body stroke rehabilitation. Tracking shoulder, elbow, and wrist joints was challenging due to occluded depth data. The researchers suggested that leveraging temporal information and extrapolating from prior frames could enhance the accuracy of joint location predictions.

IMUs are favoured for their portability, affordability, and precision in modelling participant movements. Smartphones equipped with built-in IMUs and optical cameras are another widely used approach [72, 73], as they are integral to daily life, making them ideal tools for research on human activity. Optical cameras, frequently used for recognizing human motion, are also essential. Compared to standard optical cameras, depth cameras offer the added benefit of providing depth information. Microsoft Kinect (Microsoft et al., USA) is a popular, low-cost motion sensor that measures posture and balance during physical activities [74]. Several studies have identified the Kinect RGB-D sensor as an effective and practical tool for motion analysis in rehabilitation and industrial settings, particularly when professional-grade marker-based systems are unavailable. A multi-view approach, utilizing several Kinect cameras synchronised to a single computer, offers another solution for capturing a more complete human pose [75]. Human action recognition is typically addressed through traditional algorithms or deep learning-based methods. Deep learning-based recognition algorithms use neural networks to learn object features and produce recognition results directly. In contrast, conventional methods employ a two-step process, “feature extraction and expression + feature matching”, to identify human behaviour. These traditional algorithms focus on intrinsic properties of human actions, including motion features, temporal and spatial points of interest, and geometric details. They can also analyse human shape for biometric information, such as determining sex [76] or assessing health status. Many contemporary human position estimation techniques rely on depth sensors to gather skeletal or skeletal point data [77, 78]. However, identifying movements, including direction, angles, and joint centres, is essential for rehabilitation. Although several posture estimation methods and exergames have been developed, most systems primarily focus on extracting skeletal points from depth sensors and estimating human pose, rather than determining joint movement, direction, and angle [79]. Ayed et al. [80] introduced a method to assess the Functional Reach Test (FRT), a widely used clinical balance test for predicting falls, using the Kinect V2. Their findings supported the use of Kinect V2 to calculate conventional FRT measurements. Saini et al. [81] proposed a bidirectional long- and short-term memory neural network (BLSTM-NN) that utilises a Kinect sensor to monitor individual interactions, thereby aiding in action recovery assessments.

Several methodological approaches have been proposed in the literature to classify and recognize human posture in VR environments. These approaches can be broadly categorised into rule-based techniques, statistical methods, and machine learning-driven solutions, which are discussed in the following sections.

1.5.1. Statistical Approaches for Posture Categorisation

Statistical approaches to posture recognition rely on mathematical models that capture structural or temporal patterns in human movement using handcrafted features derived from sensor or video data. Unlike rule-based systems, which are built on predefined expert logic, or machine learning models that infer patterns from large datasets, statistical models offer a middle ground by applying established classifiers such as Support Vector Machines (SVM), k-nearest neighbours (KNN), or decision trees to features specifically engineered from motion capture or pose estimation data. These techniques are often favoured where interpretability is needed.

A practical example of this approach was demonstrated in a system designed for the automated evaluation of physical therapy exercises using monocular video input [82]. The study implemented a pipeline to process exercise recordings using OpenPose to extract 2D skeletal key points. These joint coordinates were then converted into spatial and temporal features, such as joint angles, displacement vectors, and signal smoothness the which served as input to an SVM classifier employing a radial basis function (RBF) kernel. The classification framework is depicted in [Fig. 3] in [82] illustrating the sequential data flow from video frame to pose estimation, feature construction, and final classification. The system was evaluated on a dataset comprising 48 video sequences collected from 12 participants performing five rehabilitation exercises. Each motion class was treated as a separate classification task. According to the experimental results summarized in [Table 1] of [82] showing the sequential data flow is from video frame to pose estimation, feature construction, and final classification. The system achieved an overall classification accuracy of 96.6%. Individual activity recognition rates ranged from 94.3% to 98.7%, with precision and recall values exceeding 0.93 across all categories. The authors highlight that the classifier performed well despite variability in participant execution styles and minor differences in camera viewpoint, indicating potential for real-world applicability in uncontrolled environments.

One of the primary strengths of this approach lies in its independence from depth data or specialized sensors; only standard RGB video and a lightweight pose estimation library are required, making the solution accessible for home-based rehabilitation. Nonetheless, the study also notes key limitations. The exclusive use of 2D pose data restricts the system's ability to model out-of-plane or occluded movements. The classifier’s generalizability was not tested beyond the five selected exercises, limiting the scope of conclusions about system robustness in broader rehabilitation contexts. Additionally, although OpenPose provides reliable keypoint estimation under many conditions, tracking inaccuracies caused by lighting variation or partial occlusions can degrade feature extraction and, by extension, classification accuracy. The authors acknowledge these potential shortcomings as areas for future refinement and validation. It also illustrates how statistical classification, combined with pose estimation and feature engineering, can deliver reliable performance in posture assessment applications without relying on large-scale training data or computationally intensive architectures. Such approaches bridge the gap between rule-based logic and modern deep learning models, offering practical benefits in resource-constrained or decentralized rehabilitation scenarios.

A study compares two distinct technologies: an IMU coupled with Arduino Uno and Kinect V2, using OpenCV, for estimating human torso posture. The primary objective is to assess which system provides greater accuracy and reliability for posture analysis, with implications for applications in health monitoring, sports, and ergonomics. Both approaches are implemented in controlled experiments in which human movement is tracked, and postural angles are computed using geometric methods. For the IMU-based system, joint motion is captured via accelerometers and gyroscopes, and real-time orientation data is visualized using Arduino's Serial Plotter. The Kinect-based system, on the other hand, utilises RGB, IR, and depth imaging to generate skeletal joint coordinates, which are then processed through MATLAB’s Image Acquisition and Deep Learning toolboxes. To quantify lateral (Y-axis) and anteroposterior (X-axis) trunk movements, the study employs trigonometric functions applied to 3D joint coordinates captured by both systems. The specific equations for calculating tilt angles and 3D distance measures (e.g., frontal flexion) are explicitly detailed [see Equations 1–3 and Figs. 4–5 in [83]. These equations model torso inclination using key anatomical reference points that the system detects the spine base and shoulder centre. The results (Figs. 6–10 in [83]) compare real-time angular estimates from both systems across multiple postural conditions: straight standing, left/right lateral flexion, forward protraction, and backward retraction. The IMU method generally shows broader angle estimations, for example, -30° vs. -25° during left flexion, while Kinect’s estimates remain narrower. IMU reported 50–55° for forward protraction while Kinect reported 60–65°, indicating systematic offsets between sensor modalities. These quantitative differences are attributed to the IMU's susceptibility to drift and environmental interference, while Kinect is affected by occlusion and illumination variability. The strengths of the IMU approach include its high temporal resolution and portability, which enable reliable measurements even in uncontrolled lab conditions. However, the authors note that its primary drawback is gyroscope drift over time, which degrades accuracy in long-duration measurements. Additionally, the IMU system requires calibration and lacks spatial joint mapping capabilities. Kinect V2, conversely, excels in full-body skeleton reconstruction, non-invasive tracking, and higher positional accuracy under static conditions, though it suffers in dynamic or obstructed settings.

The authors conclude that the IMU provides slightly better accuracy in joint angle estimation when integrated with sensor fusion algorithms, with an error range of 3° to 8° for both systems, values consistent with the related literature. They recommend integrating both modalities in the future to exploit their complementary strengths, particularly in clinical, rehabilitative, and interactive VR environments where hybrid precision is required.

1.5.2. Machine Learning-Based Posture Recognition

Machine learning (ML)-based posture recognition systems use annotated datasets and algorithms to classify and evaluate human postures from sensor or visual inputs. These systems differ from rule-based or statistical approaches by relying on supervised learning or deep learning architectures to extract and learn patterns from data, rather than predefined logic or parametric heuristics. Various studies included in this section demonstrate the application of ML algorithms for posture recognition, including CNNs, GRUs, and LSTM networks for temporal sequence modelling, and decision tree-based classifiers such as RFs for static posture classification using joint angles. While a wide range of ML techniques has been proposed in the literature, the present dissertation focuses on CNN- and RF-based architectures, as they align most closely with this research's methodological scope and experimental framework. Other approaches, including GRU-, SVM-, or ensemble-based models, remain relevant in broader applications but are beyond the scope of the current implementation and evaluation effort.

1.5.2.1. Random Forest

An example of using RF for a posture recognition framework was developed to monitor ergonomic behaviour in innovative classroom environments using angle-based biomechanical features and conventional machine learning classifiers [84]. The system aims to identify poor postural habits in students during extended sitting periods, a scenario that, although distinct from clinical rehabilitation, aligns closely with virtual environments where continuous seated interaction and posture maintenance are relevant. The system uses a multi-camera configuration with depth sensors to acquire real-time 3D skeletal joint coordinates. Instead of relying on raw position vectors or deep learning-based feature extraction, the approach calculates static joint angles (e.g., neck, trunk, upper limbs). It uses these as interpretable classification features [Fig. 3 in [84]]. The authors trained and tested multiple machine learning models — RF, KNN, SVM, and XGBoost — on a dataset containing nine posture categories collected from 20 participants. These postures included upright, left and right lean, slouching, and head tilt. Among the models evaluated, RF achieved the highest recognition accuracy of 96.72%, followed by XGBoost, as detailed in [Table 3 in [84]]. The evaluation employed macro-averaged metrics for accuracy, precision, recall, and F1-score. The system was validated in a real classroom scenario, confirming its feasibility for continuous ergonomic monitoring. A key strength of the method is its balance between computational efficiency and interpretability, making it suitable for real-time applications on resource-constrained systems. Unlike deep neural network-based systems, which often require significant training data and opaque architectures, manually engineered joint angle features allowed for reduced training complexity and more precise postural interpretation. However, the authors acknowledge that classification accuracy may decline in environments with frequent occlusions or abrupt motion, and they recommend integrating temporal smoothing or sequence-aware models in the future to improve robustness under dynamic conditions. Overall, the method provides a viable real-time solution for posture assessment using biomechanical indicators and interpretable machine learning classifiers.

In one research project, an RF algorithm was used to analyse multimodal sensor data collected from a posture monitoring framework that integrates IMUs, Kinect skeletal tracking, and MediaPipe keypoint extraction [85]. The system processes data from five static posture classes (e.g., upright sitting, forward leaning) performed by 10 participants, comprising over 1,000 annotated samples. Handcrafted features were extracted from raw IMU time-series signals and skeletal joint data for the RF model, including statistical measures such as mean, standard deviation, range, signal magnitude area (SMA), and temporal moments. These features were selected to capture posture-relevant dynamics, including joint stability, angular deviations, and motion smoothness. The RF classifier was trained using early- and mid-fusion strategies, where early fusion involved concatenating raw sensor channels. In contrast, mid-fusion used separate sub-models for each sensor modality, whose outputs were merged. Experimental results demonstrated that RF achieved an accuracy of 93.1% under mid-fusion conditions, providing interpretable decision trees suitable for deployment in low-latency or embedded applications [Table 2, Fig. 7 in [85]]. While not as performant as CNN-based methods, RF offered computational efficiency and robustness to sensor dropout. However, the authors noted that synchronisation issues between asynchronous sensor streams (e.g., IMU and camera) could influence classification quality and suggested incorporating adaptive temporal alignment in future designs.

In another study, the authors analysed a systematic literature review of 16 posture recognition studies published between 2010 and 2020, focusing on the use of machine learning algorithms in medical, wellness, and virtual environments [86]. Among the algorithms discussed, RF was repeatedly identified as a high-performing method in constrained posture classification settings, particularly in scenarios involving limited computational resources or static postures. One reviewed study involved a smart chair equipped with 8 capacitive pressure sensors, designed to record the 7 seated postures of 41 participants across multiple sessions. These postures included slouching, forward lean, upright sitting, and side bends. The RF classifier trained on this sensor data achieved 90.9% recognition accuracy, significantly outperforming decision trees and Naïve Bayes classifiers. In another study reviewed, a pressure-sensitive cushion installed on a wheelchair collected distribution data from users in four seated positions, which were then classified using an RF model with 99.03% accuracy, demonstrating its suitability for clinical monitoring of postural stability and user comfort. A third example utilised RF for human posture detection in a smart city context, employing a LoRa-based distributed sensing framework. Here, multiple wearable and stationary sensors collected biomechanical and environmental data, and the RF model achieved 95.06% accuracy, outperforming five other machine learning methods evaluated under cross-validation. The review notes that RF's robustness stems from its ensemble-based learning design, which effectively handles noise and non-linearity in sensor-derived posture data. These systems often rely on structured features, such as joint angle calculations, spatial pressure maps, or time-averaged load vectors, that decision trees can cleanly separate. This makes RF particularly suitable for posture recognition tasks focusing on discrete classifications within relatively stable contexts (e.g., seated office environments, wheelchair monitoring, or VR calibration sessions). However, the authors also emphasize that RF's performance declines in temporally complex or highly dynamic scenarios, such as continuous full-body motion in virtual reality environments. The absence of memory components in tree-based structures limits RF's ability to learn motion dependencies across frames. As a result, while RF is well-suited for near-real-time classification of static or semi-static postures, it may be inadequate for motion prediction or real-time tracking of posture transitions, unless integrated into a hybrid architecture alongside temporal models, such as LSTM or GRU.

The research paper presents a mobile-compatible framework for evaluating posture correctness during physical exercises, such as arm raises, squats, and lunges [87]. The system uses a standard RGB camera with MediaPipe's BlazePose model to estimate 33 skeletal keypoints in real time, from which 12 biomechanically significant joint angles are extracted. These angles include metrics for shoulders, elbows, hips, and knees, which define correct execution thresholds for various exercise types (see [Table 1, p. 102] in [87]). The system applies rule-based comparisons, checking whether the current joint angles fall within predefined thresholds — for example, an elbow angle between 160° and 180° for a correct arm raise — to determine posture validity. A flow diagram of the implementation process, including angle computation and validation logic (see [Fig. 3, p. 104] in [87]). Furthermore, the system includes a repetition-counting mechanism that tracks the number of correctly executed movements by analysing transitions between validated postures

Although the current implementation relies entirely on geometric rules, the authors propose incorporating a Random Forest classifier in future versions to enhance recognition robustness and reduce reliance on fixed thresholding. Their justification stems from Random Forest's efficiency in processing structured numerical inputs, such as joint angles and inter-joint distances, especially in scenarios where computational resources are limited, as in mobile or wearable devices [Section 5, p. 107 in [87]]. The authors also highlight the algorithm's ease of interpretation and resistance to overfitting when trained on limited data. However, the system is not tested in immersive or VR environments, nor does it incorporate depth sensing or dynamic posture transitions. As such, its functionality is confined to static, camera-facing activities in well-lit, uncluttered spaces. Nonetheless, its architecture, based on joint-angle analysis and its future direction toward Random Forest integration, positions the work within the broader context of lightweight, real-time posture assessment systems for fitness and ergonomic monitoring.

1.5.2.2. Convolutional Neural Network

A CNN is a multilayered neural network that uses at least one convolutional layer to process input data. A convolutional layer (conv) is a type of artificial neural network layer that performs cross-correlation by combining two data samples. This operation transforms the function describing the input data by reducing its dimensionality while retaining critical spatial or temporal features. Using convolutional layers is essential in any CNN model, as it reduces the number of parameters and shortens the learning time. Pooling layers operate similarly to convolutional layers by reducing the dimensionality of data. They retain the essential values of the data segment, such as the maximum (max pooling) or average (average pooling). This reduction helps focus the learning process on critical features. A dropout layer is introduced into the CNN to prevent overfitting and improve generalization. As described by [88], dropout randomly deactivates a fraction of neurons during training. This makes the model less dependent on specific features in the input data, thereby improving its robustness. The flattening operation converts multidimensional feature maps into one-dimensional vectors, making them suitable for dense layers performing the final classification. Finally, dense (fully connected) layers process the extracted features. Each neuron in the thick layer receives input from all neurons in the previous layer, and these operations are crucial for combining features to achieve accurate predictions.

We proposed a deep learning-based posture recognition model utilizing CNNs to extract spatial patterns from fused sensor inputs [85]. Unlike the RF model, which relied on feature engineering, the CNN was designed to process structured inputs derived from sensor matrices reshaped into image-like grids. These matrices encoded either joint coordinates or changes in joint angles over time, allowing the network to learn posture-specific spatial configurations directly via convolutional layers. The authors evaluated various fusion techniques, including early fusion by stacking joint-angle maps and late fusion by aggregating independent CNN outputs per modality. The CNN trained using late fusion achieved the highest classification accuracy of 96.5%, outperforming both RF and other CNN fusion variants [Fig. 10, Table 2 in [85]. Its performance was robust in distinguishing subtle postural deviations, thanks to the model's capacity to generalize learned spatial dependencies across joints. However, the study acknowledges that all evaluations were conducted offline and that no streaming latency tests were reported. Additionally, the system’s reliance on CNNs resulted in higher computational cost and model complexity than RF, making deployment in real-time or constrained environments more challenging. The authors proposed future improvements through sensor-specific calibration, transfer learning, and adaptation for continuous posture transitions in immersive VR systems.

Similarly, recent studies have applied rehabilitation systems to address the problem of compensatory upper limb movements in stroke patients undergoing motor recovery exercises, particularly those at or beyond Brunnstrom Stage III [89]. The study identified that such patients often exhibit maladaptive behaviours, such as trunk tilt, shoulder elevation, or irregular joint coupling, during repetitive training, which compromises the quality of motor relearning. To mitigate this issue, the system combines a Kinect-based skeletal tracking interface within a Unity 3D environment with a dual-module architecture comprising a patient-facing training component and a clinician-facing management platform. Real-time compensation detection uses angle-based thresholding and trajectory comparison via the Fréchet distance. Upon detecting compensation behaviours, the system automatically adapts task difficulty by adjusting the spatial placement of virtual targets and providing visual corrective feedback. The rehabilitation protocol includes structured tasks such as elbow and shoulder joint rotations and object manipulation, designed according to clinical occupational therapy practices. Performance is evaluated through two core metrics: one measures spatial trajectory similarity using Fréchet distance across three spatial axes [Fig. 4 in [89]], and the other evaluates task completion accuracy based on angular achievement and collision detection [Fig. 3 in [89]]. Quantitative evaluation revealed angular error margins consistently below 2° [Table 2 in [89]] and low Fréchet distances, indicating high conformity between the target and actual movement paths [Table 3 in [89]]. The system was validated using Perception Neuron data and tested on six participants, who provided qualitative feedback on usability, clarity, and system responsiveness. Reported limitations include the use of simplified capsule colliders for object interactions, which may reduce the precision of collision-based evaluations, and reduced detection reliability during high-load activities that alter natural motion patterns. The authors concluded that while the system is currently stable for rehabilitation use, further refinement is required to improve compensation modelling and to adapt training intensity based on patient-specific recovery dynamics.

In innovative manufacturing environments, a deep learning-based ergonomic evaluation model was proposed for posture classification using immersive VR and full-body skeletal tracking [90]. The system architecture utilises the HTC Vive Pro HMD and five Vive trackers to capture the joint trajectories of participants performing simulated manufacturing tasks within a virtual factory cell. Using inverse kinematics calibration and SteamVR's built-in skeletal tracking system, the setup captures 25 body joints, including head, neck, spine, shoulders, elbows, wrists, hips, knees, and ankles, as shown in [Fig. 4] in [90]. Participants were asked to perform object manipulation tasks (e.g., picking up bricks, rotating valves) under various ergonomic conditions (e.g., different object heights and weights). These movements were recorded and then segmented into discrete time-windowed samples. These served as input to a deep convolutional neural network (CNN) model for classifying posture risk across seven RULA-based ergonomic categories. The CNN was trained using over 3,000 manually labelled data segments and evaluated using 5-fold cross-validation. The CNN achieved a macro average accuracy of 89.49%, outperforming other tested models, including SVM and LSTM-based classifiers (see Table 5 in [90]). The authors emphasized that the CNN architecture effectively identified postural differences associated with arm elevation, trunk bending, and asymmetric loading —

key risk indicators in ergonomic assessments. Unlike traditional vision-based classifiers, the skeletal model leveraged spatial joint relationships and temporal motion features, making it more resilient to occlusion and lighting variability. However, the model was trained and evaluated offline, meaning all data was collected and labelled before classification occurred. Thus, it does not provide real-time feedback during task execution. Additionally, although the HTC Vive tracking system enables accurate full-body capture in controlled environments, its applicability may be limited in cluttered or constrained industrial settings. The study does not report usability validation or user feedback on in-field deployment, and there is no discussion of adaptability to rehabilitation or health-care scenarios. Nonetheless, the authors propose the model as a foundational system for automated posture evaluation in immersive industrial training contexts, with plans to explore real-time extensions and transfer learning across users.

Another study proposes a human posture classification method designed for performing arts education environments to enhance training feedback and motion correction for students [91]. The system combines OpenPose for keypoint detection and a BP (Backpropagation) neural network for posture classification. Although the context is not explicitly VR, the structure and techniques applied (camera-based motion capture, neural classification, and feedback) are conceptually transferable to immersive environments such as VR-based education or rehabilitation scenarios. The authors first acquire human skeletal joint data using OpenPose, which extracts 2D coordinates of body keypoints (25 joint positions) from RGB video frames. This data is then standardized and normalized to remove variations in input image size or position. After normalization, the data is used to train a BP neural network to classify posture into one of four categories: “good posture,” “head down,” “leaning forward,” and “slouching.” The neural model consists of an input layer with the same number of normalized keypoint features, a hidden layer (with an unspecified structure), and an output classification layer. The training set includes 3,000 annotated posture samples, and the system achieves over 90% classification accuracy during evaluation. The authors note that OpenPose’s robustness to clothing and lighting variation helps generalize recognition performance.

Despite the successful classification results, several limitations are acknowledged. The model processes offline data only and is not optimised for a real-time application. No performance metrics for latency or system responsiveness are provided. Additionally, the dataset is constrained to a controlled environment (performing arts classrooms), and postural classes are limited to four discrete categories. The system lacks implementation of dynamic posture tracking or continuous motion analysis, which limits its use in VR or rehabilitation settings where postural transitions and micro-adjustments are crucial. Furthermore, while OpenPose provides robust skeletal extraction, its lack of depth data introduces ambiguity in evaluating forward and backward lean and joint angles.

The authors report an overall classification accuracy exceeding 90% and provide a schematic diagram of the system pipeline [Fig. 1 in [91]], showing the integration of OpenPose, normalization, and neural network classification.

Another posture classification model based on a CNN was aimed at automating the analysis of body pose correctness and technique quality, which traditionally rely on expert visual observation and subjective evaluation. The model utilises RGB video frames captured from pre-recorded dance sequences. OpenPose extracts skeletal keypoints of 18 major joints, including the head, neck, shoulders, elbows, wrists, hips, knees, and ankles. These skeletal features are encoded as spatial vectors and fed into a CNN that learns posture representations and classifies them into specific movement categories relevant to standardized dance routines. The CNN was trained and evaluated on a dataset labelled by domain experts, achieving a classification accuracy of 96.88%, which was notably higher than alternative classifiers such as SVM (Support Vector Machines), which scored 93.13%, and k-nearest neighbours (KNN) at 87.5% [Table 2 in [92]]. While the model provides high posture recognition accuracy, it operates exclusively offline, as all skeletal data was collected and labelled before training and inference. There is no integration with real-time VR systems, and the model has not been validated in immersive environments or in feedback-oriented rehabilitation contexts. The authors emphasize that their objective was to build a tool to assist instructors and students in quantitatively evaluating posture form, using a neural network that generalizes well across performers of different skill levels.

A supervised CNN-based posture recognition system was developed for seated posture classification using data captured via a 3D RGB-D camera in an office-like environment. The research addresses the need for non-intrusive, accurate, and real-time monitoring of upper-body posture among sedentary workers. This context significantly overlaps with VR-based seated interactions, where prolonged poor posture may degrade health and performance. The proposed system uses a compact, multi-branch CNN to balance computational cost and classification performance. Each branch processes different normalized body segments (e.g., neck, shoulders, torso, and arms) derived from 3D joint coordinates, and the final decision is made by concatenating the intermediate features. The input data consists of 18 skeletal joint positions, normalized in both spatial scale and position to remove inter-subject variance [Fig. 2 and Fig. 4 in [93]]. The classification task targets eight distinct postures, upright, slouching, torso leaning, arm crossing, and side bends, labelled based on ergonomic relevance. The model was trained on a dataset comprising over 13,000 labelled frames collected from 40 participants and evaluated using a 10-fold cross-validation approach. The reported results showed an overall accuracy of 94.91%, with the CNN outperforming traditional machine learning baselines, such as Random Forest and k-Nearest Neighbour (see Table 2 in [93]). The model's compact architecture (three branches, each with two convolutional layers) allows real-time inference at 26 FPS on a CPU, with the authors explicitly stating a focus on real-time deployability [Section 4.4 and Table 1 in [93]]. Misclassification rates were higher for adjacent postures with similar joint distributions, particularly for slouching versus leaning forward, which was attributed to the inherent ambiguity in these cases. Although the system was not implemented in a VR environment, the authors suggest that the method's generalizability to immersive systems is feasible because it relies on body-joint input rather than RGB data, enabling potential integration with VR skeletal-tracking middleware. Limitations include sensitivity to camera calibration and difficulty generalizing across extreme body types, which the authors propose to address through transfer learning and user-specific normalization in future work. The authors also note that their approach currently does not handle temporal smoothing or dynamic transitions, which are essential in real-world VR interactions, and that these remain future directions.

A machine learning-based system for posture recognition and similarity evaluation was developed to address the need for automated ergonomic assessment in remote learning, dance, or yoga instruction scenarios, where posture correctness must be gauged without direct supervision [94]. The study introduces a *Posture Similarity Index (PSI)* framework that quantifies posture accuracy by comparing the learner's pose against a reference posture, leveraging joint-angle metrics and neural network processing. The architecture employs a deep CNN for initial object detection, using the COCO dataset as the foundational training set ([Table I, p. 3 in [94]]). With pretrained CNN weights, person detection is achieved through bounding box localization ([Fig. 2, p. 3] in [94]), after which background noise is masked, and MediaPipe is used to extract 32 joint landmarks and skeletal edges ([Fig. 4, p. 4]; [Fig. 5, p. 4] in [94]). This skeletal representation forms the input for further normalization and similarity evaluation. Two main similarity measures are implemented: cosine similarity, computed as the normalized inner product of joint vectors, and joint angle similarity, based on Pearson correlation of corresponding joint angles ([Section F, p. 4]). The architecture can process live-streaming data at 45 frames per second, demonstrating potential for real-time operation, although most evaluations in the paper are based on static posture image pairs. Evaluation results demonstrate cosine similarity of 0.96 and joint-angle similarity of 0.88 for selected posture pairs ([Table II, p. 5]; [Fig. 8, p. 5] in [94]). The system achieved a 0.77 F1 score in posture classification and an Intersection over Union (IoU) exceeding 0.93 during object detection on the validation dataset ([Fig. 6 and Fig. 7, p. 5] in [94]). The authors highlight that the PSI-based model can be tuned for various threshold levels, enabling automatic feedback when posture deviations fall below acceptable ergonomic standards. While the system demonstrates strong offline detection accuracy, its performance in dynamic or highly variable environments (e.g., with occlusions or varied camera angles) has not been assessed. The authors propose future extensions involving 3D coordinate acquisition, multi-angle streaming, and the detection of internal body rotations to improve spatial posture alignment ([Section V, p. 5]).

The authors report that the system operates at 45 frames per second using MediaPipe for joint extraction. The paper does not include experiments conducted under real-time streaming or live feedback conditions.

1.5.3. Hybrid Posture Classification Techniques

Recent studies have demonstrated an increasing trend toward hybrid posture recognition systems that integrate multiple computational paradigms, rule-based logic, statistical models, and machine learning algorithms within a unified framework. These systems are designed to mitigate the limitations of individual approaches, such as the rigidity of rule-based classifiers, the oversimplification of statistical heuristics, and the high data-dependency of deep learning models. Combining complementary techniques, hybrid systems aim to improve posture recognition accuracy and contextual awareness, though their impact on real-time responsiveness depends on the system's computational design and hardware capabilities.

An automated Kinect-based posture evaluation method was developed to assess work-related musculoskeletal disorder (WMSD) risks, addressing the limitations of manual ergonomic supervision in dynamic environments, such as construction sites [92]. The system builds on the Rapid Upper Limb Assessment (RULA) framework and integrates a hybrid computational approach combining fuzzy-logic inference with a Faster R-CNN deep learning model. The primary function of the fuzzy logic module is to classify posture risk in real time using input parameters such as skeletal joint angles, ambient temperature, and estimated object weight. These parameters are captured by an RGB-D Microsoft Kinect sensor, which provides continuous tracking of workers' postural data. To obtain object weight estimates without manual input, the system uses a Faster R-CNN model to identify objects carried by workers, such as bricks. It infers their weight based on visual characteristics. This estimated weight is passed to the fuzzy inference system, which determines a WMSD risk score using predefined membership functions and inference rules. The output is categorised into five discrete risk levels ranging from "extremely low" to "extremely high." Validation tests were performed by comparing the system's automatically generated RULA scores with expert evaluations, showing close alignment. In one experimental condition involving a high ambient temperature of 31°C and a manually assigned RULA score of 7, the system generated a WMSD risk score of 5.52 (see [92] Table 7). The object detection module achieved a 95% confidence rate in identifying and estimating carried loads. According to the authors, the system supports continuous, autonomous ergonomic assessment and provides real-time feedback to users. However, limitations were also reported, including potential inaccuracies in Kinect-based joint tracking caused by occlusions or suboptimal lighting, as well as the complexity of designing practical fuzzy membership functions. The method was explicitly validated within a construction context, and while the computational design allows for expansion, further adaptation would be required for applications in other fields.

1.6. Summary of VR Input Data Processing Techniques

Chapter II provided a comprehensive review of the current landscape of data processing methods for input devices used in VR systems, focusing on their implications for immersive experience, motion tracking, and human posture analysis.

The analysis highlighted the strengths and limitations of off-the-shelf VR and peripheral input devices, underscoring the trade-offs between usability, accuracy, and integration complexity. Off-the-shelf VR input devices, such as the HTC Vive tracker and Oculus Touch controllers, offer high levels of precision and ease of use in standard VR environments. However, their adaptability to domain-specific tasks, especially those requiring real-world tool integration or fine motor control, remains limited due to challenges such as tracking jitter, occlusion, and insufficient haptic feedback. Conversely, peripheral input devices offer opportunities for more natural, context-specific interaction in VR. However, their integration is frequently hindered by latency, mismatches in data transmission frequency, and the need for sophisticated data processing algorithms.

The review of motion tracking technologies further illustrated the trade-offs between marker-based and markerless systems, noting that while high-end systems, such as Vicon, deliver gold-standard accuracy, more accessible consumer-grade solutions, like the HTC Vive, offer practical alternatives with acceptable error margins for many applications. Nevertheless, maintaining low latency and high fidelity in posture monitoring is critical, particularly for rehabilitation and biomechanical assessment scenarios. The chapter also explored predictive data processing methods, including interpolation, extrapolation, and filter-based techniques, as well as the emerging role of machine learning and neural networks in motion prediction and anomaly detection. These methodologies are crucial for addressing data discontinuities, minimizing latency, and ensuring seamless user interaction in VR environments, particularly when integrating non-native devices.

In summary, the literature has made significant progress in developing VR input and data processing solutions. However, current technologies still face notable limitations in terms of integration flexibility, latency management, and application-specific adaptability. These challenges establish a clear motivation for developing predictive and machine learning-based data processing methods, which are explored in subsequent chapters.

2. EVALUATION OF HTC VIVE TRACKER SENSOR ACCURACY

To evaluate the spatial tracking accuracy of the HTC Vive Tracker (V2.0), an additional experimental validation was conducted by comparing its positional output with that of a Qualisys motion capture system, which served as the reference standard. This validation ensured the reliability of the HTC Vive Tracker's measurements and established confidence in its use for the subsequent experimental procedures presented in this dissertation.

The experimental setup consisted of a treadmill operating at a constant speed of 3 km/h, with the Vive Tracker placed on the moving belt together with a Qualisys infrared marker affixed to the centre of the tracker on top to ensure both systems recorded the same physical point in space. The tracker was carried forward by the treadmill and directed toward a fixed hit platform (see Fig. 7). After each run, the tracker was manually repositioned at the start of the treadmill for subsequent repetitions.

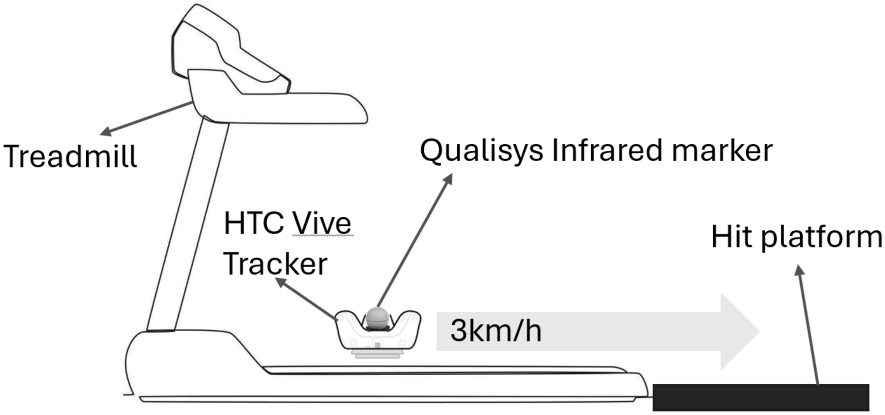


Fig. 7. Illustration of the Experimental Setup

The results shown were analysed across three experimental conditions to ensure robust evaluation of positional accuracy: (1) full signal duration, incorporating all phases including motion and impact; (2) isolated clean motion-only segments, capturing continuous movement without external disturbances; and (3) impact-specific intervals. For each of these conditions, the Mean Squared Error (MSE), RMSE, Mean Percentage Error (MPE), and Mean Absolute Percentage Error (MAPE) were calculated across all three spatial axes (X, Y, Z) to quantify positional tracking deviations between the HTC Vive Tracker and the Qualisys motion capture system.

Table 1. Positional Tracking Errors during Full Motion Signal Between HTC Vive Tracker and Qualisys System Across X, Y, Z Axes

Error	**x-axis**	**y-axis**	**z-axis**
MSE [cm²]	0,47	3,43	0,52
RMSE [cm]	0,69	1,85	0.72

MPE [%]	215,06	190,79	1128,5
MAPE [%]	2,15	1,92	30,35

Table 2. Positional Tracking Errors during Clean Motion Signal Between HTC Vive Tracker and Qualisys System Across X, Y, Z Axes

Error	**x-axis**	**y-axis**	**z-axis**
MSE [cm²]	0,67	0,01	0,11
RMSE [cm]	0,82	0,09	0,34
MPE [%]	2,71	2,06	19,7
MAPE [%]	0,03	0,03	0,24

Table 3. Positional Tracking Errors for Signals taken during Impacts Between HTC Vive Tracker and Qualisys System Across X, Y, Z Axes

Error	**x-axis**	**y-axis**	**z-axis**
MSE [cm²]	0,83	8,87	1,04
RMSE [cm]	0,91	2,98	1,02
MPE [%]	492,83	260,02	2355,1
MAPE [%]	4,93	5,65	72,07

The results in Table 1 and Fig. 8 highlight the performance differences between the HTC Vive Tracker and the Qualisys motion capture system during full signal tracking across the X, Y, and Z axes (X – lateral, Y – vertical, Z – depth). The error metrics suggest that positional discrepancies are most pronounced along the Z-axis, where the HTC Vive Tracker exhibited an exceptionally high MPE of 1128.5%. This is primarily attributed to large deviations during impact moments (see Fig. 10). In contrast, MAPE values for the X and Y axes remained below 3%, indicating relatively stable alignment during most movement. The box plots confirm this discrepancy, particularly in the Z-axis, where the HTC Vive Tracker exhibits numerous outliers and a wider range of values. These results suggest that although the tracker performs reasonably well in lateral and vertical motion, depth (Z-axis) tracking suffers significantly due to impact-induced noise.

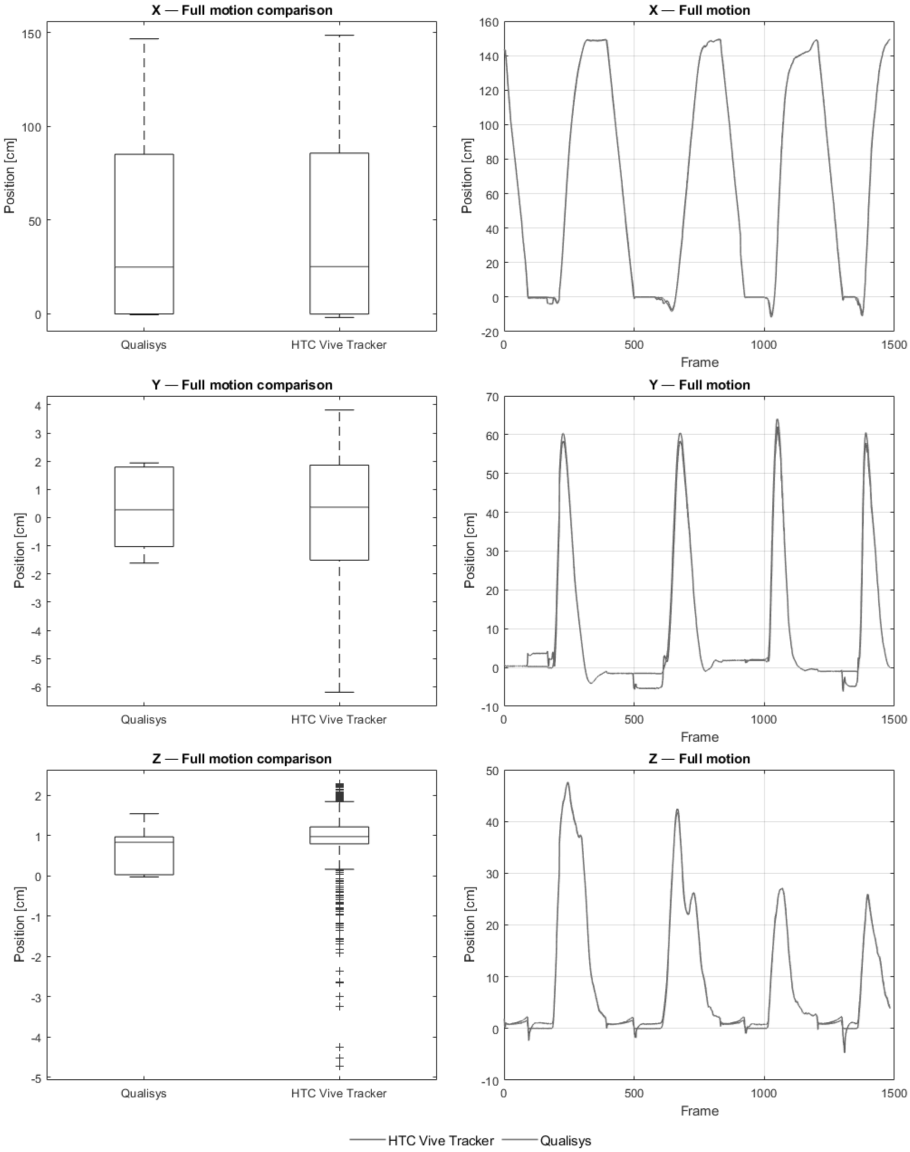


Fig. 8. Comparative Analysis of Positional Data Between Qualisys and HTC Vive Tracker Across Full Signal

To statistically validate these observations, a Kruskal-Wallis test was conducted to assess whether the median differences between the two systems were significant (see Table 4). The results show that the null hypothesis of equal medians was rejected for the Z-axis ($p < 0.05$), confirming a statistically significant difference in positional data on that axis. Conversely, the X and Y axes showed no significant difference ($p > 0.05$), reinforcing the observation that performance variation is axis-dependent.

Table 4. Kruskal-Wallis Test Results for Full Motion Signal

Axis	p-value	Reject HO
X-axis	0,22477	false
Y-axis	0,20515	false
Z-axis	2,434e-25	true

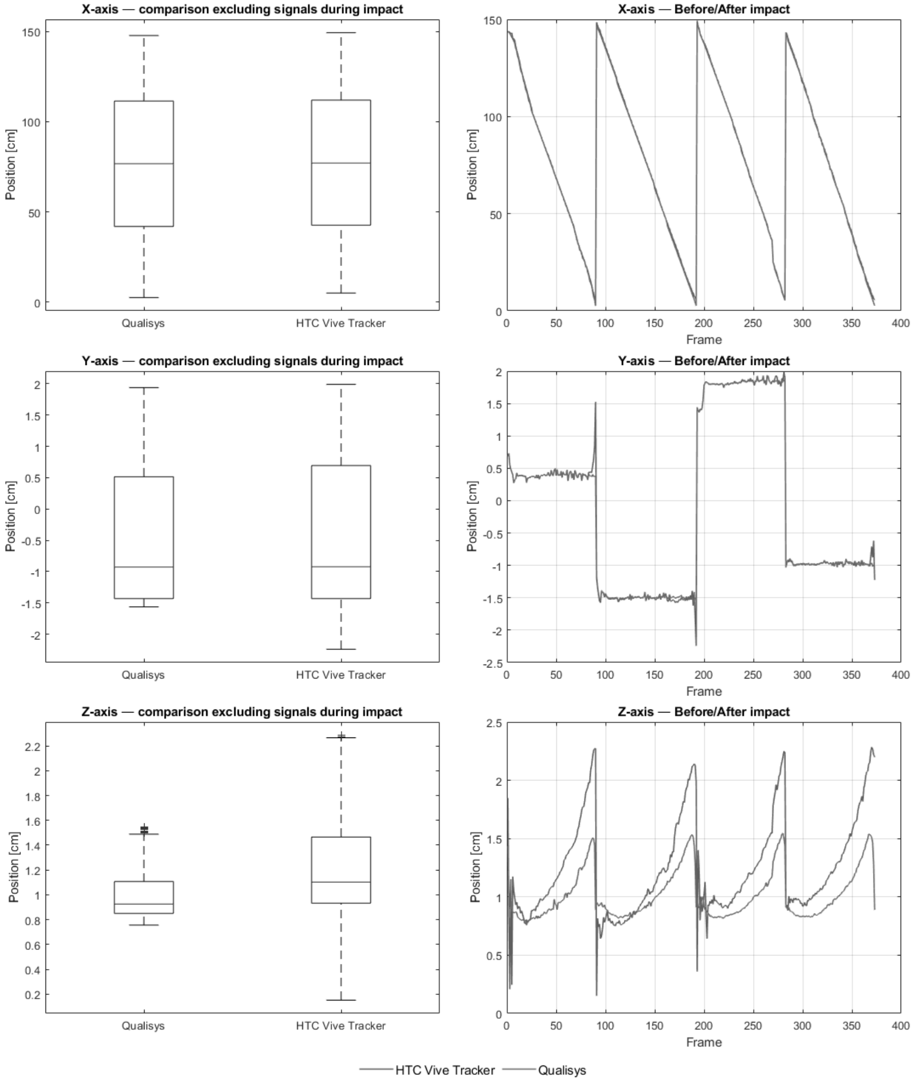


Fig. 9. Comparative Analysis of Positional Data Between Qualisys and HTC Vive Tracker, excluding the Signals during Impact

Building upon the earlier findings, a separate analysis was performed on a filtered signal segment that excludes frames with heavy impact to assess system behaviour during uninterrupted motion (see Fig. 9). As shown in Table 2, the overall error values across all axes dropped substantially, especially along the Z-axis, where the MAPE fell from over 30% to just 0.24%. This improvement confirms that transient impact artifacts largely drove the extreme deviations observed in the full signal. This further reflects the refinement, with tighter distributions and fewer outliers, particularly on the Z axis. Additionally, the cropped motion signals exhibit strong temporal alignment between the two systems, indicating that the HTC Vive Tracker reliably captures gradual and consistent movement patterns.

Table 5. Kruskal-Wallis Analysis of Motion Segments Excluding Impacts

Axis	p-value	Reject HO
X-axis	1,5715e-123	true
Y-axis	0,59499	false
Z-axis	4,807e-15	true

The Kruskal-Wallis test results reinforce these observations (Table 5) accompanying Fig. 9, which shows no statistically significant difference in the Y-axis medians ($p = 0.595$). At the same time, the X and Z axes still demonstrate considerable divergence. This remaining discrepancy may stem from minor systematic drifts or baseline calibration mismatches but is far less impactful under smooth motion conditions. These results emphasize that the HTC Vive Tracker provides acceptable tracking fidelity in calm, continuous motion scenarios. Its limitations are primarily exposed during dynamic or high-impact events.

As shown in Table 3 error rates surged dramatically compared to both full-signal and motion-only segments, particularly along the Z-axis, where MAPE reached 72.07% and MPE exceeded 2355%. These extreme values reflect the HTC Vive Tracker's difficulty in accurately registering rapid depth changes during impact. Similarly, elevated errors were observed in the X-axis, while the Y-axis remained comparatively more stable. Visualisations shown in Fig. 10 reinforce these findings, where motion curves during impacts show erratic fluctuations, latency lags, and substantial deviation from the Qualisys baseline.

Table 6. Kruskal-Wallis Analysis of Motion Segments During Impacts

Axis	p-value	Reject HO
X-axis	0,00084505	True
Y-axis	0,31282	False
Z-axis	2,1676e-53	True

Kruskal-Wallis tests were again performed and summarized in Table 6 to validate these observations. The results confirm statistically significant differences in median positions for the X-axis ($p = 0.0008$) and Z-axis ($p < 0.001$), indicating that the two systems diverge substantially during impacts. Interestingly, the Y-axis did not show a significant difference ($p = 0.3128$), suggesting vertical motion remains relatively reliable even under forceful events.

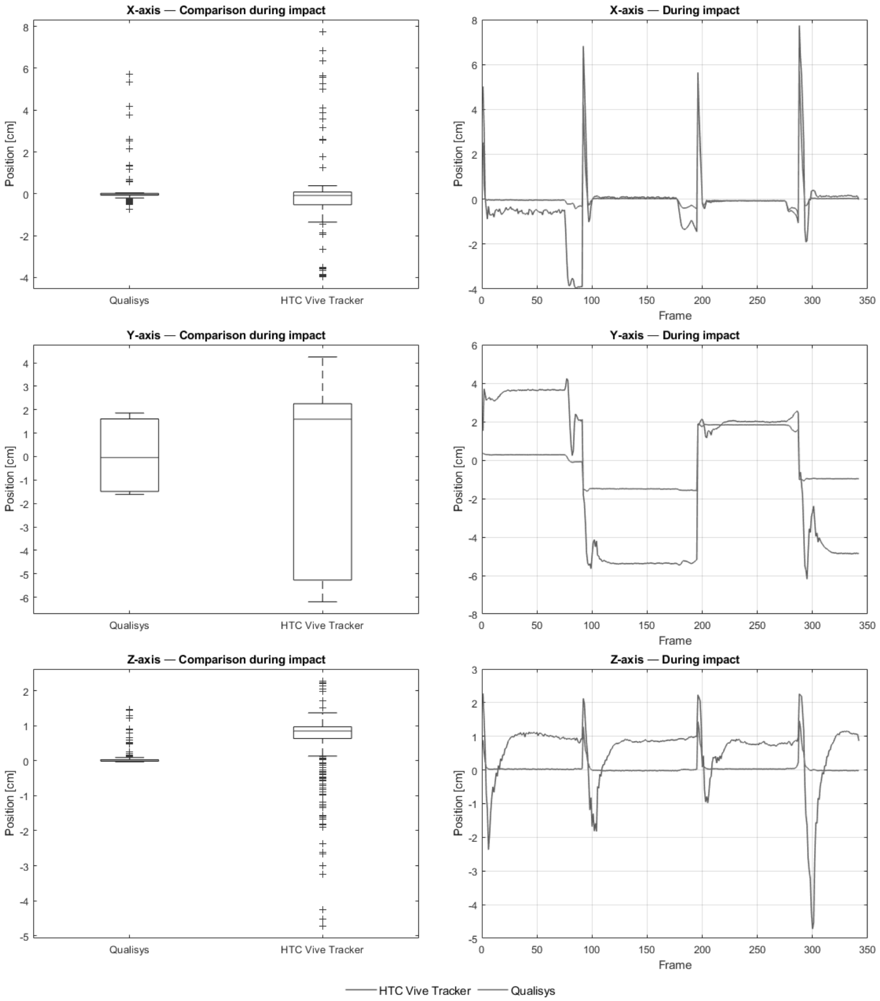


Fig. 10. Comparative Analysis of Positional Data Between Qualisys and HTC Vive Tracker for the Signals during Impact

These results highlight that while the HTC Vive Tracker offers acceptable performance in gradual or smooth motion, it faces critical limitations in reliably tracking sudden impacts, particularly in depth and lateral axes. Such behaviour is expected due to inertial sensor noise, IMU overshoot, and a lower sampling resolution than the Qualisys optical system.

Despite these trade-offs, the analysis of these sources indicates that markerless systems, such as the HTC Vive Tracker, remain viable options for extended-use applications when their limitations are effectively managed. To overcome these limitations, the research conducted by Chan et al. [22] suggests applying machine learning models to correct motion inaccuracies over time, thereby enhancing stability during long-duration rehabilitation tasks. These correction techniques offer promising directions for future systems, particularly where precise movement replication is crucial, such as in gait analysis and motor skill training.

Further innovation in VR locomotion systems includes Walking-In-Place (WIP) techniques, which allow users to navigate virtual environments without requiring large physical spaces. A study by Suma et al. [95] demonstrated a WIP implementation using positional and orientation tracking, achieving step recognition accuracy above 99% across various head positions and zero error during squat movements. This method improves user safety and spatial awareness, making it suitable for indoor VR scenarios with constrained mobility.

3. INTEGRATING A PERIPHERAL INPUT DEVICE IN VIRTUAL REALITY

3.1. Description of the Proposed System

The system developed for this research aims to deliver a stutter-free, accurate, and immersive rowing experience by processing data from the Concept-II rowing machine and rendering it in VR. The major challenge in integrating peripheral sports equipment into VR stems from the data transmission rate between these two hardware components. Both devices are developed for specific uses and perform their functions adequately within their respective domains.

Smartphones can generally refresh at a rate of 60 frames per second (FPS) [96]. The Performance Monitor, however, transmits data to the VR rowing application at 10 Hz. This frequency disparity means the VR rowing application cannot receive new data with every frame, leading to stuttering during the VR experience.

$$n = [0, 1, 2, \dots]; \qquad 3.1\text{—}1$$

here n represent the number of frames rendered by the smartphone (VR mobile).

$$m = \frac{n+1}{6}; \qquad 3.1\text{—}2$$

here m denotes the number of data packets sent from the PM to the smartphone.

Based on the given definition, the data transmission rate between the VR mobile application and the Performance Monitor can be represented in terms of time. Here, t_i denotes the timestamp for the VR mobile and t_j represents the timestamp of the data received from the Performance Monitor.

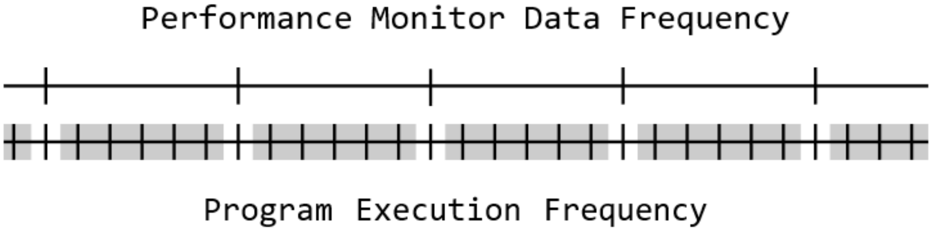

Performance Monitor Data Frequency
Program Execution Frequency

Fig. 11. Representation of Performance Monitor Data Frequency versus Execution Frequency of Application

If the represented execution times are shown in Fig. 11 than the t_i can be represented as:

$$t_i = [t_0, t_1, t_2, t_3 \dots, t_n], i = [0,1,2, \dots, n]; \qquad 3.1\text{—}3$$

here the t_j can be expressed as:

$$t_j = [t_1, t_2, t_3 \dots, t_m], j = \frac{i+1}{6} \qquad 3.1\text{—}4$$

The application renders at a higher frequency than the data sent by PM (Fig. 11). As the application waits for more data from PM (grey area in Fig. 11), the virtual rowing shell remains in VE until new data arrives. This discrete data transmission causes latency and stuttering in the VR mobile's VE. The proposed methods for predicting pending data aim to achieve more accurate, stutter-free rowing sessions. This research suggests three prediction methods using linear interpolation and extrapolation techniques to provide that.

3.2. System Setup

The single-user smartphone system comprises two primary components: a smartphone equipped with a virtual boot system and a Performance Monitor computer. The VR Rowing application interfaces with and retrieves data from the Performance Monitor via a Bluetooth Low Energy (BLE) connection. This setup explicitly requires the fifth Performance Monitor computer model, as it is currently the only model that supports Bluetooth wireless communication using BLE.

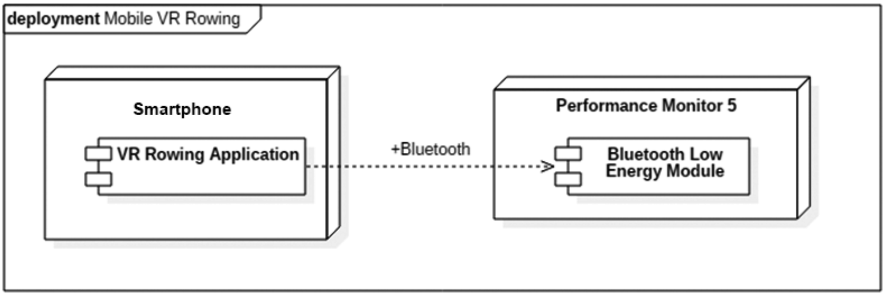


Fig. 12. System Deployment Diagram

During the system's implementation, a "Bluetooth" plug-in was developed for use within the Unity 3D Engine. This plug-in enables the Concept-II Performance Monitor to connect with the smartphone application and seamlessly transfer data parameters.

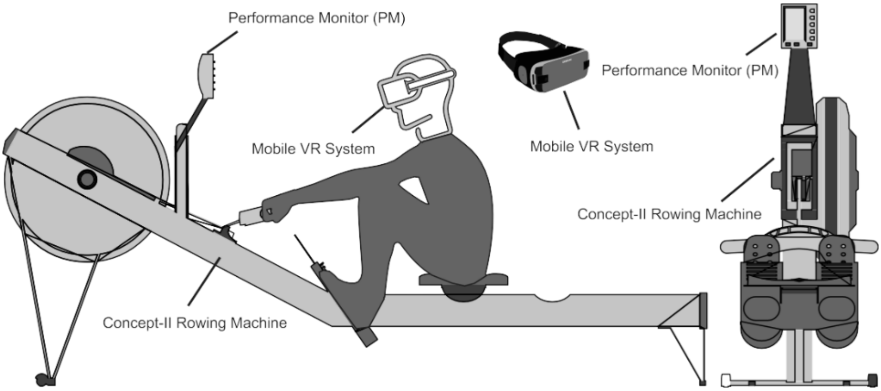


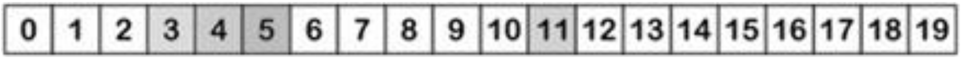
Fig. 13. Virtual Rowing Simulator System Components

Description of each hardware component and its respective role in the system:

- **Performance Monitor –** Stores the following metrics, including travelled distance, average strokes per minute, average power output, and calories burned. Strode data can be transferred via BLE or a USB interface using the CSAFE protocol.
- **Mobile VR System –** The device used in this research is the Samsung Gear VR. The developed VR application is installed on a compatible mobile phone and then attached to the Samsung Gear VR. Data transfer between the mobile phone and the Concept-II Performance Monitor is facilitated via Bluetooth Low Energy.
- **Concept-II Rowing Machine –** The Concept-2 rowing machine, commonly referred to as the Concept-2 Rower, is a high-quality indoor rowing machine designed for both home and professional use.

3.3. Data Management of the System

The rowing workout data is stored in the Performance Monitor (PM) attached to the Concept-II device. This computer records various metrics such as travelled distance, average strokes per minute, average power output, and calories burned. This data can be transferred to other devices via BLE or a USB interface using the CSAFE protocol. Two critical parameters are the rowed distance and the stroke state. The rowed distance moves the virtual rowing shell, while the stroke state is used to animate the virtual rowing shell along with the rower's avatar in the virtual environment. The necessary data is extracted by analysing the characteristics of the Concept-II rowing data, which is stored in a 20-byte array.



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

Fig. 14. Rowing General Status Feature Pack

The rowing distance is stored in the array as a centimetre unit across 3 bytes, starting from the fourth index. The stroke state is recorded in one byte at the 12th index and can have five statuses:

- Waiting until the rowing machine reaches the minimum speed (state value 0).
- Waiting for the rowing machine to reach the simulator speed (state value 1).
- Row driving state (state value 2).
- Rower completes the row (state value 3).
- Rower returns to the start of the row (state value 4).

The Performance Monitor continuously updates these data characteristics. Each time the values are updated, they are sent to the smartphone via its BLE connection. By default, the PM updates values every 0.5 seconds (500 milliseconds). However, this sending frequency can be adjusted by altering the "C2 rowing general status and additional status sample rate" characteristics:

- 0 – 1 second update period.
- 1 – 500ms update period (default).
- 2 – 250ms update period.
- 3 – 100ms update period.

Since the VR rowing application requires real-time data from the PM, it is beneficial to set the default value to 3, corresponding to a 100-millisecond update period.

In VR applications, latency and data loss can significantly degrade system responsiveness, leading to micro stuttering and inaccurate spatial representation. This study introduces three predictive methods to address these challenges, compensating for missing or delayed data in real-time environments. Additionally, the Kalman filter was evaluated as a probabilistic state-estimation method, simulated offline using real-time session data to assess its potential performance under realistic asynchronous input conditions. These approaches aim to improve the continuity and accuracy of signal processing inputs, ensuring consistent object positioning and motion tracking.

3.4. Linear Regression Methods

- **Prediction Using Linear Interpolation by Position:** Prediction Using Linear Interpolation by Position estimates the next position of a virtual object based on a linear relationship between two successive positional data points. This technique assumes a consistent rate of movement and is most effective in scenarios involving predictable, low-variance motion.
- **Prediction Using Extrapolation by Speed:** This method derives the object's instantaneous velocity from consecutive positions and projects the next position accordingly. While it enables continuous prediction, it is susceptible to accumulating error, particularly during abrupt speed changes.
- **Prediction Using Extrapolation by Speed with Correction:** This approach extends the previous one by incorporating an adaptive correction factor. This factor dynamically adjusts the predicted value to reduce discrepancies between the estimated and actual positions, enhancing prediction accuracy during variable-speed conditions.

Each method addresses specific signal irregularities typical in non-native device integration. The subsections explain their mathematical basis, implementation strategies, and performance outcomes.

3.4.1. Prediction Using Linear Interpolation by Position

The activity diagram (Fig. 15) outlines the sequential steps required to calculate the next position of the virtual rowing shell within the VR environment. The process begins with obtaining positional data from the Performance Monitor. This data includes the current and previous positions of the rowing shell. The method then calculates the difference (delta) between these positions to predict the next frame's position. If the time elapsed since the last data reception exceeds the data rate, the interpolation parameter is clamped to prevent the shell's position from exceeding the distance travelled. The diagram visually represents these steps, providing a clear understanding of the linear interpolation process.

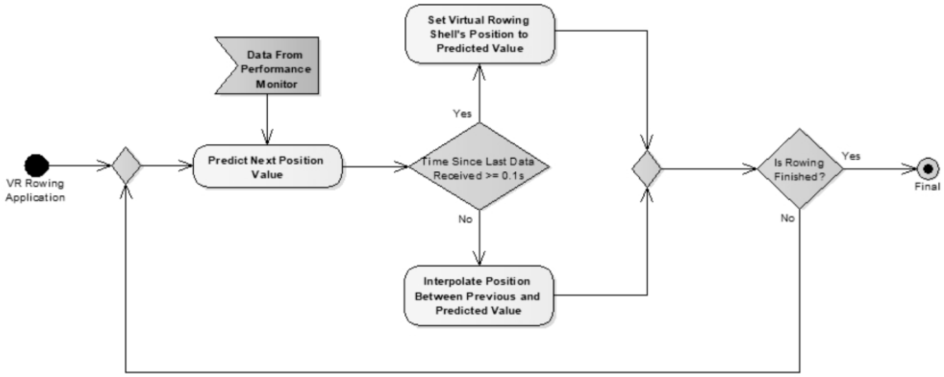

Data From Performance Monitor
Set Virtual Rowing Shell's Position to Predicted Value
Yes
VR Rowing Application
Predict Next Position Value
Time Since Last Data Received >= 0.1s
No
Interpolate Position Between Previous and Predicted Value
Is Rowing Finished?
Yes
No
Final

Fig. 15. Activity Diagram of "Prediction using linear interpolation by position" Method

In the prediction phase for the rowing shell's position within the VE, obtaining an arbitrary value from the PM to apply linear interpolation is essential. The simplest method to predict the data for the next frame is to assume that the change in distance between the current position and the future position will mirror the difference between the current position and the previous position. This difference is calculated as follows:

$$\Delta D_j = D_j - D_{j-1}; \qquad 3.4.1—1$$

here D_j is the current position and D_{j-1} is the previous position and ΔD_j is the position travelled for the given frame.

So, the predicted distance:

$$D_{j+1} = D_j + \Delta D_j \qquad 3.4.1—2$$

When determining the predicted value, the rowing shell's position is interpolated from its previous position to the estimated position until new data is received. This interpolation parameter is recalculated for every application frame. The parameter's value is determined by dividing the time elapsed since the last data was obtained from the PM by the data rate. If the elapsed time exceeds the data rate, the interpolation parameter is capped at 1

$$f(T) = \begin{cases} 1, & ,t_i - t_j \geq 0.1 \\ \dfrac{t_i - t_j}{0.1} & ,otherwise \end{cases} \qquad 3.4.1—3$$

Implementing this conditional function guarantees that the virtual rowing shell's position does not exceed the actual distance travelled on the rowing machine in real-time.

$$BP_i = (1 - T) \times D_j + T \times D_{j+1} \qquad 3.4.1—4$$

here BP_i represents the virtual rowing shell's position within the VR mobile application.

This method relies on an ideal data transfer rate of 0.1 seconds. Suppose the data is not received at this perfect interval. In that case, micro-stuttering may occur, potentially leading to VR sickness for the user, which disrupts the continuous and convincing movement of the virtual rowing shell. To address this, the starting position for interpolation must be the last interpolated position up to the current time. Additionally, delays in incoming data can cause the interpolation to complete before new data is received, leaving the rowing shell stationery for a few frames. Therefore, alternative prediction methods were explored and proposed to overcome these issues.

3.4.2. Prediction Using Extrapolation by Speed

Another method for predicting data focuses on the speed parameter of the virtual rowing shell rather than its position. This approach calculates the rowing shell's position difference between the current and previous frames, then divides it by the data transmission period. $TP = 0.1$ to determine the current speed of the rowing shell.

$$\Delta D_j = D_j - D_{j-1} \qquad 3.4.2—1$$

$$v_j = \frac{\Delta D_j}{TP} \qquad 3.4.2—2$$

To update the position of the virtual rowing shell BP in each frame within the VR mobile application,

$$\Delta t = t_i - t_{i-1} \qquad 3.4.2—3$$

To predict the position of the virtual rowing shell in the next frame, the following extrapolation function is used:

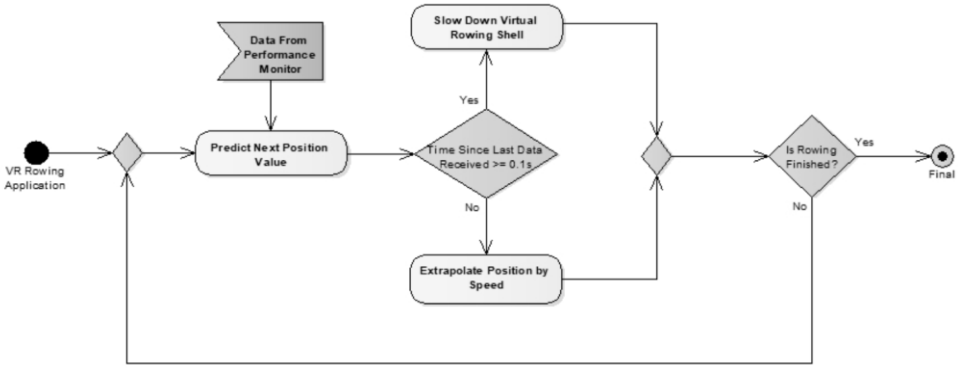
$$BP_{i+1} = BP_i + v_j \times \Delta t \qquad 3.4.2—4$$

This method ensures continuous movement of the virtual rowing shell even when the data reception period from the PM varies. However, frequent changes in the rower's tempo during the session can lead to an increasing error between the rowed distance recorded by the PM and the virtual rowing shell's position in the VR mobile application. Since temporal variations are common and natural in rowing, an additional solution was needed to minimize distance errors between the PM and the

VR mobile application. The following proposed method significantly reduces this error margin compared to the previously described methods.

3.4.3. Prediction Using Extrapolation by Speed with Correction

The activity diagram (Fig. 16) illustrates predicting the virtual rowing shell's position within the VR environment. The diagram starts with the initial data collection from the PM, including the current and previous positions of the rowing shell. The method then calculates the shell's speed and applies a correction factor to adjust it based on whether the predicted position lags or leads the actual data. This adjustment helps maintain the shell's position accuracy throughout the rowing session. The diagram visually represents these steps, highlighting how the correction factor is applied to refine the speed parameter and reduce positional errors.



Data From Performance Monitor
Slow Down Virtual Rowing Shell
Yes
VR Rowing Application
Predict Next Position Value
Time Since Last Data Received >= 0.1s
No
Is Rowing Finished?
Yes
No
Final
Extrapolate Position by Speed

Fig. 16. Activity Diagram of "Prediction using extrapolation by speed with correction" Method

Almost all prediction algorithms include one or more parameters for performance optimisation, and an essential aspect of selecting a suitable algorithm is adjusting these parameter values effectively. In our scenario, to optimise the rowing shell's position in the virtual environment, we use a constant (C) value of 0.25. This constant adjusts the virtual rowing shell's speed in each frame and attempts to correct it. If the virtual rowing shell in the VR mobile application lags behind the Performance Monitor data, the speed is constantly increased. Conversely, if it exceeds, the speed is reduced.

$$f(BP_{i+1}) = \begin{cases} BP_i + v_j \times \Delta t(1 - C), & BP_i \geq D_j \\ \dfrac{t_i - t_j}{0.1}, & otherwise \end{cases} \qquad 3.4.3\text{—}1$$

In this approach, the position of the virtual rowing shell oscillates around the input data, thereby preventing error accumulation.

3.5. Kalman Filter

The Kalman filter is a probabilistic approach for estimating the true position of an object when measurements are delayed, noisy, or irregularly sampled. Unlike deterministic linear regression methods, which directly interpolate or extrapolate between data points, the Kalman filter maintains an internal model of the moving object's position and velocity. It alternates between a prediction step, which estimates future states based on motion dynamics, and a correction step, which refines the estimate when new sensor data becomes available. In this study, the Kalman filter was implemented offline using real-time recordings from the rowing experiments. Although the filter did not operate inside the live VR system, it was driven by the same timing and data irregularities observed during those sessions, replicating the conditions of real-time asynchronous data arrival.

In real time:

- The rowing machine's PM sends cumulative distance data $D_i$at irregular timestamps τ_i.
- The VR system must render the virtual boat position at a fixed 60 Hz ($t_k = k/60$).
- The system must predict the user's distance at each frame between PM packets.

The Kalman filter uses a physical model assuming nearly constant rowing velocity to maintain a smooth, continuous motion trajectory until the next true PM update arrives.

At any time, the filter maintains two internal quantities:

$$x = \begin{bmatrix} p \\ v \end{bmatrix}; \tag{3.5—1}$$

here p is estimated cumulative distance (m) and v is estimated rowing velocity (m/s).

The system assumes nearly constant velocity motion between updates, modelled as:

$$x_k = \underbrace{\begin{bmatrix} 1 & \Delta t_k \\ 0 & 1 \end{bmatrix}}_{F(\Delta t_k)} \ x_{k-1} + W_k; \tag{3.5—2}$$

here $\Delta \mathrm{t}_\mathrm{k}$is the time between VR frames and w_k is the process noise capturing small unmodeled variations (rowing acceleration, stroke irregularity).

Noise covariance is applied on position and velocity in the Kalman filter as:

$$Q(\Delta t_k) = \begin{bmatrix} q_p \Delta t_k & 0 \\ 0 & q_p \Delta t_k \end{bmatrix} \tag{3.5—3}$$

here $q_p$and $q_v$define how much uncertainty accumulates per second in position and velocity.

The PM sends distance measurements such as:

$$z_i = Hx + r_i, H = [1\ 0], r_i \sim N(0, R); \tag{3.5—4}$$

here $z_i$is the cumulative distance $D_i$received from the PM, and R is the expected variance of PM noise measured.

When the session starts, the first two PM data points initialize position and approximate velocity:

$$X_{0|0} = \begin{bmatrix} D_1 \\ \dfrac{D_2 - D_1}{\tau_2 - \tau_1} \end{bmatrix}, P_{0|0} = \begin{bmatrix} \sigma_p^2 & 0 \\ 0 & \sigma_v^2 \end{bmatrix} \tag{3.5—5}$$

For each VR frame t_k:

$$X_{k|k-1} = F(\Delta t_k)x_{k-1|k-1} \tag{3.5—6}$$

$$P_{k|k-1} = F(\Delta t_k)P_{k-1|k-1}F(\Delta t_k)^T + Q(\Delta t_k) \tag{3.5—7}$$

If a new PM packet (τ_i, D_i) arrives at or before the current time t_k correction step is applied, otherwise the prediction continues and the prediction of rowing distance goes as:

$$\widehat{D}(t_k) = H\, x_{k|k} = P_{k|k} \tag{3.5—8}$$

This provides a continuous, low-latency trajectory synchronised to the VR system's 60 Hz frame rate, even when the PM data arrives irregularly or is briefly delayed.

3.6. Experimental Results

Measurement data can be represented in different ways, depending on the type of data obtained [97]. Presenting all session results in one or separate graphs often leads to a noisy outcome, making it difficult to differentiate between sessions. To address this, two sessions were selected and designated as the "Fastest Rowing Session" (FRS (session 9)) and the "Slowest Rowing Session" (SRS (session 5)) to analyse the characteristics of the best and worst-case scenarios (see Table 7 and Table 8). The FRS corresponds to the rowing session that reached the 250-meter distance in the shortest time, whereas the SRS represents the session that took the longest duration to complete the same distance.

The experimental results aim to evaluate the effectiveness of the proposed prediction methods in maintaining distance accuracy over time while ensuring a stutter-free VR experience for the user. The experiment comprises ten rowing sessions, each covering a distance of 250 meters.

Two primary factors are used to evaluate the performance of each method: (1) the time difference between the completion of a session on the VR mobile application and on the rowing machine, and (2) the occurrence of micro-stuttering during the VR rowing experience. The effectiveness of each method is assessed across all rowing sessions simultaneously at corresponding rowing instances. All timing and distance comparisons use the rowing machine as the reference (ground truth).

- The time difference between the two components after completing a session is caused by distance errors within the VR mobile application during a rowing session. Calculation is performed by taking the rowing machine as the reference (ground truth) for both distance and time.
- Micro-stutter measurement is another critical aspect that affects the smooth, continuous motion of the virtual rowing shell within the VR mobile application. In this experiment, micro-stutter S is measured by verifying if the distance of the current frame D_i is the same as the previous frame D_{i-j} for all proposed methods.

$$f(S) = \begin{cases} S + 1, & if\ D_i = D_{i-1} \\ S = S, & otherwise \end{cases} \qquad 3.6\text{—}1$$

3.6.1. Synchronisation Accuracy Results Between the VR Mobile Application and the Rowing Machine

The time discrepancy arises from a distance error between the VR mobile application and the rowing machine. This distance error results in a time difference at the end of a rowing session, where the session in the VR mobile application may conclude either earlier or later than the actual rowing performed on the machine (see Table 7).

Table 7. Time Difference in Milliseconds (ms) after Completing a Session

	Methods			
Session	Interpolation by position	Extrapolation by speed	Extrapolation by speed with correction	Kalman Filter
1	+39.9	-6828	-62	+43
2	+77.5	-8469	-18.4	+173
3	+57.4	-7492	+0.2	+201
4	+34	-4894	-5	+5
5 (SRS)	+94.3	-8453	-63.7	+217
6	+81.4	-6193	-14	+186
7	+43.9	-6700	-33.5	+6
8	+34	-7064	+16.6	+52
9 (FRS)	+100	-4758	-43.3	+226
10	+85.2	-7362	+32.8	+189
Sessions Average	+64.7ms	-6821 ms	+28.95 ms	+129.8 ms

The results presented in Table 7 indicate that the time difference between the VR mobile application and the rowing machine varies depending on the applied prediction method.

The interpolation-by-position method consistently produced small, positive discrepancies, indicating that the VR application typically finished slightly later than the rowing machine. This behaviour results from the nature of interpolation, in which the VR system continuously waits for the next valid data point from the performance monitor before updating the rowing shell's position in the virtual environment. As a result, the VR application maintains temporal accuracy but introduces a slight delay at the end of each session because it relies on previously received data.

In contrast, extrapolation by speed shows large negative discrepancies (from −4,894 *ms* to −8,469 *ms*, averaging −6,821 *ms*), meaning that the VR application consistently finishes sessions earlier than the actual rowing machine. This cumulative overshoot arises from minor yet consistent speed overestimations, as the extrapolation method assumes uniform velocity between data updates and ignores the natural deceleration that occurs during each rowing stroke. The significant negative offset observed across all sessions demonstrates that uncorrected extrapolation is not suitable for precise synchronisation tasks.

The constant introduced nearly eliminated the timing offset. The discrepancies ranged narrowly around zero, with a mean of +28.95 *ms*, showing that the VR session concluded almost simultaneously with the physical rowing machine. The correction factor continuously compensates for accumulated prediction drift, adjusting each subsequent estimation based on the measured difference between predicted and actual positions. This adaptive behaviour preserves the smoothness of extrapolation while maintaining precise temporal alignment. The minimal residual error confirms that the corrected extrapolation method provides the most reliable and synchronized performance among all tested techniques.

The Kalman filter method produced moderate positive discrepancies, ranging from +5 *ms* to +226 *ms*, with an average of +129.8 *ms*, indicating that the VR application finished slightly later than the rowing machine. This delay results from the filter's smoothing mechanism, which combines the predicted state with the most recent measurement to estimate each position. While this process reduces measurement noise and ensures consistent motion estimation, it also slightly delays the response to rapid velocity changes, resulting in a small temporal offset. The positive discrepancies observed across all sessions show that the Kalman filter maintains stable timing behaviour, with a consistent but minor lag relative to the rowing machine's completion time.

3.6.2. Motion Smoothness Results Between the VR Mobile Application and the Rowing Machine

The motion smoothness results evaluate how continuously the rowing shell moves within the VR environment during each physical rowing session

Table 8. Micro Stutter Count and Average Duration for each Session with different Applied Methods

			Methods			
Session	Measured metrics	Input	Interp. by position	Kalman Filter	Extrap. by speed	Extrap. by speed with correction
S1	N of stutters	478	263	135	0	0
	Avg. duration	110.8 ms	37.9 ms	48.7 ms		
S2	N of stutters	524	289	162		
	Avg. duration	111.2 ms	39.8 ms	53.6 ms		
S3	N of stutters	452	260	138		
	Avg. duration	111.2 ms	37 ms	52.5 ms		
S4	N of stutters	426	227	128		
	Avg. duration	111.1 ms	38.9 ms	53.6 ms		
S5(SRS)	N of stutters	667	354	210		
	Avg. duration	111.3 ms	38.2 ms	53.7 ms		
S6	N of stutters	447	238	131		
	Avg. duration	111.3 ms	40 ms	56.6 ms		
S7	N of stutters	437	253	134		
	Avg. duration	111.2 ms	37.8 ms	53.6 ms		
S8	N of stutters	535	273	160		
	Avg. duration	111.2 ms	40 ms	51.4 ms		
S9(FRS)	N of stutters	405	202	116		
	Avg. duration	111.1 ms	40.8 ms	53 ms		
S10	N of stutters	448	247	140		
	Avg. duration	111.1 ms	39.4 ms	52.5 ms		
Sessions Average	N of stutters	481.9	260	145.4		
	Avg. duration	111.15 ms	38.98 ms	52.92 ms		

Table 8 presents the number and average duration of stutters recorded for each prediction method, allowing comparison of their visual continuity performance. The input refers to data transmission behaviour from the Performance Monitor to the VR mobile application without applying any method.

The interpolation-by-position method significantly reduced the total number of stutters compared to the raw input signal from the performance monitor. Across all sessions, the number of stutters decreased from an average of approximately 482 events in the input data to about 260 events, representing a 46% reduction. This improvement shows that interpolation effectively smooths the motion compared to the raw input received from the performance monitor in the VR application. The average stutter duration also decreased from about 111 ms to 39 ms, which corresponds to a 65% reduction, indicating that the individual pauses were significantly shorter and less noticeable. Although interpolation minimizes motion discontinuities, it still exhibits some stuttering because it remains dependent on the timing of new data packets. When packets arrive late or irregularly, the interpolation loop completes before new information is available, causing brief standstills until the next valid update.

The Kalman filter method further improved motion continuity, reducing the average number of stutters by nearly 70% compared to the input data and by about 44% relative to the interpolation method. It produced an average of 145 stutter events per session. However, the average stutter duration increased slightly compared to interpolation (from 39 ms to 53 ms) but remained about 52% shorter than in the raw input data. This trade-off indicates that the Kalman filter favours stability and continuous updates over rapid, small corrections, resulting in fewer interruptions that last slightly longer.

The speed-based extrapolation produced no recorded stutters across all sessions, with both the number and duration of events measured as 0. This outcome indicates that the predicted position of the rowing shell was continuously updated, without interruption, throughout the simulated rowing sessions. The absence of stutters is a direct result of the method's design, which projects the shell's position forward at a constant velocity between data updates. Because the motion prediction does not rely on the arrival of new data packets, the virtual shell continues to advance at a steady rate. While this approach maintains continuous movement, it does not incorporate natural variations in rowing speed, and therefore, timing discrepancies were observed in synchronisation results (see Table 7).

The extrapolation by speed with the correction method also showed no stuttering events, with both the number and duration of pauses remaining at zero. The application of a correction term to compensate for accumulated prediction drift did not introduce any interruptions in the movement of the virtual shell. This indicates that the correction process operates gradually enough to preserve motion continuity while refining alignment with the actual rowing data. When considered together with the synchronisation results in Table 7, the method maintained continuous motion and minimized end-of-session timing differences.

3.7. Visualisation of the Results

The plotted data reflect the error distribution between the distance covered in the virtual environment in the VR mobile application and the actual input distance from the rowing machine. Each histogram is plotted with 32 bins, following Sturge's Rule [98]. Bins below zero indicate how much the virtual rowing shell lags behind the input distance parameter. In contrast, bins above zero show how far the virtual rowing shell has travelled compared to the input distance parameter from the rowing machine. Negative distance error values indicate that the virtual rowing shell in the VE lags behind the physical distance by the corresponding amount, whereas positive error values indicate that it moves ahead of the physical rowing distance.

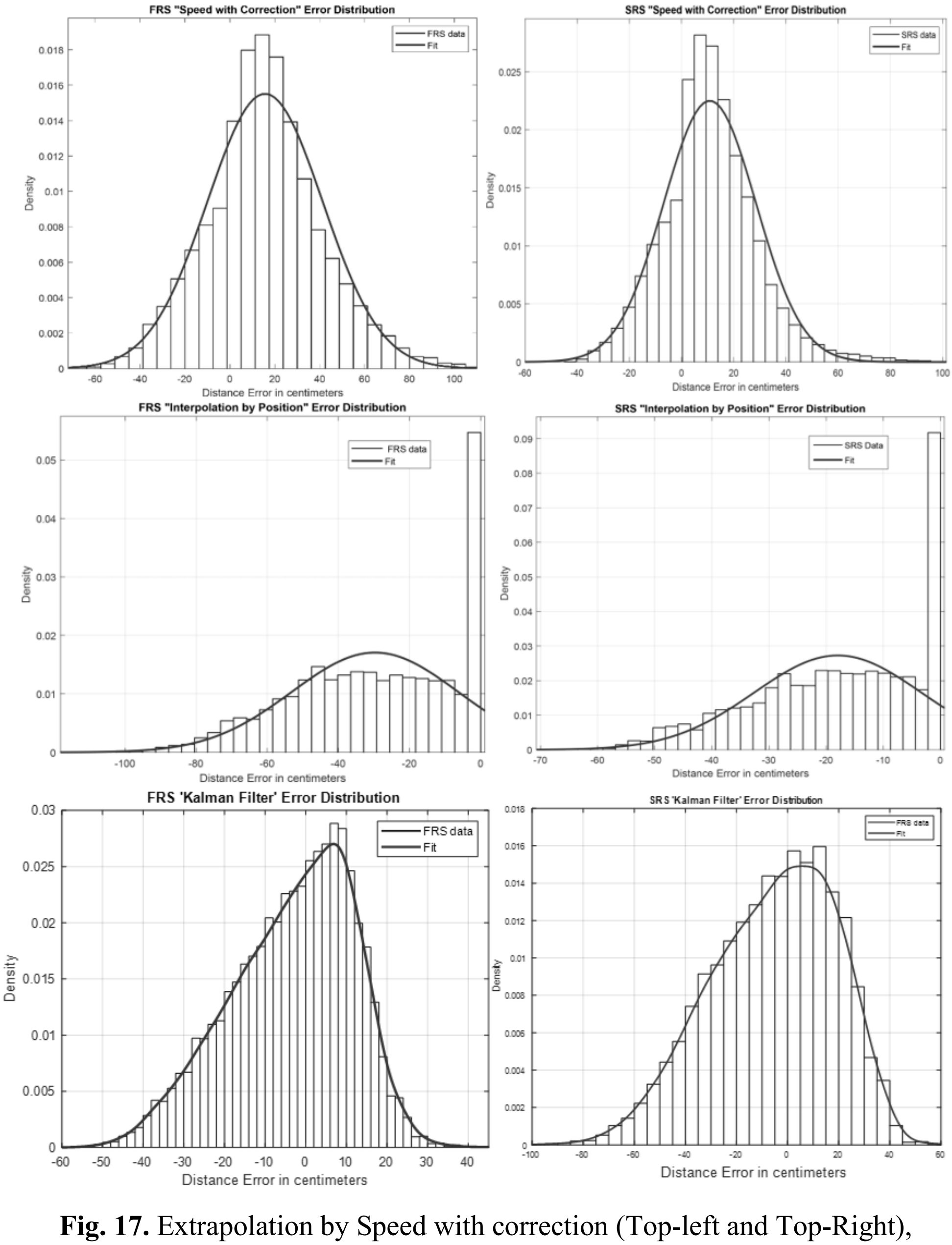


Fig. 17. Extrapolation by Speed with correction (Top-left and Top-Right), Interpolation by position (Middle-Left and Middle-Right), Kalman Filter (Bottom-left and Bottom-Right)

The extrapolation by speed with the correction method in the FRS graph (see Fig. 17 – top left) shows characteristics similar to those of SRS (see Fig. 17 – top right). There is a 51.2% probability that the data will fall within −20 to 20 centimetres of the input distance during the VR mobile application run-time, and a 25.42%

probability that the distance data in the VR mobile application will remain within −10 to 10 centimetres of the rowing machine's input distance. The SRS plot shows a symmetrically distributed distance error, with a higher probability of rowing approximately 0 to 23 centimetres beyond the input distance data (see Fig. 17 – top right). This pattern reflects consistent and accurate proximity to the input distance values transmitted by the rowing machine. For the SRS, there is a 70.19% probability that the data will be within −20 to 20 centimetres of the input distance during run-time, and a 38.69% probability that it will fall within −10 to 10 centimetres, closely following the rowing machine's input distance.

The prediction using the linear interpolation by position method exhibits a left-skewed distribution (see Fig. 17 – middle left and middle right). A large portion of the data is concentrated around a normalized value of approximately 0.02, with a higher frequency in the −30 to −5 centimetre range. The FRS plot displays characteristics similar to those observed in the SRS (see Fig. 17 – middle left), although the magnitude of error occurrence is slightly higher in the FRS. Based on the graphs, the data ranges −50 to −10 centimetres and −5 to 0 centimetres are of particular interest due to their higher data density. The probability is 58.18% that the distance data within the VR mobile application will be −50 to −10 centimetres behind the input distance parameter, and 21.09% that it will be −5 to 0 centimetres behind. In the SRS plot, the method's distance consistently trails the input distance parameter, meaning that the VR mobile application completes the virtual rowing session slightly after the physical session ends. To quantify this behaviour, two intervals were analysed—−30 to −5 centimetres and −5 to 0 centimetres—where data occurrences are most frequent. The probability is 53.07% that the distance data within the VR mobile application will be −30 to −5 centimetres behind the input distance parameter, and 25.13% that it will be −5 to 0 centimetres behind (see Fig. 17 – middle right).

The Kalman filter method displays a right-skewed pattern across both the FRS and SRS rowing sessions (see Fig. 17 – bottom left and bottom right). In the FRS graph, the values are concentrated between −25 and +25 centimetres, and the highest density occurs just above zero, around +5 to +10 centimetres. This suggests that during faster rowing, the virtual shell tends to advance farther than the physical rowing distance. Approximately 70% of the data lies within −20 to +20 centimetres, and nearly 40% falls inside the −10 to +10 centimetre interval (see Fig. 17 – bottom left). The SRS graph shows a wider, more symmetric distribution spanning roughly −80 to +50 centimetres, with its peak near +10 centimetres. Despite the broader spread, the density remains cantered near zero, indicating that the Kalman filter maintains general synchronisation even when the rowing motion slows and input updates become less uniform. Around 65% of the data fall within the −20 to +20 centimetre range, while 30% remain within −10 to +10 centimetres, showing a moderate increase in variability but no systematic bias (see Fig. 17 – bottom right).

In contrast, the extrapolation by speed method produced much more volatile results in estimating the input distance parameter from the rowing machine (see Fig.

18). This implementation represents the unmodified version of the “extrapolation by speed with correction” method.

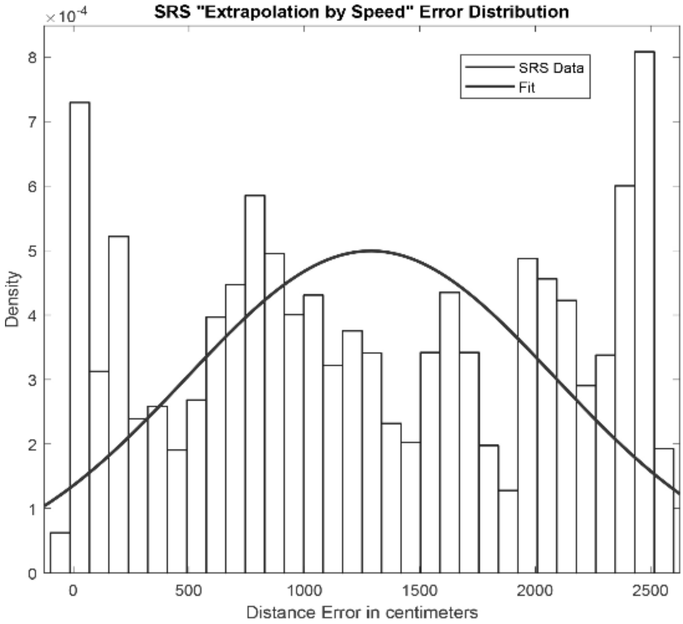


Fig. 18. SRS - Error Distribution for the Extrapolation by Speed Method

The two methods differ considerably in their ability to predict the input distance data from the rowing machine. Plotting this model provides valuable insights for other researchers, illustrating how a minor yet precise adjustment to an algorithm can yield a substantially more effective outcome. Since this model accumulates distance error exponentially over time, comparing the SRS and FRS scenarios does not offer meaningful conclusions. Therefore, only the SRS scenario is presented, as it provides a concise overview of the method’s performance in a single graph (see Fig. 18).

3.8. Conclusion

This research investigated two key issues affecting VR rowing immersion: micro-stuttering and time-based distance errors, both resulting from the differing data transmission rates between the rowing machine and the VR mobile application. To address these challenges, four prediction algorithms were developed and evaluated. The experimental findings indicate that:

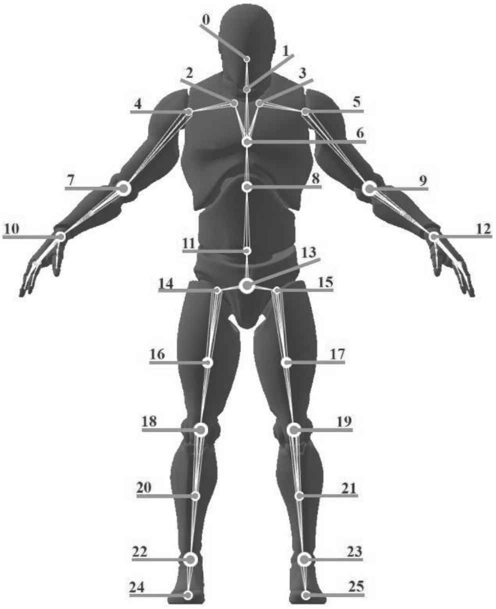
- The interpolation by position method provides accurate distance values throughout the virtual rowing session, consistently tracking the input distance parameter. However, it introduces significant stuttering, which can lead to an unpleasant VR experience, disrupt immersion, and potentially cause VR sickness. Because the method always lags slightly behind the input distance parameter, the session in the virtual environment consistently concludes a little later than the user's real-time rowing.

- The extrapolation by speed method results in an increasing distance error throughout the virtual rowing session, causing the session in the VR environment to finish much earlier than the actual completion time on the rowing machine. The error becomes more pronounced as the session duration increases to complete the 250-meter rowing distance. Although this algorithm provides a stutter-free rowing experience, it does not produce a satisfactory or accurate outcome.
- The Kalman filter method provides a balanced compromise between motion smoothness and synchronisation accuracy. It significantly reduces the number of stutters compared to the interpolation method but introduces a moderate delay and slightly longer individual stutter durations. Overall, it performs better than interpolation in motion stability but remains less precise than the extrapolation with the correction method in both timing and smooth VR experience.
- The extrapolation by speed with the correction method significantly reduces distance error, allowing the process to closely follow the input distance parameter. Across all sessions, the distance error remains within −20 to +20 cm for 50–70% of the time and within −10 to +10 cm for 25–40%, depending on session speed. This approach outperforms all other methods near the input data, minimizing the end-of-session time difference and delivering a completely stutter-free VR experience throughout all sessions.

Although the proposed prediction algorithms were experimentally validated using the Concept-II rowing machine, the same mathematical framework can be applied to other peripheral devices that produce time-series motion data with different sampling frequencies (e.g., cycling, running, rehabilitation robots). Since the algorithm operates on motion parameters such as position, velocity, and time step, it can be recalibrated for devices with different motion amplitudes or frequencies.

4. AI-SUPPORTED POSTURE EVALUATION USING OFF-THE-SHELF VR CONTROLS

The research in this section aims to diagnose human posture using HTC-Vive trackers attached to the patie's body. The UE4 mannequin is a standardized 3D model featuring a predefined skeleton with specific bone nodes, as shown in Fig. 19, facilitating consistent animation and motion capture across various applications. By aligning our VR sensors with the UE4 mannequin's bone nodes, we accurately mapped real-world movements to the virtual model. This alignment enabled measurement of joint angles and detection of compensatory movements during exercises. For each exercise, the nodes to be examined were selected, and the required changes in the nodes' angles during both correct and incorrect execution were described.



0. head
1. neck
2. clavicle_r
3. clavicle_l
4. upper arm_r
5. upper arm_l
6. spine_03
7. elbow_r
8. spine_02
9. elbow_l
10. hand_r
11. spine_01
12. hand_l
13. hips
14. thigh_r
15. ithigh_l
16. femur_r
17. femur_l
18. calf_r
19. lower_leg
20. tibia_r
21. tibia_l
22. calcaneus_r
23. calcaneus_l
24. foot_r
25. foot_l

Fig. 19. Unreal Engine 4 Skeletal Structure of Skeletal Nodes

The developed system, built upon the proposed framework shown in Fig. 20, provides angle-oriented data output that feeds CNN and RF models designed to yield distinctive results for anomalies in patient posture. The motivation for this approach arises from clinical assessment practices in which the therapist assesses the patient from the front, back, and sides. However, it is essential to assess static and dynamic postures to evaluate the patient's functional mobility and ability to self-correct static postural habits. Conditions such as scoliosis, postural decompensation, anatomical leg-length discrepancies, previous trauma or surgery, impaired trunk control (e.g., after a stroke), or specific segmental somatic dysfunctions, often related to asymmetry, may arise from postural misalignment. There are crucial elements to consider at the onset of a rehabilitation program [99]. Active range of motion (ROM) in all peripheral joints, especially the shoulder and upper limbs, should be monitored and recorded during postural assessments and training to ensure exercises are performed accurately. Pain, weakness, muscle shortening, and edema can all restrict joint mobility. Deficiencies in shoulder ROM, muscle performance, and strength can contribute to postural deviations or incorrect posture, leading to improper execution of exercises during training. For this reason, we conducted a comprehensive evaluation of the active ROM of the shoulder (including extension/elevation, adduction/abduction, horizontal adduction/abduction, and rotation), as these are the factors most likely to affect postural changes or trigger compensatory mechanisms. For instance, full shoulder elevation can influence trunk control, potentially leading to hyperextension of the back [100, 101]. Building on these clinical considerations, the next section details how the proposed workflow captures and interprets motion data to support posture analysis using machine learning models, including CNNs and RFs.

4.1. Proposed Workflow for Human Posture Analysis

The proposed workflow presents a systematic approach for capturing, processing, and interpreting human motion data to support postural and exercise analysis within a rehabilitation context (see Fig. 20). The system integrates VR tracking technology with computational modelling to evaluate the type and quality of the exercises performed. The workflow begins with the setup of VR trackers at predefined body locations, followed by the capture of positional data during exercise performance. The collected data are then processed to compute vectors and corresponding joint angles based on the bone node structure. These calculated angles serve as inputs to a trained deep learning CNN model, which first detects the performed exercise and subsequently classifies the posture quality as correct or incorrect.

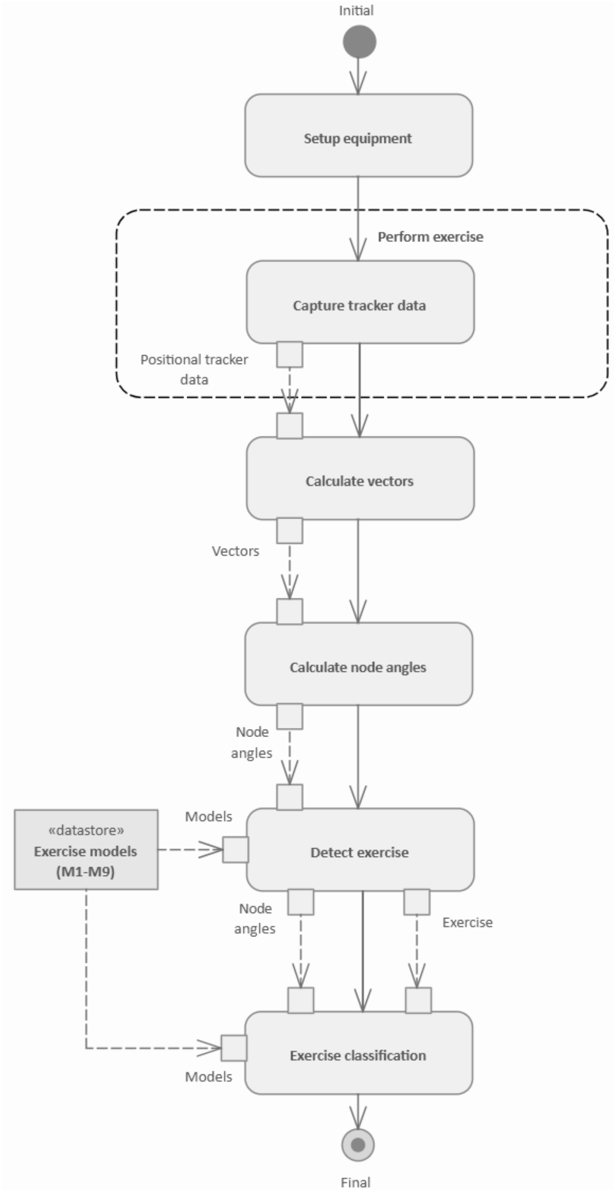


Fig. 20. Proposed Human Posture Analysis System Workflow

In summary, the proposed workflow provides a structured process for automated exercise identification and posture evaluation based on motion-tracking data. By integrating biomechanical angle extraction with a deep learning classification model, the system translates movement data into measurable outcomes that support rehabilitation assessment.

4.1.1. System Setup (Setup Equipment)

This section outlines the configuration of the human exercise tracking system for recording exercises shown in the section 4.2. The system was developed using the HTC Vive platform and at least eight second-generation HTC Vive sensors. These sensors enable the tracking of both spatial position and rotation angles. The system requires at least 2 HTC Vive base stations, though 4 are recommended for improved ease of use and more precise tracking. The arrangement of the sensors is illustrated in Fig. 21.

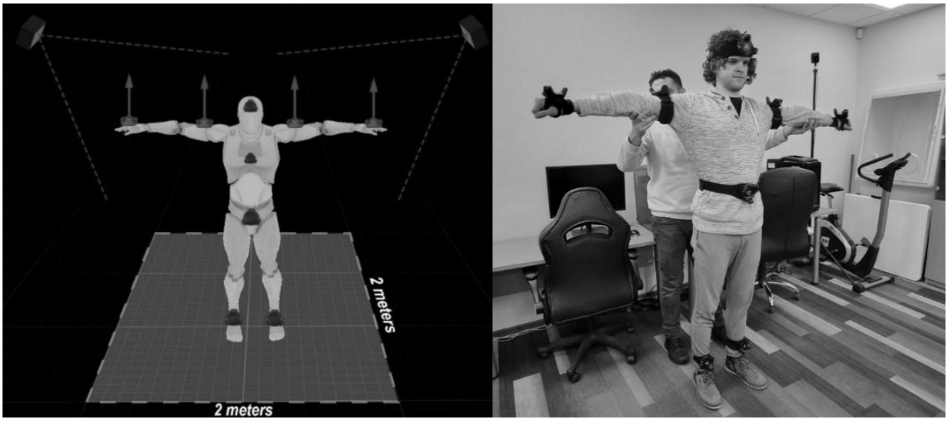


Fig. 21. System and Sensor Layout Overview on the left, and the Actual Patient view on the right

The sensors are positioned as follows:

- Two sensors are attached to the hands, facing upward;
- Two sensors are placed on the arms, also facing upward;
- Two sensors are positioned on the legs, facing forward;
- One sensor is mounted on the hips, facing forward;
- One sensor is placed on the head, facing forward;
- An optional sensor can be placed on the chest, facing forward.

Before starting the exercise tracking session, the sensors must be correctly placed on the patient, as shown in Fig. 22. Ensuring that the sensors' position and orientation are accurate is crucial. The sensors should be secured tightly to prevent movement or distortion during exercise. Once the sensors are installed on the subject, they must be aligned with the virtual avatar during the system calibration process.

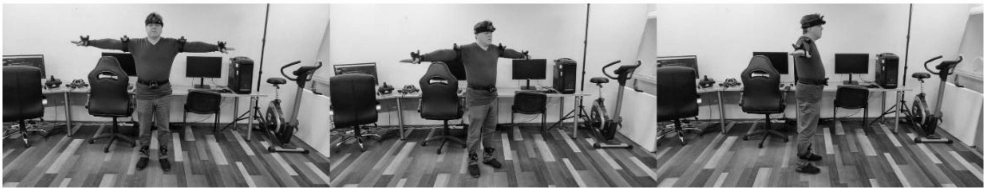


Fig. 22. Example of Sensor Placement from 3 different Angles

The subject begins by standing in a 'T' position (standing upright with arms extended outward) and extending both hands forward after a 10-second hold. For post-stroke patients, holding the "T" position for 10 seconds may be challenging due to hemiparesis or decreased muscle tone, strength, or coordination. In such cases, assistance from a caregiver or nurse is required to maintain the correct arm positions during calibration (see the right picture in Fig. 21). Because the assistant does not wear sensors, the system does not detect them, and they do not interfere with skeleton reference registration, even if their hands temporarily cover the patient's sensors. VR sensor calibration aims to link VR trackers with the human subject and a virtual avatar. During calibration, each sensor is assigned to a specific body part, and after performing a few movements, the person is synchronized with their virtual avatar. Once calibration data is recorded, the exercise session can proceed.

4.1.2. Positional Data Acquisition from VR (Capture Tacker Data)

Positional data acquisition, or, in brief, the recording of positional data during exercise sessions, is performed using HTC Vive sensors integrated into Virtual Reality. These sensors enable tracking of spatial position (x, y, z) for each sensor (Fig. 23).

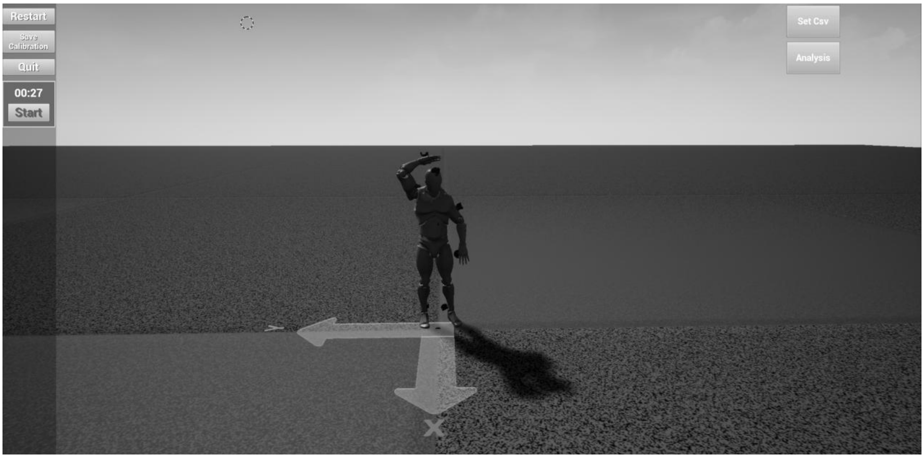
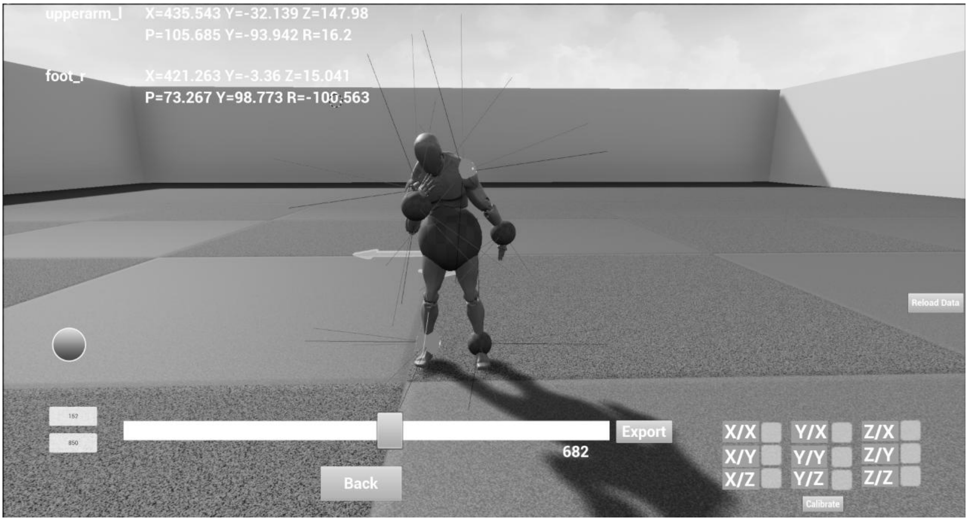


Fig. 23. Recording/Capturing Phase of the Patient in the System

Each recorded sensor's positional data is exported as CSV files, which can later be imported into the system for further manual or ML analysis of each desired bone in vector form. The next section describes this import and vector conversion phase in the system framework.

4.1.3. Conversion of Positional Data to Vectoral Data (Calculate Vectors)

In human-motion analysis using motion-capture systems and digital skeletal models, raw spatial information, such as the three-dimensional position and orientation of each bone, must be transformed into vectorial representations before further biomechanical calculations can be performed (see Fig. 24). The angle to be extracted can be selected by clicking on the ball place holders, which allows the calculation of pre-defined angular measurements between specified vectors (see Fig. 24).



```
upperarm_l   X=435.543 Y=-32.139 Z=147.98
             P=105.685 Y=-93.942 R=16.2

foot_r       X=421.263 Y=-3.36 Z=15.041
             P=73.267 Y=98.773 R=-100.563
```

Fig. 24. Imported Patient data enables Vector representation and the calculation of Angles for export to ML

These vectoral representations provide a consistent mathematical description of bone or segment directions, enabling comparisons across different body parts over time.

In the Unreal Engine skeleton, each bone node provides either positional coordinates $\mathrm{p}_B = [x_B, y_B, z_B]^T$ in world space or a complete rotation $R_B^W \in \mathbb{R}^{3\times3}$describing the bone node's local axes relative to the world coordinate system. Both types of data can be converted into unit vectors that describe the bone node's local orientation in space.

In most cases, Unreal Engine provides the orientation of each bone node rather than explicit joint positions. Each bone node's local rotation relative to the world frame is represented by the rotation matrix:

$$R_B^W = [x_B^W \ y_B^W \ Z_B^W] \qquad 4.1.3\text{—}1$$

here x_B^W, y_B^W, Z_B^W are the three column vectors corresponding to the bone node's local forward, right, and up directions, respectively, expressed in world coordinates.

From this rotation, the principal local axes can be extracted as:

$$\begin{aligned} \mathrm{f}_B &= R_B^W \begin{bmatrix}1\\0\\0\end{bmatrix} \quad \text{(forward vector)} \\ \mathrm{r}_B &= R_B^W \begin{bmatrix}0\\1\\0\end{bmatrix} \quad \text{(right vector)} \;\; ; \\ \mathrm{u}_B &= R_B^W \begin{bmatrix}0\\0\\1\end{bmatrix} \quad \text{(up vector)} \end{aligned} \tag{4.1.3—2}$$

here, for every time frame k, the bone node's rotation $R_B^{W,(k)}$updates to reflect the participant's motion. So the forward vector is obtained as:

$$\mathrm{f}_B^{(k)} = R_B^{W,(k)} \begin{bmatrix}1\\0\\0\end{bmatrix} \tag{4.1.3—3}$$

This transformation expresses each bone node vector in a hierarchical reference frame, ensuring that angular relationships reflect pure joint motion.

4.1.4. Angle Calculation Between Vectors (Calculate Node Angles)

After converting the skeletal bone node data into vector form, the next step in the motion analysis process is to compute the angular relationships between selected bone nodes (see example in Fig. 25). These angular measures describe how one bone node segment moves relative to another for further analysis in ML models.

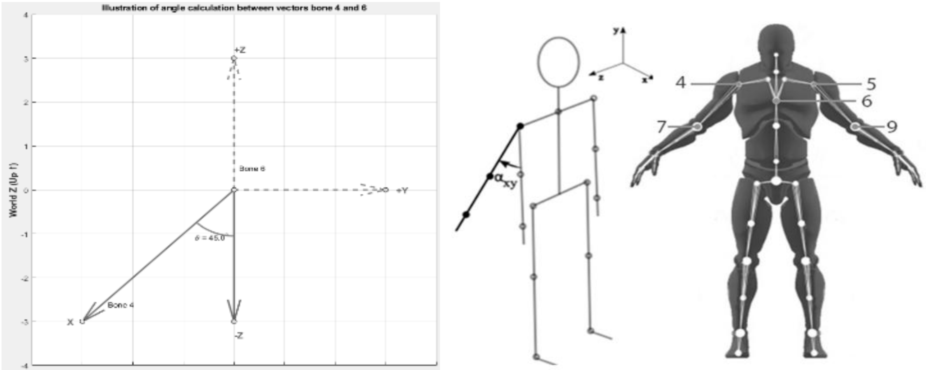


Fig. 25. Illustration of Angle Calculation between Bone Node 4 forward and Bone Node 6 downward vectors

Each bone node provides a rotation matrix $R_B^W \in \mathbb{R}^{3\times3}$, which defines its local orientation relative to the world coordinate system. From each matrix, the bone's direction vectors (forward, right, up) are extracted ($+X$ = forward, $+Y$ = right, $+Z$ = up). These vectors are then compared directly in world space to determine the angular difference between any two bones. Once bone vectors are derived from their local orientations or world-space transformations, the relative motion between two bones can be represented as the angular displacement between their respective direction vectors.

To determine the angular relationship between two bones:

$$\theta^{(k)} = atan2(n \cdot \left(v_i^{(k)} \times v_j^{(k)}\right), v_i^{(k)} \cdot v_j^{(k)}) \qquad 4.1.4—1$$

here B_i and B_j, the corresponding directional vectors $\mathrm{v}_i^{(k)}$and $\mathrm{v}_j^{(k)}$ within the same reference frame k.

Angles to be calculated for each exercise are presented in sections 4.4.2, 4.4.3, 4.4.4, 4.4.5, 4.4.6, 4.4.7, 4.4.8, 4.4.9.

4.1.5. Detection of the Exercises (Detect Exercise)

Once the node angles have been computed, the next stage of the workflow is the exercise detection process. This module identifies which exercise the participant is performing based on dedicated angles. For this purpose, the framework maintains a set of pre-trained CNN models, each specialized for a single exercise (see section 4.2 for the full list of exercises). These models are trained independently using data collected from multiple participants, ensuring that each CNN learns the characteristic temporal and kinematic patterns of its corresponding movement type. When a new recording is received, the system passes the processed node angle data through each exercise model.

The angular data are pre-processed using the scaling and feature alignment parameters associated with each model. This ensures that the input features match the model's expected configuration. Each CNN then outputs a set of probabilities indicating how likely the input motion is to correspond to its trained exercise. The model producing the highest overall confidence or classification accuracy is selected as the detected exercise.

Formally, if $M_i(X)$ represents the softmax output of model i for the input data X, the selected exercise index i^* is obtained as:

$$i^* = \arg \max_i \mathrm{Conf}(M_i(X)) \qquad 4.1.5—1$$

here Conf(·) denotes the maximum predicted class probability or mean prediction confidence across all test windows.

4.1.6. Exercise Classification

Following exercise detection, the framework proceeds to exercise classification, which determines whether the patient's movement was correct or incorrect. Once the most probable exercise model M_{i^*}has been selected, the same CNN is used to evaluate the motion's quality.

Each exercise dataset consisted of:

- **Class 0 (Correct Movement)**: The participant performed the exercise as intended, ensuring precise alignment of angles and posture.
- **Class 1 (Incorrect Movement 1)**: The participant performed the exercise with deliberate errors, simulating common compensatory behaviours.
- **Class 2 (Optional Incorrect Movement 2)**: A different incorrect variation was performed, introducing a second type of error for analysis.

The predicted movement quality is obtained by taking the class with the highest softmax probability:

$$\text{Quality} = \arg\,\max\,(\text{Softmax}(M_{i^*}(X))) \qquad 4.1.6—1$$

4.2. Experimental Setup

This section describes the experimental results of the analysed exercises. Healthy participants were included in the study and performed correct and incorrect movements during predefined exercises. Each movement was tracked using VR sensors placed on critical anatomical points, referred to as bone nodes (Fig. 19).

The data obtained after the experiments were then analysed. A total of 19 execution variations were recorded for both correct and incorrect movements. The following tools were used for the analysis:

- **Confidence Intervals**: These intervals indicate that 95% of the measured values fall within the specified range, meaning there is only a 5% probability that a particular value lies outside this interval. Confidence intervals provide a reliable measure of data variability and accuracy.
- **Bean Plots**: Bean plots highlight data density through “bean-shaped” distributions. Each “bean” represents the data density contours, mirrored to form a symmetrical polygon. The thickest part of the bean represents regions of higher data density, while narrower sections indicate lower density. A colour gradient visualizes the spread of the data, with lighter shades indicating higher concentrations.
- **Histograms**: These are bar charts that graphically display the statistical distribution or density of the data. Histograms provide an intuitive visualisation of how values are distributed across the measurement range.
- **Student’s t-test**: This test evaluates whether the means of two samples are equal. A null hypothesis is proposed, stating that the means of the two groups are equal. If the obtained p-value is more significant than the significance level of 0.05, the null hypothesis cannot be rejected. Conversely, the null hypothesis is rejected if the p-value is less than 0.05.

- **F-test**: The F-test checks whether the variances of two groups are equal. A null hypothesis is proposed, assuming that the variances of the two groups are equal. If the obtained *p*-value exceeds the significance level of 0.05, the null hypothesis cannot be rejected. Otherwise, it is rejected if the *p*-value is less than 0.05.

The project partner, VilniusTech University, provides the experimental exercises. The exercises defined are as follows:

- Frontal elevation of the arm
- Sideways elevation of the arm
- Horizontal Abduction/Adduction of the Arm
- Overhead Reach of the Arm
- Lifting and Overhead Placement of a Heavy Object
- Axial Rotation of the Extended Arm
- Nose Touch Coordination
- Forearm Supination/Pronation

The study was conducted by the Declaration of Helsinki and was ethically approved by the Institutional Review Board of VilniusTech University Faculty Committee (No. 64-2221).

4.3. Methodology

A range of classification and prediction techniques can be applied to health assessment, enabling the identification of potential pathologies based on digital images, biological signals, motion data, survey responses, and similar sources [102, 103]. Machine Learning (ML) uses advanced statistical and probabilistic techniques to develop systems that automatically learn from data. Due to their ability to rapidly and accurately analyse complex datasets, ML algorithms have become highly prevalent in research on various health disorders, supporting improved patient outcomes and a deeper understanding of physiological conditions and their regulation [104]. Depending on the amount of data or the information available on the data sample itself, an algorithm category or several algorithms shall be selected for the study. After testing, the model that best describes the data is selected.

4.3.1. Random Forest

An RF model is composed of multiple decision trees (DTs), which can be trained either sequentially or simultaneously using random data samples [105]. Within each tree, nodes have specific conditions defining how data is split into two sets based on specific properties. Analysing recorded signals, such as motion angles, in healthy individuals can reveal notable differences compared to those with movement disorders. The RF model includes several parameters that must be predefined. Since there is no universal rule for selecting the parameter for a particular dataset, determining the best configuration can be time-intensive. A random grid search is employed to streamline this process, in which 1,000 random combinations of parameter values are evaluated. The following hyperparameters of the RF model are adjusted:

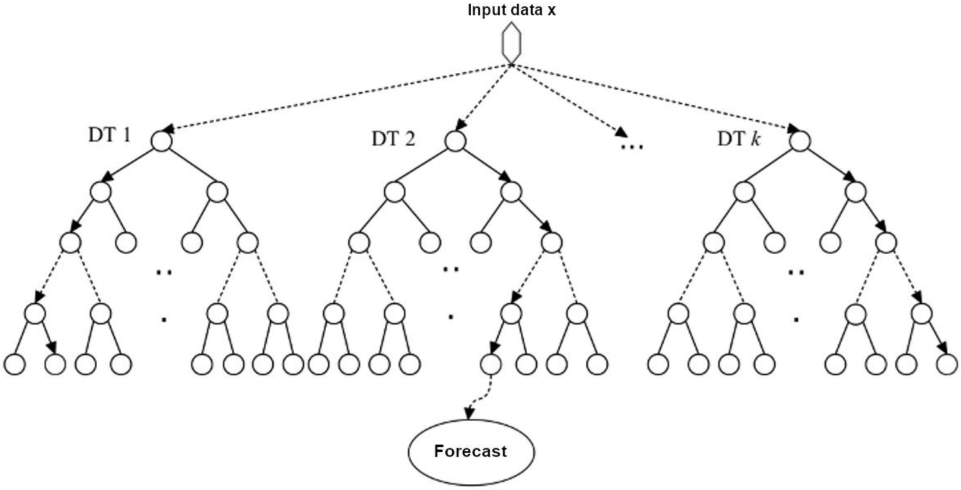


Fig. 26. Classification Scheme using Random Forest

Including all raw signals obtained during movement in the creation of an RF model can lead to challenges, such as overly complex decision trees or inaccurate final predictions due to excessive and redundant information. Three statistical features — mean, standard deviation, and range (distance) —were selected for each measured angle to address this issue. These features, calculated using the provided formulas, effectively summarize the data while retaining the essential characteristics for accurate classification.

1. Mean formula: The arithmetic mean (average) of the angles for a given measurement window:

$$m_i = \frac{\sum_{j=1}^{n_i} A_{i,j}}{n_i} \qquad 4.3.1—1$$

here m_i is the mean value for the i-th feature (angle). $A_{i,j}$ is the individual angle measurement for the i-th feature in the j-th observation. n_i is the total number of measurements for the i-th feature and $\sum_{j=1}^{n_i} A_{i,j}$ summation of all measurements for the i-th feature.

2. Standard Deviation formula to calculate the variability of angle measurements around the mean.

$$s_i = \sqrt{\frac{1}{n_i}\sum_{j=1}^{n_i}\left(A_{i,j} - m_i\right)^2} \qquad 4.3.1—2$$

here s_i is the standard deviation for the i-th feature (angle). $A_{i,j}$ is the individual angle measurement for the i-th feature in the *j*-th observation. m_i is the mean value for the i-th feature and $\sum_{j=1}^{n_i}\left(A_{i,j} - m_i\right)^2$ is the summation of squared deviations from the mean.

3. Distance formula to calculate the distance between the most extensive and smallest angle measurements.

$$d_i = \max_j A_{i,j} - \min_j A_{i,j} \qquad 4.3.1—3$$

here d_i is the distance (range) for the i-th feature. $\max_j A_{i,j}$ is the maximum observed value for the i-th feature across all observations. $\min_j A_{i,j}$ is the minimum observed value for the i-th feature across all observations.

In this dissertation, 70% of the data sample is randomly selected to train and validate the RF model, while the remaining 30% is reserved for testing.

4.3.2. Convolutional Neural Network

A Convolutional Neural Network (CNN) is a type of multilayer neural network that includes at least one convolutional layer. A convolutional (conv) layer consists of artificial neurons that perform mathematical cross-correlation operations by combining two separate data inputs. This process replaces the original data representation by reducing the dimensionality of the input data. Such a layer is essential in any CNN architecture, as it reduces the number of parameters required to describe the data and shortens training time. Pooling functions, such as convolutional layers, reduce the data size while retaining the most significant numerical values from each segment, typically via average or max pooling. To prevent overfitting in a CNN, a dropout layer can be added. This layer introduces controlled randomness during training, making the learning process noisier and reducing the model's reliance on specific input data [88]. After the input data passes through all the previously mentioned layers, a flattening process is applied, during which the matrix-shaped data are converted to a vector. This vector then serves as the input for the artificial neural network (ANN). Within the ANN, a “dense” operation takes place; each neuron in the given layer receives outputs from all neurons in the preceding layer. As the data move to the next layer, which contains fewer neurons, a matrix-vector product is performed. In this study, convolutional neural networks with varying sizes and layer configurations were examined. Ultimately, a CNN architecture comprising two convolutional layers, one pooling layer, one connecting layer, and smoothing, along with two compression steps, was selected to classify individuals with and without mobility impairments. The sequence of operations, along with the input and output data and their visual representation, is shown in Figures 27 and 28.

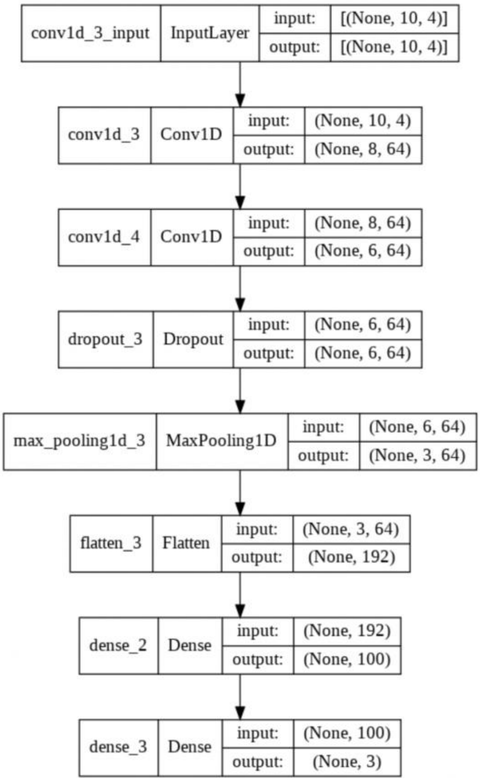

conv1d_3_input | InputLayer | input: [(None, 10, 4)] | output: [(None, 10, 4)]
conv1d_3 | Conv1D | input: (None, 10, 4) | output: (None, 8, 64)
conv1d_4 | Conv1D | input: (None, 8, 64) | output: (None, 6, 64)
dropout_3 | Dropout | input: (None, 6, 64) | output: (None, 6, 64)
max_pooling1d_3 | MaxPooling1D | input: (None, 6, 64) | output: (None, 3, 64)
flatten_3 | Flatten | input: (None, 3, 64) | output: (None, 192)
dense_2 | Dense | input: (None, 192) | output: (None, 100)
dense_3 | Dense | input: (None, 100) | output: (None, 3)

Fig. 27. Sequence of layers in a Convolutional Neural Network

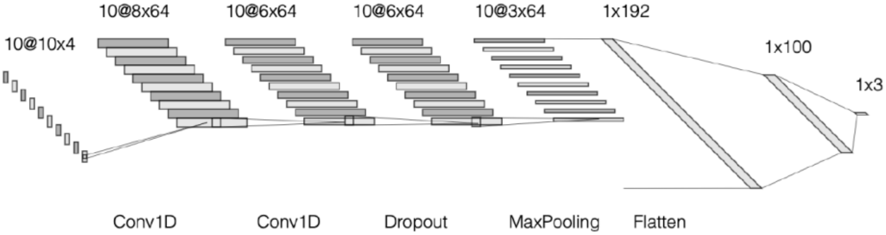


Fig. 28. Layout Diagram of a Convolutional Neural Network

Our network uses a two-step approach: first, it predicts 2D joint positions from the input images, and then it estimates the 3D pose based on those positions. For the 2D joint estimation module, we utilise the state-of-the-art stacked hourglass method [106]. To generate multiple 3D pose hypotheses, we employ a custom-designed processor that includes both a feature-extraction component and a hypothesis-generation module [107]. Each hourglass consists of an encoder–decoder structure, where the encoder performs down-sampling via convolution and pooling, while the decoder performs up-sampling via bilinear interpolation. Since multiple hourglasses are stacked to iteratively execute bottom-up and top-down processing, this architecture is referred to as a stacked hourglass network [108]. The model collects data at various input sizes. In addition, interim supervision is applied to the heatmaps produced by each stack. In addition to the fact that a CNN can have various structures (layers), it is also necessary to define the parameters required for computations, such as the number of epochs, batch size, and the amount of data used for validation. As with the random forest model, a random search grid is created to automatically construct the CNN model for each analysed movement. After 100 iterations, the model that classifies the data most accurately is selected.

- **Number of epochs (epochs)** – indicates how many times the entire input dataset is used for training. The grid uses the following array of values: [6, 8, 10, 12, 14, 16, 18, 20, 21, 22, 50];
- **Batch size (batch_size)** – indicates the size of the input data batch used during the training process before updating the model and moving to the next epoch. The grid uses the following array of values: [20, 28, 34, 40, 48, 55, 68, 74, 80];
- **Validation split (validation_split)** – indicates what portion of the dataset is used for training and what portion is used for validation. The grid uses the following array of values: [0.05, 0.1, 0.15, 0.2, 0.25, 0.3].

In binary supervised learning classification, each element in the validation (or test) set is assigned to either the positive or negative class (typically 0 or 1). In this study, most exercises have two negative classes labelled 1 and 2, both indicating incorrectly performed movements. The machine learning algorithm is trained to distinguish between these two or three classes based on the given data. Ultimately, a prediction is made for each element in the test dataset. Based on these predictions, the algorithm assigns each element to one of the categories listed in Table 9.

Table 9. Categories of ML Elements obtained after prediction

Name	**Label**	**Description**
True Negative	TN	Element is predicted as incorrect, and is incorrect
True Positive	TP	Element is predicted as correct and is correct
False Positive	FP	Element is predicted as correct, but is incorrect
False Negative	FN	Element is predicted as incorrect, but is correct

If, during testing/validation of the ML algorithm, many elements fall into the TP or TN categories, it means the algorithm correctly classifies elements as positive when they are (TP) and as negative when they are (TN). This entire table is also known as a confusion matrix. When only two classes are distinguished, the confusion matrix consists of the four values presented. To understand how well the algorithm performs in general, the overall model accuracy can be calculated using the formula:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \qquad 4.3.2—1$$

For angle-based classification, we utilise a Single-input CNN, which is both accurate and straightforward for extracting movement-related patterns from joint angles.

1. Layers (Conv1D): Convolutional layers extract spatial features from time-series data using filters. The convolution operation is defined as:

$$Z_{i,j} = \sum_{k=1}^{K} W_k \cdot X_{i+k-1,j} + b \qquad 4.3.2—2$$

here $Z_{i,j}$ is the output feature map. $X_{i+k-1,j}$ is the input data at time i and feature j. W_k are the learnable weights of the filter. b is the bias term. K is the filter size.

2. MaxPooling Layer: MaxPooling reduces the dimensionality of feature maps while maintaining the most significant values. For a pooling size p, the output is:

$$P_{i,j} = \max(X_{i:i+p,j}) \qquad 4.3.2—3$$

here $P_{i,j}$ is the pooled feature. $X_{i:i+p,j}$ represents a segment of the input feature.

3. Dropout Layer: Dropout randomly sets a fraction p of the weights to zero during training:

$$W_{dropped} = W \cdot M, M \sim Bernoulli(1-p) \qquad 4.3.2—4$$

here M is a binary mask applied to the weigths.

4. Dense Layers: Fully connected layers are used to perform the classification task. The dense operation is:

$$Z = W \cdot X + b \qquad 4.3.2—5$$

here Z is the output of this dense layer. W and b are the weights and biases of the dense layer.

4.3.2.1. Decision-Based Classification

For Decision-based classification, we combine CNN’s feature extraction with LSTM's sequential modelling to classify pseudo-participants.

1. Convolutional Layers (Conv1D): As described above, convolutional layers extract spatial and temporal features.

2. Bidirectional LSTM: LSTMs to capture sequential dependencies. The forward LSTM computes:

$$h_t^{\rightarrow} = f(W_x X_t + W_h h_{t-1}^{\rightarrow} + b) \qquad 4.3.2—6$$

The backwards LSTM computes:

$$h_t^{\leftarrow} = f(W_x X_t + W_h h_{t-1}^{\leftarrow} + b) \qquad 4.3.2—7$$

The final output:

$$h_t = [h_t^{\rightarrow}, h_h^{\leftarrow}] \qquad 4.3.2—8$$

here $h_t^{\rightarrow}, h_h^{\leftarrow}$ are hidden states in forward and backward directions. W_x, W_h are the weights for input and hidden states. b is the bias term.

3. MaxPooling, Dropout, and Dense Layers, as described above, reduce dimensionality, prevent overfitting, and perform classification.

In two-class supervised learning, each element in the validation (or test) set is assigned to either the positive or negative class, typically represented as 0 or 1. In this study, most exercises involve two negative classes, labelled as 1 and 2, indicating incorrect movements. Based on the provided data, the machine learning algorithm is trained to distinguish between these two or three classes. Ultimately, the model predicts each item in the test sample, and the algorithm assigns each item to one of the specified categories based on those predictions.

4.4. Analysis of Statistical Data and Results of the Exercises

This section presents the statistical visualisation and results for the eight exercises performed during the study. Each exercise is analysed individually, with detailed visualisations and statistical evaluations of the movements. The data collected includes angles measured between specific body nodes during correct and incorrect movement scenarios. The results are presented using statistical plots, such as confidence intervals, bean plots, and histograms, alongside machine learning classification outcomes from Random Forest and Convolutional Neural Network models.

4.4.1.1. Description of Shared Parameters in Each Exercise

- **Angle** – The angle statement in each exercise refers to the specific joint or body segment angles analysed to assess movement correctness. These angles are carefully selected to reflect the critical aspects of each exercise and identify deviations that indicate compensatory behaviours or incorrect movements.
- **Class** – Class classifications categorise participants' correct or incorrect moments to perform. Each class represents a distinct movement condition that explains whether a movement is correct or incorrect.

The graph parameters that are presented in the feature importance analysis for the Random Forest model:

- A – stands for angle
- The number after A - is the angle number (e.g., A1 is "angle 1").
- The d, s, and m after the number:

- d (derivative): Refers to the rate of change of the angle. This is often used to measure how quickly an angle changes during a movement, providing insights into the dynamics of motion (e.g., joint speed or acceleration).
- s (standard deviation): Represents the angle variability during the exercise. A higher standard deviation suggests a greater inconsistency in the movement, which may indicate compensatory behaviours or a lack of control.
- m (mean): Denotes the average value of the angle throughout the movement. This provides an overall sense of how the angle behaves during the exercise and helps identify whether it remains within expected ranges for both correct and incorrect movements.

Each CNN classifier construction process utilised complete movement signals, with angles measured every 16.6 milliseconds.

4.4.2. Frontal Elevation of the Arm Exercise

The frontal elevation of the arm exercise involves raising the arm vertically upward using only the arm muscles. The arm is moved from a neutral, lowered position to a fully elevated position in front of the participant. This exercise evaluates the participant's ability to maintain proper posture and isolate arm movement without compensating with the torso or shoulder.

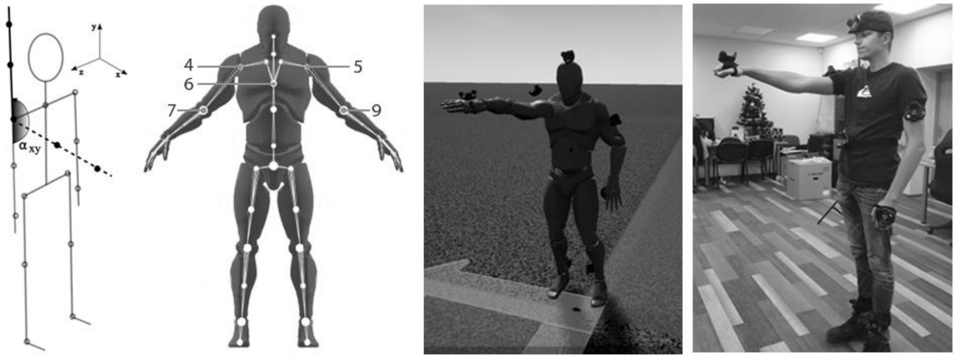


Fig. 29. From left to right: Frontal elevation of the arm exercise movement angle, analysed nodes, 3D mannequin in the system following the participant's movements, participant

This exercise analysed nodes n4, n5, n7, n9, and n6. The following criteria were considered to evaluate the exercise:

- **Angle 1** - The arm must remain fully extended without bending at the elbow. The elbow elevation angle measured at nodes n7 or n9 should be approximately 180°.

- **Angle 2** - The arm must be raised forward and upward. The angle between the shoulder line and the extended arm, measured at nodes n4 or n5, should be approximately 90°.
- **Angle 3** - The participant should maintain an upright posture. The spinal tilt angle at node n6 should be close to 90° relative to the horizontal plane.

In the experiment, data were collected from 19 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 95 individual datasets have been formed for each class.

- **Class 0 (Correct Movement)**: Executed the arm elevation correctly, maintaining proper form and alignment. This group is represented in blue on Fig. 30 and green on Fig. 31.
- **Class 1 (Incorrect Movement 1)**: Executed the movement incorrectly by compensating with the torso, deviating from the correct technique. This group is represented in green on Fig. 30 and in red on Fig. 31.
- **Class 2 (Incorrect Movement 2)**: Executed the movement incorrectly by compensating with the shoulder, further straying from the proper form. This group is represented in yellow on Fig. 30 and in blue on Fig. 31.

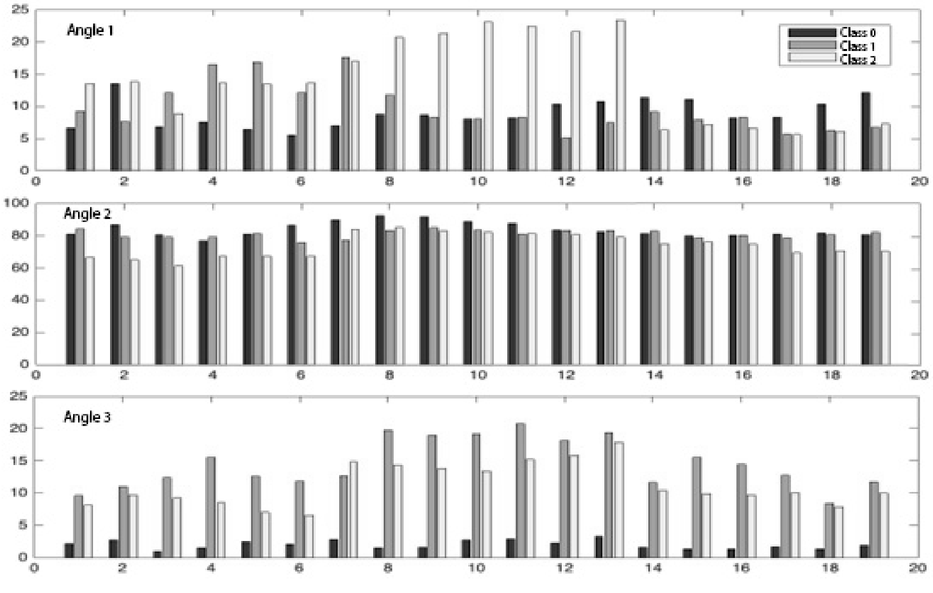


Fig. 30. Angle mean values of participants for the Frontal Elevation of the Arm Exercise

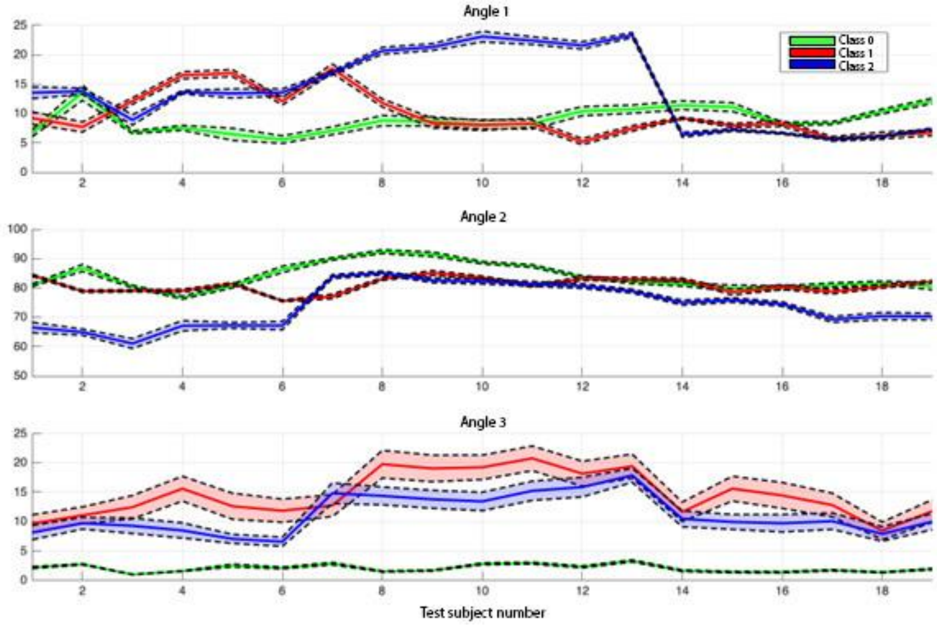


Fig. 31. Confidence Intervals of Participants for the Frontal Elevation of the Arm Exercise

In the analysis of Fig. 30 and Fig. 31 for the Frontal Elevation of the Arm exercise, distinct patterns emerged across all angles and classes. For Angle 1, Class 0 (blue/green) consistently displayed values between ~10° and ~15°, with tight confidence intervals indicating proper elbow extension, while Class 1 (green/red) exhibited higher means ranging from ~15° to ~20°, and Class 2 (yellow/blue) showed the largest deviations, peaking at ~25° and demonstrating greater inter-subject variability. For Angle 2, Class 0 maintained values tightly cantered at ~90°, reflecting accurate arm elevation, while Class 1 showed significantly reduced averages (~70°–85°) with broader confidence intervals, and Class 2 performed slightly better with means around ~80°–90°, though still below the target. Angle 3 revealed the clearest differentiation of posture, with Class 0 maintaining values close to ~0°–5°, demonstrating excellent upright posture and minimal variability, while Class 1 showed substantial deviations ranging from ~10° to ~20° in Fig. 30, with confidence intervals in Fig. 31 reaching up to ~30°, indicating significant torso misalignment. Class 2, though moderately improved compared to Class 1, showed deviations of ~10° to ~25°, highlighting compensatory movements at the shoulders. Across all angles, Class 0 consistently showed minimal variability and correct movement, while Classes 1 and 2 exhibited increasing deviations and variability, particularly in Angle 3, which proved the most sensitive indicator of compensatory movements.

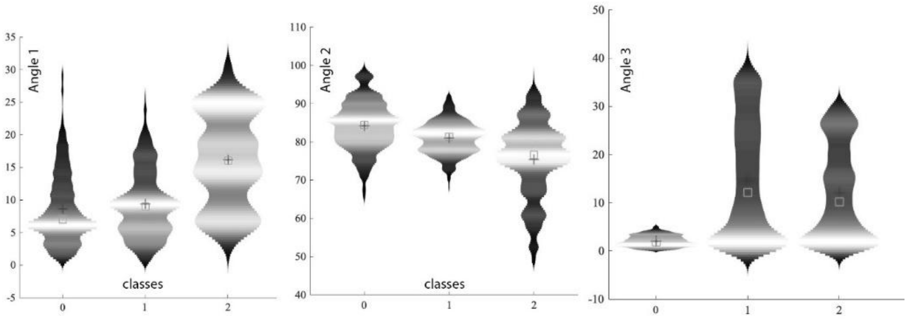


Fig. 32. Bean Plots of Evaluated Angles for the Frontal Elevation of the Arm Exercise

The bean plots in Fig. 32 The Frontal Elevation of the Arm exercise reveals distinct patterns across Angle 1, Angle 2, and Angle 3, with varying degrees of distinguishability among Classes 0 (correct movement), 1 (torso compensation), and 2 (shoulder compensation). For Angle 1, Class 0 maintains a tight distribution between ~10° and ~15°, indicating consistent and proper elbow extension. Class 1 exhibits slightly higher mean values, ranging from ~15° to 20°, with moderate variability, making the distinction between Classes 0 and 1 identifiable but not particularly pronounced. Class 2 exhibits the broadest distribution, with mean values peaking at ~25°, clearly distinguishing it from the other classes. For Angle 2, Class 0 shows a tightly clustered distribution at ~90°, representing correct arm elevation. While Class 1 shows lower averages of ~70° to ~85°, the overlap in distributions with Class 2, which ranges from ~80° to ~90°, makes the separation between these two compensatory classes less clear and harder to definitively distinguish. Angle 3, however, provides the sharpest differentiation among classes. Class 0 exhibits a highly concentrated distribution around 0° to 5°, reflecting excellent upright posture, while Class 1 shows significant deviations from 10° to 25°, with high variability due to torso compensation. Similarly, Class 2 exhibits a comparable range of ~10° to 25°, highlighting shoulder compensation; however, the broader spread suggests it is slightly less consistent. Overall, the results confirm that Angle 3 is the most sensitive and reliable indicator for detecting compensatory movements due to its sharply defined separations. In contrast, Angles 1 and 2 are moderately effective, with less distinguishable patterns between Classes 1 and 2, particularly in Angle 2, where overlaps in compensatory movements are observed.

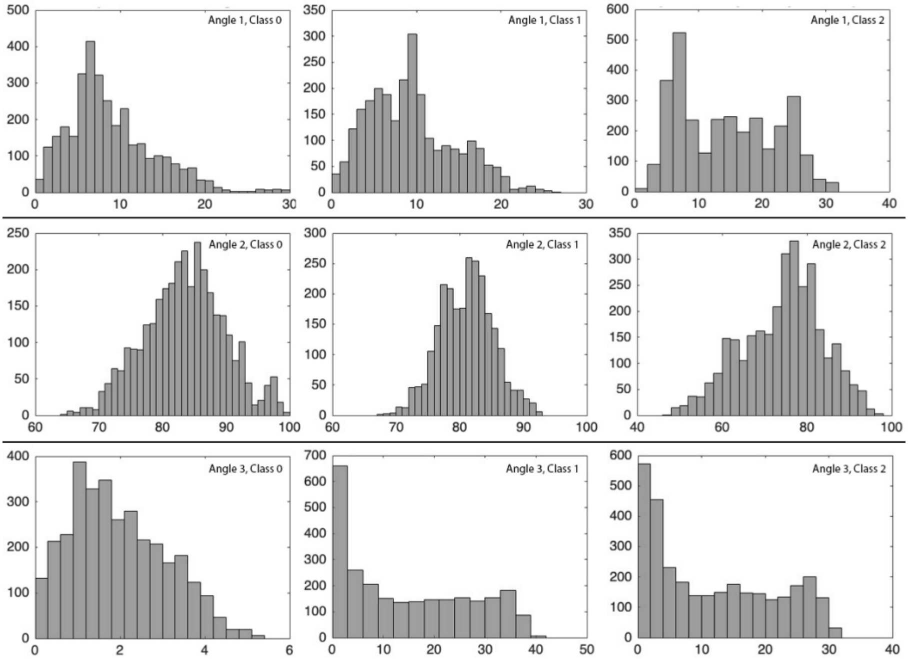


Fig. 33. Histograms of Evaluated Angles for the Frontal Elevation of the Arm Exercise

The histograms (Fig. 33) demonstrate that Class 0 movements consistently show tight distributions across all angles, reflecting correct execution with minimal deviations. For Angle 1, Class 0 is concentrated below 20°, while Classes 1 and 2 exhibit progressively higher peaks (up to 30°–35°) due to compensatory behaviours, with Class 2 showing the most significant deviations. Angle 2 reveals the most consistent alignment for Class 0 at 85°–90°, whereas Classes 1 and 2 show overlapping distributions peaking around 75°–85°, with Class 2 exhibiting a broader spread toward lower values. Angle 3 shows the most apparent differentiation: Class 0 remains tightly centred at 0°–5°, while Classes 1 and 2 deviate significantly, with peaks around 10°–25° and extending up to 40° for Class 2, highlighting substantial posture misalignments. These findings emphasize Angle 3 as the most sensitive metric for distinguishing compensatory movements, particularly between Classes 0 and the incorrect movement classes.

Table 10. Statistical Analysis of Frontal Elevation of the Arm Exercise Angles

Angle	Compared samples	t-test		F-test	
		p-value	Rejected?	p-value	Rejected?
Angle 1	Class 0 vs Class 1	0	Yes	0,8375	No
	Class 0 vs Class 2	0	Yes	0	Yes
Angle 2	Class 0 vs Class 1	0	Yes	0	Yes
	Class 0 vs Class 2	0	Yes	0	Yes
Angle 3	Class 0 vs Class 1	0	Yes	0	Yes

	Class 0 vs Class 2	0	Yes	0	Yes

The table's results indicate statistically significant differences in the means and variances between healthy individuals and those with movement disorders. In only one instance, the null hypothesis could not be rejected; however, even in this case, the incorrect movement was performed adequately, as shown in the histograms (see Fig. 33). The study revealed that movement disorders are most easily detected during arm elevation during arm exercise, using data from angle 3 (spinal tilt angle at node 6).

As mentioned earlier, features with calculated accuracy values below the threshold of 0.05 are removed and excluded from further analysis. In this case, no statistics related to "Angle 1" were used in the creation of the Random Forest model, as all three associated metrics were below the selected threshold (see Fig. 34).

4.4.2.1. Frontal Elevation of the Arm Exercise Decision Classification

The RF classifier identified Angle 3 and Angle 2 as the most influential features for classifying the disorder, as evident from the feature importance analysis (Fig. 34), with the cumulative feature contribution indicating their critical role in distinguishing between the exercise classes. Both RF and CNN achieved 100% classification accuracy for patient-based disorder detection for this exercise, as shown in Fig. 35. The confusion matrices substantiate this result, with RF correctly classifying all test samples across three classes (0, 1, 2) without any misclassification. Similarly, CNN produced a perfect classification with no errors in its confusion matrix.

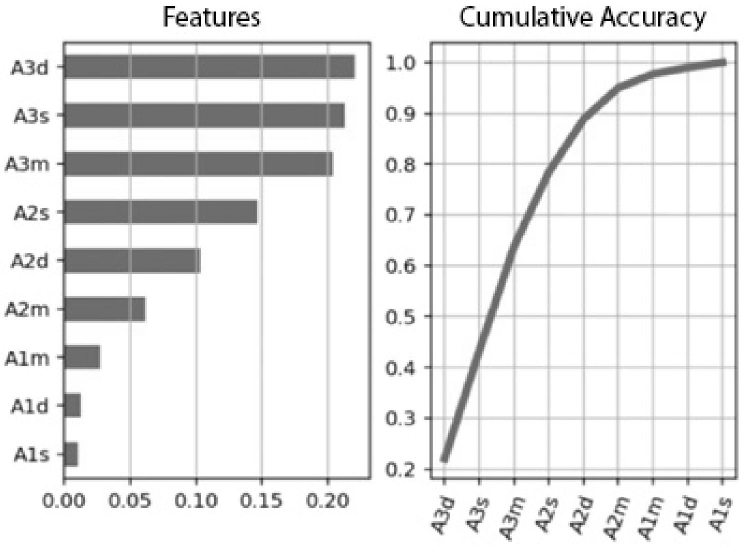


Fig. 34. Feature Importance Analysis for Frontal Elevation of the Arm Exercise

The execution time for RF was 0.03 seconds, demonstrating its computational efficiency. In contrast, CNN required 46.75 seconds due to its computationally intensive iterative process, involving 36860 iterations over 38 epochs. RF required only 10 trees, determined through OOB-based early stopping, with a maximum depth of 3, whereas CNN employed a batch size of 6 and a 20% validation split.

Table 11. Metrics of the Decision Classification for the Frontal Elevation of the Arm Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	10	Number of iterations	36860
Maximum depth	3	Number of Epochs	38
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.03 s	46.75s	
Accuracy	100%	100%	

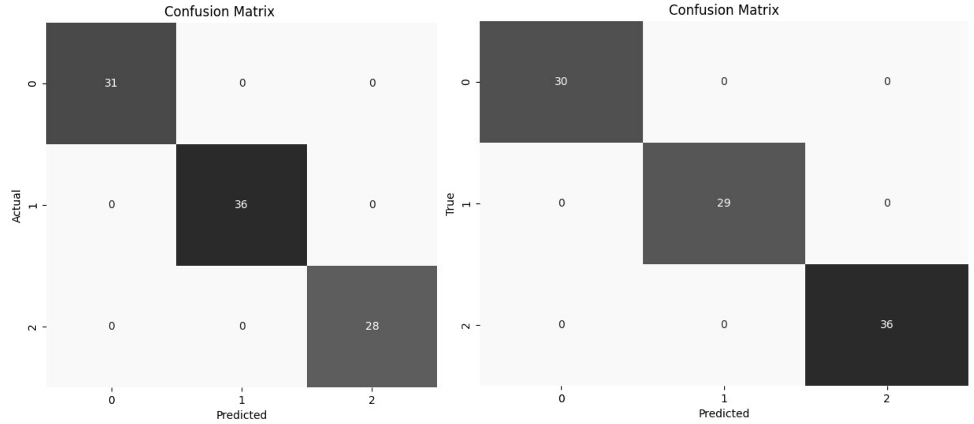


Fig. 35. CNN vs RF Confusion Matrix of the Frontal Elevation of the Arm Exercise

4.4.2.2. Classification of Angle Data for CNN

For the angle-based classification, CNN achieved a test accuracy of 90.17% with an execution time of 29.89 seconds (Fig. 36). The confusion matrix revealed some misclassifications: Class 0 had 27 false positives and 25 false negatives, Class 1 had 90 false positives and 34 false negatives, and Class 2 had 38 false positives and 45 false negatives. CNN training parameters included 41000 iterations over 25 epochs, utilizing a batch size of 6 and a 20% validation split. This demonstrates that while CNNs can effectively classify angles with high accuracy, they incur higher computational cost and some degree of error when dealing with angle-level data.

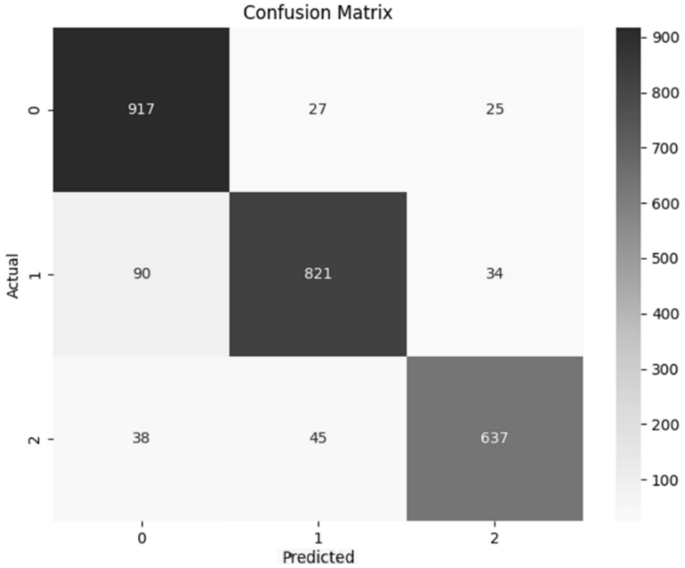


Fig. 36. Confusion Matrix of the Frontal Elevation of the Arm Exercise

Table 12. Metrics of Angle Classification for the Frontal Elevation of the Arm Exercise

CNN Classifier	Value
Number of iterations	41000
Number of Epochs	25
Batch size	6
Validation split	20%
Execution time	29.89s
Test accuracy	90.17%

4.4.3. Sideways Elevation of the Arm Exercise

During this exercise, the following nodes are analysed: 4, 5, 6, 7, 9. The evaluation of the exercise performance considers the following criteria:

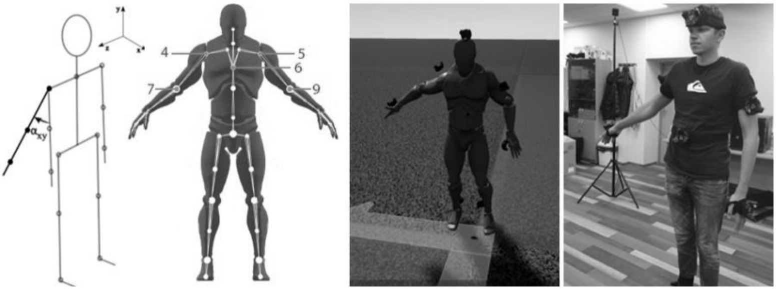


Fig. 37. From left to right: Sideways Elevation of the Arm Exercise Movement Angle, analysed nodes, 3D Mannequin in the system following the Participant’s movements, participant

- **Angle 1:** The arm must remain fully extended, meaning no bending at the elbow. The elbow flexion angle measured at nodes 7 or 9 should be approximately 0 degrees.
- **Angle 2:** When raising the arm upward, it must be extended to the side. The angle between the frontal vector and the extending arm, measured at nodes 4 or 5, should be approximately 90 degrees.
- **Angle 3:** The participant must maintain an upright posture throughout the exercise. The spinal tilt angle measured at node 6 should be close to 90 degrees relative to the horizontal plane.
- **Angle 4:** The shoulder line should remain horizontal during the exercise. The vertical angle measured at nodes 4 or 5 should be approximately 0 degrees.

In the experiment, data were collected from 21 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 105 individual datasets have been formed for each class.

- **Class 0 (Correct Movement)**: Executed the arm elevation correctly, maintaining proper form and alignment. This group is represented in blue on Fig. 38 and in green on Fig. 39.
- **Class 1 (Incorrect Movement 1)**: Executed the movement incorrectly by compensating with the torso, deviating from the correct technique. This group is represented in green on Fig. 38 and in red on Fig. 39.
- **Class 2 (Incorrect Movement 2)**: Executed the movement incorrectly by compensating with the shoulder, further straying from the proper form. This group is represented in yellow on Fig. 38 and in blue on Fig. 39.

After analysing all four angles and comparing the exercises performed by healthy individuals with those performed incorrectly, it was observed that, for Angle 1, depending on the participant's physiology, the arm could remain straight or be slightly bent (when performed incorrectly). However, the average values did not always show significant differences. From the data for participants marked as 16 or 19 in Fig. 38, no significant differences could be identified. Similar results are observed in Fig. 39.

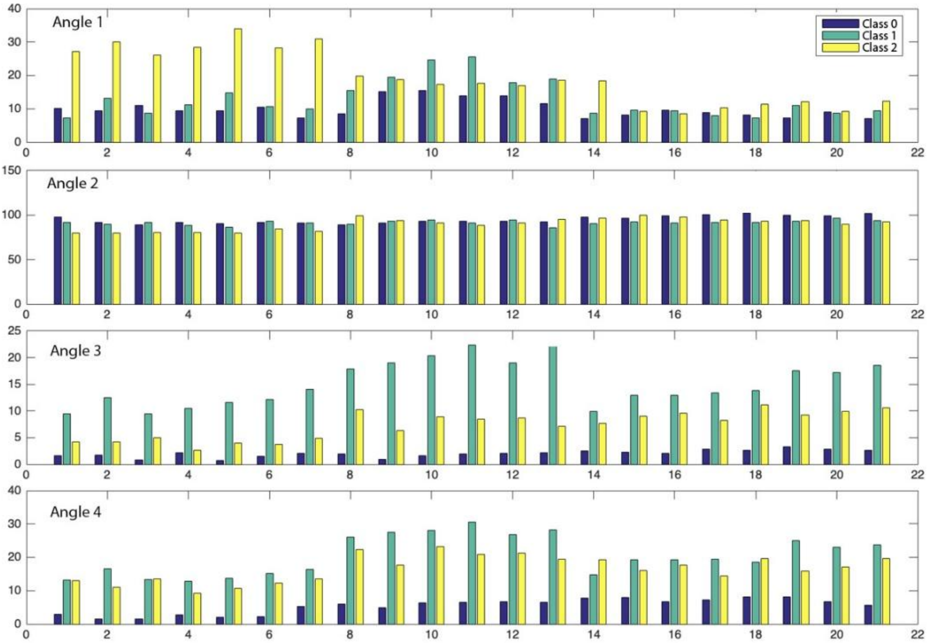


Fig. 38. Sideways Elevation of the Arm angle mean values

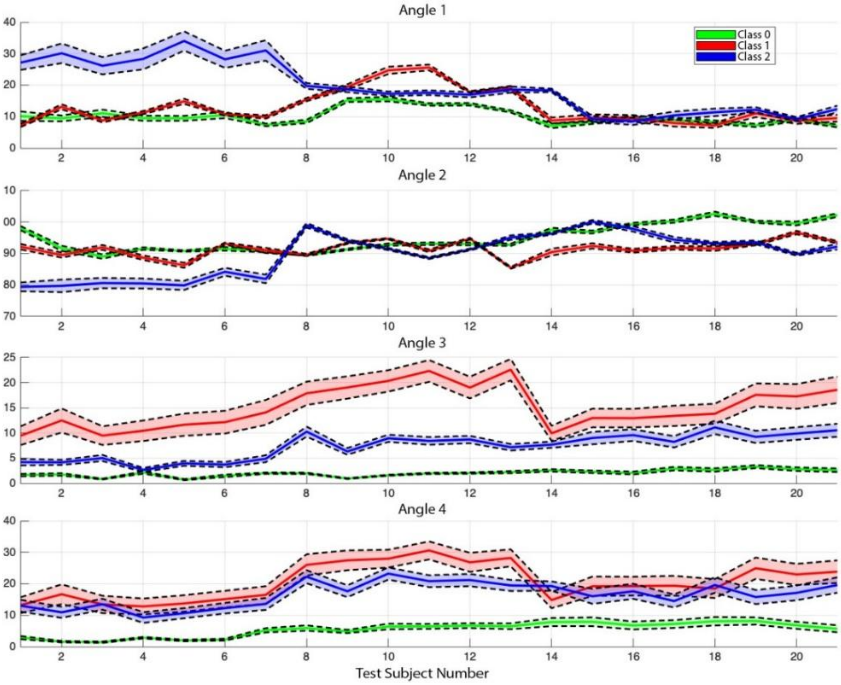


Fig. 39. Confidence Intervals of Participants for the Sideways Elevation of the Arm Exercise

Class 0 consistently exhibits mean values near the targets (Angle 1: ~0°, Angle 2: ~90°, Angle 3: ~90°, Angle 4: ~0°) with minimal variability, reflecting correct execution. Class 1 (torso compensation) shows notable deviations in all angles, particularly in Angle 3, where mean values range from ~10° to ~20°, highlighting significant posture misalignment. Class 2 (shoulder compensation) exhibits the most important deviations and variability, particularly in Angles 1 (up to ~30°) and 4 (up to ~30°), highlighting improper shoulder alignment and excessive elbow bending. Confidence intervals in Fig. 39 reinforce these findings, showing that Angles 3 and 4 are the most sensitive for detecting compensatory movements, as they exhibit the most significant deviations and variability across Classes 1 and 2.

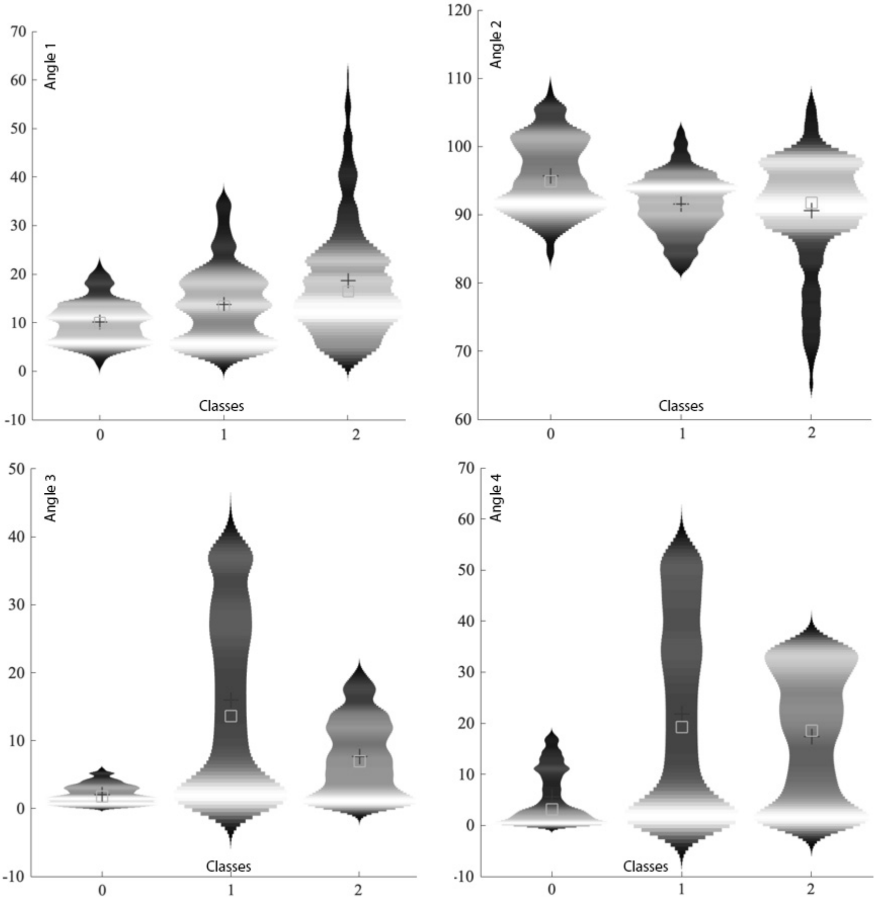


Fig. 40. Bean Plots of Evaluated Angles for the Sideways Elevation of the Arm Exercise

Fig. 40 demonstrates that Class 0 achieves the desired target values across all angles, with minimal variability, reflecting correct execution of the exercise. Class 1 (torso compensation) shows moderate deviations, particularly in Angles 3 and 4, where posture misalignments are evident, with mean values ranging from ~10° to

~30°. Class 2 (shoulder compensation) exhibits the greatest deviations and variability, particularly in Angles 1 (Elbow Flexion, up to ~40°) and 4 (Shoulder Line Alignment, up to ~40°), highlighting improper shoulder alignment and significant compensatory behaviours. These findings underscore Angles 3 (Spinal Tilt) and 4 (Shoulder Line Alignment) as the most sensitive indicators of compensatory movements, as they exhibit the greatest errors and variability in Classes 1 and 2.

The histograms in Fig. 41 demonstrate that Class 0 consistently achieves target values across all angles, with narrow distributions and minimal variability, reflecting correct execution. Class 1 (torso compensation) exhibits moderate deviations, particularly in Angle 3 (Spinal Tilt), where values peak between ~20°–30°, and in Angle 4 (Shoulder Line Alignment), where values extend up to ~40°, indicating significant posture misalignments. Class 2 (shoulder compensation) exhibits the most important deviations and variability, particularly in Angle 1 (Elbow Flexion, up to ~60°) and Angle 4 (Shoulder Line Alignment, up to ~40°), highlighting severe compensatory behaviours. These findings highlight Angles 3 and 4 as the most sensitive metrics for detecting compensatory movements, given their pronounced deviations and broader distributions in Classes 1 and 2.

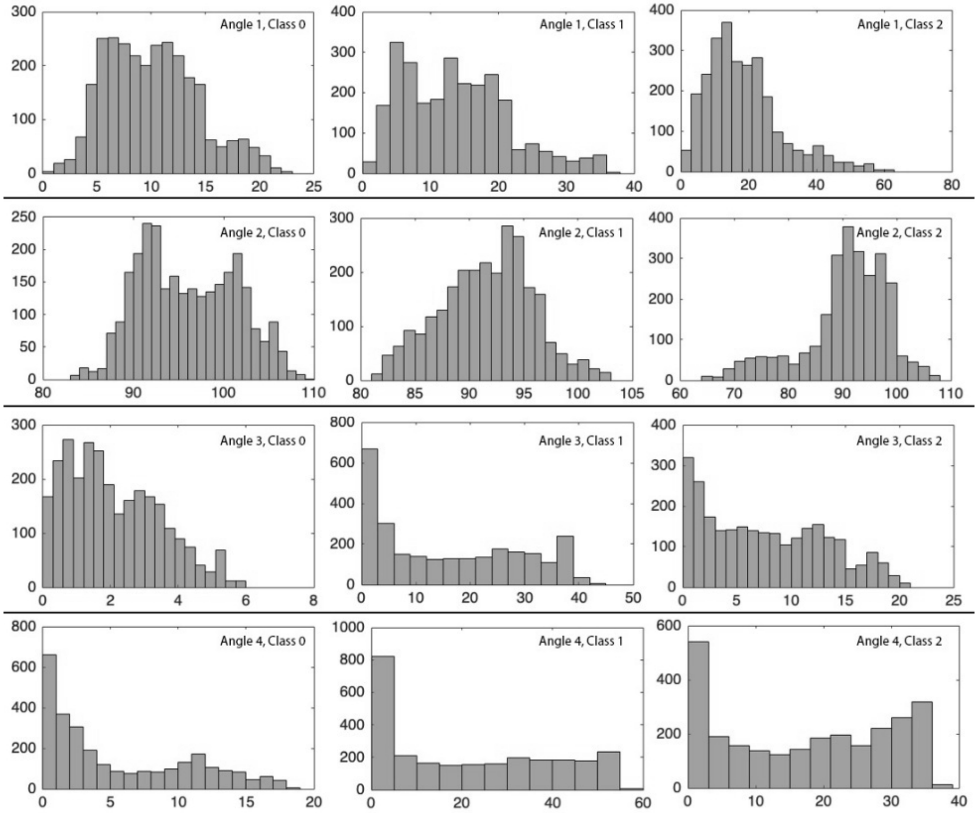


Fig. 41. Histograms of Evaluated Angles for the Sideways Elevation of the Arm Exercise

In this case, the RF model was built using all three statistical measures of Angle 3 and Angle 4, as they exceeded the selected threshold (see Fig. 42).

4.4.3.1. Sideways Elevation of the Arm Exercise Decision Classification

The RF classifier identified Angles 3 and 4 as the most significant features for distinguishing between exercise classes in the Sideways Elevation of the Arm Exercise, as demonstrated by the feature importance analysis (Fig. 42). The cumulative feature contribution confirmed that these angles provided the most meaningful insights for classification. Both RF and CNN achieved 100% classification accuracy for patient-based disorder detection for this exercise (Fig. 43). The confusion matrices indicate that RF correctly classified all test samples across the three exercise classes (0, 1, 2), with no misclassifications. Similarly, the CNN classifier achieved perfect accuracy with no misclassifications. The RF classifier demonstrated exceptional computational efficiency, completing the task in 0.032 seconds with only 10 trees and a maximum depth of 3. On the other hand, CNN required 11.57 seconds to process 9416 iterations over 11 epochs, leveraging a batch size of 6 and a validation split of 20%.

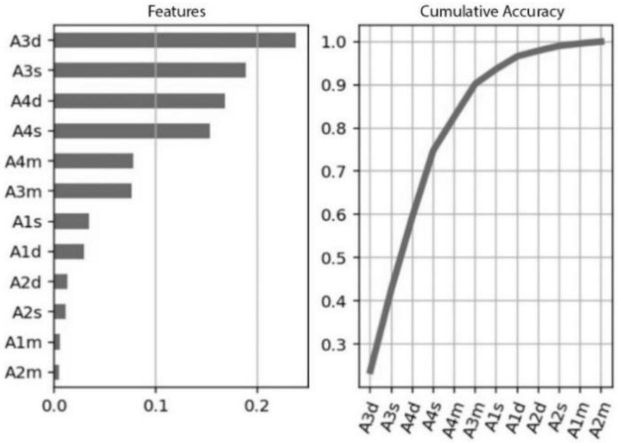


Fig. 42. Feature Importance Analysis of the Sideways Elevation of the Arm Exercise.

Table 13. Metrics of the Decision Classification for the Sideways Elevation of the Arm Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	10	Number of iterations	9416
Maximum depth	3	Number of Epochs	11
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.032 s	11.57s	
Test Accuracy	100%	100%	

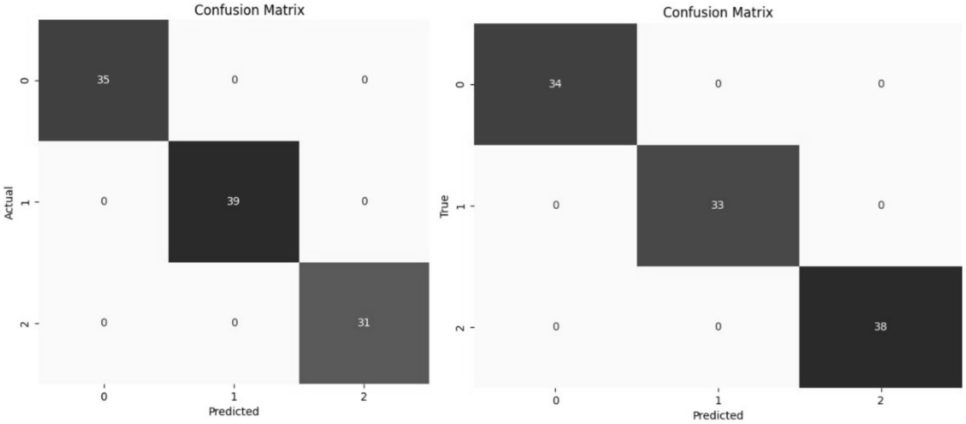


Fig. 43. CNN vs RF Confusion Matrix Comparison of Sideways Elevation of the Arm Exercise.

4.4.3.2. Sideways Elevation of the Arm Exercise Angle Classification

In angle-based classification using CNN for the Sideways Elevation of the Arm Exercise, the classifier achieved a test accuracy of 92.20%. The confusion matrix (Fig. 44) shows that while most angles were correctly classified, minor misclassifications occurred between the three classes. Specifically, classes 0 and 2 had a few samples misclassified, whereas class 1 showed better distinction. The CNN processed the classification task efficiently in 16.62 seconds, executing 36200 iterations over 15 epochs. It used a batch size of 6 and a 20% validation split, demonstrating its ability to accurately distinguish subtle variations in angle data within a reasonable computational time.

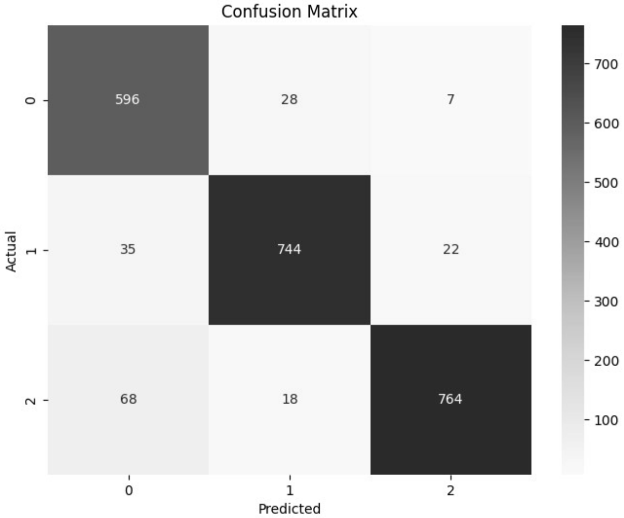


Fig. 44. Confusion Matrix of angles classified on CNN for the Sideways Elevation of the Arm

Table 14. Metrics of Angle Classification for the Sideways elevation of the Arm Exercise

CNN Classifier	Value
Number of iterations	36200
Number of Epochs	15
Batch size	6
Validation split	20%
Execution time	16.62s
Test accuracy	92.20%

4.4.4. Horizontal Abduction/Adduction of the Arm Exercise

This exercise involves moving the arm horizontally, away from or toward the body's centreline. It is performed in the transverse plane about the vertical axis, focusing on shoulder mobility and stability during the movement. During this exercise, the nodes n4, n5, n7, n9, n22, n23, and n6 are analysed**.** The evaluation of the exercise performance is based on the following criteria:

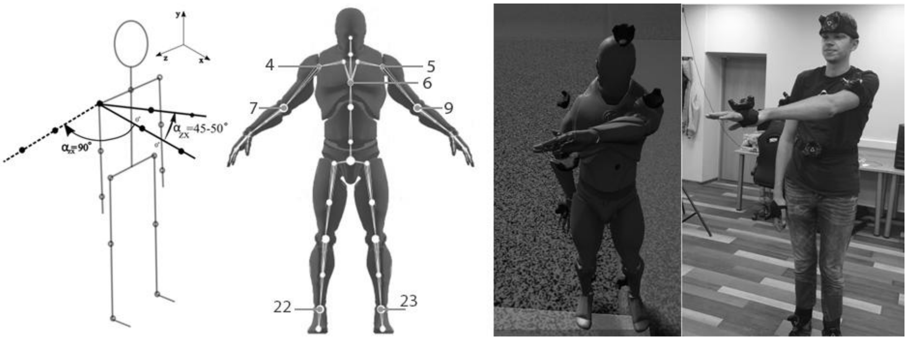


Fig. 45. From left to right: Horizontal Abduction/Adduction of the Arm exercise movement angle, analysed nodes, 3D Mannequin in the system following the Participant's Movements, participant

- **Angle 1:** The elbow flexion angle, measured at nodes n7 or n9, must be approximately 0 degrees to ensure the arm remains fully extended without bending.
- **Angle 2:** The vertical angle, measured at nodes n4 or n5, should transition from 180 degrees to 90 degrees, with values concentrating around 90 degrees during the forward and upward extension of the arm.
- **Angle 3:** The angle of the shoulder line, assessed using nodes n4 and n5, must remain close to 0 degrees to ensure the shoulder line stays horizontal throughout the exercise.
- **Angle 4:** The angle between the vectors formed by n4, n5 (shoulders), and n22, n23 (feet) must stay close to 0 degrees, ensuring no torso rotation during the movement.

- **Angle 5:** The spinal tilt angle, measured at node n6, must remain close to 90 degrees relative to the horizontal plane to maintain an upright posture.

In the experiment, data were collected from 17 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 85 individual datasets have been formed for each class.

- **Class 0 (Correct Movement)**: The exercise is performed correctly, with the arm fully extended, proper posture maintained, and no compensatory movements observed. For Angle 1, the arm remains straight, and the confidence interval bands (green in Fig. 47) show minimal variation and align closely with the ideal values.
- **Class 1 (Incorrect Movement)**: The participant performs the exercise incorrectly, either by compensating with the torso, shoulder, or posture. Significant deviations are observed in Angle 3 (spinal tilt), Angle 4 (shoulder alignment with the feet), and Angle 5 (spinal posture). This group is represented in yellow on Fig. 46 and in red on Fig. 47.

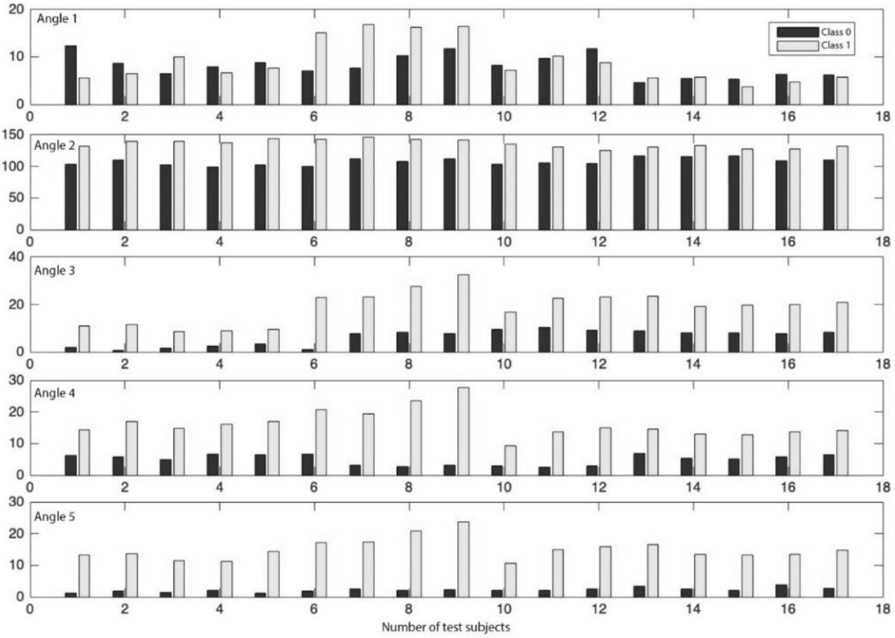


Fig. 46. Angle mean values of participants for the Horizontal Abduction of the Arm Exercise

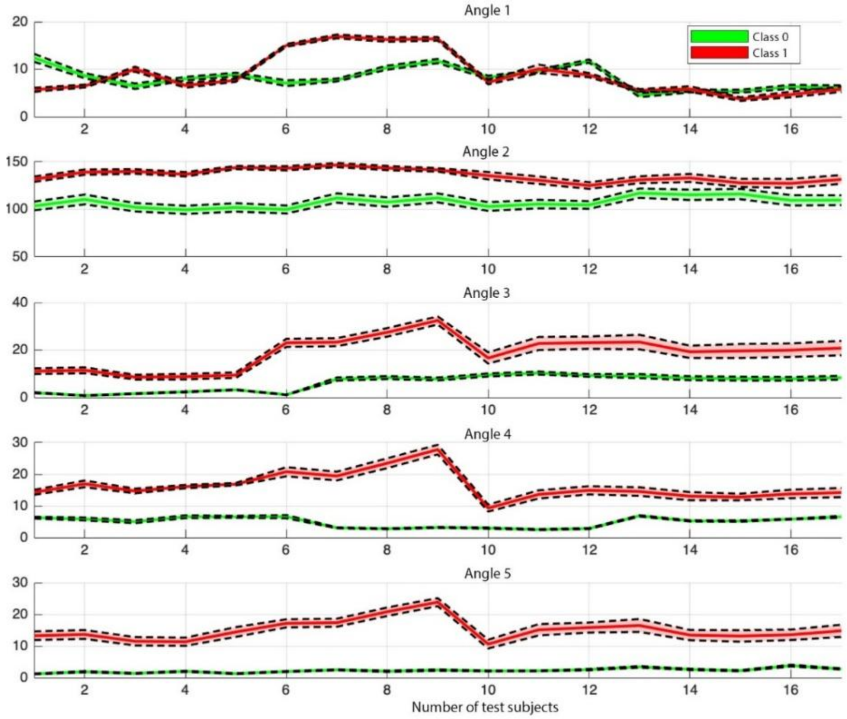


Fig. 47. Confidence Intervals of Participants for the Horizontal Abduction of the Arm

The analysis on Fig. 46 and Fig. 47 reveals distinct differences between Class 0 (correct movements) and Class 1 (compensatory movements). Class 0 consistently achieves target values across all angles, with minimal variability and tight confidence intervals, reflecting proper execution of the exercise. Class 1 exhibits significant deviations, particularly in Angles 3 (Shoulder Line Alignment), 4 (Torso Rotation), and 5 (Spinal Tilt), where mean values deviate by up to ~30° and show greater variability, indicating compensatory behaviours involving the torso and shoulders. These findings highlight the importance of monitoring Angles 3, 4, and 5 as sensitive indicators of compensatory movements during the exercise. The confidence interval patterns in Fig. 47. To further reinforce these distinctions, we highlight consistent alignment in Class 0 and significant variability in Class 1.

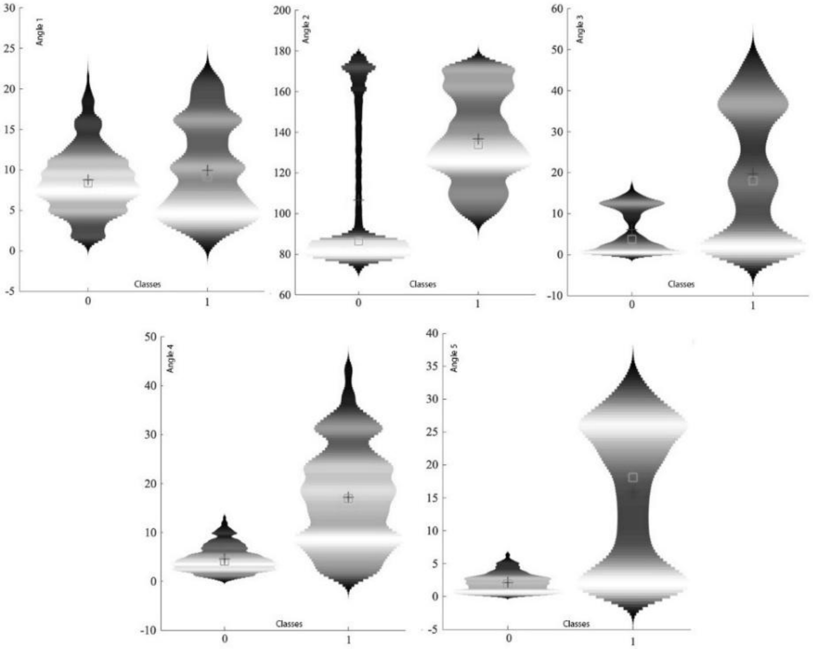


Fig. 48. Bean Plots of Evaluated Angles for the Horizontal Abduction of the Arm Exercise

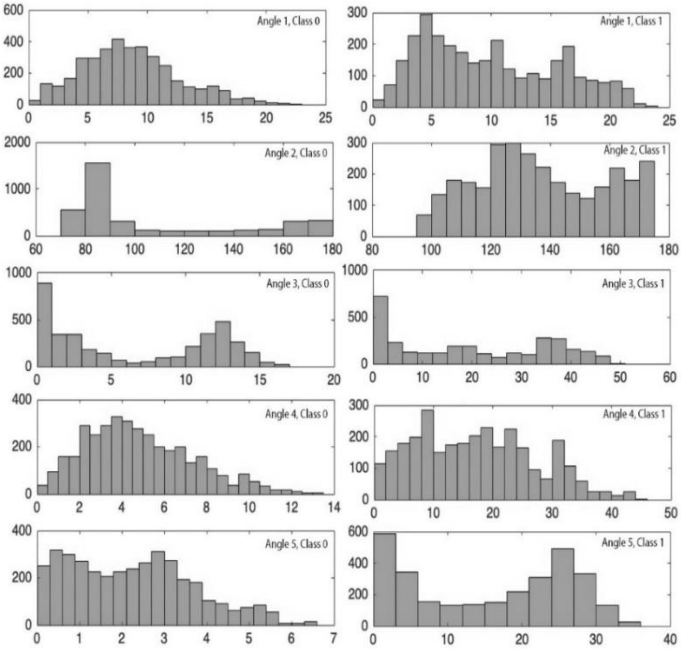


Fig. 49. Histograms of Evaluated Angles for the Horizontal Abduction of the Arm Exercise

The bean plots for the Horizontal Abduction/Adduction of the Arm Exercise (Fig. 48) reveal distinct differences in movement quality between Class 0 (Correct Movements) and Class 1 (Compensatory Movements) across all five evaluated angles. Class 0 consistently exhibits tight distributions with minimal variability, achieving mean values close to target ranges: ~10° for Angle 1, ~80° for Angle 2, ~5°–10° for Angle 3, ~5° for Angle 4, and ~5° for Angle 5. These results reflect consistent, proper execution of the exercise with minimal deviations. In contrast, Class 1 exhibits broader distributions and significant deviations across all angles, with mean values of ~15° for Angle 1, 140° for Angle 2, 20° for Angle 3, 20° for Angle 4, and 20° for Angle 5. These deviations highlight compensatory behaviours, such as moderate elbow flexion (Angle 1), improper arm elevation (Angle 2), poor shoulder alignment (Angle 3), noticeable torso rotation (Angle 4), and significant spinal tilt (Angle 5). Variability is particularly pronounced in Class 1 for Angles 3, 4, and 5, with values reaching ~40°, highlighting inconsistent posture and alignment during compensatory movements. Overall, the data underscores that Class 0 achieves correct movement patterns with high consistency. In contrast, Class 1 exhibits significant compensatory deviations, making Angles 3 (Shoulder Alignment), 4 (Torso Rotation), and 5 (Spinal Tilt) the most sensitive indicators for detecting and distinguishing compensatory movements. The histograms in Fig. 49 further emphasize the differences in movement quality between Class 0 (Correct Movements) and Class 1 (Compensatory Movements) for the Horizontal Abduction/Adduction of the Arm Exercise, showcasing the distribution of angles across participants for each class. Class 0 consistently demonstrates tightly centred distributions across all angles, reflecting proper execution of the exercise with minimal deviations. For Angle 1, most values fall between ~5° and 15°, with a clear peak around 10°, indicating consistent arm extension with minimal elbow flexion. In Angle 2, the majority of values concentrate tightly around ~80°, with very few deviations, indicating accurate arm elevation. Angle 3 (Shoulder Line Alignment) shows values clustered around ~5°, highlighting proper shoulder alignment. Similarly, for Angle 4 (Torso Rotation), the distribution peaks around ~5°, indicating minimal torso movement. Angle 5 (Spinal Tilt) also exhibits a narrow distribution around ~5°, reflecting correct upright posture. In contrast, Class 1 exhibits broader distributions and significant deviations across all angles. For Angle 1, the values are more widely distributed, peaking around 10°–15°, with some extending beyond 20°, reflecting moderate elbow flexion. In Angle 2, values are distributed broadly from ~80° to 160°, indicating inconsistent arm elevation. For Angle 3, values extend up to ~50°, with a peak around ~20°–30°, reflecting poor shoulder alignment. Angle 4 exhibits substantial deviations, with values up to ~40° and a peak in the 20°–30° range, indicating significant torso rotation. Finally, Angle 5 shows wide variability, with values up to ~40° and a peak around 20°–30°, indicating significant spinal tilt due to compensatory behaviours. These histograms reinforce that Class 0 maintains precise and correct execution across all angles with minimal variability, while Class 1 exhibits greater variability and substantial compensatory deviations, particularly in Angles 3, 4, and 5, where deviations are most pronounced. This highlights the sensitivity of these angles in identifying compensatory movements during the exercise.

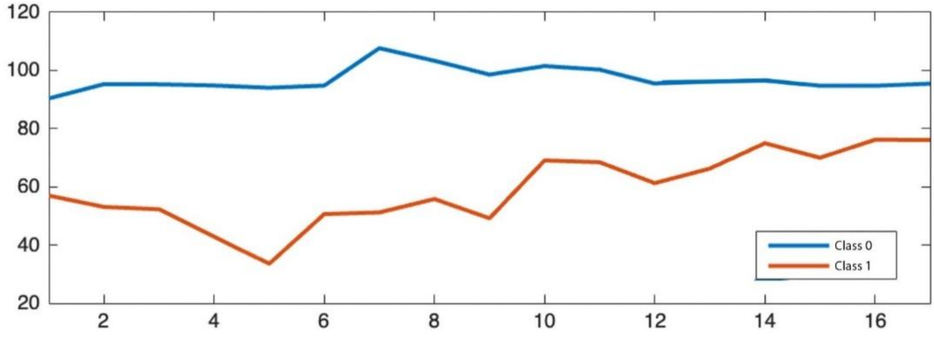


Fig. 50. Max/Min Range of Angle 2

Fig. 50 illustrates the variation in the range of values (maximum minus minimum) for Angle 2 during the Horizontal Abduction/Adduction of the Arm Exercise across test subjects, comparing Class 0 and Class 1. For Class 0, the distance remains consistently higher, averaging around ~90° to 100° throughout all test subjects. This indicates that participants in Class 0 consistently achieve the full range of arm motion required for the exercise, with minimal deviations from the ideal vertical arm positioning. The steady trend highlights a uniform execution pattern across all participants, with correct movements being performed. In contrast, Class 1 shows a significantly narrower range, averaging around 50° to 60°, with noticeable fluctuations. This reduced distance reflects restricted or inconsistent arm elevation during compensatory movements. The trend's variability further underscores inconsistent execution, with some participants deviating more than others.

4.4.4.1.Horizontal Abduction/Adduction of the Arm Exercise Decision Classification

For the Horizontal Abduction/Adduction of the Arm Exercise Decision Classification, the RF classifier identified Angles A2m and A2s as the most significant features, as illustrated by the feature importance chart (Fig. 51). The cumulative accuracy curve further supports this observation, showing a sharp rise in accuracy contributed by these angles. Both RF and CNN achieved 100% classification accuracy for patient-based exercise decision classification, as shown in the confusion matrices (Fig. 52).

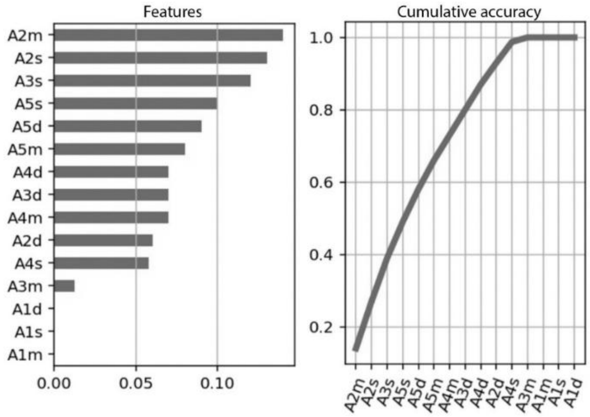


Fig. 51. Feature Importance in the RF Model for the Horizontal Abduction of the Arm

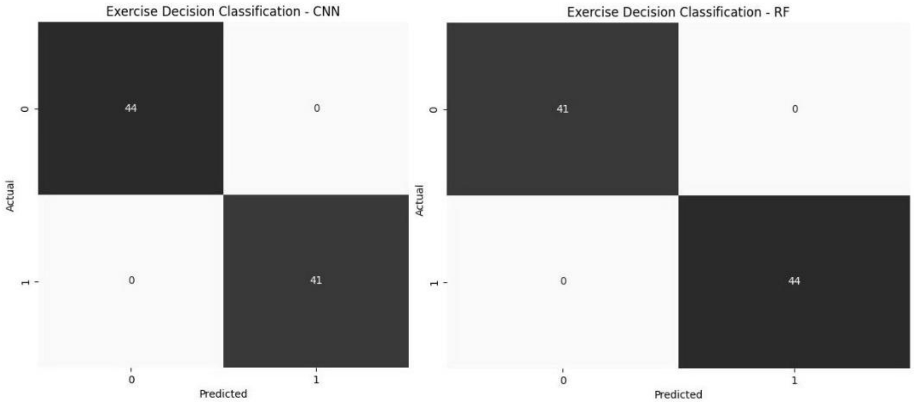


Fig. 52. Confusion Matrix of the Horizontal Abduction of the Arm

The RF classifier completed its execution in 0.03 seconds, requiring only 10 trees with a maximum depth of 3. This highlights its computational efficiency and its ability to achieve accurate classifications. The CNN classifier, on the other hand, required 9416 iterations over 11 epochs and took 11.57 seconds to run. Despite the computationally intensive training process, CNN also achieved perfect classification. Both models demonstrated robustness and reliable classification of exercise decisions for this task.

Table 15. Metrics of the Decision Classification for the Horizontal Abduction/Adduction of the Arm Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	10	Number of iterations	9416
Maximum depth	3	Number of Epochs	11
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		

Total execution time	0.03 s	11.57s
Test Accuracy	100%	100%

4.4.4.2. Horizontal Abduction/Adduction of the Arm Exercise Angle Classification

For angle-based classification in the Horizontal Abduction/Adduction of the Arm Exercise (Fig. 53), CNN achieved a test accuracy of 97.77%. The confusion matrix revealed minor misclassifications, with most errors occurring between classes 0 and 1. CNN processed this task efficiently within 11.39 seconds, executing 31050 iterations over 12 epochs. These results underscore CNN's capability to handle detailed angle data while maintaining high accuracy.

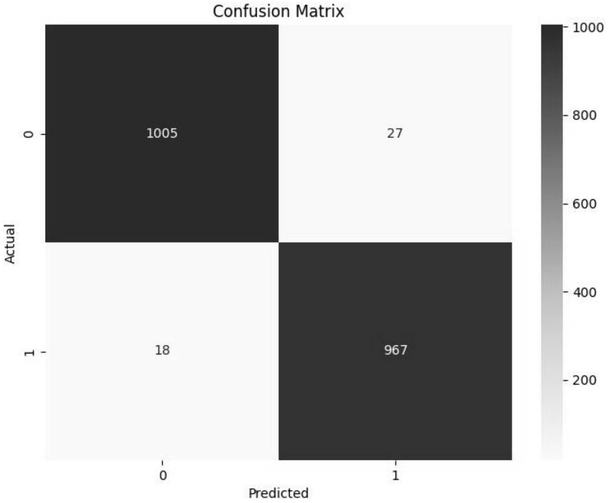


Fig. 53. Angle Classification for the Horizontal Abduction/Adduction of the Arm Exercise

Table 16. Metrics of Angle Classification for the Overhead Reach of the Arm Exercise

CNN Classifier	Value
Number of iterations	31050
Number of Epochs	12
Batch size	6
Validation split	20%
Execution time	11.39s
Test accuracy	97.77%

4.4.5. Overhead Reach of the Arm

During this exercise, the nodes n4, n5, n22, n23, n0, and n6 are analysed. The performance of the exercise is evaluated based on the following conditions:

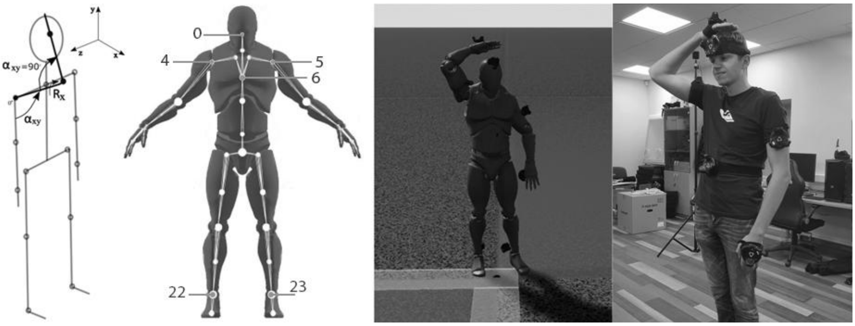


Fig. 54. From left to right: Overhead Reach of the Arm exercise movement angle, analysed nodes, 3D Mannequin in the system following the participant's movements, participant

- **Angle 1**: The participant's head must remain stationary. The vertical tilt angle of the head at node 0 should be approximately 0 degrees.
- **Angle 2**: As the arm is lifted upward, it bends at the elbow. The elbow flexion angle, measured at nodes 4 or 5, ranges from 0 to 90 degrees.
- **Angle 3**: When the arm is raised, the shoulder angle changes from the initial position (180 degrees) to the final position (~90 degrees). During correct execution, the highest concentration of values at nodes 4 or 5 should occur near 90 and 180 degrees.
- **Angle 4**: The participant must avoid rotating the torso sideways. The shoulder line angle, calculated between nodes 4 and 5 and nodes 22 and 23, should remain close to 0 degrees.
- **Angle 5**: The participant must maintain an upright posture throughout the exercise. The spinal tilt angle, measured at node 6, should be approximately 90 degrees relative to the horizontal plane.

In the experiment, data were collected from 19 healthy participants, who performed each classified exercise 5 times to have larger data set. In total 95 individual data set for each class have been formed.

- **Class 0 (Healthy Movement):** For Angle 1, clear differences are noticeable. As shown in Fig. 55, the average values for healthy individuals are significantly closer to 0 degrees compared to those with movement impairments performing the exercise incorrectly. Presented in blue on Fig. 55 and presented in green on Fig. 56.
- **Class 1 (Incorrect Movement 1):** For Angles 2 and 3, the overlapping confidence interval bands for healthy (green), compensatory (red), and incorrectly performed movements (blue) in Fig. 56 make it challenging to distinguish between healthy and incorrect movements. These angles are

variable during the movement (elbow flexion and arm elevation), which adds complexity. Presented in green on Fig. 55 and presented in red on Fig. 56.

- **Class 2 (Incorrect Movement 2):** Similar challenges are seen in Class 2 movements due to the variability of Angles 2 and 3, where uneven changes during movement do not necessarily indicate poor motor function. Presented in yellow on Fig. 55 and presented in blue on Fig. 56.

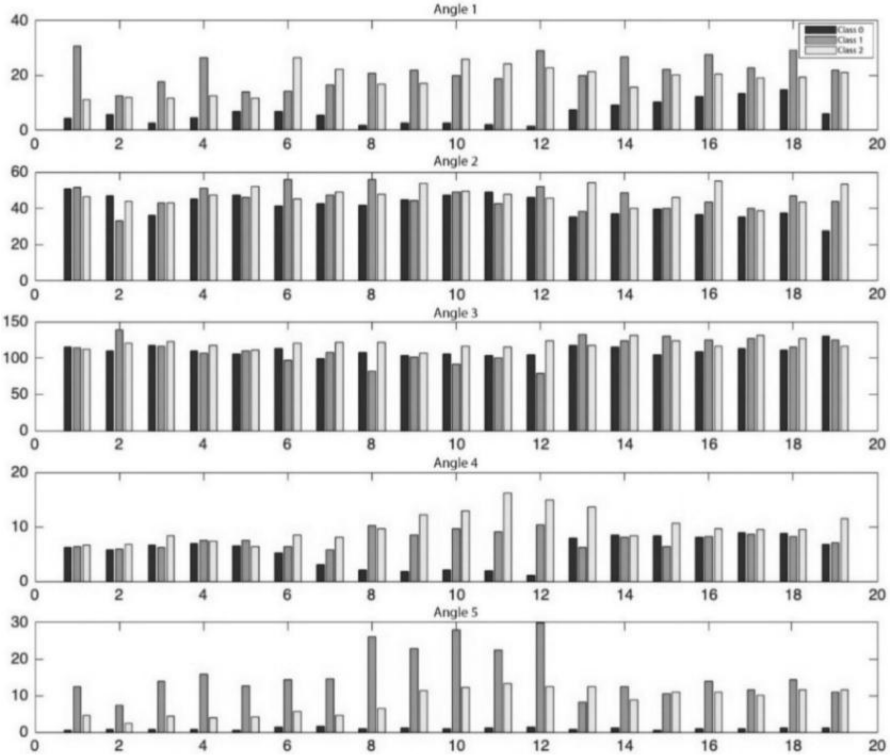


Fig. 55. Angle mean values of participants for the Overhead Reach of the Arm Exercise

The Overhead Reach of the Arm Exercise was analysed in Fig. 55 and Fig. 56 demonstrates clear distinctions in movement quality across evaluated angles and classes, highlighting the differences between Class 0 (Healthy Movements), Class 1 (Compensatory Movements), and Class 2 (Additional Errors). Class 0 consistently achieves mean values near target ranges with minimal variability across all angles: ~0° for Angle 1 (Head Stability), ~0°–90° for Angle 2 (Elbow Flexion), ~90°–180° for Angle 3 (Shoulder Angle), ~0° for Angle 4 (Torso Rotation), and ~90° for Angle 5 (Spinal Tilt). In contrast, Class 1 and Class 2 exhibit broader distributions and significant deviations, particularly in Angles 2 and 3, where overlapping confidence intervals and variability complicate the differentiation of compensatory behaviours.

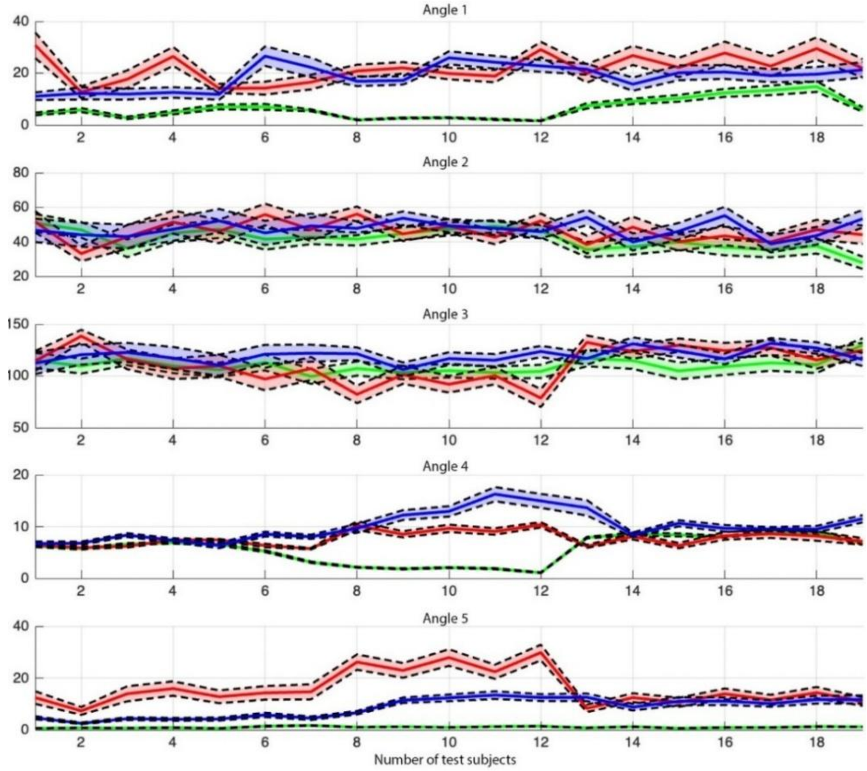


Fig. 56. Confidence intervals of participants for the Overhead Reach of the Arm Exercise

Class 1 shows mean values extending to ~10°–20° for Angle 1, ~40°–140° for Angle 2, ~150°–180° for Angle 3, ~10°–20° for Angle 4, and ~10°–30° for Angle 5, reflecting head tilt instability, inconsistent elbow flexion, improper arm elevation, and torso rotation errors. Class 2 demonstrates even greater deviations, with mean values near ~120°–140° for Angle 2 and broader variability across all angles. Angles 4 (Torso Rotation) and 5 (Spinal Tilt) are the most sensitive indicators of compensatory errors, with pronounced deviations in Class 1 and Class 2, indicating significant torso instability and spinal misalignment. These findings underscore Class 0's ability to consistently perform correct movements with high accuracy, while highlighting the challenges of analysing Angles 2 and 3 due to their variability and overlapping behaviours in incorrect movement patterns. Overall, Angles 4 and 5 are crucial for distinguishing healthy movements from compensatory behaviours, as they provide the most reliable indicators of movement quality during this exercise.

The bean plots for the Overhead Reach of the Arm Exercise (Fig. 57) reveal both strengths and limitations in distinguishing movement quality across the three classes: Class 0 (Healthy Movements), Class 1 (Compensatory Movements), and Class 2 (Additional Errors). Class 0 generally performs better, with distributions that are tighter and closer to ideal target values, particularly for Angles 1 (Head Stability) and

5 (Spinal Tilt). For Angle 1, Class 0 achieves mean values around 10°, slightly deviating from the ideal 0° but remaining significantly lower than Classes 1 and 2, which show means of ~20° and 30°, respectively, and exhibit much broader variability. Angle 5 further highlights Class 0's consistent posture, with a mean of ~5° and minimal variability, compared to ~50° in Class 1 and ~30° in Class 2, both of which show pronounced deviations indicating improper spinal alignment. Angles 2 (Elbow Flexion) and 3 (Shoulder Angle), however, expose challenges for all classes. Class 0 struggles to reach target values, with mean values of ~40° for Angle 2 and 100°–120° for Angle 3, deviating from the expected ~90°. Classes 1 and 2 display even greater inconsistencies, with means of ~60° and ~80° for Angle 2, and ~120° and ~140° for Angle 3, respectively, and significant overlap between the two incorrect movement classes. These overlaps reduce the sensitivity of these angles for distinguishing compensatory errors. Angle 4 (Torso Rotation) provides moderate differentiation, with Class 0 maintaining mean values near ~5° compared to ~10° for Class 1 and ~20° for Class 2, but the broader distributions in Classes 1 and 2 highlight torso instability in compensatory movements. Overall, Class 0 consistently exhibits tighter distributions and mean values closer to targets, particularly for Angles 1 and 5, which serve as reliable indicators of correct movements and compensatory errors. However, the substantial overlap between Classes 1 and 2 in Angles 2 and 3, combined with Class 0's deviation from target values in these angles, underscores the need for further refinement to improve differentiation and accuracy in assessing elbow flexion and shoulder alignment. While Angles 4 and 5 show promise as sensitive measures for detecting compensatory behaviours, especially in Class 2, the data highlight areas where even healthy movements need improvement to better align with the ideal execution of the exercise.

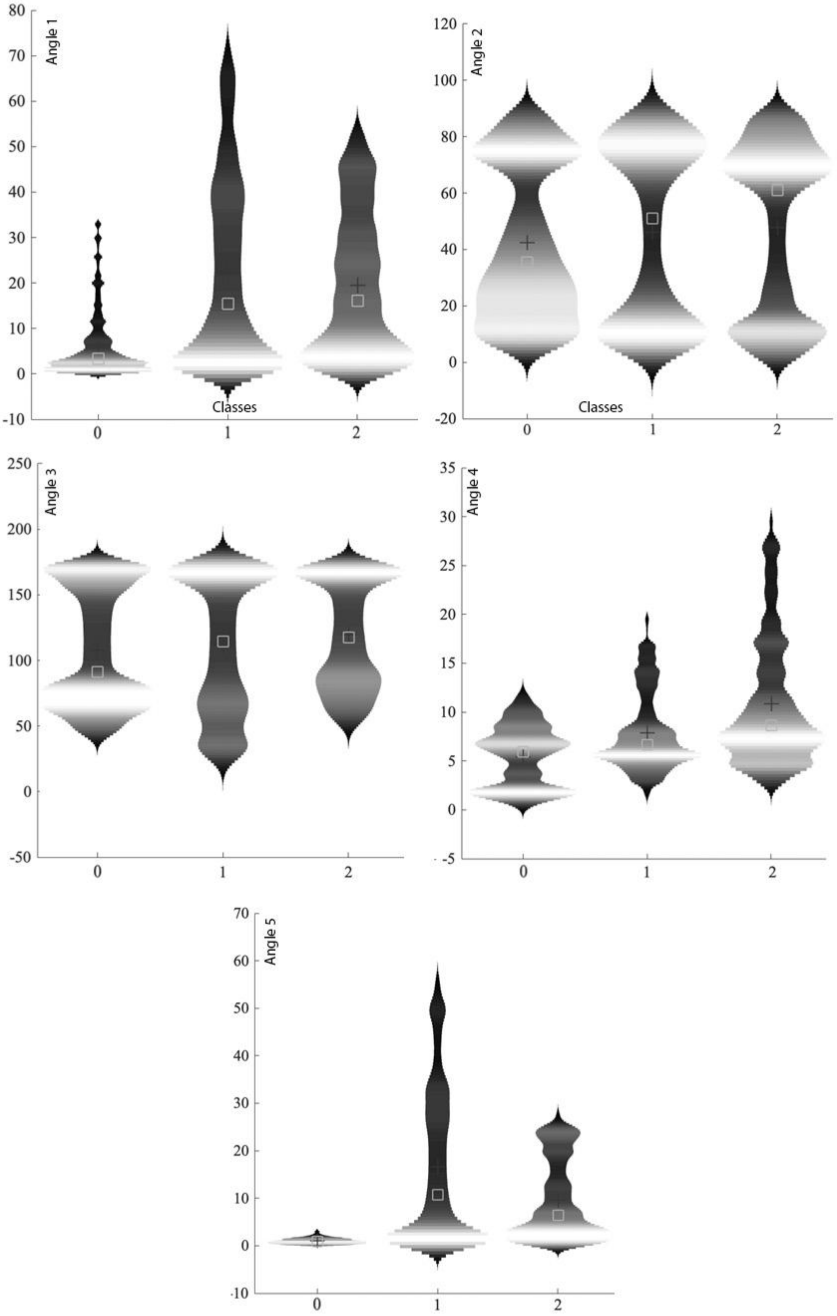


Fig. 57. Bean Plots of Evaluated Angles for the Overhead Reach of the Arm Exercise

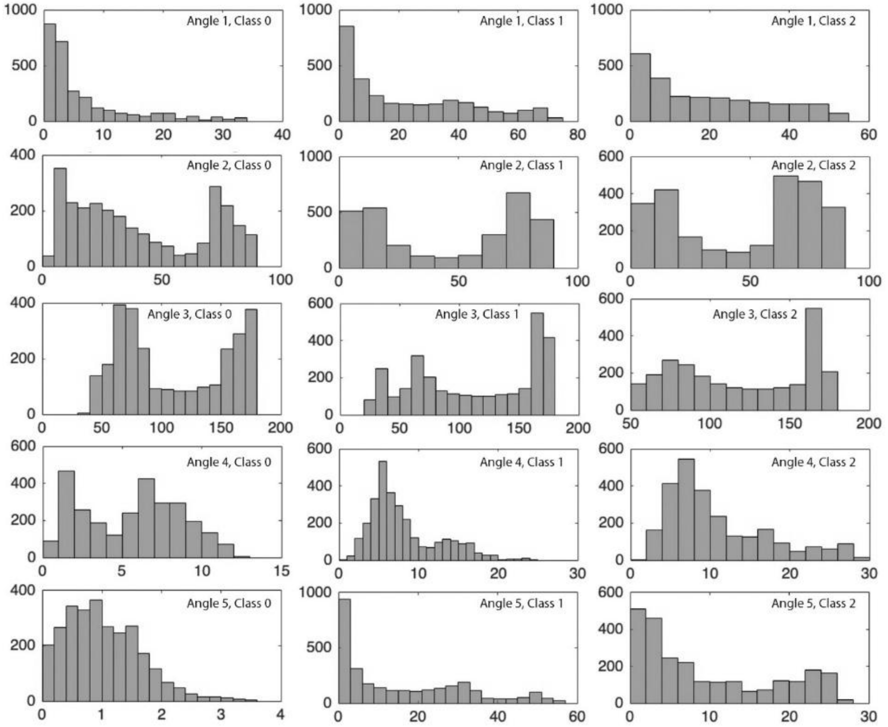


Fig. 58. Histograms of Evaluated Angles for the Overhead Reach of the Arm Exercise

The Overhead Reach of the Arm Exercise reveals significant differences in movement quality and deviations across Class 0 (Healthy Movements), Class 1 (Compensatory Movements), and Class 2 (Additional Errors), as evidenced by the histograms in Fig. 58. Class 0 consistently exhibits tight distributions and mean values within expected ranges, particularly for Angle 1 (Head Stability), where values cluster around 5°–10°, reflecting minimal head tilt. In contrast, Classes 1 and 2 exhibit broader distributions of this angle, with means ranging from ~20° to ~30°, indicating greater head instability during incorrect movements. Similarly, for Angle 5 (Spinal Tilt), Class 0 shows a concentrated distribution near ~5°, while Class 1 shows substantial deviations, with a mean around ~30° and values extending to ~60°. Class 2 performs slightly better than Class 1, peaking at ~20°, but remains outside the healthy range, highlighting poor spinal alignment in compensatory movements. Angles 4 (Torso Rotation) and 5 (Spinal Tilt) emerge as the most reliable indicators of movement quality, with Class 0 tightly clustered near ~5°, while Class 1 and Class 2 show progressive increases in variability and mean values, reaching up to ~30° in Class 2 for torso rotation. These findings indicate that compensatory movements are characterized by increasing torso and spinal deviations, which are particularly sensitive to errors. Angle 2 (Elbow Flexion) and Angle 3 (Shoulder Angle) show moderate differentiation; while Class 0 peaks near ~40°–50° for Angle 2 and ~120°–180° for Angle 3, the broader and overlapping distributions in Classes 1 and 2 complicate their utility in distinguishing incorrect movement types. However, they still provide insight into deviations from healthy patterns, with Class 1 and Class 2 peaking at ~60°–70° for Angle 2 and spreading between ~100°–200° for Angle 3.

Overall, the histograms indicate that Class 0 maintains tight distributions across all angles, demonstrating consistent and accurate execution of the exercise. In contrast, Classes 1 and 2 exhibit broader variability and significant deviations, particularly in Angles 4 (Torso Rotation) and 5 (Spinal Tilt), which are the most effective at distinguishing compensatory behaviours. While Angles 2 and 3 provide less separation between incorrect movements, they still reflect deviations from correct execution. These findings emphasize the importance of focusing on Angles 4 and 5 for accurate detection of compensatory behaviours, while acknowledging the challenges posed by variability and overlap in Angles 2 and 3. In this case, the RF model construction utilised all three statistical measures of Angle 3, as well as the ranges (A5d and A1d) and standard deviations (A5s and A1s) for Angles 5 and 1, since they exceeded the selected threshold (see Fig. 59).

4.4.5.1. Overhead Reach of the Arm Exercise Decision Classification

The RF classifier identified Angle 5 as the most critical feature for distinguishing the Overhead Reach of the Arm exercise, as highlighted by the feature importance chart (Fig. 59). The cumulative accuracy plot shows that Angle 5 alone significantly contributed to classification, followed by supporting features such as Angle 1 and Angle 3. RF achieved 95.78% accuracy, while the CNN achieved 100%. This is evident from the confusion matrices (Fig. 60), where the RF model showed minor misclassifications in class 2, whereas the CNN correctly classified all instances. Regarding computational efficiency, RF took only 0.06 seconds to process the data with 20 trees and a maximum depth of 3. In contrast, CNN required 14.63 seconds, with 11492 iterations across 13 epochs. The CNN utilised a batch size of 6 and a 20% validation split.

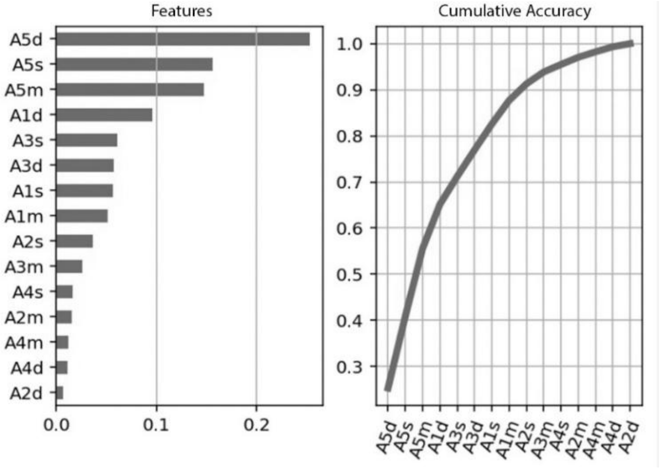


Fig. 59. Feature Importance for the Overhead Reach of the Arm Exercise

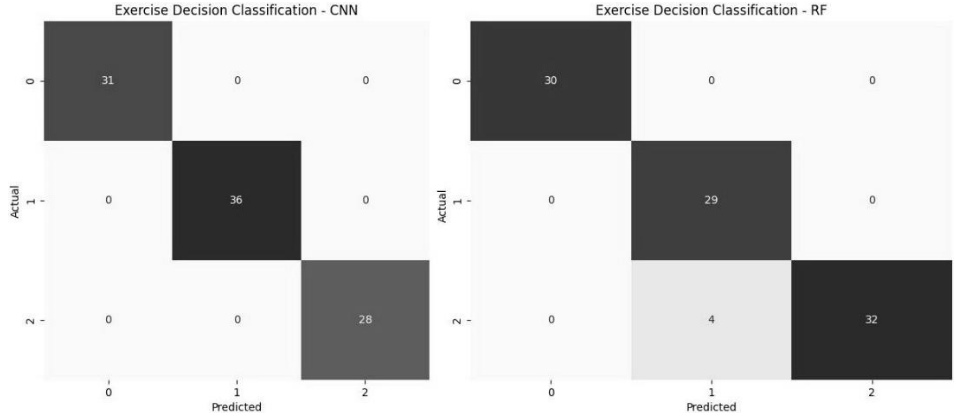


Fig. 60. CNN vs RF Confusion Matrix of the Overhead Reach of the Arm Exercise

Table 17. Metrics of the Decision Classification for the Overhead Reach of the Arm Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	20	Number of iterations	11492
Maximum depth	3	Number of Epochs	13
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.06 s	14.63s	
Test Accuracy	95.78 %	100%	

4.4.5.2. Overhead Reach of the Arm Exercise Angle Classification

For angle-based classification of the Overhead Reach of the Arm exercise, CNN achieved a test accuracy of 93.86%. The confusion matrix (Fig. 61) shows some minor misclassifications across the three classes, particularly in distinguishing between class 1 and class 2. CNN classification took 36.88 seconds, with 37100 iterations over 33 epochs. Similar to previous exercises, the CNN used a batch size of 6 and a 20% validation split, ensuring robust training and validation of the angle-based data.

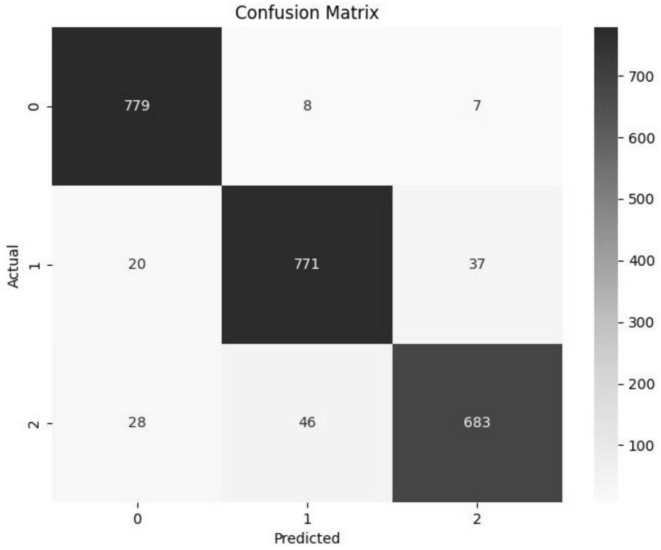


Fig. 61. Confusion Matrix of the Overhead Reach of the Arm Exercise

Table 18. Metrics of Angle Classification for the Overhead Reach of the Arm exercise

CNN Classifier	Value
Number of iterations	37100
Number of Epochs	33
Batch size	6
Validation split	20%
Execution time	36.88s
Test accuracy	93.86%

4.4.6. Lifting and Overhead Placement of a Heavy Object

This exercise involves lifting a heavy object from the ground and placing it overhead. It is performed in multiple planes, requiring coordinated movements of the arms, legs, and spine. The exercise focuses on upper- and lower-body strength, balance, and stability. During this exercise, the nodes n4, n5, n6, n7, n9, n18, and n19 are analysed. The performance of the exercise is evaluated based on the following conditions:

- **Angle 1 and Angle 2:** Represent the angles between the chest and the arms (nodes m4 and m5). These angles should range from approximately 0° to 110° during the movement for the left and right arms, respectively.
- **Angle 3 and Angle 4:** Represent the elbow flexion angles at nodes m7 and m9. These angles range from approximately 0° to 100° as the left and right arms are initially extended, bent, and then extended again.

- **Angle 5 and Angle 6:** Represent the knee flexion angles at nodes m18 and m19. These angles should change from approximately 0° to 90° during the squatting portion of the movement for the left and right knees, respectively.
- **Angle 7:** Represents the spinal tilt angle at node m6, which varies from approximately 0° to 90°. Incorrect performance results in reduced flexion, indicating compensation through squatting rather than proper bending.

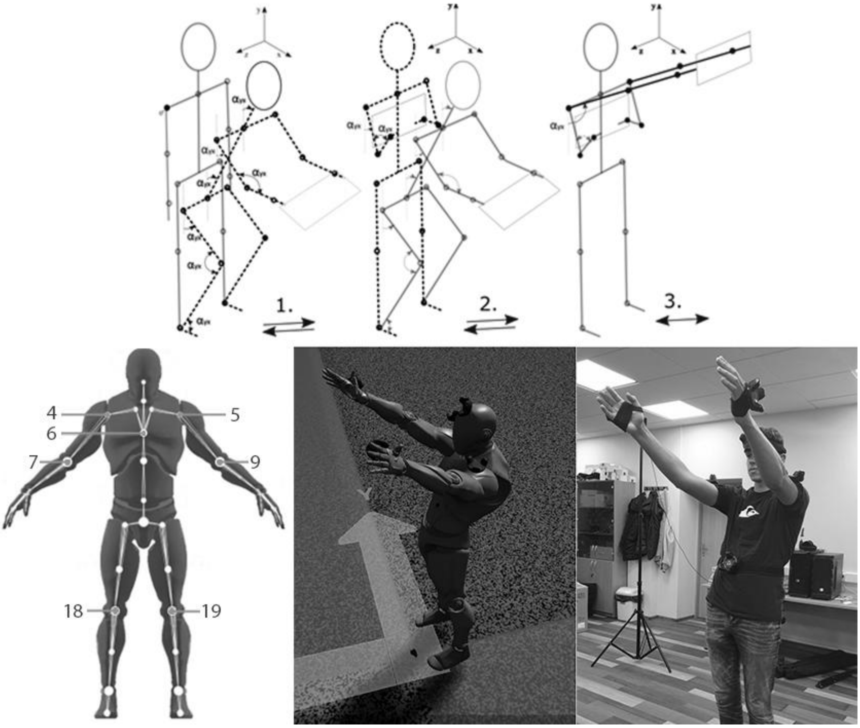


Fig. 62. On top: Lifting and Overhead Placement of a Heavy Object of the arm exercise movement angles. Left bottom: Analysed nodes. Bottom Middle: 3D mannequin in the system following the participant's movements. Bottom right: Participant

In the experiment, data were collected from 14 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 70 individual datasets for each class have been formed.

- **Class 0 (Correct Movement):** Participants performed the exercise correctly, maintaining coordinated movements of their arms, legs, and spine. The object was lifted and placed overhead smoothly without excessive leaning, forward bending, or imbalance. Movements were controlled, and posture remained stable throughout the exercise. Presented in blue on Fig. 63 and presented in green on Fig. 64.

- **Class 1 (Incorrect Movement):** Participants demonstrated improper form, often compensating with excessive forward bending or leaning, poor coordination of arm and leg movements, or instability during the lift. These deviations resulted in improper object placement, compromising balance and posture. Presented in yellow on Fig. 63 and presented in red on Fig. 64.

The analysis of the Lifting and Overhead Placement of a Heavy Object exercise highlights apparent differences in movement quality between Class 0 (Correct Movements) and Class 1 (Incorrect Movements), as reflected in Fig. 63 and Fig. 64. Across Angles 1 and 2, which represent the chest-arm angles, Class 0 demonstrates smooth transitions within the expected range of 0° to 110°, with tight distributions and well-defined confidence intervals (green in Fig. 64). This reflects proper arm coordination during the lift. Conversely, Class 1 exhibits broader variability, with values frequently falling outside the desired range, indicating poor arm movement coordination and improper object positioning. For Angles 3 and 4, representing elbow flexion, Class 0 maintains consistent control throughout the flexion-extension phases, as shown by the uniform distributions in Fig. 63 (blue) and narrow confidence intervals in Fig. 64 (green). In contrast, Class 1 displays scattered values and irregular transitions, indicative of reduced control over arm movements during the lift. Angles 5 and 6, representing knee flexion, further emphasize these differences. Class 0 adheres closely to the expected range of 0° to 90°, maintaining consistent and balanced squatting mechanics. However, Class 1 shows significant variability and frequent deviations from the expected range, suggesting instability and an over-reliance on compensatory behaviours, such as excessive back bending. Finally, for Angle 7 (Spinal Tilt), Class 0 achieves values within the desired range of 0° to 90°, reflecting proper spinal alignment and minimal forward bending. In contrast, Class 1 exhibits pronounced deviations, characterized by broader confidence intervals and frequent values exceeding acceptable limits, which highlights excessive forward bending or insufficient engagement of the spine. Overall, the results show that Class 0 consistently maintains proper execution across all evaluated angles, with minimal variability and adherence to expected ranges. In contrast, Class 1 demonstrates substantial deviations, particularly in spinal tilt (Angle 7) and knee flexion (Angles 5 and 6). These findings underscore the importance of spinal stability and coordinated knee movements for safe and effective performance of this exercise, while also identifying key areas for targeted training to address compensatory behaviours and instability observed in Class 1.

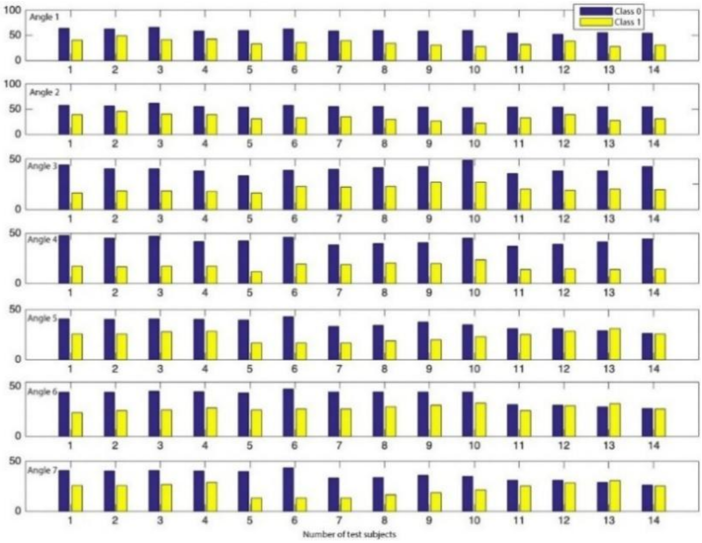


Fig. 63. Angle mean values of participants for the Lifting and Overhead Placement of a Heavy Object Exercise

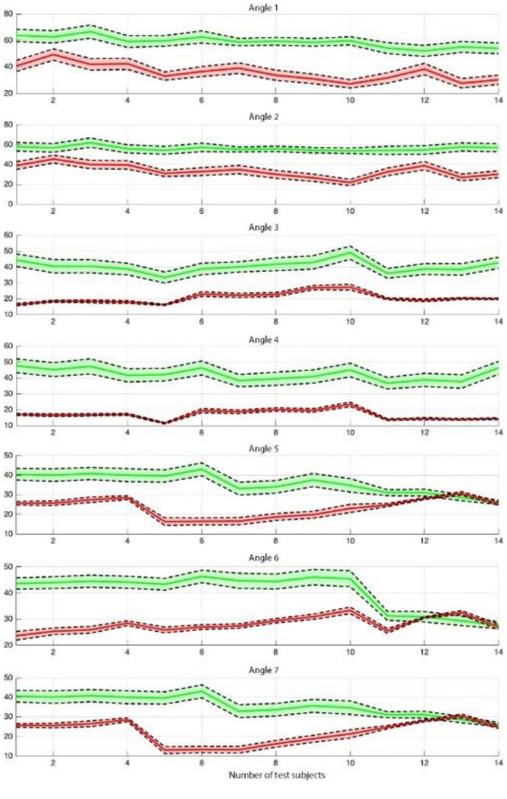


Fig. 64. Mean Values of Evaluated Angles and Confidence Intervals of Participants for the Lifting and Overhead Placement of a Heavy Object Exercise

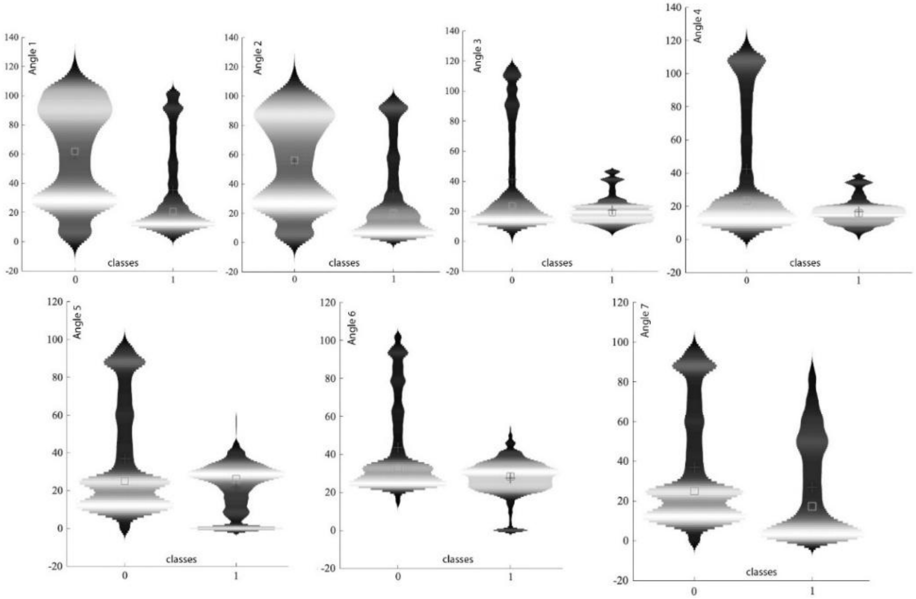


Fig. 65. Bean Plots of Evaluated Angles for the Lifting and Overhead Placement of a Heavy Object Exercise

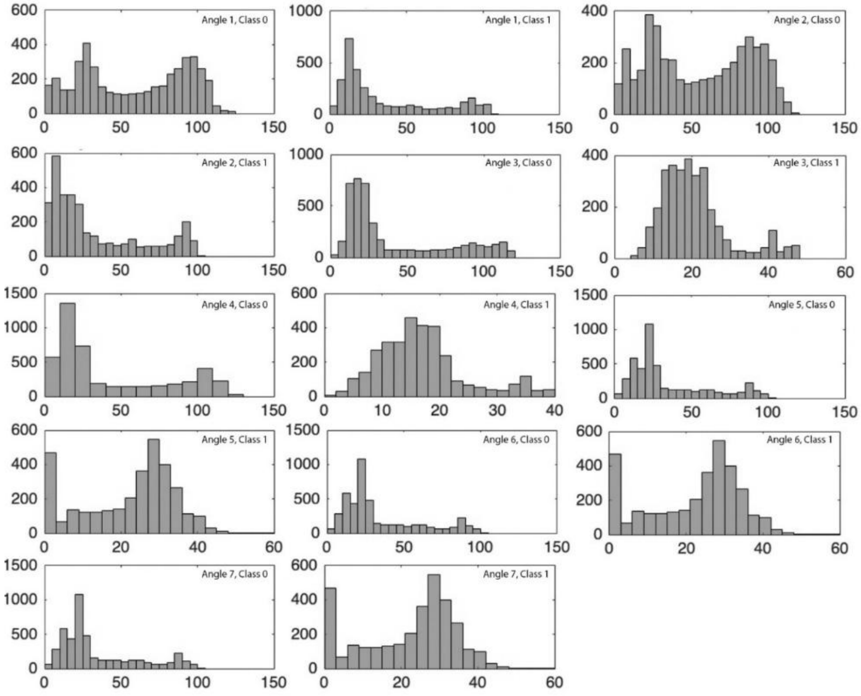


Fig. 66. Histograms of Evaluated Angles for the Lifting and Overhead Placement of a Heavy Object Exercise

From the analysis represented by the bean plots Fig. 65, Class 0 demonstrates relatively wider distributions, and mean values align with the expected ranges for most angles, reflecting proper coordination of the arms, legs, and spine. Angles 1 and 2, which measure chest-to-arm angles, exhibit well-aligned mean values (~60°) with stable variability in Class 0. In contrast, Class 1 shows narrower distributions and significantly lower mean values (~30°–40°), indicating restricted arm movement during incorrect execution. Similarly, Angles 3 and 4 (elbow flexion angles) display appropriate ranges and higher mean values (~40°) in Class 0, while Class 1 consistently exhibits lower mean values (~20°), suggesting incomplete or compensatory arm flexion. Angles 5 and 6, representing knee flexion, reveal critical differences between the classes. Class 0 exhibits mean values near 40°, demonstrating proper squatting mechanics, while Class 1 displays narrower distributions with mean values clustered around 20°, indicative of inadequate knee flexion and improper use of leg muscles during the lift. Finally, Angle 7 (spinal tilt) highlights significant deviations; Class 0 shows mean values around 30°–40° with balanced distributions, whereas Class 1 shows much lower mean values (~15°) and constrained variability, suggesting compensatory strategies such as reduced spine bending. Overall, the bean plots reveal that Class 0 consistently maintains target-aligned ranges and variability across all angles, reflecting proper execution of the exercise with coordinated arm, leg, and spinal movements. In contrast, Class 1 demonstrates significant deviations, particularly in Angles 5 (knee flexion) and 7 (spinal tilt), which emerge as the most sensitive indicators of compensatory movements. The narrower distributions and lower mean values across most angles in Class 1 highlight reduced flexibility, instability, and incorrect movement patterns during the exercise. These findings underscore the importance of monitoring knee flexion and spinal tilt as critical metrics for assessing proper execution and identifying compensatory behaviours during heavy-object lifting exercises.

The histograms in Fig. 66 illustrate distinct movement patterns between Class 0 (Correct Movements) and Class 1 (Incorrect Movements) during the Lifting and Overhead Placement of a Heavy Object Exercise. For chest-to-arm angles (Angles 1 and 2), Class 0 shows broad distributions centred around 60°–110°, reflecting proper arm coordination and elevation, while Class 1 peaks at narrower ranges of 30°–50°, indicating restricted arm movement and incomplete extension. In elbow flexion angles (Angles 3 and 4), Class 0 demonstrates consistent distributions near 40°–100°, representing appropriate flexion and extension, whereas Class 1 is concentrated around 10°–30°, highlighting insufficient elbow movement and reduced efficiency in object handling. Knee flexion angles (Angles 5 and 6) further distinguish the classes: Class 0 is tightly clustered between 40° and 90°, indicative of proper squatting mechanics, and Class 1 is skewed toward lower ranges of 10°–40°, signalling an over-reliance on other body parts, such as the spine. For spinal tilt (Angle 7), Class 0 displays a broader range of 30°–50°, indicating adequate spinal flexion and stability, whereas Class 1 is concentrated below 20°–40°, suggesting compensatory behaviours and reduced spinal movement, which may increase the risk of injury.

4.4.6.1. Lifting and Overhead Placement of a Heavy Object Decision Classification

For the decision classification task in the "Lifting and Overhead Placement of a Heavy Object" exercise, the RF classifier identified A1m, A4d, and A3d as the most critical features (Fig. 67). These angles contributed significantly to the model's ability to distinguish between the exercise classes.

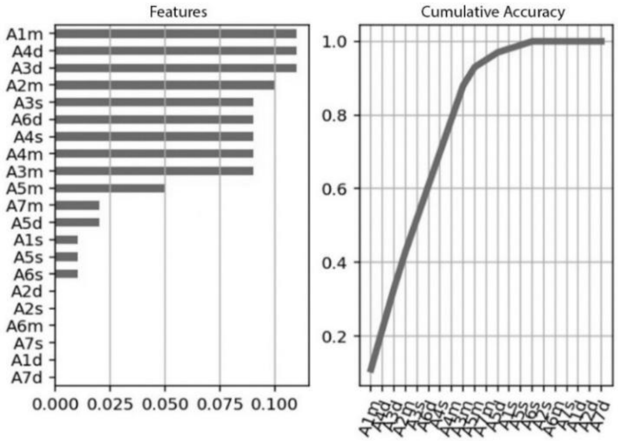


Fig. 67. Feature Importance for the Lifting and Overhead Placement of a Heavy Object Exercise

Both RF and CNN achieved 100% accuracy on this task (Fig. 67). RF correctly classified all test samples with 10 trees and a maximum depth of 3, and its execution time was remarkably low at 0.031 seconds. The CNN classifier required 8722 iterations over 14 epochs, with an execution time of 11.57 seconds. Despite the computational differences, both methods proved equally effective at classifying exercise decisions.

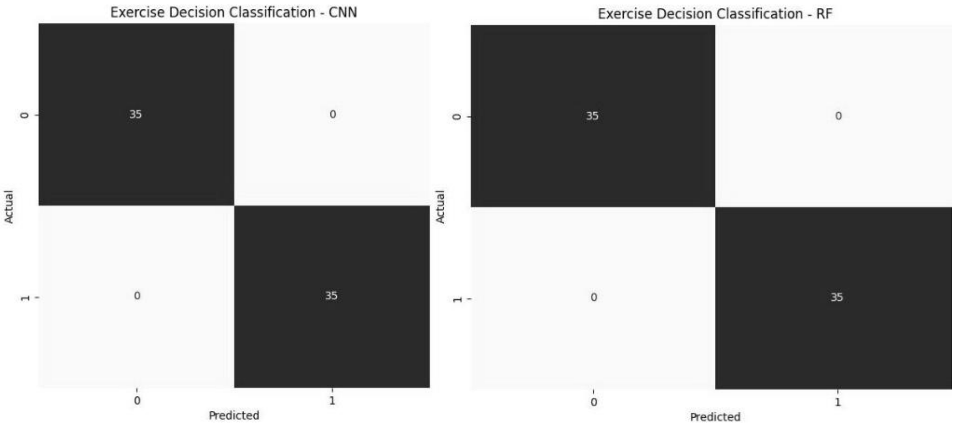


Fig. 68. CNN vs RF Confusion Matrix of the Lifting and Overhead Placement of a Heavy Object Exercise

Table 19. Metrics of Decision Classification for the Lifting and Overhead Placement of a Heavy Object exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	10	Number of iterations	8722
Maximum depth	3	Number of Epochs	14
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.031 s	11.57s	
Test Accuracy	100%	100%	

4.4.6.2. Overhead Placement of a Heavy Object Exercise: Angle Classification

For angle classification during the "Lifting and Overhead Placement of a Heavy Object" exercise, the CNN classifier achieved a test accuracy of 94.34%, as shown in the confusion matrix (Fig. 69). While most angles were classified accurately, there were minor misclassifications, particularly between classes 0 and 1.

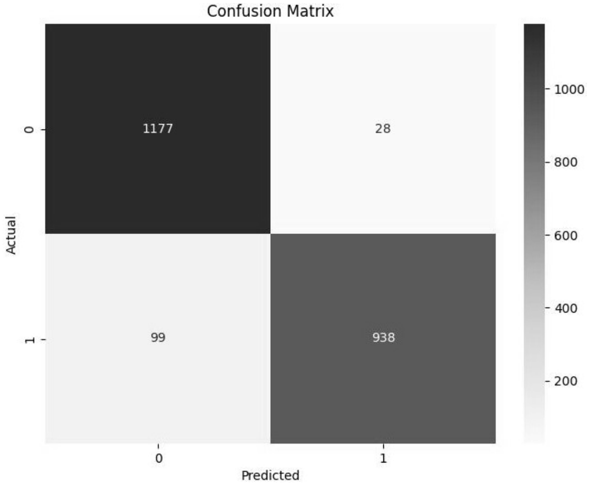


Fig. 69. Confusion Matrix of the Lifting and Overhead Placement of a Heavy Object

The CNN completed this task in 12.8 seconds, executing 34850 iterations over 11 epochs. The model utilised a batch size of 6 and a 20% validation split, demonstrating its capability to perform detailed angle-based classification efficiently.

Table 20. Metrics of Angle classification for the Lifting and Overhead Placement of a Heavy Object exercise

CNN Classifier	Value
Number of iterations	34850
Number of Epochs	11

Batch size	6
Validation split	20%
Execution time	12.8s
Test accuracy	94.34%

4.4.7. Axial Rotation of the Extended Arm

Nodes 4, 5, 7, 9, 10, and 12 are analysed in this exercise. The performance of the exercise is evaluated based on the following conditions:

- **Angle 1 and Angle 2:** Represent the elbow flexion angles at nodes 7 (left arm) and 9 (right arm), respectively. These angles should remain close to 0°, indicating that the arms are fully extended. In this experiment, correct movements are performed with the right arm (Angle 2), while the left arm (Angle 1) simulates both correct and incorrect movements.
- **Angle 3 and Angle 4**: Represent the rotation angles at nodes 10 (left arm) and 12 (right arm). These angles should range from approximately 0° to 180° during the exercise, encompassing the full rotation of the arms along their axes.
- **Angle 5**: Represents the shoulder alignment angle measured at nodes 4 and 5. This angle should remain close to 0°, keeping the shoulder line horizontal throughout the movement.

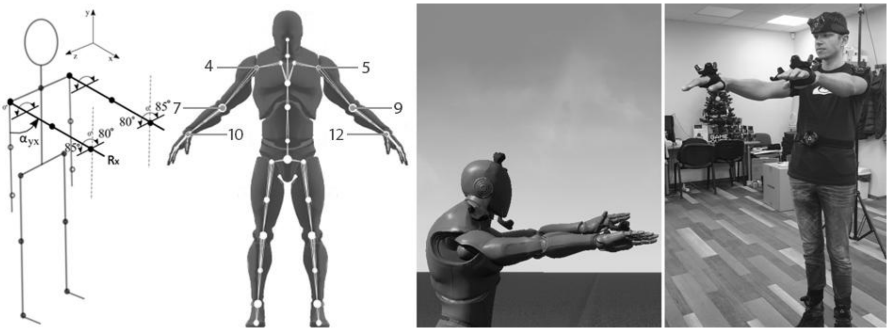


Fig. 70. From left to right: Axial Rotation of the Extended Arm exercise movement angle, analysed nodes, 3D mannequin in the system following the participant's movements, participant

In the experiment, data were collected from 18 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 90 individual datasets have been formed for each class.

- **Class 0 (Correct Movement):** Both arms correctly executed the exercise, maintaining proper form and alignment. The shoulder line remained straight, and the rotation angles followed the expected range. Presented in blue on Fig. 71 and presented in green on Fig. 72.

- **Class 1 (Incorrect Movement 1):** The left arm simulated an incorrect movement, with deviations in form and alignment. In some cases, these deviations were subtle, leading to overlaps with the correct data range. Presented in green on Fig. 71 and presented in red on Fig. 72.
- **Class 2 (Incorrect Movement 2):** The left arm simulated a different type of incorrect movement, with more pronounced deviations from the correct form, particularly in the rotational angles and alignment. Presented in yellow **on** Fig. 71 and presented in blue on Fig. 72.

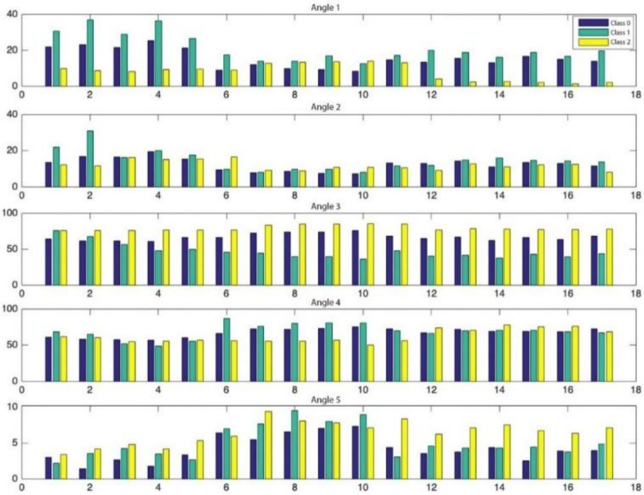


Fig. 71. Angle mean values of participants for the Axial Rotation of the Extended Arm Exercise

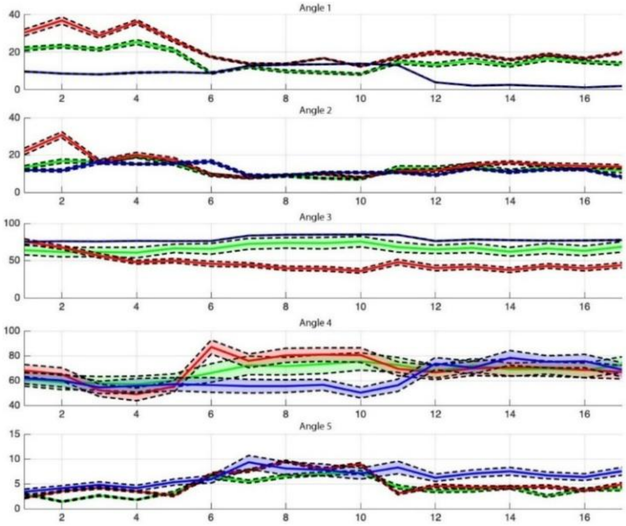


Fig. 72. Confidence intervals of participants for the Axial Rotation of the Extended Arm Exercise

From the data provided in Fig. 71 and Fig. 72, clear differentiation between Class 0 (Correct Movements), Class 1 (Incorrect Movements 1), and Class 2 (Incorrect Movements 2) is challenging across most angles. At the same time, slight trends are observable for Angles 3 and 4, where Class 0 tends to maintain higher rotational ranges; the overlaps in confidence intervals and variability within Classes 1 and 2 significantly reduce the reliability of these angles for distinguishing movement quality. For Angles 1, 2, and 5, the overlap is extensive, rendering these angles ineffective for separating the classes. This highlights the need for either enhanced data representation or additional complementary metrics to better capture the distinctions in arm-rotation exercises.

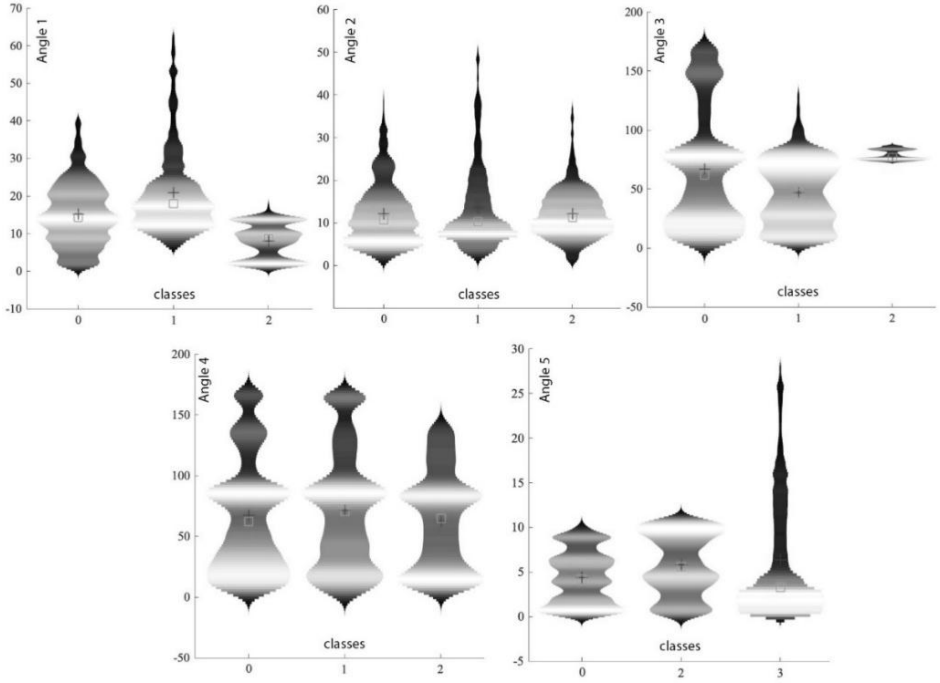
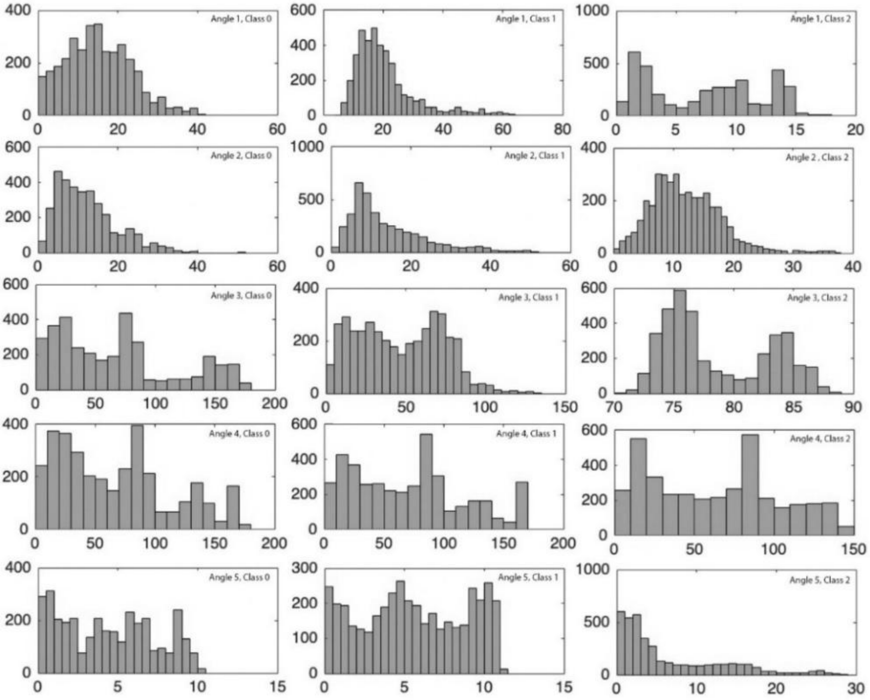


Fig. 73. Bean Plots of Evaluated Angles for the Axial Rotation of the Extended Arm Exercise



s

Fig. 74. Histograms of Evaluated Angles for the Axial Rotation of the Extended Arm Exercise

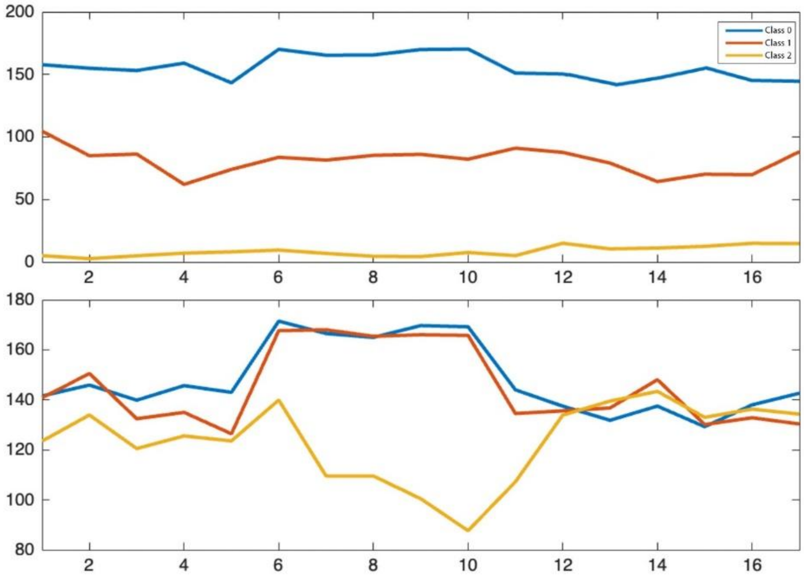


Fig. 75. Max/Min Range: Angle 3 (top), Angle 4 (bottom)

Based on Fig. 75, which depicts the distance between the maximum and minimum values for Angle 3 (top) and Angle 4 (bottom) across the three classes, clearly shows distinct differences in variability between the classes. For Angle 3, Class 0 (correct movements) consistently exhibits the highest range, remaining above approximately 150 units across most samples, indicative of a broad, controlled movement range. Class 1 (compensatory movements) exhibits moderate variability, maintaining values around 100 units, indicating a reduced but relatively consistent range of motion compared to Class 0. Class 2 (additional errors) exhibits a significantly reduced range, averaging well below 50 units, which is indicative of restricted or improper movements. For Angle 4, the trends are less distinct. While Class 0 starts with higher variability, around 150–160 units, it converges closely with Class 1 and Class 2 around samples 12–16, reducing the differentiation between these classes for this angle. Initially, Class 1 and Class 2 exhibit slightly lower ranges (120–140 units), but these differences become less pronounced toward the end of the series. This suggests that Angle 3 is a more reliable indicator of correct versus incorrect movements, as it maintains consistent class separation. In contrast, Angle 4 exhibits overlapping trends, which reduce its discriminative power in later samples.

The Random Forest model included all three statistical measures of Angle 3, as well as the distance measures for Angles 5 and 1 (A5d and A1d), and their standard deviations (A5s and A1s). These features were selected because their importance scores exceeded the predefined threshold (see Fig. 76). During testing, the RF model achieved 100% classification accuracy, demonstrating its reliability in classifying movement data for this exercise.

4.4.7.1. Axial Rotation of the Extended Arm Exercise Decision Classification

For the Axial Rotation of the Extended Arm Exercise Decision Classification, the RF classifier utilised 20 trees with a maximum depth of 3, achieving a 100% classification accuracy. Both classifiers displayed perfect classification performance, as evidenced by their confusion matrices, with no misclassifications.

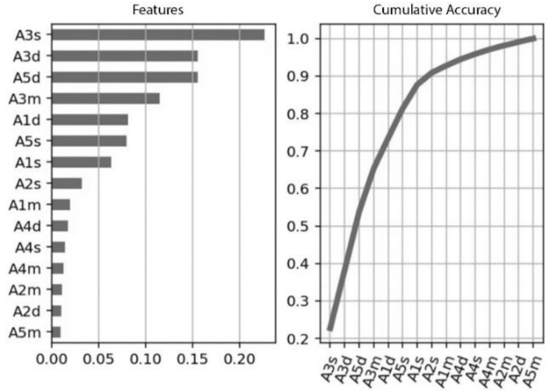


Fig. 76. Feature Importance for the Axial Rotation of the Extended Arm Exercise

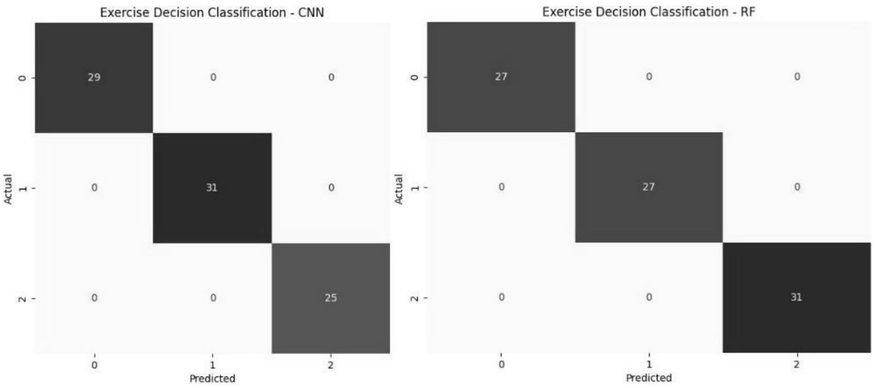


Fig. 77. CNN vs RF Confusion Matrix of the Axial Rotation of the Extended Arm Exercise

The CNN classifier, trained over 17360 iterations and 14 epochs, also achieved 100% accuracy, demonstrating its effectiveness in this task. The execution time for RF was significantly faster at 0.031 seconds compared to CNN's 20.84 seconds, reflecting RF's computational efficiency.

Table 21. Metrics of Decision Classification for the Axial Rotation of the Extended Arm

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	20	Number of iterations	17360
Maximum depth	3	Number of Epochs	14
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.031 s	20.84s	
Test Accuracy	100%	100%	

4.4.7.2. Axial Rotation of the Extended Arm Exercise Angle Classification

For the Angle Classification for the Axial Rotation of the Extended Arm Exercise, the CNN classifier achieved a test accuracy of 95.7%, demonstrating its ability to classify the majority of angles correctly. The confusion matrix shows that 93.05% of Class 0 samples were correctly classified, with a misclassification rate of 6.95%, with most errors predicted as Class 1. Similarly, 94.54% of Class 1 samples were accurately classified, with a 5.46% misclassification rate, predominantly misclassified as Class 0. For Class 2, the model performed exceptionally well, correctly classifying 99.20% of samples, with only 0.80% misclassified.

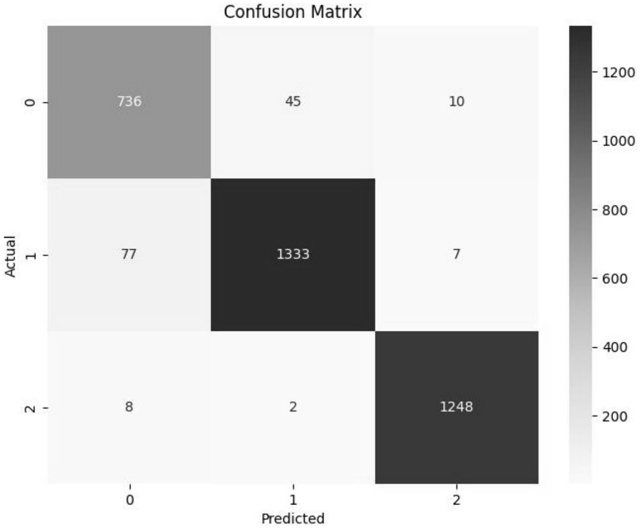


Fig. 78. Confusion Matrix of the Axial Rotation of the Extended Arm Exercise

The execution time for this task was 21.83 seconds, during which the CNN processed 52300 iterations over 14 epochs, using a batch size of 6 and a 20% validation split. These results underscore the model's efficiency and accuracy, with minor errors observed between closely related classes.

Table 22. Metrics of Angle Classification for the Axial Rotation of the Extended Arm Exercise

CNN Classifier	Value
Number of iterations	52300
Number of Epochs	14
Batch size	6
Validation split	20%
Execution time	21.83s
Test accuracy	95.7%

4.4.8. Nose Touch Coordination Exercise

During the Nose Touch Coordination exercise, participants performed the movement under three distinct conditions, with 16 participants assigned to each class. In Class 0, representing correct movements, participants maintained proper coordination without compensatory behaviours, ensuring the head remained stationary and dynamic angles, such as Angle 2 and Angle 3, transitioned smoothly while stationary angles, including Angle 1, Angle 4, and Angle 5, remained relatively constant. Minor deviations were observed in Angle 4 during correct movements, with values occasionally reaching 13 degrees instead of remaining closer to zero. In Class 1, representing the first type of incorrect movement, compensatory behaviours, such as head tilting or shoulder misalignment, led to significant deviations in stationary angles, as evident from the broader variability in Angle 4 values. Finally, in Class 2, representing another type of incorrect movement, participants displayed overextension or misalignment during the upward or nose-reaching phase, resulting in abnormal ranges or erratic transitions in the dynamic angles. The variability in these angles was more prominent than in Class 1, highlighting distinct movement patterns. Across all conditions, clear distinctions were observed in the analysed angles, providing a robust basis for identifying correct and incorrect movement executions.

The nodes 4, 5, 7, 9, 0, and 6 are analysed during this exercise. The performance of the exercise is evaluated based on the following conditions:

- **Angle 1**: The head must remain stationary, with the vertical tilt angle at node 0 remaining close to **0 degrees**.
- **Angle 2**: The elbow should bend to an angle of approximately 170 degrees as the participant moves the finger to touch the nose
- **Angle 3**: The arm is initially raised sideways to an upward position, with the shoulder angle ranging from ~0 degrees to ~90 degrees (or ~180 degrees to ~90 degrees in the system).
- **Angle 4**: The shoulder line must stay horizontal, with the vertical angle at nodes 4 or 5 close to 0 degrees.
- **Angle 5**: The participant must maintain an upright posture, with the spinal tilt angle at node 6 staying near 90 degrees relative to the horizontal plane.

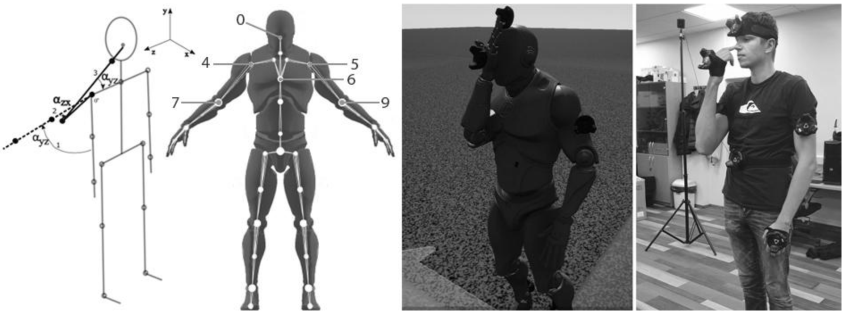


Fig. 79. From left to right: Nose Touch Coordination exercise movement angle, analysed nodes, 3D mannequin in the system following the participant’s movements, participant

In the experiment, data were collected from 18 healthy participants who performed each classified exercise five times to obtain a larger dataset. In total, 90 individual datasets have been formed for each class.

- **Class 0 (Correct Movement):** The participant performs the exercise correctly, maintaining a stationary head and achieving accurate coordination without compensatory movements. Presented in blue on Fig. 80 and presented in green on Fig. 81.

- **Class 1 (Incorrect Movement 1):** The participant performs the exercise incorrectly by introducing compensatory movements, such as tilting the head or misaligning the shoulder line. Presented in green on Fig. 80 and presented in red on Fig. 81.
- **Class 2 (Incorrect Movement 2):** The participant performs the exercise incorrectly by overextending or misaligning the arm during the upward or nose-reaching phase. Presented in yellow on Fig. 80 and presented in blue on Fig. 81.

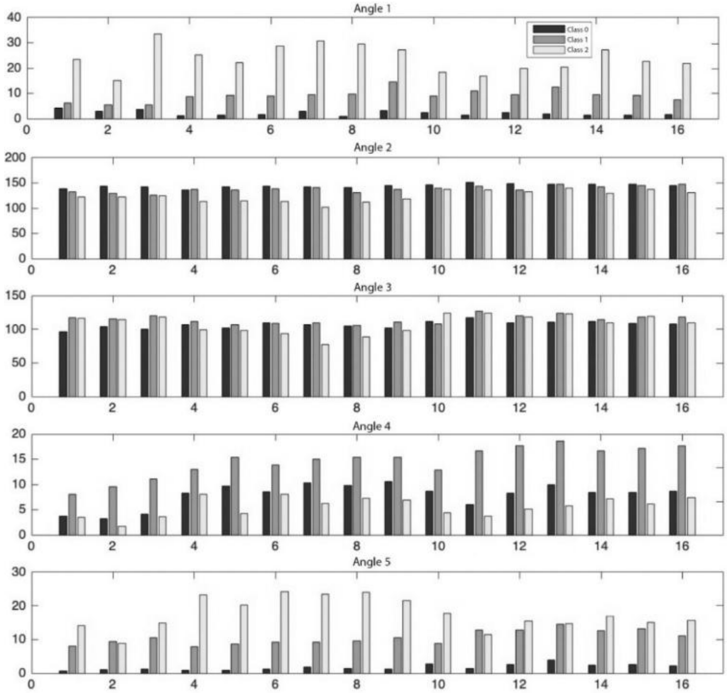


Fig. 80. Angle mean values of participants for the Nose Touch Coordination Exercise

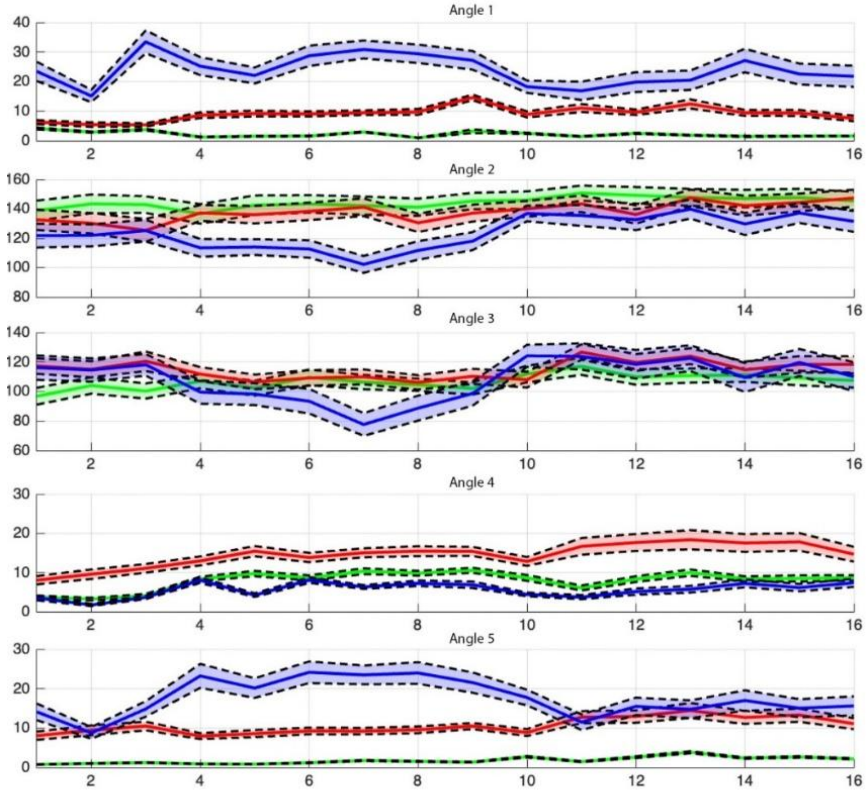


Fig. 81. Confidence intervals of participants for the Nose Touch Coordination Exercise

The Nose Touch Coordination Exercise: Mean values and confidence intervals for the five angles are analysed. Angle 1 (Head Tilt) shows high distinctiveness, with Class 0 showing values near 0–10 degrees, while Classes 1 and 2 exhibit higher means and greater variability (Fig. 80 and Fig. 81), indicating its reliability in detecting compensatory head tilting behaviours. Angle 4 (Shoulder Line Alignment) is another highly distinctive parameter, with Class 0 values consistently near 0–10 degrees and broader deviations for Classes 1 and 2, as shown by wider confidence intervals and greater mean separation. Similarly, Angle 5 (Spinal Tilt) effectively identifies incorrect movements: Class 0 maintains alignment within 0–10 degrees, while Classes 1 and 2 deviate significantly to 20–30 degrees, particularly in Class 2. In contrast, Angle 2 (Elbow Bending) and Angle 3 (Arm Raising) are less distinctive; Angle 2 exhibits overlapping values across all classes (~120–160 degrees), limiting its utility, while Angle 3 shows moderate separation but with high variability, especially in Class 2 (~80–120 degrees). Overall, Angles 1, 4, and 5 emerge as the most distinctive parameters, providing robust indicators of incorrect movement patterns, whereas

Angles 2 and 3 offer limited discriminatory power due to overlapping values and variability.

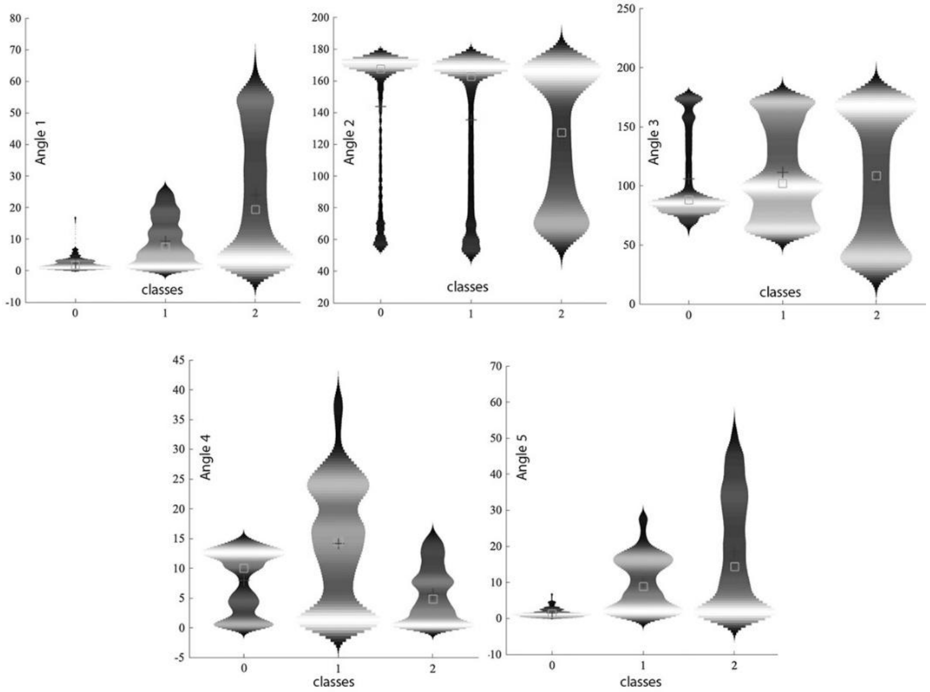


Fig. 82. Bean Plots of Evaluated Angles for the Nose Touch Coordination Exercise

The bean plots in Fig. 82 For the Nose Touch Coordination exercise, provide a clear representation of the distribution of values for Angles 1 through 5 across three classes (Class 0: Correct Movements, Class 1: Incorrect Movement 1, Class 2: Incorrect Movement 2). The median and mean markers highlight distinct behavioural differences for these angles among the classes. For Angle 1, Class 0 demonstrates low values tightly grouped near zero, consistent with proper head stability. In contrast, Class 2 shows a markedly broader distribution, with mean values around 20 and upper values extending to 70, indicating significant instability or compensatory head movements. Angle 2, however, shows minimal separation between Class 0 and Class 1, as both exhibit similar distributions and central tendencies near 170, reflecting overlapping behaviour in elbow flexion. For Angle 3, Class 0 maintains a concentrated range around 90 degrees, corresponding to proper shoulder rotation. In contrast, Class 2 exhibits greater variability and a mean value closer to 100, indicating misalignment and overextension in the arm movement. Angle 4, which captures shoulder line stability, remains close to zero for Class 0 but exhibits increased spread and higher mean values in Class 1, indicative of tilting or alignment issues. Angle 5 provides a strong distinction: Class 0 maintains tight distributions near 10, while Class 2 exhibits significantly higher mean values, often exceeding 50, highlighting compensatory behaviours or improper spinal posture. Based on these observations, Angles 1, 3, and

5 are the most effective indicators for distinguishing between correct and incorrect movements, as their distributions and central tendencies show the greatest separability across the classes.

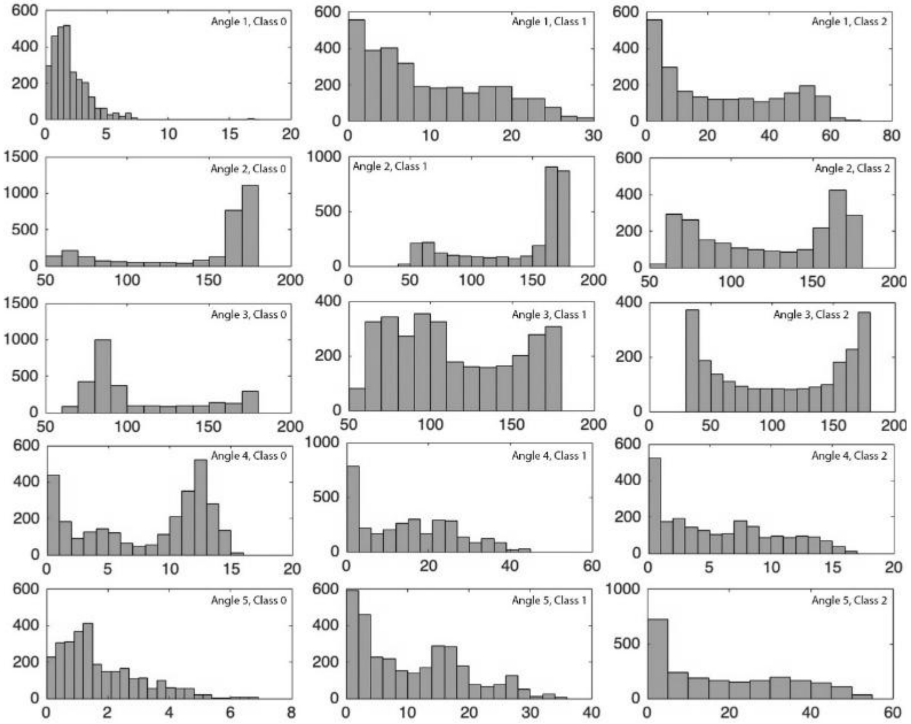


Fig. 83. Histograms of Evaluated Angles for the Nose Touch Coordination Exercise

The histogram data in Fig. 83 provides a deeper insight into the distribution of evaluated angles for each class in the Nose Touch Coordination exercise. Angle 1 shows a narrow, concentrated distribution near 0 for Class 0, with values rarely exceeding 10, confirming proper head stability. For Class 1, the distribution shifts slightly to the right, with higher values, while Class 2 exhibits a broader spread, extending up to 70, indicating head instability. Angle 2 reveals minimal distinction between Class 0 and Class 1, as both exhibit peaks around 170 degrees, with some overlap in distributions. However, Class 2 shows a notable shift in distribution, with a broader spread and higher frequency of values around 140 degrees, reflecting deviations in elbow flexion. Angle 3 highlights a distinct peak near 90 degrees for Class 0, indicative of proper shoulder rotation. Class 1 exhibits a bimodal distribution, indicating performance variability, whereas Class 2 displays a more uniform distribution, spanning the entire range, suggesting significant deviations. Angle 4 shows a tightly grouped distribution near 0 for Class 0, indicative of proper shoulder line stability. In contrast, Class 1 displays a broader spread, peaking around 10, and Class 2 exhibits a similar spread, albeit slightly more concentrated, confirming misalignments in both incorrect movement classes. Angle 5, associated with spinal posture, exhibits a distinct peak at 5 degrees for Class 0, indicating proper alignment. Class 1 exhibits a broader distribution, with values extending to 30, while Class 2 shows a significantly broader range, peaking around 50, reflecting pronounced deviations in spinal posture. These histograms reinforce the earlier observations that Angles 1, 3, and 5 show the clearest distinctions across classes, particularly in differentiating between correct and incorrect movements.

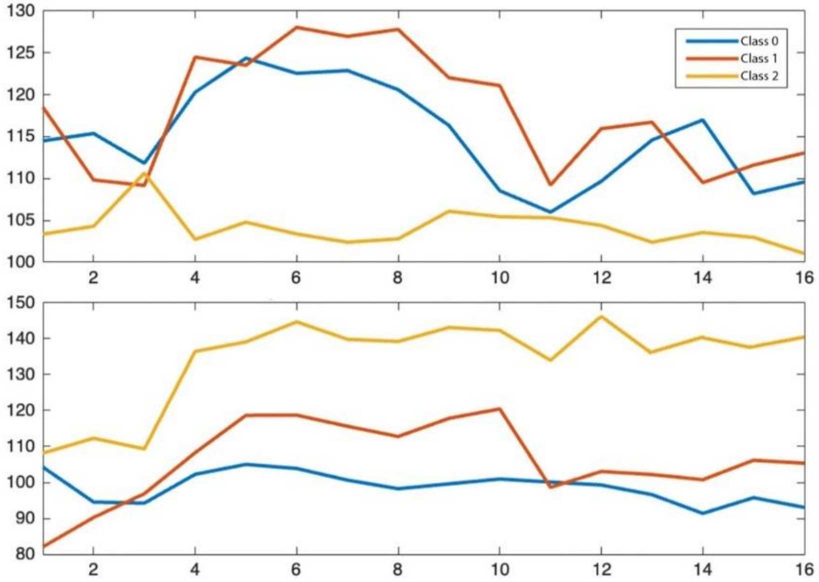


Fig. 84. Max/Min Range: Angle 2 (Top), Angle 3 (Bottom)

Fig. 84 illustrates the distance between maximum and minimum values for Angles 2 and 3 across Class 0 (Correct Movements), Class 1 (Incorrect Movement 1), and Class 2 (Incorrect Movement 2) for the Nose Touch Coordination exercise. For Angle 2 (top), the spread of values is most stable for Class 0, remaining relatively constant throughout all participants, reflecting proper elbow movement during the exercise. In Class 1, the spread shows a notable increase around participants 4–6, indicating greater variability in elbow flexion, which may be due to compensatory behaviours. Class 2 shows the smallest spread, with values consistently lower than those of Class 0 and Class 1, suggesting a restricted range of motion or significant performance deviations. For Angle 3 (bottom), Class 0 shows consistent ranges with minor fluctuations, consistent with the expected shoulder rotation. Class 1 shows an increased, fluctuating spread among participants 6–10, highlighting variability in incorrect execution. Class 2 shows the most extensive spread, with notable peaks around participants 6–8 and 10–14, suggesting pronounced deviations in shoulder movement. This visualisation highlights differences in movement patterns, with Angle 3 exhibiting the most pronounced differentiation among the three classes due to its greater dynamic range. In comparison, Angle 2 offers insights into compensatory behaviour but less differentiation between Classes 0 and 1. In conclusion, the most informative angles for the Nose Touch Coordination exercise are:

- **Angle 1**: Represents forward head tilt
- **Angle 3**: Represents arm elevation to the side
- **Angle 5**: Represents forward spinal tilt

The construction of the Random Forest model utilised all three statistical measures of "Angle 1" and "Angle 5," as well as the range (A2d) of "Angle 2," the range (A4d) and standard deviation (A4s) of "Angle 4," and the standard deviation (A3s) of "Angle 3," as these features exceeded the selected threshold (see Fig. 85). During testing, the constructed RF model achieved 100% classification accuracy with the available data.

4.4.8.1. Nose Touch Coordination Exercise Decision Classification

For the "Nose Touch Coordination Exercise," the RF classifier identified Angles A1, A1d, and A1m as the most influential features, as shown in the feature importance analysis (Fig. 85). Both RF and CNN classifiers achieved perfect decision classification accuracy of 100%, as evidenced by their respective confusion matrices (Fig. 86).

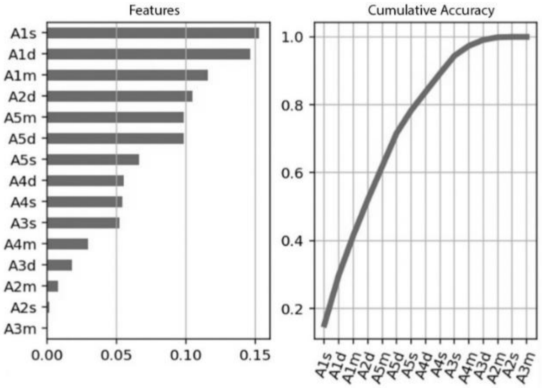


Fig. 85. Feature Importance for the Nose Touch Coordination Exercise

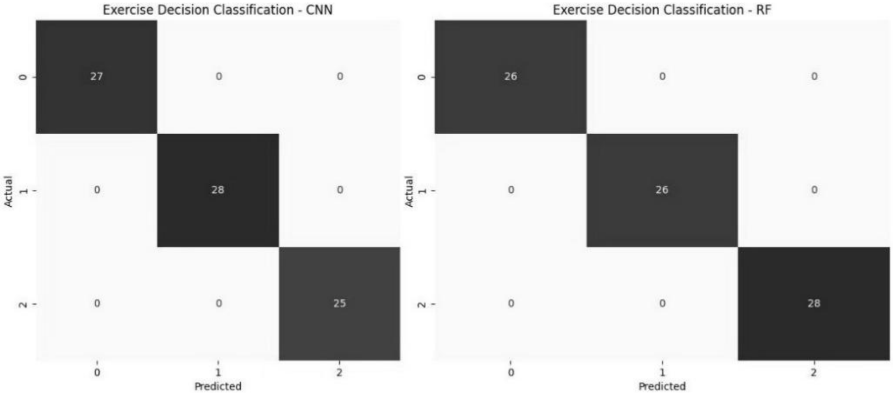


Fig. 86. CNN vs RF Confusion matrix of the Nose Touch Coordination Exercise

RF achieved this result with only 10 trees and an execution time of 0.03262 seconds, leveraging an OOB-based early stopping strategy and a maximum depth of 3. CNN, on the other hand, required 12908 iterations over 14 epochs, completing in 15.53 seconds with a batch size of 6.

Table 23. Metrics of Decision Classification for the Nose Touch Coordination Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	10	Number of iterations	12908
Maximum depth	3	Number of Epochs	14
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.03262 s	15.53s	
Test Accuracy	100%	100%	

4.4.8.2. Nose Touch Coordination Exercise. Angle Classification

In angle-based classification, the CNN achieved an accuracy of 92% (Fig. 87). The confusion matrix revealed some misclassifications across all three classes. Class 0 achieved an 88.56% correct classification rate, with 65 instances misclassified into Class 1 and 15 into Class 2. Class 1 achieved 91.48% accuracy, with 22 instances misclassified to Class 2 and 65 misclassified to Class 0. Class 2 had an 88.62% correct classification rate, with 55 instances misclassified into Class 1 and 23 into Class 0.

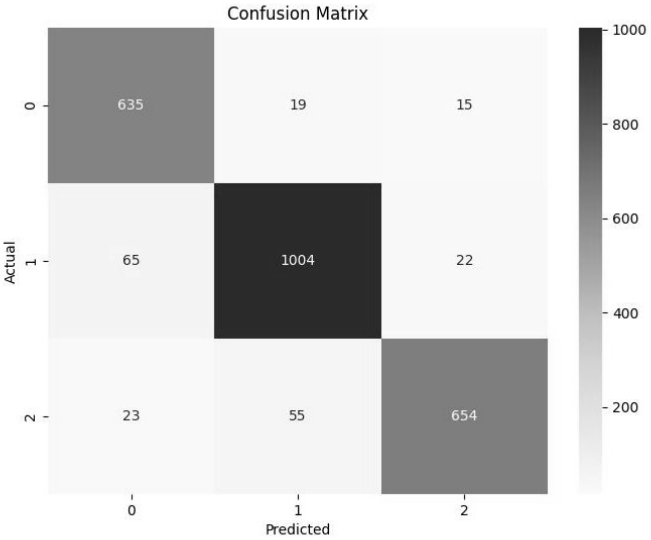


Fig. 87. Confusion Matrix of the Nose Touch Coordination Exercise

Despite these minor inaccuracies, the CNN processed the data in 27.21 seconds, completing 39050 iterations across 17 epochs, showcasing its robustness in handling angle-based classifications.

Table 24. Metrics of Angle Classification for the Nose Touch Coordination Exercise

CNN Classifier	Value
Number of iterations	39050
Number of Epochs	17
Batch size	6
Validation split	20%
Execution time	27.21s
Test accuracy	92%

4.4.9. Forearm Supination/Pronation

During the Forearm Supination/Pronation exercise, nodes 7, 9, 10, and 12 are analysed. Exercise performance is evaluated based on the following conditions:

- **Angle 1 and Angle 2**: The arms must bend approximately 90 degrees at the elbows. The elbow flexion angles at nodes 7 or 9 should range from approximately 0° to 90°. These angles correspond to the left and right arm elbow flexion, respectively. In the study, the right arm performed correct movements (Angle 2), while the left arm simulated both correct and incorrect movements (Angle 1).
- **Angle 3 and Angle 4**: The arms must rotate 180 degrees. The rotational angles at nodes 10 and 12 should vary from approximately 0° to 180°, corresponding to the left and right arm rotations, respectively.

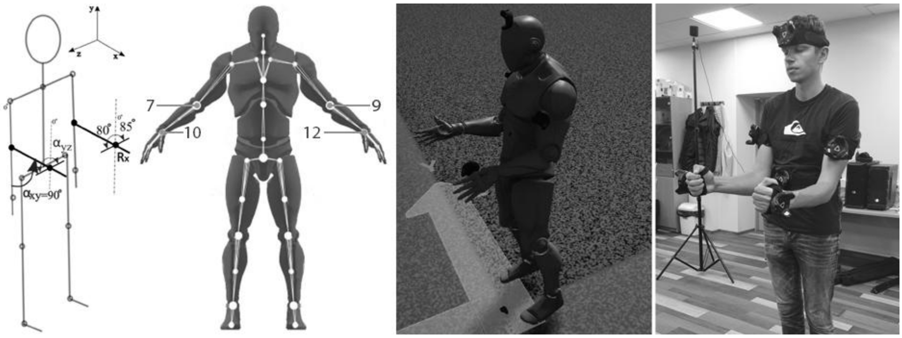


Fig. 88. From left to right: Forearm Supination/Pronation exercise movement angle, analysed nodes, 3D mannequin in the system following the participant's movements, participant

Data were collected from 15 participants performing the Forearm Supination/Pronation exercise under three distinct conditions, 5 times. In total, 75 datasets were formed. The classes are defined as follows:

- **Class 0 (Correct Movement):** Participants performed the exercise correctly with both arms, maintaining smooth and coordinated transitions in the forearm rotational angles and elbow flexion. The right arm always fell into this category, and the left arm movements showed consistent angles. Presented in blue on Fig. 89 and in green on Fig. 90.

- **Class 1 (Incorrect Movement 1):** Incorrect movements were simulated with the left arm, displaying deviations such as incomplete rotational motion or inconsistent elbow flexion angles. Variations in angular transitions were evident, often resulting in irregular movements. Presented in green on Fig. 89 and in red on Fig. 90.
- **Class 2 (Incorrect Movement 2):** The left arm exhibited a second type of incorrect movement, such as over-rotation or abrupt transitions in rotational angles. This class highlighted exaggerated angular deviations compared to Class 1, further diverging from the correct execution. Presented in yellow on Fig. 89 and in blue on Fig. 90.

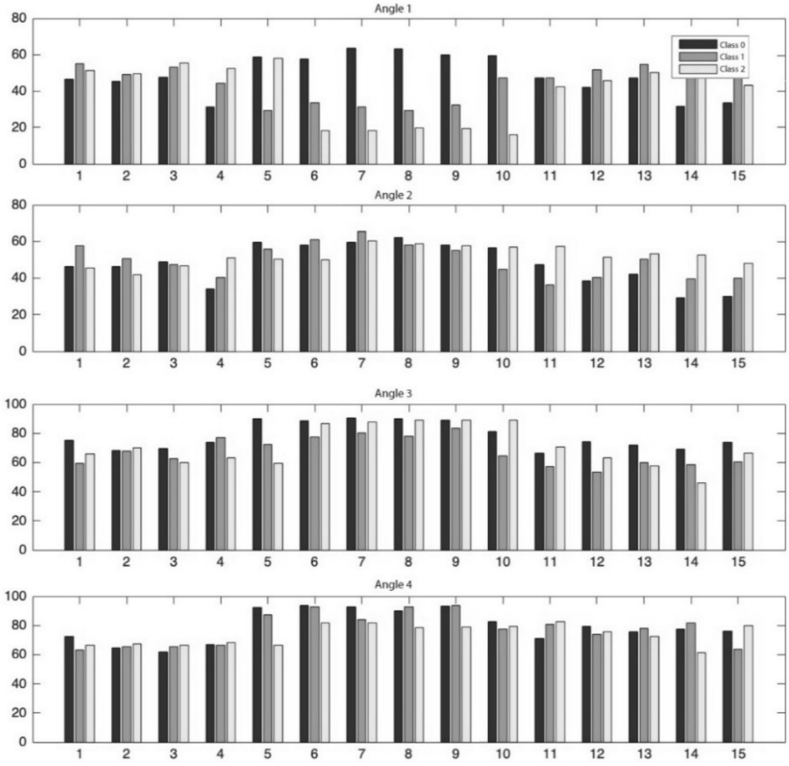


Fig. 89. Angle mean values of participants for the Forearm Supination/Pronation Exercise

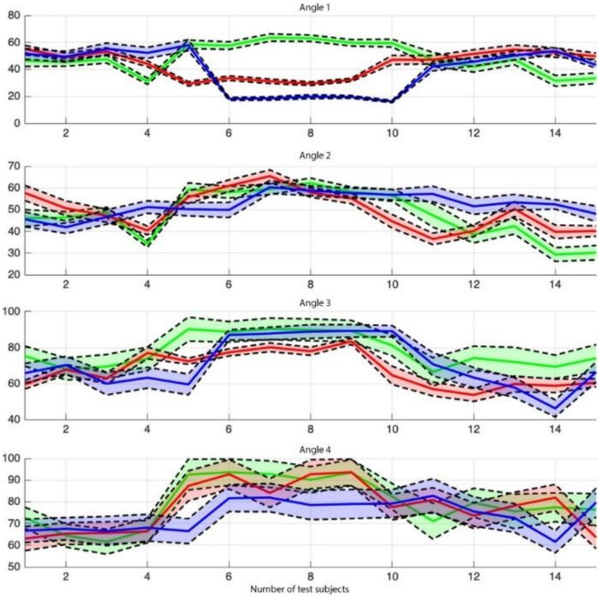


Fig. 90. Mean Values of Evaluated Angles and Confidence Intervals of Participants for the Forearm Supination/Pronation Exercise

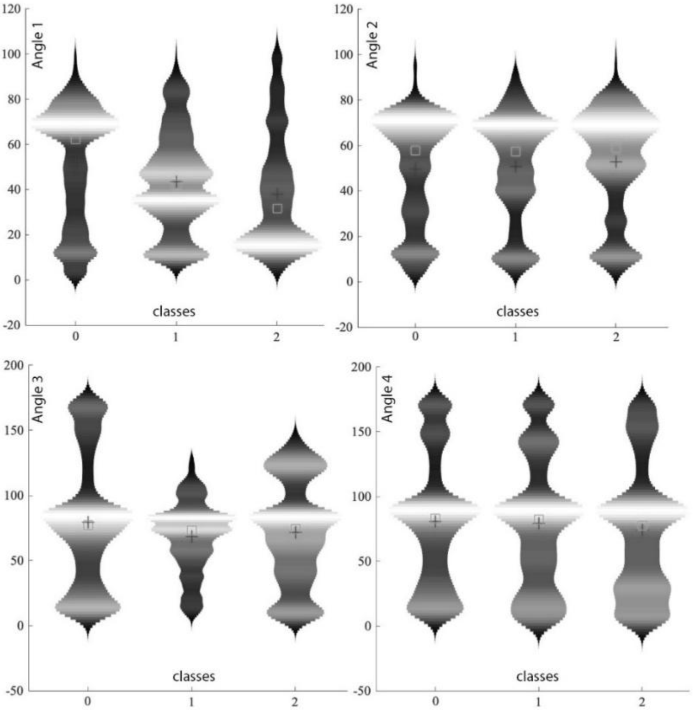


Fig. 91. Bean Plots of Evaluated Angles for the Forearm Supination/Pronation Exercise

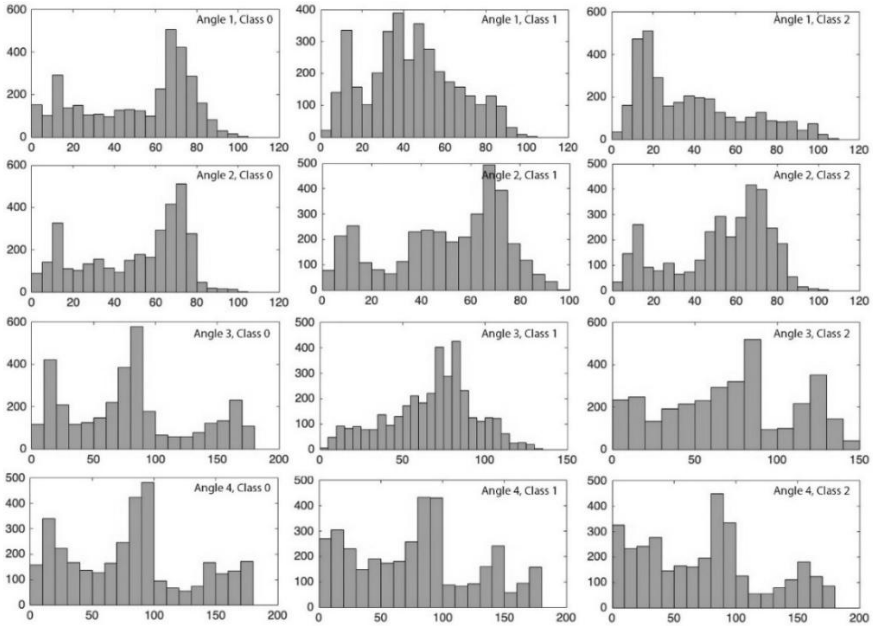


Fig. 92. Histograms of Evaluated Angles for the Forearm Supination/Pronation Exercise

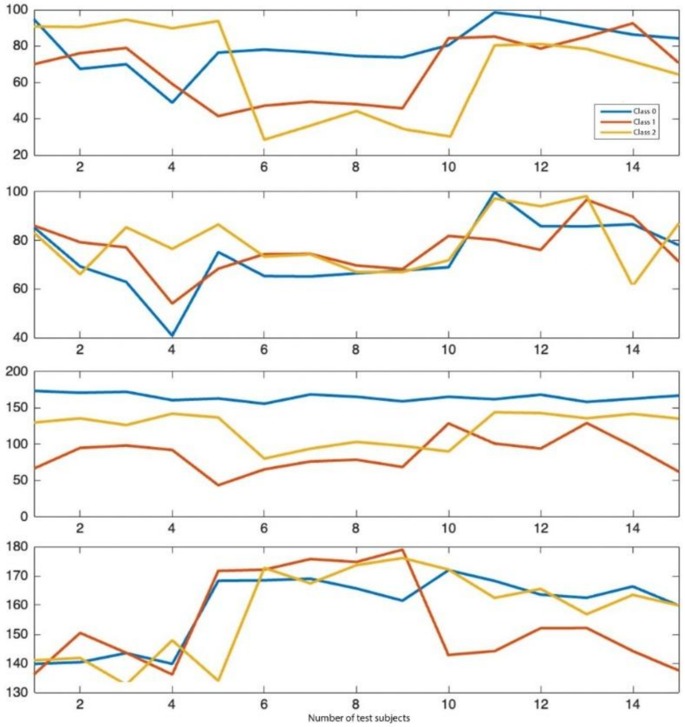


Fig. 93. Max/Min Range from top to bottom: Angle 1, 2, 3, and 4

Since all angles are dynamic, analysing their mean values or distributions is not appropriate because data may be concentrated at specific angle values when movements are performed more slowly. For this reason, additional analysis was conducted by visualizing the range between the maximum and minimum values of the respective angles. For Angle 1 (left arm) and Angle 2 (right arm), no apparent differences were observed. However, the data for Angles 3 and 4 show that both correct and incorrect movements were performed with the left arm, with correct movements involving approximately 180 degrees of rotation. It can be concluded that the most informative angles for identifying movement accuracy in the forearm supination/pronation exercise are Angle 3 (left arm rotation) and Angle 4 (right arm rotation).

Table 25. Statistical Analysis of Forearm Supination/Pronation Exercise Angles

Angle	Compared samples	t-test		F-test	
		p-value	Rejected?	p-value	Rejected?
Angle 1	Class 0 vs Class 1	0	Yes	0	Yes
	Class 0 vs Class 2	0	Yes	0,30	No
Angle 2	Class 0 vs Class 1	0,53	No	0,09	No
	Class 0 vs Class 2	0	Yes	0,01	Yes
Angle 3	Class 0 vs Class 1	0	Yes	0	Yes
	Class 0 vs Class 2	0	Yes	0	Yes
Angle 4	Class 0 vs Class 1	0,053	No	0,68	No
	Class 0 vs Class 2	0	Yes	0,91	No

The table’s results (see Table 25) indicate that there is no statistically significant difference in performing the exercise with the right arm between a healthy individual and one who executes the flexion correctly but performs the rotation poorly. However, in all other cases, there is a statistically significant difference in the mean values between healthy individuals and those with impairments. Since the exercise was always performed correctly with the right arm and correctly or incorrectly with the left arm, the results reveal statistically significant differences in the analysed angles. It is important to note that not all data samples follow a normal distribution (see Fig. 92), which may affect the accuracy of the statistics presented in the table. The Random Forest model for the forearm supination/pronation exercise was constructed using all three statistical measures (mean, standard deviation, and range) for Angle 1 and Angle 3, along with the mean value of Angle 4 (A4m), as these features exceeded the predefined threshold (see Fig. 94).

4.4.9.1. Forearm Supination/Pronation Decision Classification

For the Forearm Supination/Pronation Exercise, the RF classifier and CNN were compared for both decision and angle-based classifications. The RF model identified Angle 3 and Angle 3s as the most influential features, as depicted in the feature importance analysis (Fig. 94). For decision classification, both RF and CNN achieved high test accuracies, with CNN reaching 100% and RF slightly lower at 96%. The confusion matrices (Fig. 95) illustrate that RF misclassified three samples from Class

1 into Class 2 but performed flawlessly for Class 0 and Class 2. On the other hand, CNN was ideally able to classify all samples across the three classes.

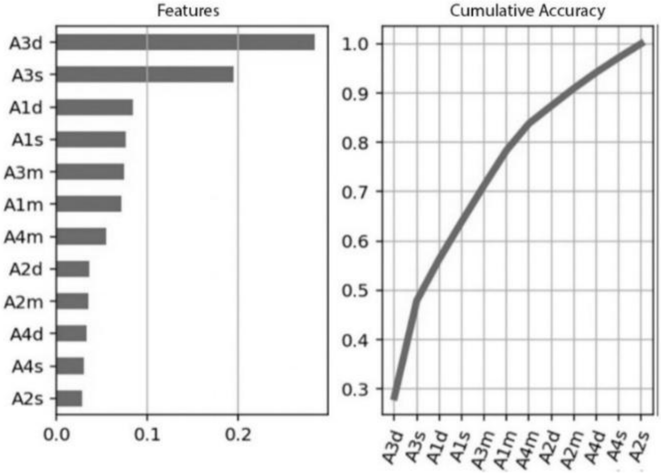


Fig. 94. Feature Importance for the Forearm Supination/Pronation Exercise

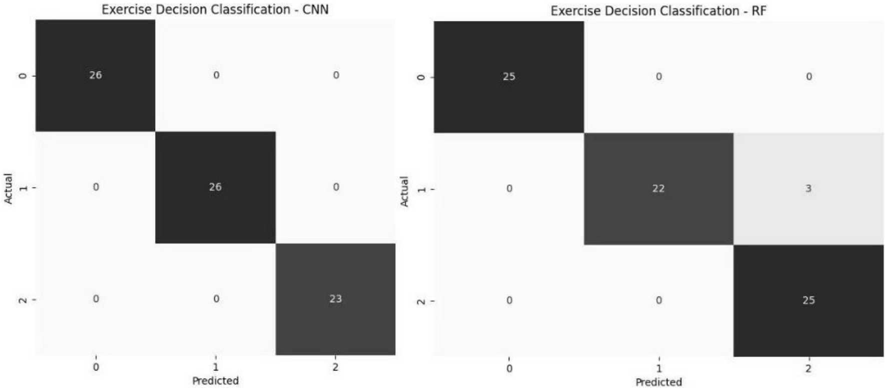


Fig. 95. CNN vs RF Confusion Matrix of the Forearm Supination/Pronation Exercise

The RF classifier utilised 30 trees with an out-of-bag (OOB) stopping criterion and a maximum depth of 3, achieving an execution time of 0.06 seconds, which is highly efficient. On the other hand, the CNN classifier executed 16212 iterations across 14 epochs with a batch size of 6, achieving a perfect test accuracy of 100%. Its execution time was 20.59 seconds, reflecting its iterative and computationally intensive nature. The confusion matrix shows that the CNN perfectly classified all samples across the three classes, with no misclassifications, demonstrating its robustness to complex data patterns. These metrics emphasize the trade-off between computational efficiency and classification performance when comparing RF and CNN.

Table 26. Metrics of Decision Classification for the Forearm Supination/Pronation Exercise

RF classifier	Value	CNN Classifier	Value
Number of trees (OOB)	30	Number of iterations	16212
Maximum depth	3	Number of Epochs	14
Minimum number of samples to split a node	6	Batch size	6
Minimum number of samples per leaf node	3	Validation split	20%
Maximum number of features considered	2		
Total execution time	0.06 s	20.59s	
Test Accuracy	96%	100%	

4.4.9.2. Forearm Supination/Pronation Exercise Angle Classification

For angle classification, the CNN achieved an accuracy of 85.73% with an execution time of 20.99 seconds (Fig. 96). While most angles were correctly classified, notable misclassifications occurred between Class 0 and Class 1, and between Class 2 and Class 1. Specifically, 87 Class 1 angles were misclassified as Class 2, and 129 Class 2 angles were misclassified as Class 0. Despite these challenges, the CNN performed classification effectively through its iterative and computational capabilities.

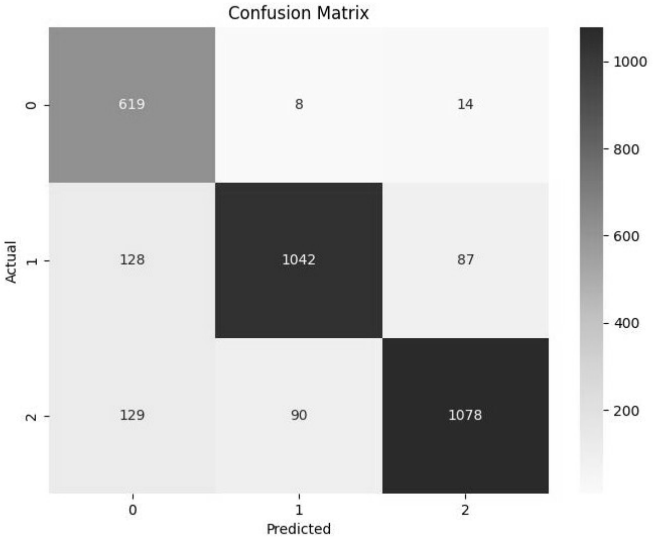


Fig. 96. Confusion Matrix of the Forearm Supination Exercise

For the Forearm Supination/Pronation Exercise Angle Classification, the CNN classifier's metrics demonstrate its ability to handle fine-grained angle classification. The model achieved a test accuracy of 85.73% after 47400 iterations over 15 epochs, with a batch size of 6 and a 20% validation split. The total execution time was 20.99 seconds, indicating the computational effort involved in this classification task.

Table 27. Metrics of Angle Classification for the Forearm Supination Exercise

CNN Classifier	Value
Number of iterations	47400
Number of Epochs	15
Batch size	6
Validation split	20%
Execution time	20.99s
Test accuracy	85.73%

4.5. Exercise Detection Experiment

The cross-exercise balanced accuracy matrix presented in Fig. 97 summarizes the performance of the CNN-based exercise detection system across eight rehabilitation exercises. Each row represents a model trained on a specific exercise, and each column shows its testing accuracy when evaluated on all other exercises. Diagonal elements correspond to the self-test accuracy of each exercise model, while off-diagonal values indicate how well a model trained on one exercise generalizes to other movement types.

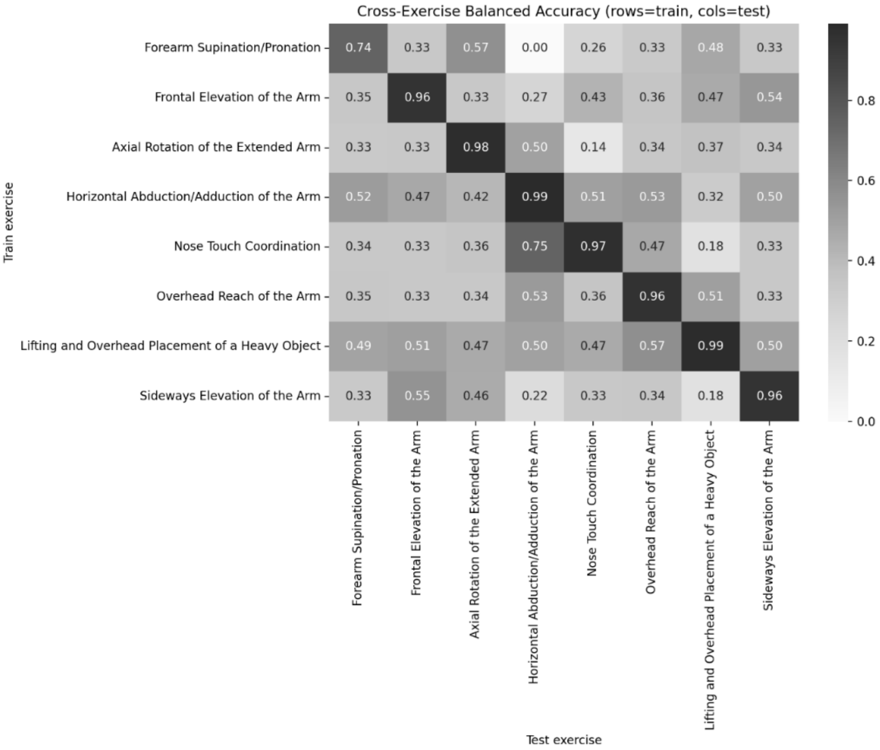


Fig. 97. Cross-Exercise Balanced Accuracy Matrix for CNN-Based Exercise Detection

For each exercise, individual datasets containing the angular relationships of the selected skeletal nodes were processed separately. Each dataset was divided into 70% training and 30% testing subsets based on unique participant–posture combinations. After training, the models were saved along with their preprocessing components (feature scalers, label encoders, and feature metadata), allowing them to be reused for evaluating unseen data or cross-exercise testing without retraining. During evaluation, each trained model was cross-tested on all exercises to assess its generalization across movement types, with predictions compared against true posture labels (Good, Incorrect 1, Incorrect 2). The resulting balanced accuracy matrix highlights both recognition capability and inter-exercise similarity patterns.

4.5.1. Results

The diagonal accuracies range from approximately 0.74 to 0.99, confirming that each CNN accurately recognizes its corresponding exercise when trained and tested on its own dataset. The mean self-recognition accuracy across all exercises was 0.92 ± 0.09, reflecting strong within-exercise consistency in angular motion patterns. Exercises such as Horizontal Abduction/Adduction of the Arm (0.99), Nose Touch Coordination (0.97), Overhead Reach of the Arm (0.96), and Sideways Elevation of the Arm (0.96) achieved the highest self-recognition scores, suggesting that these movements exhibit stable and distinctive angular trajectories across participants.

Moderate off-diagonal values reflect partial overlap between biomechanically similar movements. For example, Frontal Elevation of the Arm and Sideways Elevation of the Arm show mutual recognition around 0.57, as both involve single-arm lifting in a vertical plane with comparable shoulder and elbow angular trajectories (Angles 2–3). The distinction between the frontal and lateral planes helps explain minor confusion when spatial direction is not explicitly encoded. Similarly, Axial Rotation of the Extended Arm and Forearm Supination/Pronation exhibit cross-accuracies above 0.55, both relying on rotational motion about the upper limb's longitudinal axis (Angles 3–4), with limited shoulder displacement producing similar temporal angle profiles. In contrast, exercises with distinct kinematic structures, such as Lifting and Overhead Placement of a Heavy Object or Nose Touch Coordination, show minimal overlap with others (cross-accuracies typically below 0.35). Their multi-joint coordination across the elbow, shoulder, and trunk (Angles 2–5) generates unique angular signatures that the CNN distinguishes clearly from simpler one-joint actions.

Overall, exercises that involve similar joint movements, rotational axes, or motion ranges tend to exhibit moderate cross-recognition, whereas those characterized by distinct joint coordination patterns remain clearly differentiated. These quantitative findings indicate that the CNN models primarily learn and respond to underlying biomechanical relationships within the angular data, and that any observed misclassifications result from genuine movement similarity rather than model error.

4.6. Conclusion

Chapter 4 presented the development, deployment, and comprehensive evaluation of an AI-supported human posture diagnosis system that utilised the HTC Vive Tracker 2.0 as an accessible VR input device. The system's quantitative performance was assessed using upper-limb exercises, integrating RF and CNN models for both decision- and angle-based classifications. Both RF and CNN models demonstrated strong discriminative capability across multiple exercise scenarios, including frontal, lateral, and overhead elevations, horizontal abduction/adduction, axial rotation, and complex object manipulation. Both approaches consistently achieved test accuracies of 95.78%-100% for decision-level classification of posture quality or disorder, with RF models using as few as 10–20 trees (maximum depth 3) and CNNs converging within 11–38 epochs. Execution times for RF classifiers remained below 0.06 seconds, indicating high computational efficiency suitable for real-time feedback. CNN models, while requiring longer execution times (ranging from 11.39s to 46.75s depending on exercise complexity), also achieved perfect or near-perfect accuracy in all decision classification tasks. In angle-based classification tasks, the CNN achieved test accuracies ranging from 90.17% to 97.77% across exercises, demonstrating robust capacity for fine-grained motion analysis, albeit with minor misclassifications between similar compensatory patterns.

In addition, a cross-exercise evaluation was conducted to assess how well the CNN models generalized across different movement types and correctly identified the exercise being performed. Each exercise-specific CNN model was tested on the datasets of all other exercises, and the resulting balanced accuracy matrix showed high recognition rates for self-tests (ranging from 0.74 to 0.99). Moderate recognition overlap was observed between exercises with similar biomechanical structures, such as frontal and lateral arm elevations or upper-limb rotational movements. These results suggest that the observed confusion arises mainly from genuine physical similarities in movement patterns, rather than from limitations in the model's learning process. This confirms that each CNN effectively captures the unique angular characteristics that define its corresponding exercise.

The most significant discriminative features were systematically identified: for example, Angle 2 and Angle 3 (elbow flexion and shoulder alignment) for the frontal elevation of the arm, and Angle 5 (spinal tilt) for the overhead reach, were critical for accurate detection of compensatory movement and disorder classification. Feature importance analyses and error distributions indicated that the models could reliably distinguish between healthy and compensatory or pathological movement patterns using specific joint angle metrics. Quantitative analysis of the dataset, which included up to 85 samples per exercise class from 17 participants, indicated that the system maintained high sensitivity and specificity in classifying correct versus incorrect movements. The platform's performance was validated against gold-standard clinical interpretation metrics, with RF and CNN decision accuracies consistently matching or exceeding benchmarks for biomechanical assessment. Although some limitations were noted regarding angle-based misclassification and inter-individual variability, the overall findings establish that combining off-the-shelf VR hardware with advanced AI algorithms yields a cost-effective, scalable, and quantitatively reliable solution for human posture assessment in clinical and home-based settings. Further work could involve additional real-world clinical experiments to assess its in vivo potential in actual disease cases.

CONCLUSIONS

1. Peripheral input devices present various integration challenges, primarily related to device compatibility, latency, and the ability to provide a fully immersive simulation-like experience in virtual reality. They can enhance realism but introduce latency and synchronisation issues, leading to visual stutters that degrade the overall VR experience. Addressing these challenges requires adaptive techniques to minimize latency and improve system responsiveness while maintaining accuracy and immersiveness.
2. High-end motion capture systems offer precise tracking but are costly and require specially allocated, pre-conditioned room space. Off-the-shelf VR trackers are affordable and adaptable for exercise tracking, but they exhibit positional and angular drift, especially in extended sessions, high-speed, sudden, or impactful movements. Prior studies have shown the potential of machine learning-based motion classification in similar applications. Building on this, our research proposes a system for full-body tracking and human posture analysis based on using off-the-shelf VR systems.
3. To address integration challenges specific to peripheral input devices, four prediction methods have been proposed and evaluated for real-time data synchronisation between peripheral equipment (Concept-II) and the virtual reality. These methods predict missing or delayed input data to reduce perceived input latency and visual stuttering while maintaining rowing accuracy. Four prediction methods were used for the study: linear interpolation by position, extrapolation by speed, extrapolation by speed with correction, and the Kalman filter.
4. To ensure accurate motion tracking and classification, we propose a system that utilises HTC Vive tracker sensors positioned on key anatomical points to capture full-body movements. Joint nodes were identified for each exercise to calculate angles, which serve as input features for machine learning algorithms. The system employs adapted machine learning models—specifically Random Forest and Convolutional Neural Network architectures—to analyse joint angles, motion patterns, and compensatory behaviours to classify patient movement disorders based on the performed exercises.
5. The four prediction methods have been proposed, aiming to reduce latency, remove micro-stuttering, and improve data accuracy between peripheral equipment (Concept-II) and virtual reality. Linear interpolation had positive timing offsets averaging +64.7 ms but reduced the stuttering events from ≈ 482(111 ms) to ≈ 260 (39 ms). The Kalman filter further reduced the stutter

to ≈ 145 (53 ms) but introduced a ≈130 ms offset in timing. Extrapolation by speed eliminated stuttering but cumulatively produced a large negative timing ≈ of 6.8 seconds. The introduced correction parameter in Extrapolation by speed significantly reduced this timing offset to ≈ 29 ms, meaning the 250 m session in the virtual environment finished almost simultaneously with the peripheral input data while providing a stutter-free VR experience.

6. Across eight upper-limb exercises tested on the proposed framework. Both Random Forest and CNN models achieved decision-level accuracies between 95.78% and 100%, while CNN angle-based classifications ranged from 85.73% to 97.77%. A cross-exercise evaluation tested each CNN on all other exercise datasets, showing self-recognition accuracies between exercises. Moderate cross-recognition was found between Frontal Elevation of the Arm and Sideways Elevation of the Arm, with mutual accuracies around 0.57, likely caused by similar joint motion patterns in Angle 2 (shoulder elevation) and Angle 3 (elbow alignment). Lifting and Overhead Placement of a Heavy Object and Nose Touch Coordination had cross-recognition values below 0.35, indicating clear distinction. These results show that CNN models clearly separated most exercises, with only moderate overlap in similar movements.

SANTRAUKA

Įvesties delsimas, sinchronizacijos ir su judesiu susijusios problemos daro didelę įtaką VR programų kūrimui ir naudojimui. Šie iššūkiai dažnai kyla dėl VR valdymo technologijų apribojimų, todėl atsiranda kliūčių įtraukiančiai patirčiai sukurti ir taikyti mokslinėms taikomosioms programoms. Mokslininkai plačiai tyrinėja ir tobulina metodus, skirtus įvertinti, kas didina VR įtraukiamumą, atsižvelgiant į įvairius jutiminius stimulus ir naudotojo sąveiką [3, 4].

Atliktais tyrimais siekiama pagerinti įtraukiančią VR patirtį įvairiais aspektais, ypač kuriant į virtualiąsias aplinkas (VE) integruotas imitacijas. Sukurti imitacijas, veikiančias VR aplinkoje yra nemažas iššūkis, nes reikia rasti tinkamą įrangos integravimo būdą. Reikia paminėti, kad esamos VR sistemos pirmiausia remiasi integruotais valdymo metodais ir įrankiais, tokiais kaip „HTC Vive“ arba „Steam“ valdikliais, kad pagerintų naudotojo sąveiką virtualioje aplinkoje [5].

Nors šios sistemos yra dažniausiai veiksmingos, tačiau jos dažnai nesuteikia į imitavimą panašios patirties, kai jame naudojami valdikliai, kurie nepanašūs į minėtus. Ir atvirkščiai, integruojant išorinius periferinius valdiklius, pavyzdžiui, „Concept-II“ irklavimo treniruoklį, kyla unikalių iššūkių dėl nesuderinamo duomenų perdavimo greičio tarp komponentų. Dėl to dažnai atsiranda judėjimo netikslumų, vaizdas stringa ir naudotojas jaučia diskomfortą VR seanso metu. Šios ir kitų integracijos problemų sprendimas yra labai svarbūs norint atskleisti VR galimybes ir kitose srityse. Šioje disertacijoje siūlomi sprendimai, kaip būtų galima išspręsti tokias problemas. Tyrime siekiama sumažinti dėl duomenų vėlavimo atsirandantį vaizdo trūkčiojimą ir duomenų netikslumus, pasiūlant tris duomenų prognozavimo metodus, kad būtų užtikrintas sklandus ir tikslus duomenų pateikimas VR sistemai. Ekstrapoliacijos pagal greitį metodas taikant korekciją parodo gerai sukurtos sistemos rezultatus.

Nepaisant įrodytos VR naudos terapijoje ir reabilitacijoje, jos pritaikymas klinikinėje aplinkoje tebėra ribotas dėl kainos, prieinamumo ir gydytojų technologijų kompetencijos [1]. Daugelyje mokslinių tyrimų tiriami nestandartiniai VR įvesties įrenginiai siekiant nustatyti, ar jie galėtų būti nebrangi alternatyva tradiciniams sprendimams [6, 7]. Bendros šių tyrimų išvados, palyginus su brangiai kainuojančiais sprendimais, rodo, jog reikšmingai skiriasi tikslumas, ypač atliekant sudėtingus judesius arba, kai sekimo priemonės nėra matomumo zonoje.

Atsižvelgiant į tai, šioje disertacijoje pristatoma, kaip naudojant nebrangias VR sistemas galima registruoti žmogaus judesius ir taikyti CNN ir RF mašininio mokymosi modelius biomechaninei analizei. Siūlomoje sistemoje naudojami „HTC-Vive“ sekimo įrenginiai ir giliojo mokymosi judesių identifikavimo modeliai žmogaus laikysenos sutrikimams nustatyti. Siūlomas metodas remiasi iš anksto nustatytais scenarijais, pagal kuriuos galima klasifikuoti taisyklingus ir netaisyklingus judesius. Tai yra patogi priemonė sprendimams priimti ir fizinio lavinimo veiksmingumui vertinti pagal iš anksto nustatytus atvejus.

Šios disertacijos indėlis – naujas požiūris į integracijos iššūkius VR sistemoms, VR sistemų įtraukumo didinimą ir VR pritaikymą žmogaus laikysenai analizuoti. Disertacijoje siūlomi metodai suteikia galimybių spręsti periferinių valdiklių integravimo į VR aplinkas problemas. Taip pat, reabilitacijos arba fizinio lavinimo pratimų vertinimo tikslumas parodo, kad VR sekimo įrenginiai galėtų būti praktiškai prtaikomi klinikose.

Disertacijos objektas

Disertacijos objektas – virtualiosios realybės technologijų taikymas laikysenos sutrikimams nustatyti ir kategorizuoti, taikant prognozavimo metodus bei periferinių įrenginių įntegravimas į VR sistemas.

Disertacijos tikslas

Pagerinti žmogaus laikysenos judesių identifikavimą ir kategorizavimą naudojant standartinius VR įvesties įrenginius ir palengvinti sklandų periferinių įvesties įrenginių pritaikymą virtualiosios realybės sistemose taikant duomenų prognozavimo metodus.

Disertacijos uždaviniai:

1. išanalizuoti dabartinę periferinių įvesties įrenginių integravimo į virtualiąsias aplinkas ir dabartinę žmogaus judesių sekimo metodų, skirtų žmogaus laikysenai ir judesiams klasifikuoti, situaciją;
2. išanalizuoti metodus, leidžiančius sumažinti periferinių įvesties įrenginių integravimo problemas, susijusias su įvesties vėlavimu, duomenų tikslumu ir vaizdo trūkčiojimu virtualiose aplinkose;
3. sukurti metodus, leidžiančius sumažinti periferinių įvesties įrenginių integravimo problemas, susijusias su įvesties vėlavimu, duomenų tikslumu ir vaizdo trūkčiojimu virtualioje realybėje;
4. sukurti metodus realybės technologijos ir jutiklių taikymui pratimų sekimui ir klasifikavimui;
5. įvertinti pasiūlytų metodų, skirtų integruoti periferinius įrenginius virtualiojoje realybėje, efektyvumą;
6. įvertinti pasiūlytų pratimų sekimo ir klasifikavimo metodų, naudojančių virtualiosios realybės technologiją ir sekimo jutiklius, efektyvumą.

Mokslinis naujumas

1. Pasiūlytas greičiu pagrįstos ekstrapoliacijos korekcijos metodas leidžia spręsti tikslumo, sinchronizacijos ir imersijos (įsitraukimo) iššūkius, eliminuojant mikrotrūkčiojimus VR sistemose ir kartu išlaikant didelį tikslumą naudojant integruotus periferinių įrenginių valdiklius.
2. Siūloma reabilitacinių pratimų klasifikavimo sistema leidžia tiksliai klasifikuoti aštuonis pratimus, nustatyti atliktą pratimą ir įvertinti asmens

laikysenos būklę. Metodas naudoja VR sekiklius, išdėstytus pagrindiniuose anatominiuose taškuose laikysenai sekti, kiekvienam pratimui identifikuotus mazgas sąnarių kampams skaičiuoti, CNN pagrįstos architektūros pratimų duomenims klasifikuoti ir aptikimo metodą atliktam pratimui identifikuoti.

Praktinė vertė

Realizuoti ir ištirti keturi algoritmai, pagrįsti interpoliacijos ir ekstrapoliacijos principais dviems svarbioms problemoms spręsti: mikrotrūkčiojimo ir duomenų netikslumo dėl įvesties delsos, atsirandančios dėl duomenų perdavimo spartos skirtumo tarp išorinių periferinio valdymo įrenginių ir VR sistemų. Šie algoritmai skirti užtikrinti sklandžius ir tikslius VR seansus mažinant delsos problemas. Tyrime pristatoma virtualaus irklavimo korpuso padėties prognozavimo, naudojant įvairius metodus, svarba. Kai kurie pasiūlyti metodai teikia pirmenybę laiko ir padėties tikslumui, o kiti nukreipti į tai, kad VR patirtis būtų užtikrinta be trūkčiojimo. Pasiūlytas metodas integruoja greičio parametrų derinimą su nuolatiniu koregavimu, užtikrinant duomenų tikslumą ir VR patirtį be trūkčiojimų. Pasiūlyti metodai suteikia vertę tiems, kurie integruoja periferinius valdiklius į VR sistemas.

Galimybė sekti kūno padėtį – nuo paprastų pozų iki sudėtingų, naudojant VR sekimo įrenginius, suteikia praktinį sprendimą, skirtą pacientų judėjimo sutrikimams nustatyti. Sistema atkuria paciento judesius naudodama trimatį modelį, kuris sukuriamas pagal informaciją gautą iš virtualiosios relybės sensorių. Panaudoti CNN ir RF metodai, taikomi siekiant nustatyti žmogaus laikysenos anomalijas, gali pagreitinti analizės procesą ir gali būti panaudoti lyginamajai analizei tiriant pacientų progresą.

Disertacijos teiginiai

1. Prognozavimas naudojant tiesinę interpoliaciją pagal padėties duomenis duoda pakankamai tikslių rezultatų, tačiau negali panaikinti trūkčiojimo problemų, atsiradusių dėl skirtingos duomenų registravimo ir perdavimo spartos tarp periferinių ir VR įrenginių.
2. Prognozavimas naudojant ekstrapoliaciją pagal greitį išsprendžia trūkčiojimo problemą, tačiau atsiranda netisklumų su tarp įrenginyje užregistruoto ir VR aplinkoje užregistruoto nuirkluoto atstumo.
3. Naudojant ekstrapoliaciją pagal greitį ir korekcijos metodą, į prognozavimą įvestas pastovus modifikatorius leidžia gauti tikslius duomenis apie atstumą ir laiką. Kartu jis užtikrina, kad sesijos vyktų be trūkčiojimų. Iš pasiūlytų tai yra geriausias sinchronizacijos problemų sprendimas, taikomas integruojant periferinius valdiklius į VR.
4. Galima patikimai naudoti VR sekimo sistemas ir mašininio mokymosi metodus laikysenos sutrikimams nustatyti po fizinių pratimų atlikimo. Siūlomi VR sistemos metodai, naudojantys viso kūno sekimą, gali tiksliai aptikti laikysenos sutrikimus atlikus fizinius pratimus. CNN pasiekė didžiausią sprendimų priėmimo tikslumą nustatydama, ar dalyvis turi

laikysenos sutrikimų, ar ne. CNN metodus taip pat galima taikyti pacientų judesių modeliams klasifikuoti ir netaisyklingiems judesiams nustatyti.

VIRTUALIOJE REALYBĖJE NAUDOJAMŲ ĮVESTIES ĮRENGINIŲ DUOMENŲ APDOROJIMO METODŲ ANALIZĖ

Įprasti VR įvesties įrenginiai

Įsitraukimo (angl. *immersiveness*) laipsnis yra daugiau susijęs su tikroviškumu nei su VR produkto kokybe, o tai, galima teigti, priklauso nuo suvokimo autentiškumo ir jutiminio įsitraukimo, kurį suteikia VR aplinka [8]. Specialiai VR technologijai sukurtų valdiklių integravimas yra geras sprendimas, duodantis praktinių ir pritaikomų rezultatų. Kita vertus, standartinių VR priemonių (pvz. „HTC Vive“ sekimo įrenginių) taikymas skirtingų techninių priemonių atkūrimui gali sumažinti įsitraukimo jausmą, nes naudotojai turi prisitaikyti prie valdymo schemos, kuri tiesiogiai neimituoja realaus pasaulio sąveikos. Taigi, norint pagerinti naudotojų sąveiką ir imitavimo tikroviškumą, būtina priderinti nestandartinius VR valdymo įrenginius [9].

Viename iš tyrimų [5], kaip ir šioje diesrtacijoje, nagrinėjamas irklavimo imitavimas panaudojant „HTC Vive“ sekimo jutiklius, skirtus VR. Jutikliu, pritvirtintu prie irklavimo rankenos galima tiksliai sekti ir fiksuoti irkluotojo judesius. Viena iš svarbiausių šio tyrimo siekių – techninės problemos, susijusios su irklavimo treniruoklio kalibravimu. Nors integravimas iš esmės buvo sėkmingas, tačiau nustatant reikėjo tikslaus kalibravimo, o tai turėjo įtakos jo panaudojimui. Sistema reguliuoja greitį ir kryptį remdamasi informacija iš „HTC Vive“ sekimo įrenginių. Tačiau reikia paminėti, kad šį metodą buvo galima apgauti, nustačius sekimo įrenginį taip, kad būtų galima imituoti greitą judesį. Tokiu būdu galima pasiekti didelį greitį tam skiriant mažiau pastangų. Šioje disertacijoje siūlomame sprendime naudojami neapdoroti prietaiso duomenys, t. y. renkami duomenys tik iš „Concept-II “ irklavimo treniruoklio ir taikomi prognozavimo metodai. Šis metodas užtikrina sistemos vientisumą, nes pasikliaujama ne tik VR sekimo įrenginiais, o ir techninės priemonės „Concept-II“ pateikiama informacija.

Viename iš tyrimų buvo vertinamas įvairių rankinių įrankių tinkamumas neurochirurgijos mokymams virtualioje aplinkoje, panaudojant įprastines VR įvesties technologijas [15]. Iš visų prietaisų didžiausią vidutinį tikslumą atskleidė VR tušinukas – 63 %, po jo buvo standartinis valdiklis – 60,34 %. Prasčiausiai pasirodė kraniotomas – pasiekta tik 53,60 %. Buvo nustatytas pastebimas kraniotomo drebėjimas, kurį lėmė „HTC Vive“ sekimo įrenginio ir bazinių stočių konfigūracija, dėl to sumažėjo sekimo stabilumas.

Šiuose ir kituose tyrimuose pabrėžiami techniniai apribojimai, susiję su įprastinių VR sekimo įrenginių pritaikymu fiziniams prietaisams. Išanalizuota teorinė literatūra ir atvejų tyrimai parodė, kad nestandartiniai VR įvesties įrenginiai leidžia įgyti įtraukiančios ir interaktyvios patirties, dėl to pasirenkami kompromisai tarp patogumo, tikslumo ir pritaikomumo. Paminėtina tai, kad pasikartojančios problemos, tokios kaip ribotas tikslumas atliekant smulkiosios motorikos užduotis, jautrumas akloms zonoms ir trūkčiojimui bei realaus „haptinio“ grįžtamojo ryšio trūkumas, riboja jų tikroviškumą ir veiksmingumą didelio tikslumo srityse. Todėl šioje disertacijoje pagrindinis dėmesys skirtas periferinių įrenginių integracijai siekiant išspręsti problemas, susijusias su VR sistemų ir periferinių įrenginių veikimo skirtumais.

Judesio sekimo įvesties įrenginiai

„HTC Vive Tracker“ pristatomas kaip plačiai paplitęs sekimo įrenginys, leidžiantis įgyvendinti sekimą šešių žingsnių (6 DoF) programoms, skirtoms fizinei terapijai ir įtraukiam mokymui [19]. Nors jis gerai veikia, tačiau kontroliuojamoje aplinkoje yra netikslumų, tokių kaip padėties poslinkis, atsirandantis atliekant greitus ar ilgalaikius judesius [19].

Signalo (nuo judesio iki jo atvaizdavimo VR akiniuose) vėlavimas taip pat mažina įtraukumą; nors vidutinis vėlavimas svyruoja nuo 21 iki 42 ms, prognozavimo metodai gali sumažinti šias vertes iki 2–13 ms [20]. Pažangūs metodai, pavyzdžiui, gilieji aido būsenų tinklai federacinėse mokymosi sistemose, pasirodė esantys veiksmingi išlaikant sekimo stabilumą nestabiliomis tinklo sąlygomis [21]. Platesnis žymekliais ir nežymekliais grindžiamų sistemų palyginimas atskleidžia, kad nors „Vicon“ sistema užtikrina didesnį tikslumą, techninė įranga ir jos parengimas mažina naudojimo galimybes. Tuo metu nenaudojančios žymeklių siūlo natūralų judėjimą, tačiau atsiranda sekimo artefaktų ir drebėjimo [22]. Empiriniai vertinimai, lyginant HTC „Vive“ su „Vicon“, rodo, kad „Vive“ pasiekia 0,58 ± 0,89 cm padėties ir 1,46 ± 0,62 ° sukimo RMSE [23]. Nors 30 ° ir 60 ° sulenkimo kampu padėtis yra tiksli, neatitikimai 45 ° kampu ir kryžkaulio pasukimo pervertinimas rodo, kad reikia koreguoti algoritmus. Tokie trūkumai kaip drebėjimas ir mažesnis dažnis (90 Hz, palyginti su „Vicon“ 100–200 Hz) dar labiau riboja patikimumą atliekant didelės spartos užduotis.

Su sekimo tikslumu susiję disertacijos tyrimai (žr. III skyrių) parodė, kad „Vive Tracker“ RMSE rodikliai yra pakankamai geri ir statinėmis, ir dinaminėmis sąlygomis, todėl „HTC Vive“ ir jų sekimo jutikliai yra tinkami laikysenai sekti.

Periferiniai įvesties įrenginiai skirti VR

Šiame tyrime VR periferiniai įvesties įrenginiai – tai fizinė įvesties įranga, kuri iš pradžių nebuvo sukurta virtualiajai realybei, bet buvo pritaikyta, kad veiktų VR sistemose. Vėlavimas yra vienas iš esminių iššūkių integruojant periferinius įvesties įrenginius. Mokslinėje literatūroje teigiama, kad įvairiose mokslinių tyrimų srityse, įskaitant kompiuterių ir elektrotechnikos inžineriją, skirtingų dažnių sričių integravimas įgyvendinamas taikant interpoliavimo ir ekstrapoliavimo metodus. Pavyzdžiui, signalų apdorojimo srityje interpoliacija ir ekstrapoliacija paprastai naudojama trūkstamiems signalams prognozuoti iš esamų signalų pavyzdžių [24]. Tikėtinos problemos, susijusios su išorinio valdymo integravimu į VR, gali lemti patirties kokybės (QoE) pablogėjimą dėl vėlavimo tarp valdiklio ir VR sistemos [25]. Vėlavimas reiškia laiką, per kurį pranešimas pasiekia sistemą [26]. Šis parametras dažnai pabrėžiamas kaip reikšmingas VR taikomųjų programų apribojimas [26, 27]. Priimtina riba VR sprendimams ne ilgesnė kaip 20 milisekundžių [27]. Valdymo įvesties vėlavimai gali lemti ekrano atnaujinimo vėlavimą, kuris gali svyruoti nuo 0 iki 30 ms. Mažo vėlavimo valdymo sistemos yra labai svarbios siekiant išlaikyti veiksmingas ir efektyvias operacijas VR modeliavime. Tai ypač svarbu integruojant išorinius valdymo įrenginius į VR, kai net ir nedideli vėlavimai gali gerokai sutrikdyti naudotojo patirtį.

Svarbus Iskandar'o ir kt. darbas [27] apima išsamią algoritmų prognozavimo, skirtų vėlavimo poveikiui neutralizuoti, analizę. Šie algoritmai skirstomi į penkis tipus: interpoliaciją, ekstrapoliaciją, filtrais pagrįstus metodus, neuroninių tinklų metodus ir kiti. Interpoliavimo metodais apskaičiuojamos vertės tarp žinomų duomenų taškų, o ekstrapoliavimo metodais prognozuojamos vertės už esamo diapazono ribų. Filtrais pagrįsti metodai, pavyzdžiui, Kalmano filtras, išplėstinis Kalmano filtras, neakcentuotas Kalmano filtras ir Wienerio filtras, naudojami padėties duomenims stabilizuoti. Neuroninių tinklų metodai, įskaitant perceptroninius neuroninius tinklus ir atgalinio sklidimo tinklus, adaptyviai mokosi iš naudotojo įvesties modelių. Šiame darbe naudojamos Iskandaro ir kt. [27] pateiktos prognozavimo strategijos, ypač daug dėmesio skiriant interpoliacijos ir ekstrapoliacijos metodams. Tyrimais prisidedama prie vėlavimo valdymo strategijos, apibūdinamos kaip „registravimas nepaisant vėlavimo“ ir užtikrinama, kad naudotojo veiksmai išliktų tiksliai sinchronizuoti su VR grafinės aplinkos atnaujinimu, net ir esant vėlavimo sąlygoms.

The Bluefruit BLE sistema atskleidė palyginti nedidelį atsako vėlavimą, kurio tipinė vertė buvo 80 ms. *Elite Axiom* sistema pasižymėjo gerokai didesniu ir labiau kintančiu atsako vėlavimu, kuris siekė iki 2000 ms. Šis didelis vėlavimas pirmiausia buvo susijęs su sistemos stabdymo mechanizmu. Tyrime pabrėžiama, kad norint užtikrinti sklandžią VR sąveiką, vien tik didelio duomenų perdavimo dažnio nepakanka. Išlaikant vidutinį 80 FPS kadrų per sekundę dažnį ir 25–30 ms uždelsimą, kalnų dviračio VR imitaciniame modelyje išvengiama bendrų uždelsimo problemų, paprastai susijusių su periferiniais VR įvesties įrenginiais, nes duomenų perdavimo dažnis yra optimizuotas. Bėgimo takelio sistemos „Hex-Core-MK1“ tyrimai parodė, kad vidutinis vėlavimas tarp naudotojo judesio ir VR atsako yra 36 ms. Tyrime nepateikti išsamūs paklaidų rodikliai, kurie gali būti labai svarbūs konkrečioms VR taikomosioms programoms, reikalaujančioms tikslios kūno koordinacijos sekimo. Nepaisant to, „Hex-Core-MK1“ išlieka vertas dėmesio pavyzdys, kaip periferiniai VR įvesties įrenginiai gali būti veiksmingai pritaikyti įtraukiančiai VR patirčiai. Tyrimas su „Myo armband“ parodė problemą, susijusią su nuoseklaus gestų atpažinimo tikslumo išlaikymu VR. Nors šiame tyrime nepateikti išsamūs delsos rodikliai, jis parodė, kad gestams valdyti VR gali būti naudojama periferiniams įrenginiams tam, kad sąveika būtų sklandi.

Virtualiosios realybės įvesties duomenų apdorojimo metodai

Duomenų apdorojimas VR programose yra labai svarbus norint tiksliai parodyti naudotojo veiksmus virtualioje aplinkoje, išvengti vėlavimo efekto ir išlaikyti įsitraukimą. VR įvesties duomenys gali būti gaunami iš judesių sekimo, gestų atpažinimo arba įvairių, sąveika pagrįstų įrenginių, kuriems visiems reikia specialaus duomenų apdorojimo etapo, kad juos būtų galima perkelti į kompiuterio aplinką. Prognozavimo metodai, įskaitant interpoliaciją ir ekstrapoliaciją, paprastai naudojami naudotojo veiksmams įvertinti, kai negalima laiku gauti duomenų, taip užtikrinant nenutrūkstamą judėjimą ir sąveiką. Filtravimo metodai taikomi siekiant sumažinti jutiklio triukšmą ar pablogėjus signalui. Paminėtina tai, kad mašininiu mokymusi pagrįsti metodai naudoja neuroninius tinklus veiksmams prognozuoti, gestams atpažinti ir jutiklio duomenims tikslinti. Dažnai hibridinės strategijos sujungia šiuos metodus, kad būtų optimizuotas veikimas įvairiais atvejais. Virtualios ir realaus pasaulio sąveikos laikinis pobūdis reiškia, kad VR visada apima tam tikrą vėlavimo laipsnį [41]. Labai svarbi vieta tenka vizualiniam įsitraukimui ir įvesties įrenginių kokybei bei tikslumui. Tokios, jau parengtos arba užsakomos judesių fiksavimo sistemos, yra taikomos sveikatos priežiūros, kariuomenės, švietimo ir sporto srityse [42, 43]. Sistemų suderinamumas, vėlavimas, kalibravimas ir duomenų interpretavimas yra nuolatiniai iššūkiai, todėl vėlavimo valdymas yra labai svarbus, kad nesutrikdytų įtraukimo ar nesukeltų judesio klaidų. Tokios strategijos kaip ribinė, kompensacinė ir numatančioji sinchronizacija, pasirenkamos atsižvelgiant į taikomąją programą, įvesties įrenginius ir naudotojo patirties tikslus.

Iš prognozavimo duomenų apdorojimo metodų, interpoliacija ir ekstrapoliacija įvertina nežinomus duomenų taškus. Keliuose tyrimuose vėlavimas VR aplinkoje nagrinėjamas taikant šiuos metodus [26, 44]. Nagrinėjami tiesinės ir polinominės regresijos metodai, iš kurių tiesinė regresija yra efektyvi, tačiau ne tokia tiksli dinamiškiems judesiams. Kvadratinė regresija efektyviau fiksuoja netiesiškumus, o kubinė regresija, nors ir lanksti, dažnai atmetama dėl jautrumo triukšmui. Šie bendrieji modeliai rodo pamatinį prognozavimo modeliavimo VR efektyvumą [26, 44]. Pažangiose taikomosiose programose gali prireikti pritaikymo ir svertinių daugialypių įvesčių. Šioje disertacijoje regresijos modeliai adaptuoti VR duomenims ir buvo patvirtinti eksperimentiškai. Regresija išlieka vertinga judesiams prognozuoti net ir atsiradus hibridiniams ir mašininio mokymosi modeliams. Labai svarbus yra realaus laiko kriterijaus išpildymas, įskaitant vėlavimą ir skaičiavimo apkrovą. Virtualų irklavimą aprašančiuose darbuose nustatyti atstumo netikslumai ir trūkčiojimai dėl asinchroninės įvesties. Interpoliavimas pagal padėtį užtikrino tikslumą, bet sukėlė trūkčiojimą, o ekstrapoliavimas pagal greitį sukėlė dreifą. Koreguotas ekstrapoliacijos metodas sumažino laiko paklaidas ir pagerino judėjimo tęstinumą [45].

Filtrais pagrįsti duomenų koregavimo metodai patikslina įvestį iš įrenginių, pašalindami triukšmą, sumažindami drebėjimą ir ištaisydami dreifą ar vėlavimą, o tai ypač svarbu naudojant asinchroninę aparatinę įrangą. Duomenims stabilizuoti realiuoju laiku naudojami paprasti išlyginimo metodai, pavyzdžiui, eksponentinis slenkamasis vidurkis (EMA) ir Kalmano filtrai, o prireikus – hibridinės filtravimo strategijos, kuriose derinami keli metodai [47–50]. Mašininio mokymosi modeliai, tokie kaip LSTM, CNN ir hibridinės architektūros, fiksuoja jutiklių duomenų laikines priklausomybes, kad būtų galima numatyti veiksmus, atpažinti gestus ar įvertinti fizines savybes [52–56]. Šie metodai yra pagrįsti duomenimis ir adaptuojami, tačiau jiems gali prireikti daug skaičiavimo išteklių ir didelių sužymėtų duomenų rinkinių.

Nepaisant sprendimų įvairovės, literatūroje mažai dėmesio skiriama periferinių VR įvesties įrenginių tyrimams – dauguma sprendimų grindžiami standartinėmis konfigūracijomis ir nešiojamaisiais VR įrenginiais. Ši disertacija praplečia tokius tyrimus, eksperimentiškai patvirtindama prognozavimo funkcijas ir korekcijos metodus, pritaikytus periferinei VR įvesties įrangai, transformuojančius fizines sąveikas į virtualius veiksmus su minimaliu vėlavimu ir dideliu VR atsako tikslumu.

Laikysenos stebėsenos, analizės ir atpažinimo metodai taikant VR priemones

Sveikatos priežiūros srityje kineziterapijos pratimai, įskaitant aktyvius judesių diapazono judesius, pavyzdžiui, ištiesimą, pakėlimą, sulenkimą, pasukimą, raumenų jėgos ir ištvermės lavinimą – yra labai svarbūs pacientams, sveikstantiems po insulto. Pacientų fizioterapijos pratimų atlikimas reikalauja daug laiko, pastangų ir išlaidų [59], todėl siekiant atkurti viršutinių galūnių motoriką, pusiausvyrą, eiseną, laikyseną ir vaikščiojimą [60], taikyta telereabilitacija, kad pacientai galėtų gydytis nuotoliniu būdu [61], imtasi kompiuterinėmis technologijomis ir VR paremto gydymo. Siekiant mažinti sveikatos priežiūros išlaidas, vis dažniau naudojamas sveikatinimas namuose, ypač nuotolinės sveikatos priežiūros srityje [62, 63]. Reikia pasakyti, kad išlaikyti motyvaciją ir palaikyti kokybinę medicininę priežiūrą ir toliau išlieka iššūkiu. Funkcinio atsigavimo stebėsenos metodai, pavyzdžiui, po insulto, yra labai svarbūs siekiant užtikrinti pratimų užbaigimą ir pacientų motyvaciją. Šių pratimų kokybė paprastai vertinama analizuojant krūtinės ląstos kryptį, klubo ir kelio sąnarių pasisukimą. Judesių stebėjimo technologijos dėl savo prieinamumo ir kompaktiškumo yra tinkamos klinikose, sporto salėse ir namuose. Paminėtina tai, kad laikysenos stebėjimas taip pat būtinas sveikiems asmenims, ypač tiems, kuriems kyla rizika dėl netinkamų darbo įpročių ir ergonomikos [64, 65]. Reabilitacijos sistema, kurioje žaidybinė virtuali aplinka integruota su viršutinių galūnių reabilitacijos technologijomis, užtikrina interaktyvų požiūrį, didina pacientų motyvaciją ir reabilitacijos efektyvumą [66].

Neapdorotiems jutiklių duomenims rinkti, siekiant stebėti žmogaus fizinę veiklą, taikomi įvairūs metodai. Dėvimi jutikliai dažnai naudojami klinikiniuose tyrimuose [67, 68], o RGB-D jutikliai, pavyzdžiui, „Kinect“, siūlo praktines judesių analizės priemones reabilitacijos aplinkoje, ypač kai nėra galimybės naudoti žymekliais pagrįstų sistemų [69–75]. Žmogaus judesių atpažinimo klausimai sprendžiami taikant tradicinius algoritmus (požymių išskyrimas ir atitikimas) arba giliuoju mokymusi pagrįstus metodus, o pastarieji, objektų požymių mokosi neuroniniais tinklais [76, 78–81]. Daugelis šiuolaikinių metodų remiasi gylio jutikliais skeleto duomenims rinkti. Tikslus sąnarių judesių, kampų ir krypčių nustatymas yra labai svarbus reabilitacijai. Sukurta keletas laikysenos įvertinimo metodų ir tam skirtų žaidimų, tačiau dauguma sistemų orientuojasi į skeleto taškų išskyrimą, o ne į sąnarių judesių, krypties ir kampo nustatymą [79]. Tokių metodų, kaip „Kinect V2“ naudojimas funkciniam pasiekiamumo testui [80] ir BLSTM-NN veiksmų atkūrimo vertinimui [81], atskleidžia jutiklių ir neuroninių modelių naudojimą.

Statistiniai laikysenos atpažinimo metodai remiasi matematiniais modeliais ir rankiniu būdu nustatytais požymiais, o judesių fiksavimo ar pozos įvertinimo duomenims taikomi tokie klasifikatoriai kaip SVM, KNN ar sprendimų medžiai [82]. Pavyzdžiui, naudojant „OpenPose“, skeleto raktiniams taškams išgauti ir pratimams klasifikuoti naudojant SVM, pasiektas 96,6 % tikslumas. Kitame tyrime lyginamos IMU ir „Kinect V2“ sistemos, skirtos laikysenai analizuoti. Nustatyta, kad IMU užtikrina geresnį sąnarių kampų tikslumą taikant jutiklių sintezės algoritmus, o „Kinect“ geriau atkuria viso kūno skeletą, tačiau paminėtina tai, kad kiekviena sistema turi savų trūkumų [83].

Mašininiu mokymusi grindžiamas laikysenos atpažinimas remiasi anotuotais duomenų rinkiniais ir prižiūrimo arba gilaus mokymosi architektūromis. Atsitiktinio miško (angl. *Random Forest, RF*) modeliuose naudojami tokie požymiai kaip išmatuotų kampų vidurkis, standartinis nuokrypis ir diapazonas, kurie parenkami taip, kad apibendrintų duomenis išlaikydami esmines jų savybes. RF modeliai būna apmokyti ir validuojami pagal duomenų pavyzdžius, jais pasiekiamas didelis tikslumas statinėms arba apribotoms pozoms ir užtikrinamas skaičiavimo efektyvumas bei interpretacijos galimybės [87–105]. CNN pagrįstos sistemos apdoroja struktūrizuotus įvesties duomenis, tokius kaip sąnarių koordinatės ar kampų pokyčiai, tiesiogiai mokosi konkrečiai laikysenai būdingų erdvinių konfigūracijų ir pasiekia didesnį tikslumą, nors ir su didesnėmis skaičiavimo pastangomis [85, 94]. Tyrimai rodo, kad CNN metodai yra ypač veiksmingi atskiriant subtilius laikysenos nukrypimus ir klasifikuojant ergonomines rizikas.

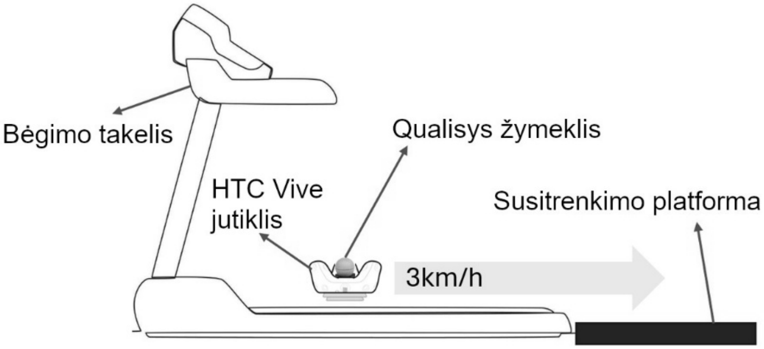
Hibridiniai metodai integruoja taisyklėmis pagrįstą logiką, statistinius modelius ir mašininio mokymosi algoritmus į bendras sistemas, kad būtų pašalinti atskirų metodų trūkumai, taip pagerindami atpažinimo tikslumą ir kontekstinį supratimą [92]. Pavyzdžiui, automatizuotuose „Kinect“ pagrįstuose vertinimo metoduose *fuzzy* logikos išvados derinamos su giliojo mokymosi modeliais, kad būtų galima klasifikuoti ergonominę riziką realiuoju laiku, taip parodant daugiaparadigminių sistemų privalumus.

Apibendrintai tariant, mokslinėje literatūroje teigiama, kad ir klasikiniai statistiniai metodai, ir pažangūs neuroniniai tinklai gali užtikrinti aukšto tikslumo vertinimą ir atpažinimą, tačiau geriausi sprendimai priklauso nuo konteksto, turimos aparatinės įrangos ir reikalavimų našumui, patikimumui ir tikslumui.

„HTC VIVE TRACKER“ JUTIKLIO TIKSLUMO VERTINIMAS

Siekiant įvertinti „HTC Vive Tracker (V2.0)“ erdvinio sekimo tikslumą, buvo atliktas papildomas eksperimentinis tyrimas, lyginami „HTC Vive“ sistemos registruojami poziciniai duomenys su „Qualisys“ optinio sekimo sistema, kuri buvo naudota kaip etalonas. Šis tyrimas patvirtino „ HTC Vive Tracker“ matavimų patikimumą esant vienodam judėjimui ir pagrindė VR jutiklių taikymo galimybę kuriamoje sistemoje.

Eksperimentui buvo naudojamas bėgimo takelis, judantis vienodu 3 km/h greičiu. Ant judančios juostos buvo padėtas „Vive Tracker“, o jo viršuje, centre, pritvirtintas „Qualisys“ infraraudonųjų spindulių žymeklis, kad abi sistemos registruotų tą patį fizinį tašką erdvėje. Jutikliai judėdavo bėgimo takeliu vienodu greičiu, kol jutiklis pasiekdavo takelio pabaigą (žr. 98 pav.) ir atsitrenkdavo į greta esančią platformą. Po kiekvieno bėgimo, sekiklis rankiniu būdu buvo grąžinamas į takelio pradžią sesijai pakartoti.



98 pav. Eksperimento iliustracija

„HTC Vive Tracker“ jutiklio tikslumo įvertinimas atliktas lyginant jo padėties sekimą su judesio fiksavimo sistemos „Qualisys“ padėties sekimu trimis eksperimentinėmis sąlygomis: visa signalo trukmė (įskaitant visas fazes, pavyzdžiui, judesį ir smūgį), atskiri švarūs judesio segmentai ir smūgiui būdingi intervalai. Klaidų rodikliai, įskaitant MSE, RMSE, MPE ir MAPE, buvo apskaičiuoti visoms trims erdvinėms ašims (X, Y ir Z) siekiant kiekybiškai įvertinti padėties sekimo nuokrypius (žr. 28, 29, 30 lenteles).

28 lentelė. Pozicijos sekimo paklaidos, kai „HTC Vive tracker“ ir „Qualisys“ sistema X, Y, Z ašyse veikia viso judesio signalo intervale

Klaida	**x- ašis**	**y- ašis**	**z- ašis**
MSE	0,47	3,43	0,52
RMSE	0,69	1,85	0,72
MPE	215,06	190,79	1128,5
MAPE	2,15	1,92	30,35

29 lentelė. Pozicinės sekimo klaidos, kai „HTC Vive“ sekimo įrenginys ir „Qualisys“ sistema X, Y, Z ašyse naudoja švarų (apkirptą) judesio signalą

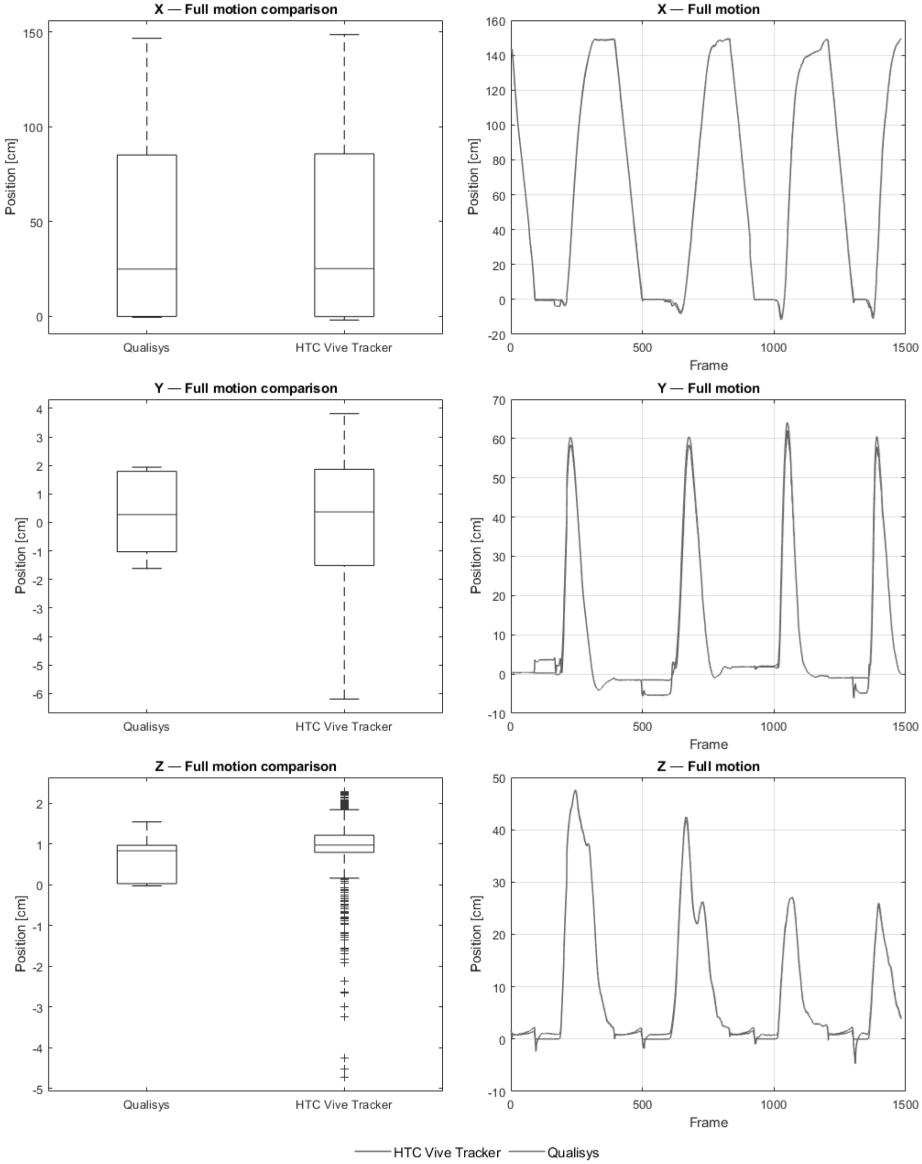
Klaida	**x-ašis**	**y-ašis**	**z-ašis**
MSE	0,67	0,01	0,11
RMSE	0,82	0,09	0,34
MPE	2,71	2,06	19,7
MAPE	0,03	0,03	0,24

30 lentelė. Smūgių metu gautų signalų padėties sekimo paklaidos tarp „HTC Vive Tracker“ ir „Qualisys“ sistemos X, Y, Z ašyse

Klaida	**x-ašis**	**y-ašis**	**z-ašis**
MSE	0,83	8,87	1,04
RMSE	0,91	2,98	1,02
MPE	492,83	260,02	2355,1
MAPE	4,93	5,65	72,07

Rezultatai rodo, kad didžiausi neatitikimai yra ties Z ašimi, kur sekiklio MAPE pasiekė itin dideles vertes, ypač smūgių metu (pvz., 1128,5 % visiško judėjimo metu ir 2355,1 % smūgių metu), daugiausia dėl didelių nuokrypių, kai sistema patiria staigius trikdžius. Priešingai, X ir Y ašių MAPE paprastai nebuvo daugiau kaip 3 %, o tai rodo stabilų suderinimą daugelyje judėjimo fazių.

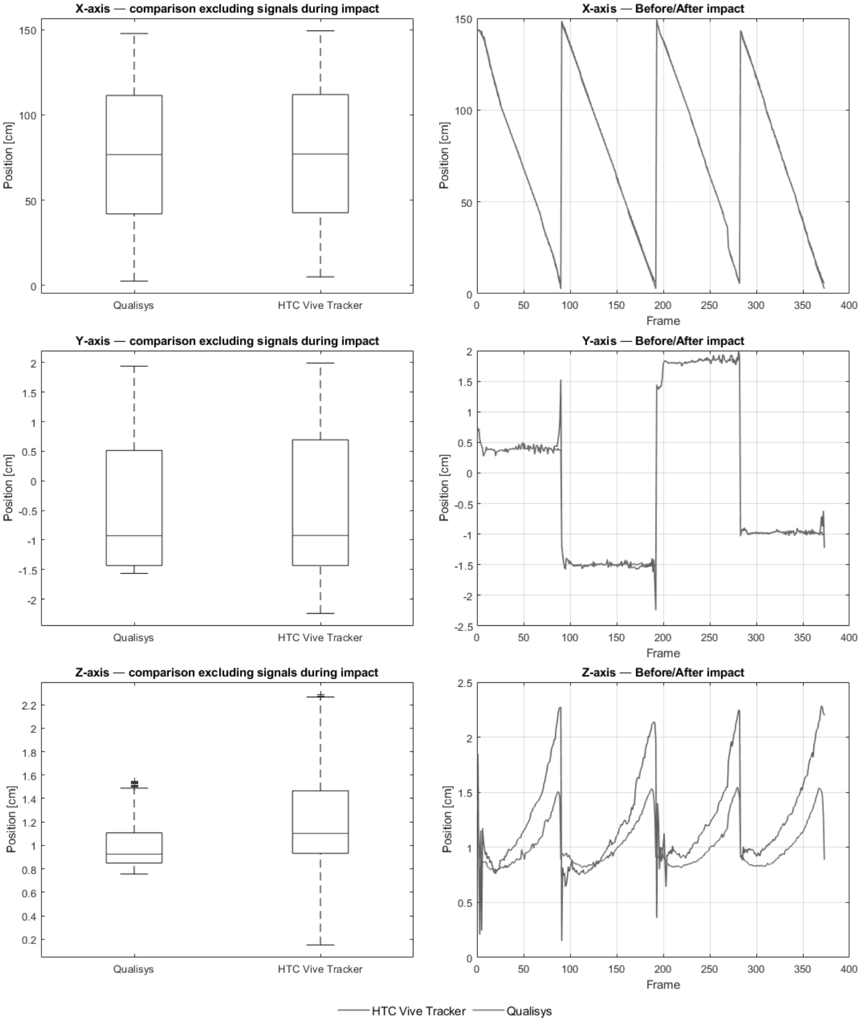
Vizualūs palyginimai, įskaitant dėžutės diagramas ir judėjimo signalų kreives (žr. 99 pav.) patvirtina šias išvadas.



99 pav. „Qualisys“ ir „HTC Vive Tracker“ padėties duomenų lyginamoji analizė per visus signalo ir poveikio segmentus

Kaip matyti iš paveiklėlių, Z ašyje atsiranda daugybė nuokrypių su daug didesniu verčių spektru, o šoninis ir vertikalus sekimas yra palyginti stabilus. Statistinis patvirtinimas naudojant *Kruskal-Wallis* testą parodė, kad tik Z ašyje abiejų sistemų medianos statistiškai reikšmingai skiriasi, ypač smūgių metu (p < 0,05), o X ir Y ašyse esant daugumai sąlygų, reikšmingų skirtumų nebuvo. Atmetus smūgių paveiktus intervalus, bendros paklaidos vertės gerokai sumažėjo, ypač Z ašyje, kur MAPE sumažėjo nuo daugiau kaip 30 % iki vos 0,24 % (žr.

29 **lentelėę** ir 100 pav.), o tai patvirtina, kad dauguma nukrypimų atsiranda dėl trumpalaikių smūgių.



100 pav. „Qualisys“ ir „HTC Vive Tracker“ padėties duomenų lyginamoji analizė, išskiriant gautus smūgio metu signalus

31 lentelė. *Kruskalio-Valiso* testo rezultatai visam signalui

Ašis	**p-reikšmė**	**Atmetimas (H_0)**
X- ašis	0,22477	Neigiamas
Y- ašis	0,20515	neigiamas
Z- ašis	2,434e-25	teigiamas

32 lentelė. *Kruskal-Wallis* segmentų analizė neįtraukiant poveikių

Ašis	**p-reikšmė**	**Atmetimas (H_0)**
X-ašis	1,5715e-123	teigiamas
Y- ašis	0,59499	neigiamas
Z- ašis	4,807e-15	teigiamas

33 lentelė. *Kruskal-Wallis* judėjimo segmentų analizė smūgių metu

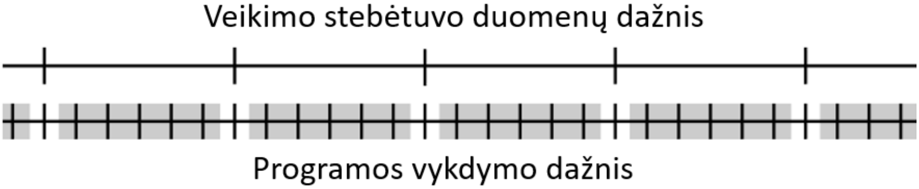
Ašis	**p-reikšmė**	**Atmetimas (H_0)**
X- ašis	0,00084505	teigiamas
Y-axis	0,31282	neigiamas
Z-axis	2,1676e-53	teigiamas

Šie rezultatai rodo, kad nors „HTC Vive Tracker“ užtikrina pakankamą sekimo tikslumą ramaus ir nepertraukiamo judėjimo atvejais, veikimas pablogėja didelio poveikio įvykių metu, ypač gylio (Z) kryptimi. Praktiniam naudojimui tai reiškia, kad sekiklis tinka ilgesniam naudojimui be žymeklių VR, jei suvaldomi smūgių sukelti nuokrypiai. Kai reikia tiksliai atkartoti greitus judesius ar smūgius, pavyzdžiui, labai tikslioje reabilitacijoje ar lavinant motorinius įgūdžius, gali prireikti alternatyvių sprendimų ar papildomų korekcijos metodų. Vis dėlto bendram VR naudojimui, susijusiam su sklandžiu, nuolatiniu judėjimu, ši sistema yra tinkamas ir praktiškas sprendimas.

PERIFERINIO ĮVESTIES ĮRENGINIO INTEGRAVIMAS Į VIRTUALIĄJĄ REALYBĘ

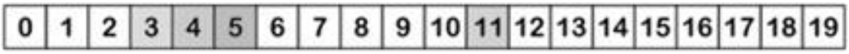
Siūlomos sistemos aprašymas

Šiam tyrimui sukurta sistema siekiama, kad iš „Concept-II“ irklavimo treniruoklio į VR perduodami duomenys būtų apdorojami taip, kad irklavimas virtualiojoje realybėje būtų perteikiamas tiksliai ir be trūkčiojimų. Pagrindinis iššūkis kyla dėl duomenų perdavimo spartos skirtumo tarp „Concept-II Performance Monitor“ (PM), kuris veikia 10 Hz dažniu ir nešiojamų VR įrenginių, kurie paprastai atnaujina duomenis 60 FPS sparta. Dėl šio neatitikimo VR programa negali gauti naujų duomenų kiekvieno kadro, todėl VR vaizde atsiranda trūkčiojimo efektas ir vėlavimas (dažnio iliustracija (žr. 101 pav.).



101 pav. PM (Performance Monitor) duomenų ir taikomosios programos vykdyto dažnio atvaizdavimas

Siekiant išspręsti šią problemą sukurta sistema, kurią sudaro du pagrindiniai komponentai: nešiojamas VR įrenginys, kuriame veikia VR irklavimo programa, ir „performance monitor“ kompiuteris, su kuriuo palaikomas ryšys per „Bluetooth Low Energy“ (BLE). VR irklavimo taikomoji programa iš „performance monitor“ gauna duomenis, į kuriuos įeina tokie rodikliai kaip atstumas, mosto dažnis, o išvystoma galia ir kalorijos perduodamos per BLE specifiniu baitų masyvo formatu (žr. 102 pav.).



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

102 pav. Irklavimo bendrosios būsenos funkcijų paketas

Kadangi VR irklavimo programai reikalingi realaus laiko duomenys iš PM, naudinga pakeisti numatytąją vertę nuo 1 iki 3, atitinkančią 100 milisekundžių atnaujinimo periodą.

Sistemos konfigūracija

- **„Performance monitor“ (PM):** įrenginys, kuris fiksuoja ir saugo treniruotės duomenis, įskaitant nuplauktą atstumą ir mosto būseną, atnaujinamus 100 ms intervalu.
- **Nešiojama VR sistema:** VR irklavimo programa, veikianti mobiliajame VR įrenginyje, kuri gauna duomenis per BLE ir juos apdoroja.
- „**Concept-II“ irklavimo treniruoklis:** fizinis prietaisas skirtas irklavimo treniruotėms.

Pagrindinis signalų numeracijos ir sinchronizavimo iššūkis kyla dėl programos vykdymo spartos ir PM duomenų dažnio neatitikimo, dėl kurio gali atsirasti galimas vėlavimas ir nenuosekli animacija virtualiojoje realybėje.

Ištirti prognozavimo metodai

Siekiant kompensuoti vėlavimą ir prarastus duomenis dėl mažesnio PM atnaujinimo dažnio, buvo įgyvendinti keturi prognozavimo algoritmai.

- Interpoliavimas pagal padėtį leido, palyginti, tiksliai sekti atstumą, tačiau atsirado nemažai mikrodilgsėjimų ir VR sesija paprastai baigdavosi šiek tiek vėliau nei fizinė sesija.

- Ekstrapoliacija pagal greitį leido išvengti mikrosekų, tačiau dėl klaidų kaupimosi VR seansas baigdavosi daug anksčiau nei tikrasis.
- Ekstrapoliacija pagal greitį su korekcija užtikrino ir didelį tikslumą, ir sklandų, neužsikertantį vaizdą, o pabaigos laikas buvo ties nuline klaida, todėl beveik nebuvo užsikirtimų visuose seansuose.
- Kalmano filtro metodas teikė kompromisą tarp judesio tolygumo ir sinchronizacijos tikslumo – reikšmingai sumažino trūkčiojimų skaičių, tačiau jo taikymas sukėlė nedidelį vėlavimą ir šiek tiek ilgesnes pavienių trūkčiojimų trukmes.

Kiekvienu metodu siekiama įvertinti kitą virtualios valties padėtį, užtikrinant sklandesnį judėjimą ir sumažinant trūkstamų ar vėluojančių duomenų poveikį.

Eksperimento rezultatai

Visų seansų rezultatų pateikimas viename arba atskiruose grafikuose dažnai sukelia triukšmingą vaizdą, todėl tampa sunku atskirti atskirus atvejus. Siekiant to išvengti buvo atrinkti du atvejai ir pažymėti kaip „Greičiausias irklavimo seansas“ (FRS, 9 seansas) ir „Lėčiausias irklavimo seansas“ (SRS, 5 seansas), kad būtų galima išanalizuoti geriausio ir blogiausio atvejų charakteristikas (žr. 34 ir 35 lentelę). FRS atitinka irklavimo seansą, per kurį 250 metrų atstumas buvo pasiektas per trumpiausią laiką, o SRS – seansą, kuriam prireikė ilgiausiai laiko tam pačiam atstumui įveikti.

Eksperimentų rezultatais siekiama įvertinti siūlomų prognozavimo metodų veiksmingumą išlaikant atstumo tikslumą laiko atžvilgiu ir užtikrinant sklandžią, be trūkčiojimų VR patirtį naudotojui. Eksperimentą sudaro dešimt irklavimo seansų, kurių atstumas – 250 metrų distancija.

Kiekvieno metodo našumas vertinamas pagal du pagrindinius veiksnius:

1. laiko skirtumą tarp seanso užbaigimo VR mobiliojoje programoje ir irklavimo treniruoklyje;
2. mikrotrūkčiojimų atsiradimą VR irklavimo metu.

34 lentelė. Laiko skirtumas milisekundėmis (ms) užbaigus sesiją

	Metodai			
Sesija Nr.	**Interpoliavimas pagal padėtį**	**Ekstrapoliacija pagal greitį**	**Ekstrapoliacija pagal greitį su pataisa**	**Kalmano filtras**
1	+39,9	–6828	–62	+43
2	+77,5	–8469	–18,4	+173
3	+57,4	–7492	+0,2	+201
4	+34	–4894	–5	+5
5 (SRS)	+94,3	–8453	–63,7	+217
6	+81,4	–6193	–14	+186

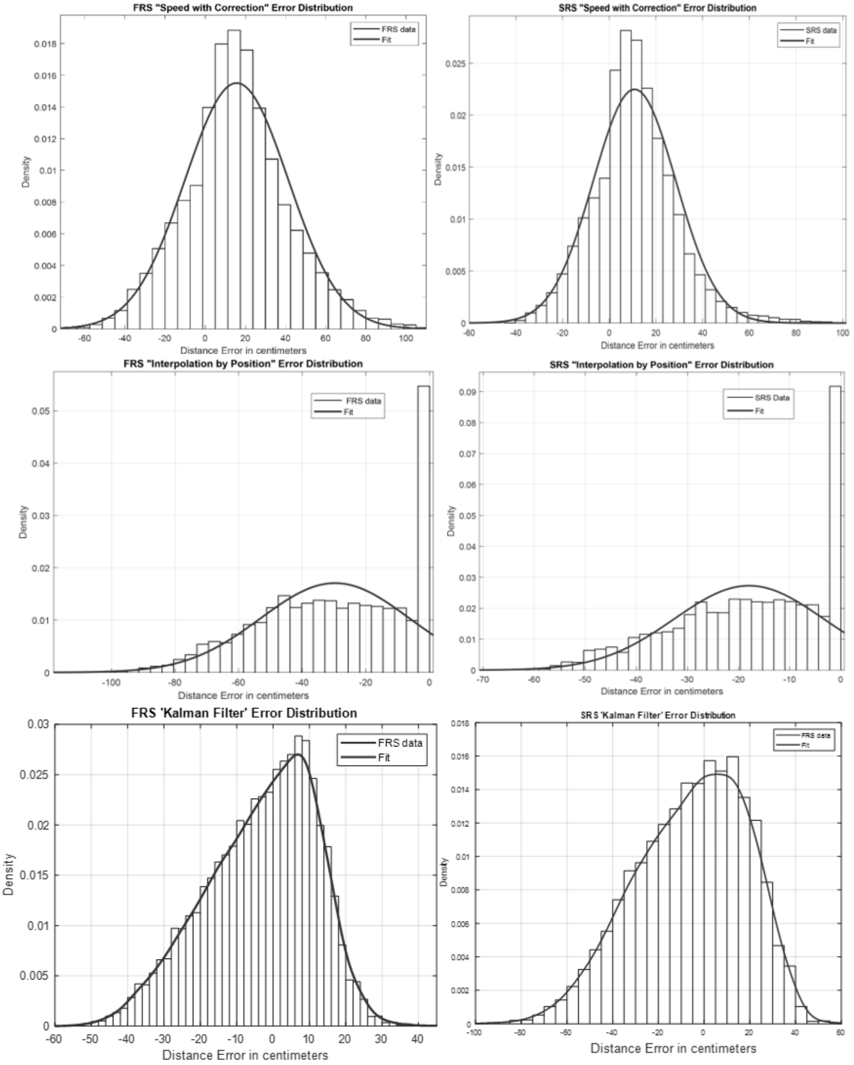
7	+43,9	–6700	–33,5	+6
8	+34	–7064	+16,6	+52
9 (FRS)	+100	–4758	–43,3	+226
10	+85,2	–7362	+32,8	+189
Sesijos vidurkis	+64,7ms	–6821 ms	+28,95 ms	+129,8 ms

35 lentelė. Mikrotrūkčiojimų skaičius ir vidutinė kiekvienos sesijos trukmė taikant skirtingus metodus

			Metodai			
Sesija Nr.	Išmatuoti rodikliai	Įvestis	Interpoliavimas pagal padėtį	Kalmano filtras	Ekstrapoliacija pagal greitį	Ekstrapoliacija pagal greitį su korekcija
S1	Užstrigimų skaičius	478	263	135	0	0
	Vidutinė trukmė	110,8 *ms*	37,9 *ms*	48,7 *ms*		
S2	Užstrigimų skaičius	524	289	162		
	Vidutinė trukmė	111,2 *ms*	39,8 *ms*	53,6 *ms*		
S3	Užstrigimų skaičius	452	260	138		
	Vidutinė trukmė	111,2 *ms*	37 *ms*	52,5 *ms*		
S4	Užstrigimų skaičius	426	227	128		
	Vidutinė trukmė	111,1 *ms*	38,9 *ms*	53,6 *ms*		
S5(SRS)	Užstrigimų skaičius	667	354	210		
	Vidutinė trukmė	111,3 *ms*	38,2 *ms*	53,7 *ms*		
S6	Užstrigimų skaičius	447	238	131		
	Vidutinė trukmė	111,3 *ms*	40 *ms*	56,6 *ms*		
S7	Užstrigimų skaičius	437	253	134		
	Vidutinė trukmė	111,2 *ms*	37,8 *ms*	53,6 *ms*		
S8	Užstrigimų skaičius	535	273	160		
	Vidutinė trukmė	111,2 *ms*	40 *ms*	51,4 *ms*		
S9(FRS)	Užstrigimų skaičius	405	202	116		
	Vidutinė trukmė	111,1 *ms*	40,8 *ms*	53 *ms*		
S10	Užstrigimų skaičius	448	247	140		

	Užstrigimų skaičius	111,1 ms	39,4 ms	52,5 ms		
Sesijos vidurkis	Trūkčiojimų skaičius	481,9	260	145,4		
	Vidutinė trukmė	111,15 *ms*	38,98 *ms*	52,92 *ms*		

Šiuos rezultatus toliau parodo klaidų pasiskirstymo histogramos. Taikant pataisytą ekstrapoliavimo metodą gautas beveik tikslus klaidų pasiskirstymas, kurio centras arti nulio, o taikant kitus metodus buvo pastebėtas nuokrypis arba didėjanti paklaida.



103 pav. Ekstrapoliacija pagal greitį su korekcija (viršaus kairėje ir viršaus dešinėje), interpoliacija pagal padėtį (vidurio kairėje ir vidurio dešinėje); Kalmano filtras (apačios kairėje ir apačios dešinėje)

Tyrimo išvados

Šiame tyrime buvo nagrinėjamos dvi pagrindinės problemos, turinčios įtakos įtraukčiai į VR irklavimą: mikrotrūkčiojimas ir atstumo klaidos pagal laiką. Abi šios problemos kyla dėl skirtingų duomenų perdavimo dažnių tarp irklavimo treniruoklio ir VR mobiliosios programos. Siekiant išspręsti šiuos iššūkius, buvo sukurti ir įvertinti keturi prognozavimo algoritmai. Eksperimentiniai rezultatai pateikiami toliau.

- Interpoliavimo pagal padėtį metodas pateikia tikslias atstumo reikšmes per visą virtualaus irklavimo seansą, nuosekliai sekdamas įvesties atstumo parametrą. Tačiau šis metodas sukelia reikšmingus trūkčiojimus, kurie gali neigiamai paveikti VR patirtį, sutrikdyti įsitraukimą ir netgi sukelti judesio pažeidimų. Kadangi metodas visada šiek tiek atsilieka nuo įvesties atstumo parametro, seansas VR aplinkoje baigiasi šiek tiek vėliau nei realiuoju laiku baigiamas irklavimas.

- Ekstrapoliavimo pagal greitį metodas lemia didėjančią atstumo paklaidą per visą virtualaus irklavimo seansą, todėl VR aplinkoje seansas baigiasi gerokai anksčiau nei realus irklavimas treniruokliu. Paklaida tampa vis ryškesnė, kai sesijos trukmė pailgėja iki 250 metrų distancijos. Nors šis metodas suteikia sklandžią, be trūkčiojimų irklavimo patirtį, jis neužtikrina pakankamo tikslumo.

- Kalmano filtro metodas suteikia balansą tarp judesio tolygumo ir sinchronizacijos tikslumo. Palyginti su interpoliacijos metodu, jis reikšmingai sumažina trūkčiojimų skaičių, tačiau sukelia nedidelį vėlavimą ir šiek tiek ilgesnes atskirų trūkčiojimų trukmes. Apskritai, šis metodas pasižymi geresniu judesio stabilumu nei interpoliacija, tačiau išlieka ne toks tikslus nei greičio ekstrapoliacija su korekcija tiek laiko, tiek VR įtraukumo požiūriu.

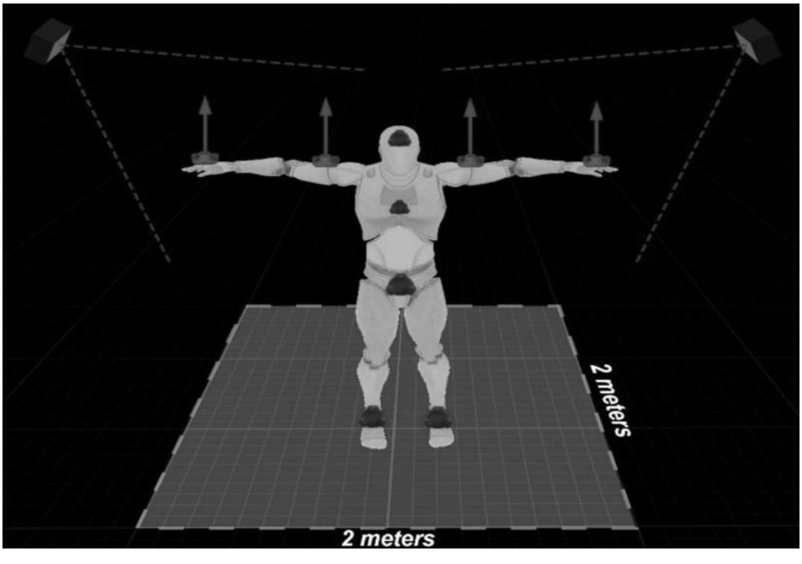
- Korekcija, taikoma ekstrapoliacijai pagal greitį metodu, gerokai sumažina atstumo paklaidą, leidžia tiksliai sekti įvesties atstumo parametrą. Šis metodas užtikrina didesnį tikslumą, atstumo paklaida dažniausiai svyruoja nuo −20 iki 20 cm ir nuo −10 iki 10 cm, o tai yra geresnis rezultatas nei kitų metodų, naudojamų įvesties duomenų atveju. Dėl to šis metodas sumažina laiko skirtumą sesijos pabaigoje ir užtikrina visiškai sklandžią VR patirtį visose sesijose.

AI PAREMTAS LAIKYSENOS ĮVERTINIMAS NAUDOJANT VR VALDIKLIUS

Šiame tyrime daugiausia dėmesio skiriama žmogaus laikysenos diagnostikai naudojant „HTC-Vive“ sekimo įrenginius, kuriais fiksuojami į kampą nukreipti duomenys, kurie analizuojami CNN ir RF modeliais, siekiant nustatyti laikysenos anomalijas. Laikysenos vertinimas yra pirmas žingsnis vertinant raumenų ir kaulų sistemos sutrikimus, apimantis tiek statinę, tiek dinaminę analizę, kad būtų galima suprasti paciento judėjimo ir kontrolės gebėjimus. Tokie sutrikimai kaip skoliozė, kojų ilgio neatitikimai, ankstesnės traumos ar neurologiniai sutrikimai, gali lemti laikysenos sutrikimus. Siekiant veiksmingos reabilitacijos, reikia stebėti aktyvią judesių amplitudę (ROM), ypač pečių ir viršutinių galūnių, nes dėl ROM, raumenų jėgos ar darbingumo trūkumų gali atsirasti netinkamas pratimų atlikimas ir kompensaciniai judesių modeliai. Todėl norint tiksliai įvertinti ir pašalinti laikysenos nukrypimus, labai svarbu visapusiškai įvertinti pečių judesius – įskaitant įvairias kryptis ir rotacijas [99–101].

Sistemos konfigūracija

Konfigūracijoje naudojama „HTC Vive" platforma su mažiausiai aštuoniais, antros kartos „HTC Vive" jutikliais, kad būtų galima fiksuoti ir erdvines padėtis ir pasukimo kampus. Veikimui reikalingos bent dvi „HTC Vive" bazinės stotys, tačiau rekomenduojamos keturios stotys, kad būtų užtikrintas didesnis sekimo tikslumas ir patogesnis naudojimas. Bendras sistemos ir jutiklių išdėstymas parodytas septintame paveiksle (žr. 104 pav.), kuriame taip pat parodyta, kad tiriamasis turi turėti bent 2 m. laisvos erdvės, kad galėtų nevaržomai judėti. Tiriamasis stovi „T" padėtyje (vertikalioje padėtyje su ištiestomis rankomis) ir po 10 sekundžių išlaikymo ištiesia abi rankas į priekį. Pacientams po insulto, kuriems gali būti hemiparezė ar sumažėjęs raumenų tonusas, slaugytojas arba slaugytoja gali padėti išlaikyti taisyklingą rankų padėtį kalibravimo metu. Sistema neseka padėjėjo, todėl jo buvimas netrukdo skeleto atskaitai registruoti, net jei jo rankos laikinai uždengia paciento jutiklius. Kalibravimo procese kiekvienas VR seklys susiejamas su konkrečia kūno dalimi, o po kelių judesių tiriamasis sinchronizuojamas su savo virtualiu avataru. Baigus kalibravimą, galima pradėti pratybų seansą.



104 pav. Sistemos ir jutiklių išdėstymas: tiriamasis turi turėti mažiausiai 2 metrų erdvę, kad galėtų laisvai ir nevaržomai judėti

Jutiklių išdėstymas:

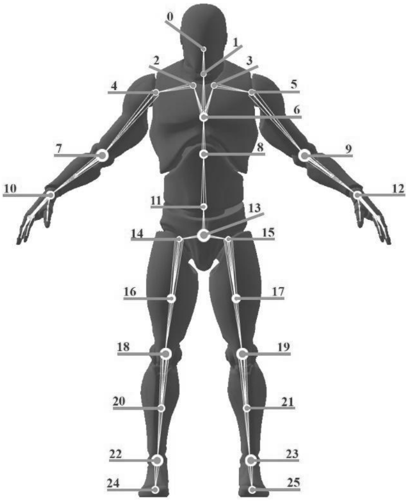
- du jutikliai tvirtinami prie rankų (nukreipti į viršų);
- du jutikliai tvirtinami prie delnų (nukreipti į viršų);
- du jutikliai yra ant kojų (nukreipti į priekį);
- vienas jutiklis tvirtinamas ant klubų (nukreiptas į priekį);
- vienas jutiklis dedamas ant galvos (nukreiptas į priekį);
- vienas (neprivalomas) jutiklis gali būti ant krūtinės (nukreiptas į priekį).

Prieš pradedant pratimų stebėjimo seansą, jutikliai turi būti pritvirtinti prie paciento ir patikrinti ar tinkamai išdėstyti ir orientuoti. Jutikliai turi būti tvirtai pritvirtinti, kad judėjimo metu nepasislinktų. Pritvirtinus jutiklius, jų suderinimas su virtualiu avataru atliekamas per sistemos kalibravimą.

Eksperimentinė konfigūracija

Eksperimente dalyvavo sveiki dalyviai, kurie atliko ir taisyklingus, ir netaisyklingus, iš anksto nustatytų pratimų variantus, o kiekvienas judesys buvo stebimas naudojant VR jutiklius, įrengtus ant anatominių orientyrų, vadinamų kaulų mazgais.

Suderinus VR jutiklius su UE4 manekeno kaulų mazgais, realaus pasaulio judesius buvo galima atvaizduoti virtualiam avatarui, todėl buvo galima tiksliai išmatuoti sąnarių kampus ir nustatyti kompensacinius judesius. Kiekvienam pratimui buvo parinkti atitinkami mazgai, nustatyti ir užfiksuoti reikiami kampiniai pokyčiai atliekant taisyklingai ir netaisyklingai pratimus. Iš viso buvo išanalizuota 19 variantų, apimančių ir taisyklingas, ir netaisyklingas formas.



0. galva
1. kaklas
2. raktikaulis_r
3. raktikaulis_l
4. petis_r
5. žastas_l
6. stuburas_03
7. alkūnė_r
8. stuburas_02
9. alkūnė_l
10.ranka_r
11.stuburas_01
12.ranka_l
13. klubai
14. šlaunis_r
15. šlaunis_l
16. šlaunikaulis_r
17. šlaunikaulis_l
18. blauzda_r
19. blauzdikaulis
20. blauzdikaulis_r
21. blauzdikaulis_l
22. kulnakaulis_r
23. kulnakaulis _l
24. pėda_r
25. pėda_l

105 pav. „Unreal Engine 4“ skeleto mazgų struktūra

Duomenims analizuoti buvo naudojami toliau aprašyti įrankiai.

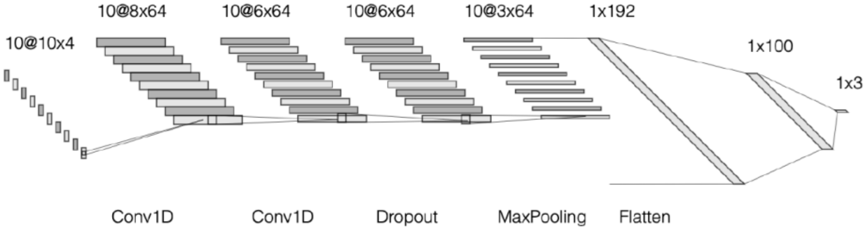
- **Patikimumo intervalai.** Naudojami siekiant nurodyti, kad 95 % reikšmių patenka į nurodytą intervalą, atspindintį duomenų kintamumą ir patikimumą.
- **Pupų (angl. *bean*) diagramos.** Tai duomenų tankį iliustruojantys simetriški pupelės formos pasiskirstymai, kuriuose spalvų gradientai išryškina koncentracijos sritis.
- **Histogramos.** Stulpelinės diagramos, vaizduojančios statistinį duomenų pasiskirstymą ir tankį.
- ***Stjudento* t-testas.** Įvertinama, ar dviejų imčių vidurkiai reikšmingai skiriasi, remiantis p vertėmis, palyginti su 0,05 reikšmingumo riba.
- **F testas.** Lyginamos grupių dispersijos taip pat naudojant p reikšmę, kai hipotezės tikrinimo riba yra 0,05.

Pratimus pateikė projekto partneris Vilniaus Gedimino technikos universitetas. Tai: priekinis rankos pakėlimas, šoninis rankos pakėlimas, horizontalioji rankos abdukcija / addukcija, rankos ištiesimas per galvą, sunkaus daikto pakėlimas ir padėjimas virš galvos, ašinė ištiestos rankos rotacija, nosies prisilietimo koordinacija, dilbio supinacija / proninacija. Kiekvieno pratimo duomenų rinkinys buvo suskirstytas į šias dalis: 0 klasė (taisyklingas judesys), 1 klasė (netaisyklingas judesys 1), 2 klasė (pasirinktinis netaisyklingas judesys 2).

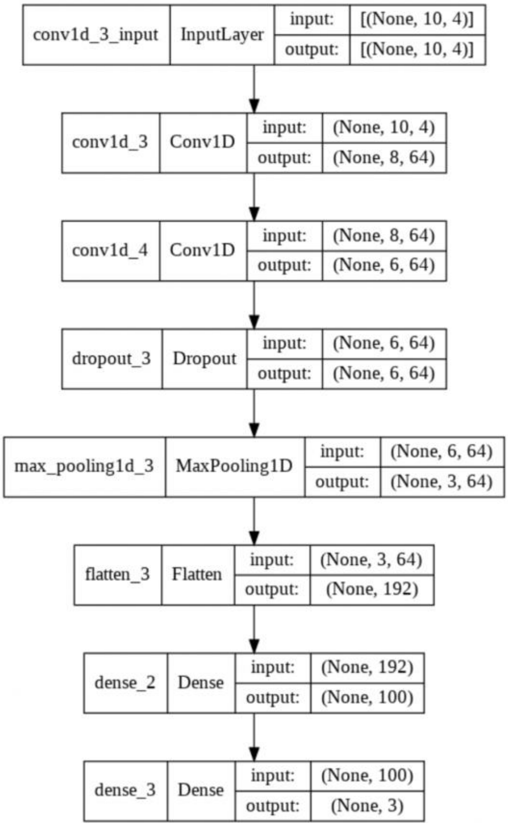
Šios priemonės ir protokolai buvo naudojami siekiant nustatyti reikšmingus sąnarių kampų skirtumus tarp taisyklingų ir netaisyklingų judesių klasių, užtikrinant griežtą statistinį vertinimą. Tyrimo scenarijais siekta atskleisti realius fizinių pratimų atlikimo rezultatus ir kasdienes kompensacines strategijas. Atliekant tyrimą laikytasi Helsinkio deklaracijos nuostatų, jam etiškai pritarė Vilniaus Gedimino technikos universiteto Institucinės priežiūros komitetas Nr. 64-2221.

Metodika

Atliktame tyrime CNN modeliai buvo naudojami dviem pagrindinėms klasifikavimo užduotims: kampu pagrįstam klasifikavimui (judesių modeliams nustatyti) ir sprendimu pagrįstam klasifikavimui. Kiekvienai užduočiai buvo naudojamos konkrečios neuronų tinklų architektūros, pritaikytos atitinkamam uždaviniui. Sutrikimams vertinti yra įvairių klasifikavimo ir prognozavimo metodų, kuriais galima nustatyti galimas patologijas naudojant skaitmeninius vaizdus, biologinius signalus, judesių duomenis ir kitus šaltinius [102, 103]. Buvo išbandytos įvairios CNN architektūros ir sluoksnių deriniai. Galutinį CNN modelį, naudojamą judesių sutrikimų turintiems ir neturintiems asmenims atskirti, sudarė: du konvoliuciniai sluoksniai, vienas jungiamasis sluoksnis, išlyginamasis sluoksnis ir du tankūs, visiškai sujungti sluoksniai. Šio tinklo struktūra parodyta devintame ir dešimtame paveiksle (žr. 106 ir 107 pav.).



106 pav. Konvoliucinio neuroninio tinklo išdėstymo schema



107 pav. Konvoliucinio neuroninio tinklo sluoksnių seka

Kuriant modelį naudota atsitiktinė tinklo paieška, kad būtų optimizuoti pagrindiniai parametrai, įskaitant epochų skaičių, partijos dydį ir validacijos padalijimą. Buvo atlikta 100 iteracijų kiekvienam judėjimo tipui ir atrinktas tiksliausias modelis. Tinklelio paieškos reikšmės buvo tokios:

epochų skaičius: [6, 8, 10, 12, 14, 16, 18, 20, 21, 22, 50];

partijos dydis: [20, 28, 34, 40, 48, 55, 68, 74, 80];

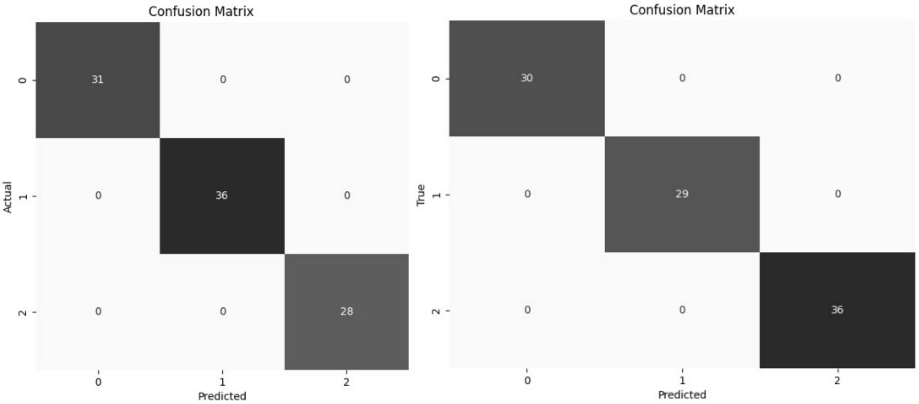
validacijos padalijimas: [0,05; 0,1; 0,15; 0,2; 0,25; 0,3].

Statistinių duomenų ir pratybų rezultatų vizualizavimas

Šiame skyriuje pateikiama statistinė analizė ir pratimų duomenų, užfiksuotų naudojant „HTC Vive“ jutiklius, vizualizavimo rezultatai. Buvo analizuojami kiekvieno pratimo sąnarių kampai ir kompensaciniai judesiai, lyginami taisyklingai ir netaisyklingai atlikti pratimai. Pagrindinis dėmesys buvo skiriamas reikšmingiems statistiniams skirtumams nustatyti ir šiems rezultatams vizualizuoti naudojant pasikliautinuosius intervalus, pupelių diagramas, histogramas ir sumaišymo matricas.

Pratimas „Priekinis rankos pakėlimas“

Analizuoti 4, 5, 7, 9 ir 6 mazgai, daugiausia dėmesio skiriant alkūnės ištiesimui, peties pakėlimui ir vertikaliai laikysenai. Dalyviai atliko priekinį rankos pakėlimą, siekdami tinkamų alkūnės (180 °), peties (90 °) ir stuburo (90 °) kampų. Duomenys apėmė 95 kiekvienos klasės pavyzdžius iš 19 sveikų tiriamųjų. 0 klasė rodo taisyklingą judesį; 1 klasė rodo kompensaciją liemeniu, o 2 klasė – kompensaciją petimi.

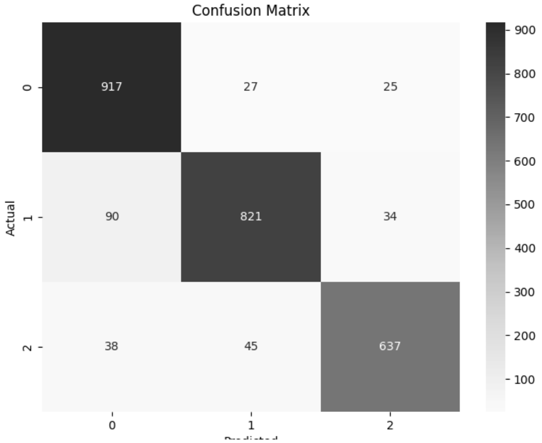


108 pav. Kiekvienos klasės, klasifikuotos atliekant pratimą „Rankos pakėlimas į priekį“, sumaišymo matrica

36 lentelė. Sprendimų klasifikavimo rodikliai, taikomi pratimui „Rankos pakėlimas į priekį “.

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	10	Iteracijų skaičius	36860
Didžiausias gylis	3	Epochų skaičius	38
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,03 s	46,75 s	
Tikslumas	100 %	100 %	

Atliekant kampu pagrįstą klasifikavimą, CNN pasiekė 90,17 % bandymo tikslumą, o jo atlikimo laikas buvo 29,89 sekundės (žr. 109 pav.) Supainiojimo matrica atskleidė keletą klaidingų klasifikacijų: 0 klasėje buvo 27 klaidingai teigiami ir 25 klaidingai neigiami rezultatai, 1 klasėje – 90 klaidingai teigiamų ir 34 klaidingai neigiami rezultatai, o 2 klasėje – 38 klaidingai teigiami ir 45 klaidingai neigiami rezultatai. CNN mokymo parametrai apėmė 41 000 iteracijų per 25 epochas, naudojant 6 partijų dydį ir 20 % validacijos padalijimą.



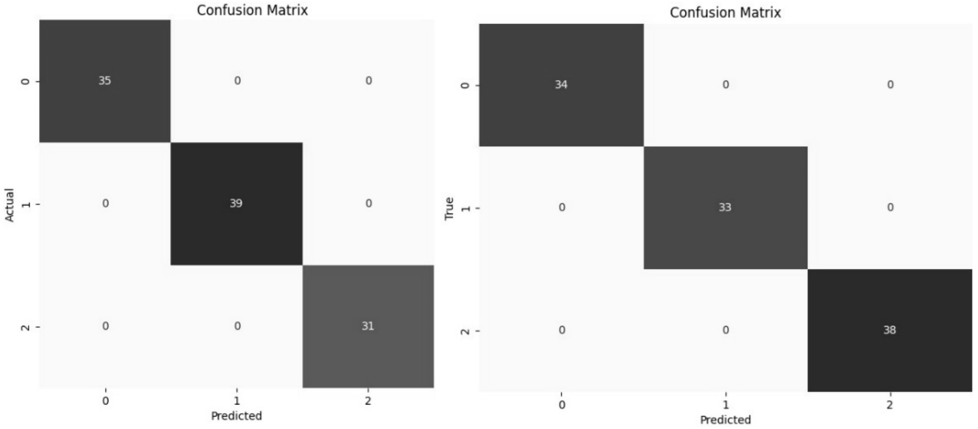
109 pav. Kiekvieno CNN klasifikuoto kampo sumaišymo matrica atliekant pratimą „Priekinis rankos pakėlimas“

37 lentelė. Kampų klasifikavimo metrikos, skirtos pratimui „Priekinis rankos pakėlimas“

CNN klasifikatorius	**Reikšmė**
Iteracijų skaičius	41000
Epochų skaičius	25
Partijos dydis	6
Patvirtinimopadalijimas	20 %
Vykdymo laikas	29,89 s
Testo tikslumas	90,17 %

„Rankos pakėlimo į šoną“ pratimas

Pratime analizuojami 4, 5, 6, 7 ir 9 mazgai, daugiausia dėmesio skiriant alkūnės ištiesimui, rankos pakėlimui į šoną, vertikaliai laikysenai ir horizontaliai pečių padėčiai. Dalyviai siekė 0 ° alkūnės sulenkimo, 90 ° rankos pakėlimo į šoną ir tinkamo stuburo bei pečių kampo.



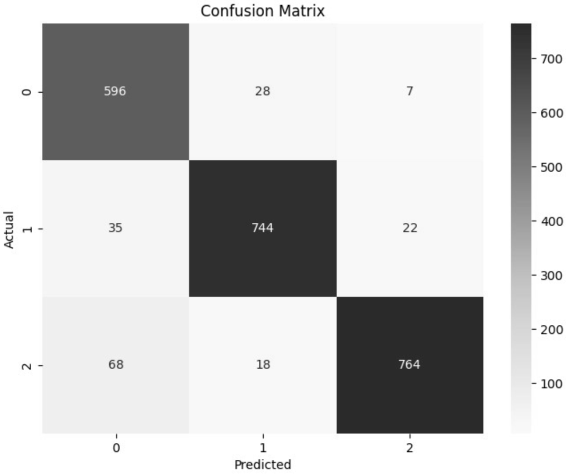
110. pav. Kiekvienos klasės, klasifikuotos atliekant pratimą „Rankos pakėlimas į šoną“, sumaišymo matrica

38 lentelė. Sprendimų klasifikavimo rodikliai, taikomi pratimui „Rankos pakėlimas į šoną“

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	10	Iteracijų skaičius	9416

Didžiausias gylis	3	Epochų skaičius	11
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,032 s	11,57s	
Testo tikslumas	100 %	100 %	

Atliekant kampu pagrįstą klasifikavimą naudojant CNN, kai buvo atliekamas pratimas „Rankos pakėlimas į šoną“, klasifikatorius pasiekė 92,20 % testo tikslumą. Maišaties matrica (žr. 111 pav.) rodo, kad nors dauguma kampų buvo gerai suklasifikuoti, tarp trijų klasių pasitaikė nedidelių klaidų. Tiksliau, 0 ir 2 klasėse buvo negerai suklasifikuota keletas pavyzdžių, o 1 klasė pasižymėjo iš dalies geresniu atskyrimu. CNN efektyviai įvykdė klasifikavimo užduotį per 16,62 sekundes, atlikdamas 36 200 iteracijų per 15 epochų. CNN naudojo 6 partijų dydį ir 20 % validacijos padalijimą, taip parodydamas gebėjimą tiksliai atskirti subtilius kampo duomenų pokyčius per priimtiną skaičiavimo laiką.



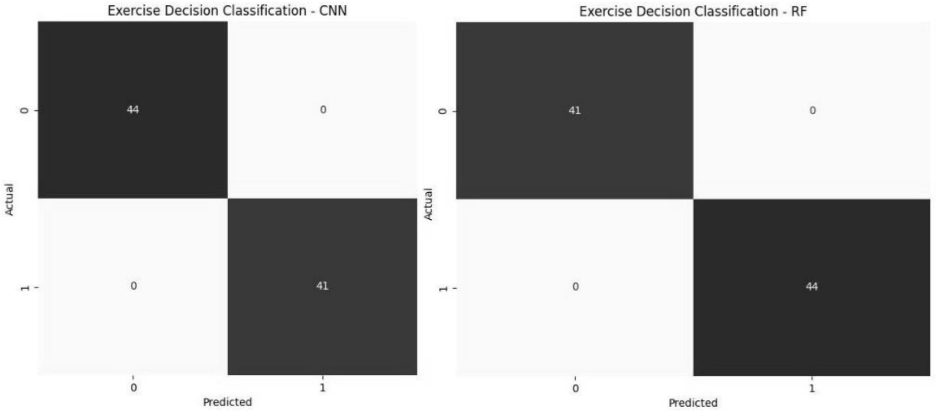
111 pav. CNN klasifikuotų kampų, skirtų pratimui „Rankos pakėlimas į šoną“, painiavos matrica

39 lentelė. Kampų klasifikavimo metrikos, skirtos „Rankos pakėlimo į šoną “ pratimui

CNN klasifikatorius	**Reikšmė**
Iteracijų skaičius	36200
Epochų skaičius	15
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	16,62 s
Testo tikslumas	92,20 %

Horizontalus rankos abdukcijos / addukcijos pratimas

Atliekant šį pratimą daugiausia dėmesio skirta 4, 5, 6, 7, 9, 22 ir 23 mazgams, vertinant alkūnės ištiesimą, vertikalią ir horizontalią pečių padėtį, liemens pasukimą ir vertikalią laikyseną. Dalyviai siekė, kad rankos būtų visiškai ištiestos, pečių perėjimo kampai būtų taisyklingi, o laikysena ir pasisukimas kuo mažiau nukryptų.



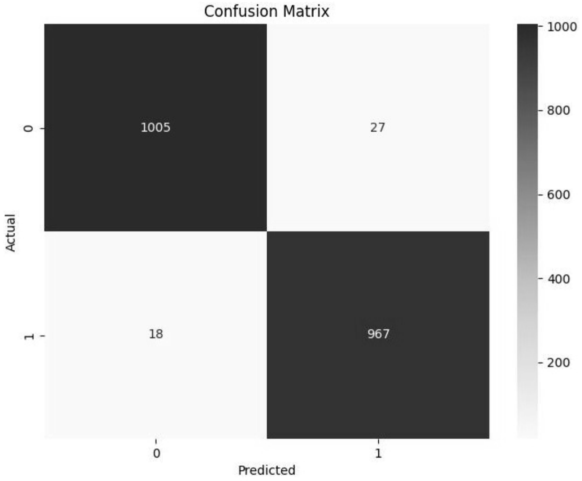
112 pav. Kiekvienos klasės, klasifikuotos atliekant horizontalaus rankos abdukcijos / addukcijos pratimą, sumaišymo matrica

RF klasifikatorius įvykdė savo užduotį per 0,03 sekundės, jam reikėjo tik 10 medžių, kurių didžiausias gylis buvo 3. Tai parodo jo skaičiavimo efektyvumą ir paprastumą siekiant tikslios klasifikacijos. Kita vertus, CNN klasifikatoriui prireikė 9 416 iteracijų per 11 epochų ir 11,57 sekundės vykdymo laiko. Nepaisant skaičiavimo požiūriu intensyvaus mokymo proceso, CNN taip pat pasiekė puikų rezultatą. Abu modeliai yra tinkami patikimai klasifikuoti tokio tipo pratimus.

40 lentelė. Horizontalaus rankos abdukcijos / addukcijos pratimo klasifikavimo rodikliai

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	10	Iteracijų skaičius	9416
Didžiausias gylis	3	Epochų skaičius	11
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,03 s	11,57s	
Testo tikslumas	100 %	100 %	

Atliekant kampu pagrįstą klasifikavimą horizontalios rankos abdukcijos / addukcijos pratime (žr. 113 pav.), CNN pasiekė 97,77 % bandymo tikslumą. Supainiojimo matrica atskleidė nedidelių klaidų, daugiausia klaidų buvo padaryta tarp klasių 0 ir 1. CNN šią užduotį efektyviai apdorojo per 11,39 s, atlikdama 31 050 iteracijų per 12 epochų.



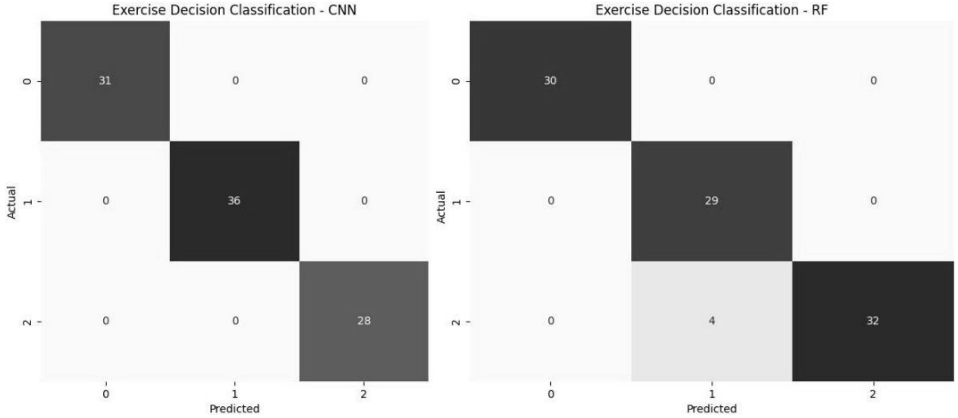
113 pav. Horizontalaus „Rankos abdukcijos“ pratimo klasifikavimo rodikliai

41 lentelė. Horizontalaus „ Rankos abdukcijos“ pratimo klasifikacijos rodikliai

CNN klasifikatorius	Reikšmė
Iteracijų skaičius	31050
Epochų skaičius	12
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	11,39 s
Testo tikslumas	97,77 %

Viršugalvio siekimas ranka

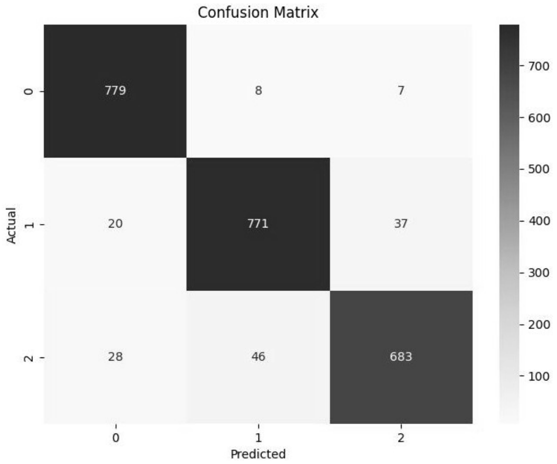
Atliekant šį pratimą buvo analizuojamas galvos stabilumas, alkūnės sulenkimas, pečių pakėlimas, liemens pasukimas ir vertikali laikysena. Daugiausia dėmesio skiriant 0, 4, 5, 6, 22 ir 23 mazgams. Dalyviai stengėsi išlaikyti stabilią galvos padėtį (0 ° pasvirimas), atlikti tinkamą alkūnės ir peties pasvirimą (0–90 ° ir 90–180 °), vengti liemens rotacijos ir išlaikyti 90 ° stuburo pasvirimą. Nulinei klasei būdingi sveiki judesiai, kai galvos pakreipimo vertės yra artimos 0 °, o 1 ir 2 klasėms būdingi persidengiantys alkūnių ir pečių kampų svyravimai, todėl čia sunku atskirti neteisingas formas.



114 pav. „Viršugalvio siekimo ranka“ klasifikavimo sumaišymo matrica

42 lentelė. Sprendimų klasifikavimo rodikliai, taikyti pratimui „Viršugalvio siekimas ranka“

RF klasifikatorius	Reikšmė	CNN klasifikatorius	Reikšmė
Medžių skaičius (OOB)	20	Iteracijų skaičius	11492
Didžiausias gylis	3	Epochų skaičius	13
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,06 s	14,63 s	
Testo tikslumas	95,78 %	100 %	



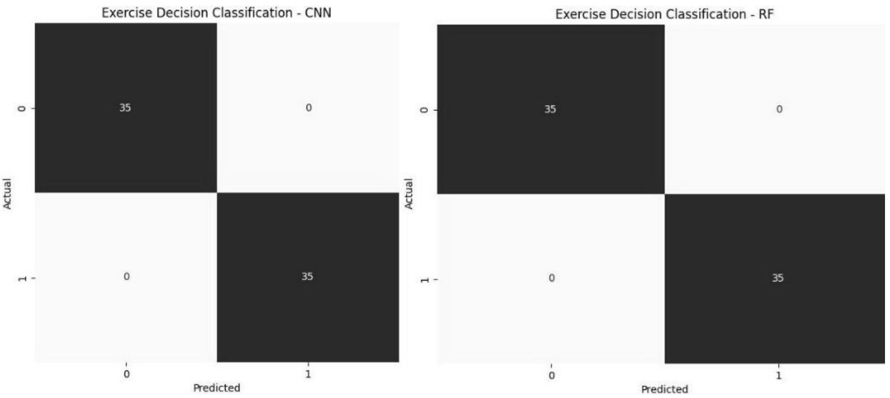
115 pav. CNN klasifikuotų kampų, skirtų pratimui „Viršugalvio siekimas ranka“, sumaišymo matrica

43 lentelė. Kampų klasifikavimo metrikos, skirtos pratimui „Viršugalvio siekimas ranka“ atlikti

CNN klasifikatorius	Reikšmė
Iteracijų skaičius	37100
Epochų skaičius	33
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	36,88 s
Bandymų tikslumas	93,86 %

Sunkaus daikto kėlimas ir padėjimas virš galvos

Atliekant šį pratimą buvo vertinami rankų (4, 5), alkūnių (7, 9), kelių (18, 19) ir stuburo (6) kampai, daugiausia dėmesio skiriant tinkamai koordinacijai keliant ir dedant sunkų objektą virš galvos. Rankų ir alkūnių kampai svyravo nuo 0 ° iki 110 ° ir nuo 0 ° iki 100 °, kelio sulenkimas – nuo 0 ° iki 90 °, o stuburo pasvirimas – nuo 0 ° iki 90 °. Nulinės klasės dalyviai taisyklingai kėlė daiktus, pasižymėjo stabilia laikysena ir sklandžiais, koordinuotais judesiais. 1 klasės dalyviai imitavo per didelį palinkimą į priekį, blogai koordinavo rankų ir kojų judesius arba buvo nestabilūs, taip pažeisdami kėlimo techniką.

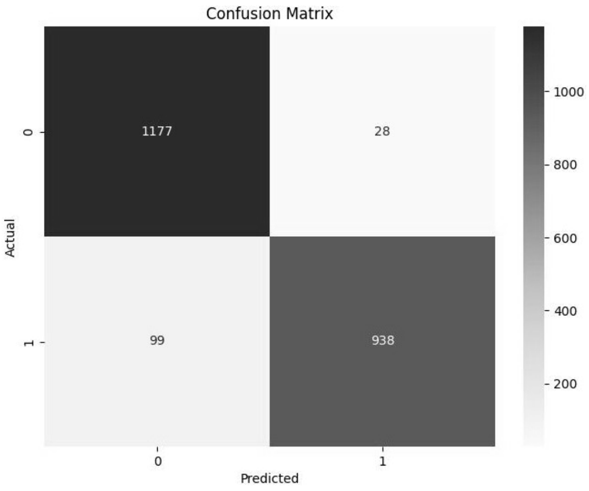


116 pav. Kiekvienos klasės, klasifikuotos atliekant pratimą „Sunkaus daikto kėlimas ir padėjimas virš galvos“, sumaišties matrica

44 lentelė. „Sunkaus daikto kėlimo ir padėjimo virš galvos“ pratimo klasifikavimo rodikliai

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	10	Iteracijų skaičius	8722
Didžiausias gylis	3	Epochų skaičius	14
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,031 s	11,57 s	
Testo tikslumas	100 %	100 %	

Atliekant kampo klasifikavimą per užduotį „Sunkaus daikto kėlimas ir padėjimas virš galvos“, CNN klasifikatorius pasiekė 94,34 % tikslumą, kaip parodyta supainiojimo matricoje (žr. 117 pav.). Nors dauguma kampų buvo klasifikuoti tiksliai, pasitaikė nedidelių klaidų, ypač tarp 0 ir 1 klasių.



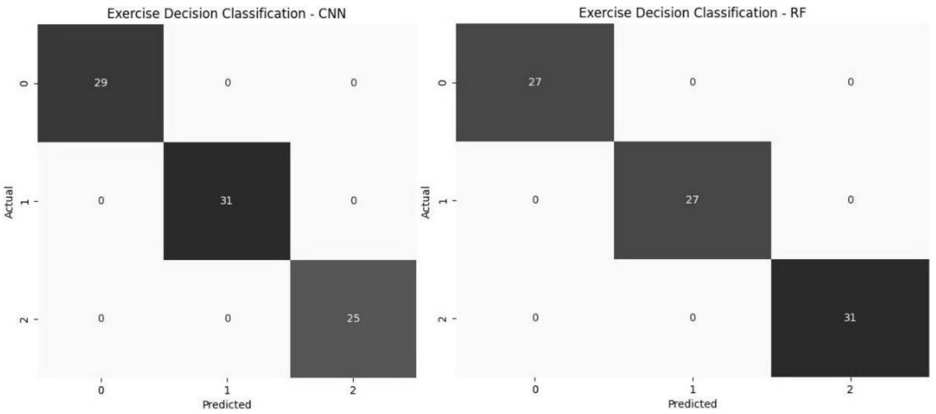
117 pav. Kampų, suklasifikuotų naudojant CNN, sumaišymo matrica atliekant „Sunkaus daikto kėlimo ir padėjimo virš galvos“ pratimą

45 lentelė. Kampų klasifikavimo metrikos atliekant „Sunkaus daikto kėlimo ir padėjimo virš galvos“ užduotį

CNN klasifikatorius	**Reikšmė**
Iteracijų skaičius	34850
Epochų skaičius	11
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	12,8 s
Testo tikslumas	94,34 %

„Ištiestos rankos ašinis sukimas“

Atliekant šį pratimą buvo analizuojami 4, 5, 7, 9, 10 ir 12 mazgai, siekiant įvertinti alkūnės ištiesimą ir rankos sukamąjį judesį, kai pečiai yra vienoje linijoje. Nulinei klasei priskiriami taisyklingai atlikti, visiškai ištiesti ir tinkamai išlyginti rankos sukimai. 1 klasė apima nežymius netaisyklingus judesius, kairioji ranka rodo nedidelius nukrypimus. 2 klasei priskiriamos ryškesnės klaidos, kai pastebimi dideli sukimosi ir padėties išlyginimo nukrypimai.

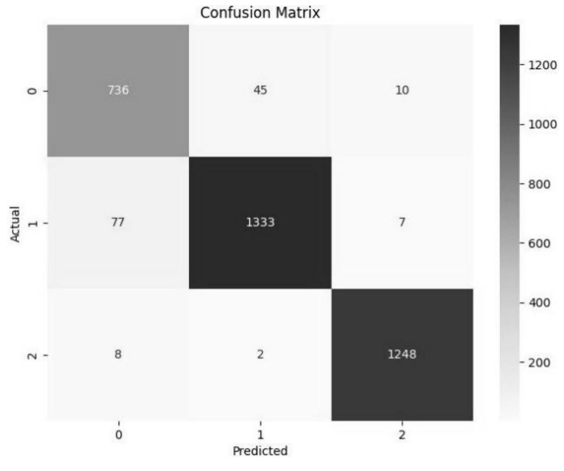


118 pav. Kiekvienos klasės, klasifikuotos atliekant pratimą „Ištiestos rankos ašinis sukimas“ sumaišymo matrica

46 lentelė. „Ištiestos rankos ašinis sukimas“ pratimo sprendimų klasifikavimo rodikliai.

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	20	Iteracijų skaičius	17360
Didžiausias gylis	3	Epochų skaičius	14
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Patvirtinimo padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,031 s	20,84 s	
Testo tikslumas	100 %	100 %	

Atliekant pratimo „Ištiestos rankos ašinis sukimas“ kampų klasifikavimą, CNN klasifikatorius pasiekė 95,7 % bandymo tikslumą, o tai rodo, kad jis geba teisingai klasifikuoti daugumą kampų. Supainiojimo matrica atskleidžia, kad 93,05 % nulinės klasės pavyzdžių buvo teisingai suklasifikuoti, o klaidingo klasifikavimo lygis – 6,95 %, nors dauguma klaidų buvo prognozuojamos kaip 1 klasė. Panašiai 94,54 % 1 klasės pavyzdžių buvo teisingai suklasifikuoti, o klaidingo klasifikavimo lygis siekė 5,46 %. Daugiausia klaidų buvo klasifikuojant kaip 0 klasę. Kalbant apie 2 klasę, modelis veikė itin gerai – teisingai klasifikuota 99,20 % mėginių, o klaidingai klasifikuota tik 0,80 %.



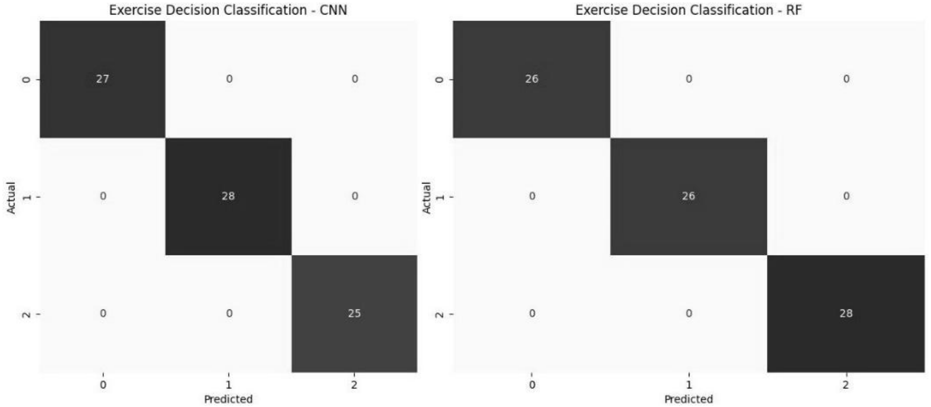
119 pav. Kampų, klasifikuojamų CNN, sumaišymo matrica pratimui „Ištiestos rankos ašinis sukimas“ sumaišymo matrica

47 lentelė. Kampų klasifikavimo metrikos pratimui „Ištiestos rankos ašinis sukimas“.

CNN klasifikatorius	**Reikšmė**
Iteracijų skaičius	52300
Epochų skaičius	14
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	21,83 s
Testo tikslumas	95,7 %

„Nosies palietimo“ koordinavimo pratimas

Atliekant šį pratimą buvo vertinami 4, 5, 7, 9, 0 ir 6 mazgai, daugiausia dėmesio skiriant stacionariai galvos padėčiai, alkūnių lenkimui, rankų judesiams ir vertikaliai laikysenai. Nulinė klasė apima taisyklingą nosies prisilietimo koordinaciją be kompensacinių judesių. 1 klasei priskiriamos tokios klaidos, kaip galvos pakreipimas ar netaisyklingas pečių padėties nustatymas. 2 klasei priskiriamos tokios klaidos, kaip rankos per didelis ištiesimas arba netaisyklingas padėties suderinimas per pakilimo arba nosies palietimo fazę.

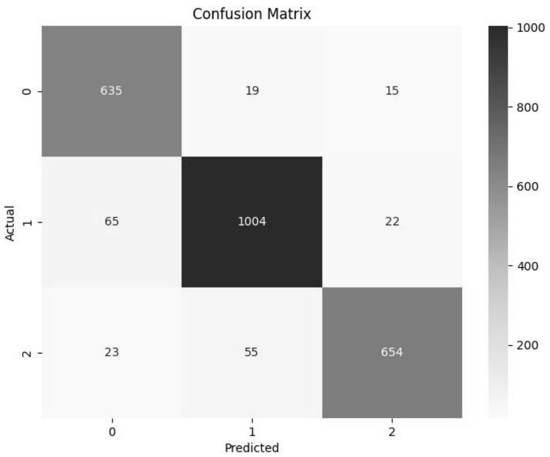


120 pav. Kiekvienos klasės, klasifikuojamos atliekant nosies palietimo koordinacijos pratimą, painiavos matrica

48 lentelė. „Nosies palietimo“ koordinavimo pratimo sprendimų klasifikavimo rodikliai

RF klasifikatorius	**Reikšmė**	**CNN klasifikatorius**	**Reikšmė**
Medžių skaičius (OOB)	10	Iteracijų skaičius	12908
Didžiausias gylis	3	Epochų skaičius	14
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Validacijos padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,03262 s	15,53 s	
Testo tikslumas	100 %	100 %	

Atliekant klasifikavimą pagal kampą, CNN pasiekė 92 % tikslumą (žr. 121 pav.). Supainiojimo matrica atskleidė keletą klaidingų klasifikacijų visose trijose klasėse. Nulinė klasė pasiekė 88,56 % taisyklingo klasifikavimo rodiklį, 65 atvejai neteisingai priskirti 1 klasei, o 15 atvejų netinkamai priskirti 2 klasei. 1 klasė parodė 91,48 % tikslumą, 22 atvejai buvo netinkamai priskirti 2 klasei ir 65 atvejai netinkamai priskirti 0 klasei. 2 klasė parodė 88,62 % taisyklingo klasifikavimo rodiklį, 55 atvejai netinkamai priskirti 1 klasei, o 23 atvejai netinkamai priskirti 0 klasei.



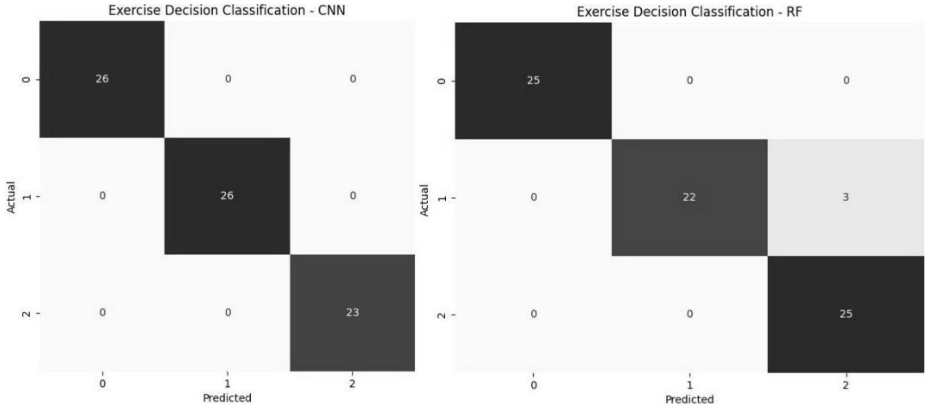
121 pav. Kampų, klasifikuotų CNN, taikant „Nosies palietimo“ koordinavimo pratimą, sumaišymo matrica

49 lentelė. Kampų klasifikavimo metrikos atliekant „Nosies palietimo“ koordinavimo pratimą

CNN klasifikatorius	Reikšmė
Iteracijų skaičius	39050
Epochų skaičius	17
Partijos dydis	6
Patvirtinimo padalijimas	20 %
Vykdymo laikas	27,21s
Testo tikslumas	92 %

Dilbio supinacija / pronacija

Atliekant „Dilbio supinacijos / pronacijos“ pratimą buvo analizuojami 7, 9, 10 ir 12 mazgai, daugiausia dėmesio skiriant alkūnės lenkimui ir dilbio rotacijai. Nulinė klasė atspindi taisyklingus, koordinuotus abiejų rankų, ypač dešiniosios, judesius su nuosekliais kampo perėjimais. 1 klasė apima netaisyklingus kairės rankos judesius su nepilnaverte rotacija arba netaisyklingu alkūnės sulenkimu. 2 klasei priskiriamos ryškesnės kairės rankos klaidos, pavyzdžiui, per didelė rotacija arba staigūs kampo perėjimai.

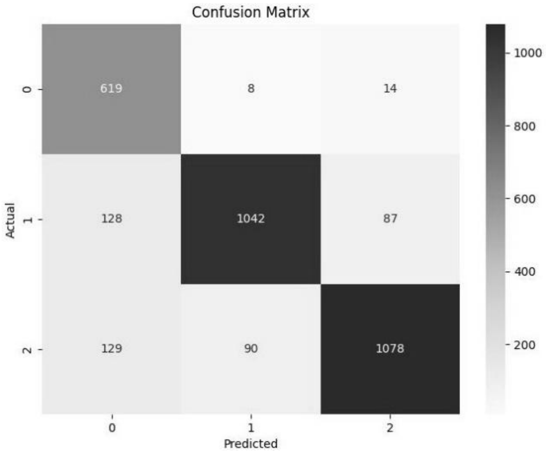


122 pav. Kiekvienos klasės, klasifikuojamos atliekant „Dilbio supinacijos / pronacijos“ pratimą, painiavos matrica

50 lentelė. „Dilbio supinacijos / pronacijos“ pratimo sprendimų klasifikavimo rodikliai

RF klasifikatorius	Reikšmė	CNN klasifikatorius	Reikšmė
Medžių skaičius (OOB)	30	Iteracijų skaičius	16212
Didžiausias gylis	3	Epochų skaičius	14
Mažiausias mėginių skaičius mazgui padalyti	6	Partijos dydis	6
Mažiausias mėginių skaičius vienam lapiniam mazgui	3	Validacijos padalijimas	20 %
Didžiausias nagrinėjamų požymių skaičius	2		
Bendras vykdymo laikas	0,06 s	20,59 s	
Testo tikslumas	96 %	100 %	

Kampų klasifikavimo atveju CNN pasiekė 85,73 % tikslumą, o vykdymo laikas buvo 20,99 sekundės (žr. 123 pav.). Nors dauguma kampų buvo klasifikuoti gerai, pastebimos klaidos tarp 0 ir 1 klasės, taip pat tarp 2 ir 1 klasės. Tiksliau, 87 kampai iš 1 klasės buvo netiksliai priskirti 2 klasei, o 129 kampai iš 2 klasės buvo netiksliai priskirti 0 klasei. Nepaisant šių netikslumų, CNN veiksmingai susidorojo su klasifikavimo užduotimi.



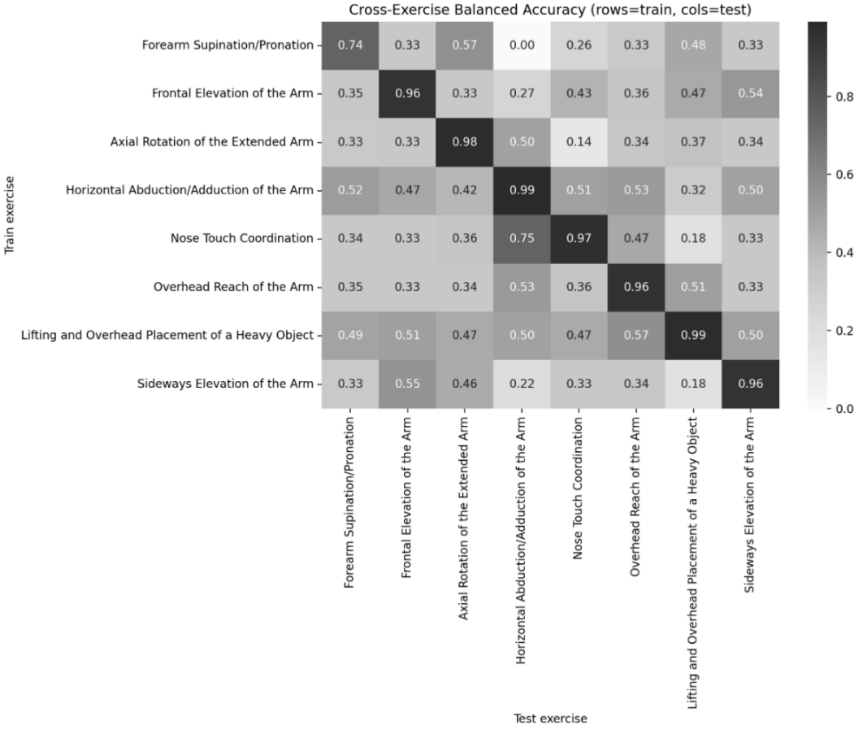
123 pav. CNN klasifikuotų kampų, atliekant „Dilbio supinacijos / pronacijos” pratimą, sumaišymo matrica

51 lentelė. „Dilbio supinacijos / pronacijos“ pratimo kampų klasifikavimo metrikos

CNN klasifikatorius	Reikšmė
Iteracijų skaičius	47400
Epochų skaičius	15
Partijos dydis	6
Validacijos padalijimas	20 %
Vykdymo laikas	20,99s
Testo tikslumas	85,73 %

Atliekamo pratimo nustatymo eksperimentas

Paveiksle pateikta maišaties (angl. *confusion matrix*) matrica (žr. 124 pav.) apibendrina konvoliacinio neuroninio tinklo (CNN) pagrindu veikiančios pratimų nustatymo sistemos tikslumą aštuoniuose reabilitacijos pratimuose. Kiekviena eilutė žymi modelį, apmokytą konkrečiam pratimui, o kiekvienas stulpelis rodo jo testavimo tikslumą vertinant visus kitus pratimus. Įstrižainės elementai atitinka kiekvieno pratimo modelio tikslumą testuojant tą patį pratimą, o reikšmės, nesančios įstrižainėje parodo, kaip vienam pratimui apmokytas modelis geba išsiskirti nuo kitų pratimų.



124 pav. CNN pagrindu veikiančios pratimų aptikimo sistemos tyrimo maišaties matrica

Kiekvienam pratimui atskirai buvo apdoroti duomenų rinkiniai, kuriuose pateikiami pasirinktų skeleto mazgų kampiniai santykiai. Kiekvienas rinkinys buvo padalintas į 70 % mokymo ir 30 % testavimo poaibius pagal unikalias dalyvio pozos kombinacijas. Po mokymo modeliai buvo išsaugoti kartu su jų išankstinio apdorojimo komponentais, kad juos būtų galima pakartotinai naudoti vertinant nematytus duomenis arba atliekant tarp pratimų testavimą be papildomo pertreniravimo. Vertinimo metu kiekvienas apmokytas modelis buvo kryžmiškai išbandytas visuose pratimuose siekiant įvertinti jo gebėjimą tiksliai veikti tik su originalaus pratimo duomenimis. Pateikta matrica patvirtina galimybę rasti atliekamą pratimą bei parodo pratimų panašumo dėsningumus.

Rezultatai

Konvoliaciniai neuroniniai tinklai (CNN) nagrinėtų aštuonių pratimų pasiekė gerą tikslumą (nuo 0,74 iki 0,99), o vidutinis originalaus pratimo atpažinimo tikslumas sudarė 0,92 ± 0,09, kas rodo pratimo identifikavimo nuoseklumą. Tokie judesiai kaip horizontali abdukcija / addukcija, nosies palietimo koordinacija, viršugalvio siekimas ir pakėlimas į šoną pasižymėjo didžiausiu tikslumu (≥ 0,96), kas rodo šių pratimų išskirtinumą.

Vidutinio tikslumo kryžminis atpažinimas pasireiškė biomechaniškai panašiuose pratimuose – pavyzdžiui, keliant ranką į priekį ir keliant ranką į šoną (~0,57) bei atliekant ašinę rotaciją ir dilbio supinaciją / pronaciją (> 0,55). Pratimai su aiškiai išreikšta daugiasąnarine koordinacija, tokie kaip objekto kėlimas virš galvos ar nosies palietimo koordinacija, rodė minimalų persidengimą (< 0,35).

Galima teigti, kad pratimai, kuriuose panašūs sąnarių judesiai, sukimosi ašys ar judesio amplitudės rodo vidutinį kryžminio atpažinimo tikslumą, o tie, kuriems naudojami skirtingi sąnariai ir judesio šablonai, išlieka aiškiai atskiriami.

Tyrimo išvados

Pristatytas žmogaus laikysenos diagnostikos sistemos, kuriose naudojamas „HTC Vive Tracker 2.0“ kaip prieinamas VR įvesties įrenginys, kūrimas, diegimas ir išsamus vertinimas. Sistemos kiekybinis našumas buvo įvertintas atliekant viršutinių galūnių pratimus, integruojant RF ir CNN modelius. Tiek RF, tiek CNN modeliai patikimai klasifikavo įvairius pratimų seansus, įskaitant priekinį rankos pakėlimą, ranko pakėlimą į šoną, viršugalvio siekimą, horizontalią abdukciją / addukciją, ašinę rotaciją ir sudėtingą manipuliavimą daiktais. Abu metodai nuosekliai pasiekė nuo 95,78 % iki 100 % tikslumą, kai buvo klasifikuojama laikysenos kokybė arba sutrikimas. RF modeliams buvo naudojama vos 10–20 medžių (didžiausias gylis – 3), o CNN modeliai konverguodavo per 11–38 epochas. RF klasifikatorių vykdymo laikas nebuvo ilgesnis kaip 0,06 sekundės, o tai rodo didelį skaičiavimo efektyvumą. CNN modeliams reikėjo ilgesnio vykdymo laiko (nuo 11,39 s iki 46,75 s, priklausomai nuo užduoties sudėtingumo), tačiau jie taip pat pasiekė gerų rezultatų vertinant klasifikavimo užduočių tikslumą. Atliekant kampu grindžiamas klasifikavimo užduotis, CNN pasiekė nuo 90,17 % iki 97,77 % testų tikslumą visuose pratimuose, o tai rodo patikimumą atliekant smulkią judesių analizę, nors ir su nedidelėmis klaidomis klasifikuojant panašius kompensacinius šablonus.

Papildomai buvo atliktas tarp pratimų (kryžminis) vertinimas siekiant įvertinti, kaip gerai CNN modeliai prisitaiko prie skirtingų judesių tipų ir kaip tiksliai atpažįsta atliekamą pratimą. Kiekvienas konkrečiam pratimui skirtas CNN modelis buvo išbandytas su visų kitų pratimų duomenų rinkiniais, o gauta maišaties matrica parodė aukštus atpažinimo rodiklius testuojant tą patį pratimą (nuo 0,74 iki 0,99). Vidutinis atpažinimo persidengimas buvo stebimas tarp pratimų, turinčių panašią biomechaninę struktūrą: pavyzdžiui, rankos kėlimo į priekį ir į šoną arba viršutinių galūnių sukamųjų judesių pratimai. Nustatyta, kad nedidelė išskirtis daugiausia kyla dėl tikrų fizinių judesio panašumų, o ne dėl modelio mokymosi proceso apribojimų. Taigi, galima teigti, kad kiekvienas CNN veiksmingai užfiksuoja unikalius kampinius požymius, apibrėžiančius atitinkamą pratimą.

Buvo nustatyti svarbiausi diskriminaciniai požymiai: pavyzdžiui, 2 ir 3 kampas (alkūnės sulenkimas ir pečių sulygiavimas), kai ranka keliama į priekį, ir 5 kampas (stuburo pasvirimas), kai ranka iškeliama virš galvos, buvo labai svarbūs tiksliai nustatant kompensacinius judesius ir klasifikuojant sutrikimus. Požymių svarbos analizė ir paklaidų pasiskirstymas atskleidė, kad modeliai gali patikimai atskirti sveikus ir kompensacinius ar patologinius judesių šablonus pagal konkrečias sąnarių kampų metrikas. Kiekybinė duomenų rinkinio, kurį sudarė iki 85 pavyzdžių kiekvienai pratimų klasei iš 17 dalyvių, analizė parodė, kad sistema išlaikė didelį jautrumą ir specifiškumą klasifikuojant taisyklingus ir netaisyklingus judesius. Nors buvo nustatyta tam tikrų apribojimų, susijusių su kampų pagrindu atliekamu neteisingu klasifikavimu ir tarpasmeniniu kintamumu, bendri rezultatai rodo, kad standartinė VR įrangą ir dirbtinio intelekto algoritmų taikymas yra pakankamas žmogaus laikysenai vertinti.

IŠVADOS

1. Periferiniai įvesties įrenginiai kelia įvairių integracijos iššūkių, daugiausia susijusių su įrenginių suderinamumu, delsa ir gebėjimu užtikrinti įtraukiamą, imitavimą primenančią patirtį virtualioje realybėje. Jie gali padidinti tikroviškumą, tačiau įneša delsos ir sinchronizavimo problemų, kurios sukelia vaizdo trūkčiojimą ir blogina bendrą VR patirtį. Šiems iššūkiams spręsti reikia adaptyvių metodų, kurie mažintų delsą ir didintų sistemos reagavimą, kartu išlaikydami tikslumą ir įtraukimą.
2. Aukštos klasės judesių fiksavimo sistemos užtikrina tikslų sekimą, tačiau yra brangios ir joms reikia specialiai paskirtų, iš anksto paruoštų patalpų. Standartiniai (angl. *off-the-shelf*) VR sekikliai yra nebrangūs ir tinkami pratimams sekti, tačiau jie pasižymi padėties ir kampiniu dreifu, ypač per ilgesnius seansus atliekant didelio greičio, staigius ar smūginius judesius. Ankstesni tyrimai parodė mašininio mokymosi pagrindu veikiančios judesių klasifikacijos potencialą panašiose taikymo srityse. Remiantis tuo, šiame tyrime siūloma viso kūno sekimo ir žmogaus laikysenos analizės sistema, paremta standartinių VR sistemų naudojimu ir mašininio mokymosi algoritmų taikymu klasifikacijai.
3. Siekiant spręsti periferinių įvesties įrenginių integracijos problemas buvo įvertinti keturi prognozavimo metodai, skirti duomenims sinchronizuoti realiuoju laiku tarp periferinės įrangos „Concept-II“ ir virtualiosios realybės programos. Šie metodai prognozuoja trūkstamus arba vėluojančius įvesties duomenis, kad sumažintų juntamą įvesties delsą ir vaizdo trūkčiojimą, kartu išlaikant irklavimo tikslumą. Tyrime taikyti keturi metodai: tiesinė interpolacija pagal padėtį, egztrapoliacija pagal greitį, pasiūlytas egztrapoliacijos pagal greitį su korekcija metodas ir Kalmano filtras.
4. Siekiant užtikrinti reabilitacijos pratimų stebėjimą ir klasifikavimą pasiūlyta sistema, kuri naudoja „HTC Vive“ sekiklius, išdėstytus pagrindiniuose anatominiuose taškuose, viso kūno judesiams fiksuoti. Kiekvienam pratimui tyrimų metu buvo nustatyti sąnarių mazgai, pagal kuriuos skaičiuojami kampai. Šie kampai naudojami kaip įvesties požymiai mašininio mokymosi algoritmams. Sistema taiko adaptuotus mašininio mokymosi metodus, konkrečiai – konvoliucinius neuroninius tinklus (*CNN*) ir atsitiktinį mišką (angl. *Random Forest*) atliekamų pratimų judesiams analizuoti ir klasifikuoti. Taip pat pasiūlytas ir integruotas atliekamo pratimo identifikavimo algoritmas.
5. Buvo įvertinti keturi prognozavimo metodai, kurių tikslas – sumažinti delsą, pašalinti mikrotrūkčiojimus ir pagerinti duomenų tikslumą tarp periferinės įrangos „Concept-II“ ir virtualiosios realybės. Tiesinė interpolacija sukėlė teigiamų laiko poslinkių, vidutiniškai +64,7 ms, tačiau sumažino trūkčiojimų skaičių nuo ≈ 482 (111 ms) iki ≈ 260 (39 ms). Kalmano filtras dar labiau sumažino trūkčiojimus iki ≈ 145 (53 ms), bet įvedė apie ≈ 130 ms laiko

poslinkį. Egztrapoliacija pagal greitį panaikino trūkčiojimą, tačiau bendrai sukaupė didelį neigiamą laiko poslinkį – apie 6,8 s. Pasiūlytas ekstrapoliacijos pagal greitį su korekcija algoritmas reikšmingai sumažino šį poslinkį iki ≈ 29 ms, t. y. 250 m sesija virtualioje realybėje baigiasi beveik tuo pačiu metu kaip ir irklavimo treniruoklyje. Paminėtina, kad šis algoritmas užtikrina VR patirtį be trūkčiojimų (eksperimento metu trūkčiojimų nenustatyta).

6. Atlikus aštuonis viršutinių galūnių pratimus ir „Random Forest“, ir CNN modeliai pasiekė sprendimų lygmens tikslumą nuo 95,78 % iki 100 %, o CNN kampų pagrindu klasifikacijų tikslumas svyravo nuo 85,73 % iki 97,77 %. Vertinant pratimus tarpusavyje, kiekvienas CNN modelis buvo patikrintas su kitų pratimų duomenimis, kad būtų įvertintas savęs atpažinimo tikslumas. Vidutinio stiprumo tarpusavio atpažinimas nustatytas tarp rankos kėlimo į priekį ir rankos kėlimo į šoną – abipusis tikslumas siekė apie 0,57, dėl panašių sąnarių judesių 2 kampu (peties kėlimas) ir 3 kampu (alkūnės išlygiavimas). Sunkaus objekto kėlimo ir uždėjimo virš galvos bei nosies lietimo koordinacijos pratimai pasižymėjo tarpusavio atpažinimu, mažesniu nei 0,35, kas rodė aiškų jų išskyrimą. Rezultatai parodė, kad CNN modeliai sėkmingai atskyrė daugumą pratimų su tik vidutiniu persidengimu tarp biomechaniškai panašių judesių.

LIST OF REFERENCES

1. WIEDERHOLD, Brenda K. and Giuseppe RIVA. Virtual Reality Therapy: Emerging Topics and Future Challenges. In *Cyberpsychology, Behavior, and Social Networking*. 2019, Vol. 22, no. 1, p. 3–6. Available from: https://doi.org/10.1089/cyber.2018.29136.bkw
2. THURNER, Thomas Wolfgang. Ekaterina DYACHENKO. and Asia MIRONENKO. Diffusion of new technologies in medical research: the case of virtual reality. In *Health and Technology*. 2020, Vol. 10, no. 5, p. 1221–1227. Available from: https://doi.org/10.1007/s12553-020-00416-7
3. SLATER, Mel. and Maria V. SANCHEZ-VIVES. Enhancing Our Lives with Immersive Virtual Reality. In *Frontiers in Robotics and AI* [online]. 2016, Vol. 3. [viewed 2025-01-12]. Available from: https://doi.org/10.3389/frobt.2016.00074
4. STEED, Anthony. Ye PAN. Fiona ZISCH. and others. The impact of a self-avatar on cognitive load in immersive virtual reality. In *2016 IEEE Virtual Reality (VR)*. 2016, p. 67–76. Available from: https://doi.org/10.1109/VR.2016.7504689
5. SHOIB, Nur. Mohd Shahrizal SUNAR. Nurshamine MOHD NOR. and others. Rowing Simulation using Rower Machine in Virtual Reality. In *2020 6th International Conference on Interactive Digital Media (ICIDM)*. 2020, p. 1–6. Available from: https://doi.org/10.1109/ICIDM51048.2020.9339603
6. MERKER, Sebastian. Stefan PASTEL. Dan BÜRGER. and others. Measurement Accuracy of the HTC VIVE Tracker 3.0 Compared to Vicon System for Generating Valid Positional Feedback in Virtual Reality. In *Sensors*. 2023, Vol. 23, no. 17, 7371. Available from: https://doi.org/10.3390/s23177371
7. VOX, Jan P. Anika WEBER. Karen Insa WOLF. and others. An Evaluation of Motion Trackers with Virtual Reality Sensor Technology in Comparison to a Marker-Based Motion Capture System Based on Joint Angles for Ergonomic Risk Assessment. In *Sensors*. 2021, Vol. 21, no. 9, 3145. Available from: https://doi.org/10.3390/s21093145
8. ABIDI, Mustufa Haider. Abdulrahman AL-AHMARI. Ali AHMAD. and others. Assessment of virtual reality-based manufacturing assembly training system. In *International Journal of Advanced Manufacturing Technology*. 2019, Vol. 105, no. 9, p. 3743–3759. Available from: https://doi.org/10.1007/s00170-019-03801-3
9. JEON, Seunggon. Seungwon PAIK. Ungyeon YANG. and others. The More, the Better? Improving VR Firefighting Training System with Realistic Firefighter Tools as Controllers. In *Sensors*. 2021, Vol. 21, no. 21, 7193. Available from: https://doi.org/10.3390/s21217193
10. CANBULUT, Cenker. and Tomas BLAŽAUSKAS. Using Virtual Reality Technologies for Full-Body Tracking in Ice Hockey Training. In NOSENKO, Y. G., Ed. *Immersive Technologies in Education: Proceedings of the IV International Scientific and Practical Conference*. Institute of Digitalization of Education, NAES of Ukraine, 2024, p. 4–8. Available from: https://doi.org/10.33407/lib.NAES.id/eprint/742688
11. BAUER, Peter. Werner LIENHART. and Samuel JOST. Accuracy Investigation of the Pose Determination of a VR System. In *Sensors*. 2021, Vol. 21, no. 5, p. 1–17. Available from: https://doi.org/10.3390/s21051622
12. KHUNDAM, Chaowanan. Varunyu VORACHART. Patibut PREEYAWONGSAKUL. and others. A Comparative Study of Interaction Time and Usability of Using Controllers and Hand Tracking in Virtual Reality Training. In

Informatics. 2021, Vol. 8, no. 3, 60. Available from: https://doi.org/10.3390/informatics8030060

13. WU, Shengzhi. Siyu CHEN. Mong-Yah HSIEH. and others. The other way: immersive VR storytelling through biking. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. 2020, p. 852. Available from: https://doi.org/10.1109/VRW50115.2020.00277
14. HETRICK, Rebecca. Nicholas AMERSON. Boyoung KIM. and others. Comparing Virtual Reality Interfaces for the Teleoperation of Robots. In *2020 Systems and Information Engineering Design Symposium (SIEDS)*. 2020, p. 1–7. Available from: https://doi.org/10.1109/SIEDS49339.2020.9106630
15. ALLGAIER, Mareen. Vuthea CHHEANG. Patrick SAALFELD. and others. A comparison of input devices for precise interaction tasks in VR-based surgical planning and training. In *Computers in Biology and Medicine*. 2022, Vol. 145, 105467. Available from: https://doi.org/10.1016/j.compbiomed.2022.105429
16. WONSICK, Murphy. and Taskin PADIR. A Systematic Review of Virtual Reality Interfaces for Controlling and Interacting with Robots. In *Applied Sciences*. 2020, Vol. 10, no. 24, 9051. Available from: https://doi.org/10.3390/app10249051
17. MONETTI, F. M. A. De GIORGIO. H. YU. and others. An experimental study of the impact of virtual reality training on manufacturing operators on industrial robotic tasks. In *Procedia CIRP*. Elsevier B.V., 2022, p. 33–38. Available from: https://doi.org/10.1016/j.procir.2022.02.151
18. KIM, Hojoong. Young-Tae KWON. Hyo-Ryoung LIM. and others. Recent Advances in Wearable Sensors and Integrated Functional Devices for Virtual and Augmented Reality Applications. In *Advanced Functional Materials* [online]. 2021, Vol. 31, no. 39, 2005692. [viewed 2025-11-20]. Available from: https://doi.org/10.1002/adfm.202005692
19. CASERMAN, Polona. Augusto GARCIA-AGUNDEZ. Robert KONRAD. and others. Real-time body tracking in virtual reality using a Vive tracker. In *Virtual Reality*. 2019, Vol. 23, no. 2, p. 155–168. Available from: https://doi.org/10.1007/s10055-018-0374-z
20. WARBURTON, Matthew. Mark MON-WILLIAMS. Faisal MUSHTAQ. and others. Measuring motion-to-photon latency for sensorimotor experiments with virtual reality systems. In *Behavior Research Methods*. 2023, Vol. 55, no. 7, p. 3658–3678. Available from: https://doi.org/10.3758/s13428-022-01983-5
21. CHEN, Mingzhe. Omid SEMIARI. Walid SAAD. and others. Federated Echo State Learning for Minimizing Breaks in Presence in Wireless Virtual Reality Networks. In *IEEE Transactions on Wireless Communications*. 2020, Vol. 19, no. 1, p. 177–191. Available from: https://doi.org/10.1109/TWC.2019.2942929
22. GANAL, Elisabeth. Andrea BARTL. Franziska WESTERMEIER. and others. Developing a Study Design on the Effects of Different Motion Tracking Approaches on the User Embodiment in Virtual Reality. In HANSEN, CHRISTIAN - NÜRNBERGER, ANDREAS - PREIM, BERNHARD, Eds. *Mensch und Computer 2020 – Tagungsband* [online]. Gesellschaft für Informatik e.V., 2020, p. 177–181. [viewed 2025-11-20]. Available from: http://dblp.uni-trier.de/db/conf/mc/mc2020w.html#GanalBW0L20.
23. VEEN, Susanne M. Van der. Martine BORDELEAU. Peter E. PIDCOE. and others. Agreement analysis between Vive and Vicon systems to monitor lumbar postural changes. In *Sensors (Switzerland)*. 2019, Vol. 19, no. 17, 3729. Available from: https://doi.org/10.3390/s19173632

24. MICHAELI, Tomer. Volker POHL. and Yonina C. ELDAR. U-invariant sampling: extrapolation and causal interpolation from generalized samples. In *IEEE Transactions on Signal Processing*. 2011, Vol. 59, no. 5, p. 2085–2100. Available from: https://doi.org/10.1109/TSP.2011.2113342
25. BRUNNSTRÖM, Kjell. Elijs DIMA. Tahir QURESHI. and others. Latency impact on Quality of Experience in a virtual reality simulator for remote control of machines. In *Signal Processing: Image Communication*. 2020, Vol. 89, 116005. Available from: https://doi.org/10.1016/j.image.2020.116005
26. GUTMANN, Gregory. and Akihiko KONAGAYA. Predictive Simulation: Using Regression and Artificial Neural Networks to Negate Latency in Networked Interactive Virtual Reality [online]. 2019. [viewed 2025-11-20]. Available from: https://arxiv.org/abs/1910.04703.
27. ISKANDAR, Yulita Hanum P. Lester GILBERT. and Gary B. WILLS. Reducing latency when using Virtual Reality for teaching in sport. In *International Symposium on Information Technology 2008 (ITSIM)*. 2008, p. 1–5. Available from: https://doi.org/10.1109/ITSIM.2008.4632076
28. GUNN, Therese. Lee JONES. Pete BRIDGE. and others. The use of virtual reality simulation to improve technical skill in the undergraduate medical imaging student. In *Interactive Learning Environments*. 2018, Vol. 26, no. 5, p. 613–620. Available from: https://doi.org/10.1080/10494820.2017.1374981
29. SCHRÖDER, Jonas. Tamaya Van CRIEKINGE. Elissa EMBRECHTS. and others. Combining the benefits of tele-rehabilitation and virtual reality-based balance training: a systematic review on feasibility and effectiveness. In *Disability and rehabilitation. Assistive technology*. 2018, Vol. 14, no. 1, p. 2–11. Available from: https://doi.org/10.1080/17483107.2018.1503738
30. KIKUCHI, Atsushi. Tatsunori TANIGUCHI. Kei NAKAMOTO. and others. Feasibility of home-based cardiac rehabilitation using an integrated telerehabilitation platform in elderly patients with heart failure: A pilot study. In *Journal of Cardiology*. 2021, Vol. 78, no. 1, p. 66–71. Available from: https://doi.org/10.1016/j.jjcc.2021.01.010
31. POURNAJAF, Sanaz. Michela GOFFREDO. Leonardo PELLICCIARI. and others. Effect of balance training using virtual reality-based serious games in individuals with total knee replacement: A randomized controlled trial. In *Annals of Physical and Rehabilitation Medicine* [online]. 2022, Vol. 65, no. 6, 101609. Available from: https://doi.org/10.1016/j.rehab.2021.101609
32. SU, Mu Chun. Pang Ti TAI. Jieh Haur CHEN. and others. A Projection-Based Human Motion Recognition Algorithm Based on Depth Sensors. In *IEEE Sensors Journal*. 2021, Vol. 21, no. 15, p. 16990–16996. Available from: https://doi.org/10.1109/JSEN.2021.3079983
33. MENG, Zhaozong. Mingxing ZHANG. Changxin GUO. and others. Recent Progress in Sensing and Computing Techniques for Human Activity Recognition and Motion Analysis. In *Electronics*. 2020, Vol. 9, no. 9, 1357. Available from: https://doi.org/10.3390/electronics9091357
34. KULIKAJEVAS, Audrius. Rytis MASKELIUNAS. Robertas DAMASEVICIUS. and others. Humannet-a two-tiered deep neural network architecture for self-occluding humanoid pose reconstruction. In *Sensors*. 2021, Vol. 21, no. 12, 3945. Available from: https://doi.org/10.3390/s21123945

35. EMURA, Satoru. and Susumu TACHI. Multisensor integrated prediction for virtual reality. In *Presence: Teleoperators and Virtual Environments*. 1998, Vol. 7, no. 4, p. 410–422. Available from: https://doi.org/10.1162/105474698565811
36. SCHRAMKA, Filip. Stefan ARISONA. Michael JOOS. and others. Development of virtual reality cycling simulator. In *Journal of Computers*. 2018, Vol. 13, no. 6, p. 603–615. Available from: https://doi.org/10.3929/ethz-b-000207175
37. HERNÁNDEZ–MELGAREJO, G. D. A. FLORES–HERNÁNDEZ. A. LUVIANO–JUÁREZ. and others. Mechatronic design and implementation of a bicycle virtual reality system. In *ISA Transactions*. 2020, Vol. 97, p. 336–351. Available from: https://doi.org/10.1016/j.isatra.2019.08.002
38. WANG, Ziyao. Chiyi LIU. Jialiang CHEN. and others. Strolling in Room-Scale VR: Hex-Core-MK1 Omnidirectional Treadmill. In *IEEE Transactions on Visualization and Computer Graphics*. 2023, Vol. 29, no. 12, p. 5538–5555. Available from: https://doi.org/10.1109/TVCG.2022.3216211
39. PARK, Hyung-Soon. Jung Won YOON. Jonghyun KIM. and others. Development of a VR-based treadmill control interface for gait assessment of patients with Parkinson's disease. In *2011 IEEE International Conference on Rehabilitation Robotics* [online]. Zurich: IEEE, 2011, p. 1–5. [viewed 2025-11-20]. Available from: https://doi.org/10.1109/ICORR.2011.5975463
40. BUCKINGHAM, Gavin. Hand Tracking for Immersive Virtual Reality: Opportunities and Challenges. In *Frontiers in Virtual Reality*. 2021, Vol. 2. Available from: https://doi.org/10.3389/frvir.2021.728461
41. RAHIMIAN, Pooya. Jodie M. PLUMERT. and Joseph K. KEARNEY. The Effect of Visuomotor Latency on Steering Behavior in Virtual Reality. In *Frontiers in Virtual Reality* [online]. 2021, Vol. 2, 727858. [viewed 2025-11-20]. Available from: https://doi.org/10.3389/frvir.2021.727858
42. ZHANG, Linyuan. Guoru DING. Qihui WU. and others. Defending Against Byzantine Attack in Cooperative Spectrum Sensing: Defense Reference and Performance Analysis. In *IEEE Access*. 2016, Vol. 4, p. 4011–4024. Available from: https://doi.org/10.1109/ACCESS.2016.2593952
43. MAROUGKAS, Andreas. Christos TROUSSAS. Akrivi KROUSKA. and others. Virtual Reality in Education: A Review of Learning Theories, Approaches and Methodologies for the Last Decade. In *Electronics*. 2023, Vol. 12, no. 13, 2832. Available from: https://doi.org/10.3390/electronics12132832
44. KUNDU, Ripan Kumar. Akhlaqur RAHMAN. and Shuva PAUL. A Study on Sensor System Latency in VR Motion Sickness. In *Journal of Sensor and Actuator Networks*. 2021, Vol. 10, no. 3, 45. Available from: https://doi.org/10.3390/jsan10030053
45. CANBULUT, Cenker. Andrius PAULAUSKAS. and Tomas BLAZAUSKAS. Prediction of pending data using interpolation and extrapolation techniques for virtual rowing. In *International Journal of Computers, Communications and Control*. 2020, Vol. 15, no. 2, p. 1–15. Available from: https://doi.org/10.15837/IJCCC.2020.2.3778
46. GAMAGE, Nisal Menuka. Deepana ISHTAWEERA. Martin WEIGEL. and others. So Predictable! Continuous 3D Hand Trajectory Prediction in Virtual Reality. In *UIST 2021*. Association for Computing Machinery, Inc, 2021, p. 332–343. Available from: https://doi.org/10.1145/3472749.3474753
47. HENDERSON, Jeffrey. Joan CONDELL. James CONNOLLY. and others. Review of Wearable Sensor-Based Health Monitoring Glove Devices for Rheumatoid Arthritis. In *Sensors*. 2021, Vol. 21, no. 16, 5352. Available from: https://doi.org/10.3390/s21051576

48. BANERJEE, Premankur. Jason CHERIN. Jayati UPADHYAY. and others. Perception and Control of Surfing in Virtual Reality using a 6-DoF Motion Platform. In *Advances in Computer Entertainment (ACE 2024)*. Springer, Cham, 2024, p. 512–536. Available from: https://doi.org/10.1007/978-3-031-70061-3_38
49. PRADA, John David Prieto. Jintaek IM. Hyondong OH. and others. Enhanced location tracking in sensor fusion-assisted virtual reality micro-manipulation environments. In *PLoS ONE*. 2022, Vol. 16, no. 12, p. 1–15. Available from: https://doi.org/10.1371/journal.pone.0260836
50. LIAO, Chung Shu. Chih Fang HUANG. and Wei Hua CHIENG. A novel washout filter design for a six degree-of-freedom motion simulator. In *JSME International Journal, Series C: Mechanical Systems, Machine Elements and Manufacturing*. 2004, Vol. 47, no. 2, p. 626–636. Available from: https://doi.org/10.1299/jsmec.47.626
51. SIM, Donghyun. Yoonchul BAEK. Minjeong CHO. and others. Low-latency haptic open glove for immersive virtual reality interaction. In *Sensors*. 2021, Vol. 21, no. 11, 3924. Available from: https://doi.org/10.3390/s21113682
52. ILLAHI, Gazi Karam. Ashutosh VAISHNAV. Teemu KÄMÄRÄINEN. and others. Learning to Predict Head Pose in Remotely-Rendered Virtual Reality. In *MMSys 2023 - Proceedings of the 14th ACM Multimedia Systems Conference*. Association for Computing Machinery, 2023, p. 27–38. Available from: https://doi.org/10.1145/3587819.3590972
53. LOI, Iliana. Evangelia I. ZACHARAKI. and Konstantinos MOUSTAKAS. Machine Learning Approaches for 3D Motion Synthesis and Musculoskeletal Dynamics Estimation: A Survey. In *IEEE Transactions on Visualization and Computer Graphics*. 2024, Vol. 30, no. 8, p. 5810–5829. Available from: https://doi.org/10.1109/TVCG.2023.3308753
54. BREMER, Gianni. Niklas STEIN. and Markus LAPPE. Predicting Future Position From Natural Walking and Eye Movements with Machine Learning. In *2021 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. 2021, p. 19–28. Available from: https://doi.org/10.1109/AIVR52153.2021.00013
55. POHL, Sebastian. Armin BECHER. Thomas GRAUSCHOPF. and others. Neural network 3D body pose tracking and prediction for motion-to-photon latency compensation in distributed virtual reality. In *Lecture Notes in Computer Science*. Springer Verlag, 2019, p. 429–442. Available from: https://doi.org/10.1007/978-3-030-30508-6_35
56. CHI, Seunggeun. Hyung-gun CHI. Qixing HUANG. and others. InfoGCN++: Learning Representation by Predicting the Future for Online Human Skeleton-based Action Recognition. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2025, Vol. 47, no. 1, p. 514–528. Available from: https://doi.org/10.1109/TPAMI.2024.3466212
57. ZHAO, Dongming. Tiantian YANG. Wen OU. and others. Autopilot Design for Unmanned Surface Vehicle based on CNN and ACO. In *International Journal of Computers Communications & Control*. 2018, Vol. 13, no. 3, p. 429–439. Available from: https://doi.org/10.15837/ijccc.2018.3.3236
58. REIS, Arsénio. Jorge LAINS. Hugo PAREDES. and others. Developing a System for Post-Stroke Rehabilitation: An Exergames Approach. In ANTONA, M. - STEPHANIDIS, C., Eds. *Universal Access in Human-Computer Interaction. Users and Context Diversity*. Cham: Springer International Publishing, 2016, p. 403–413. Available from: https://doi.org/10.1007/978-3-319-40238-3_39

59. GOELDNER, Moritz. Cornelius HERSTATT. Helena CANHÃO. and others. User Entrepreneurs for Social Innovation: The Case of Patients and Caregivers as Developers of Tangible Medical Devices [online]. Hamburg University of Technology (TUHH), Institute for Technology and Innovation Management, 2019 [viewed 2025-03-19]. Available from: https://ssrn.com/abstract=3461916.
60. JUNG, Hyunwoo. Jae Gyeong JEONG. Youn Soo CHEONG. and others. The effectiveness of computer-assisted cognitive rehabilitation and the degree of recovery in patients with traumatic brain injury and stroke. In *Journal of Clinical Medicine*. 2021, Vol. 10, no. 24, 5902. Available from: https://doi.org/10.3390/jcm10245728
61. PERETTI, Alessandro. Francesco AMENTA. Seyed Khosrow TAYEBATI. and others. Telerehabilitation: Review of the state-of-the-art and areas of application. In *JMIR Rehabilitation and Assistive Technologies*. 2017, Vol. 4, no. 2, e7. Available from: https://doi.org/10.2196/rehab.7511
62. UCCHEDDU, Francesca. Lapo GOVERNI. Rocco FURFERI. and others. Home physiotherapy rehabilitation based on RGB-D sensors: a hybrid approach to the joints angular range of motion estimation. In *International Journal on Interactive Design and Manufacturing*. 2021, Vol. 15, no. 1, p. 99–102. Available from: https://doi.org/10.1007/s12008-020-00728-y
63. VANAGAS, Giedrius. Rolf ENGELBRECHT. Robertas DAMAŠEVIČIUS. and others. EHealth Solutions for the Integrated Healthcare. In *Journal of Healthcare Engineering*. 2018, Vol. 2018, 3846892. Available from: https://doi.org/10.1155/2018/3846892
64. LI, Miaoyu. Zhuohan JIANG. Yutong LIU. and others. Sitsen: Passive sitting posture sensing based on wireless devices. In *International Journal of Distributed Sensor Networks*. 2021, Vol. 17, no. 7. Available from: https://doi.org/10.1177/15501477211024846
65. KULIKAJEVAS, Audrius. Rytis MASKELIUNAS. and Robertas DAMAŠEVIČIUS. Detection of sitting posture using hierarchical image composition and deep learning. In *PeerJ Computer Science*. 2021, Vol. 7, p. 1–20. Available from: https://doi.org/10.7717/peerj-cs.442
66. PALESTRA, Giuseppe. Mohamed REBIAI. Estelle COURTIAL. and others. Evaluation of a rehabilitation system for the elderly in a day care center. In *Information (Switzerland)*. 2018, Vol. 10, no. 1, 3. Available from: https://doi.org/10.3390/info10010003
67. KRISTOFFERSSON, Annica. and Maria LINDÉN. A Systematic Review of Wearable Sensors for Monitoring Physical Activity. In *Sensors (Basel)*. 2022, Vol. 22, no. 2, 573.
68. BOUKHENNOUFA, Issam. Xiaojun ZHAI. Victor UTTI. and others. Wearable sensors and machine learning in post-stroke rehabilitation assessment: A systematic review. In *Biomedical Signal Processing and Control* [online]. 2022, Vol. 71, 103197. [viewed 2023-12-08]. Available from: https://doi.org/10.1016/j.bspc.2021.103197
69. CAPECCI, Marianna. Maria Gabriella CERAVOLO. Francesco FERRACUTI. and others. The KIMORE Dataset: Kinematic Assessment of Movement and Clinical Scores for Remote Monitoring of Physical Rehabilitation. In *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. 2019, Vol. 27, no. 7, p. 1436–1448. Available from: https://doi.org/10.1109/TNSRE.2019.2923060
70. WANG, Lina. Jiangbo LIU. and Jian LAN. Feature Evaluation of Upper Limb Exercise Rehabilitation Interactive System Based on Kinect. In *IEEE Access*. 2019,

Vol. 7, p. 165985–165996. Available from: https://doi.org/10.1109/ACCESS.2019.2953228

71. SARSFIELD, Joe. David BROWN. Nasser SHERKAT. and others. Clinical assessment of depth sensor based pose estimation algorithms for technology supervised rehabilitation applications. In *International Journal of Medical Informatics*. 2019, Vol. 121, p. 30–38. Available from: https://doi.org/10.1016/j.ijmedinf.2018.11.001
72. STRĄCZKIEWICZ, Marcin. Peter JAMES. Marcin STRACZKIEWICZ. and others. A systematic review of smartphone-based human activity recognition for health research. In *npj Digital Medicine* [online]. Vol. 4, 148. [viewed 2023-01-04]. Available from: https://doi.org/10.48550/arXiv.1910.03970
73. MEEGAHAPOLA, Lakmal. and Daniel GATICA-PEREZ. Smartphone Sensing for the Well-Being of Young Adults: A Review. In *IEEE Access*. 2021, Vol. 9, p. 3374–3399. Available from: https://doi.org/10.1109/ACCESS.2020.3045935
74. CLARK, Ross A. Stephanie VERNON. Benjamin F. MENTIPLAY. and others. Instrumenting gait assessment using the Kinect in people living with stroke: reliability and association with balance tests. In *Journal of NeuroEngineering and Rehabilitation*. 2015, Vol. 12, 15. Available from: https://doi.org/10.1186/s12984-015-0006-8
75. RYSELIS, Karolis. Tautvydas PETKUS. Tomas BLAŽAUSKAS. and others. Multiple Kinect based system to monitor and analyze key performance indicators of physical training. In *Human-centric Computing and Information Sciences*. 2020, Vol. 10, no. 1, 51. Available from: https://doi.org/10.1186/s13673-020-00256-4
76. CAMALAN, Seda. Gokhan SENGUL. Sanjay MISRA. and others. Gender detection using 3d anthropometric measurements by kinect. In *Metrology and Measurement Systems*. 2018, Vol. 25, no. 2, p. 253–267. Available from: https://doi.org/10.24425/119568
77. ALI, Syed Farooq. Ahmed Sohail ASLAM. Mazhar Javed AWAN. and others. Pose Estimation of Driver's Head Panning Based on Interpolation and Motion Vectors under a Boosting Framework. In *Applied Sciences*. 2021, Vol. 11, no. 24, 11600. Available from: https://doi.org/10.3390/app112411600
78. SHARIF, Muhammad Imran. Muhammad Attique KHAN. Abdullah ALQAHTANI. and others. Deep Learning and Kurtosis-Controlled, Entropy-Based Framework for Human Gait Recognition Using Video Sequences. In *Electronics (Switzerland)*. 2022, Vol. 11, no. 3, 334. Available from: https://doi.org/10.3390/electronics11030334
79. CHEN, Yen Lin. Chin Hsuan LIU. Chao Wei YU. and others. An upper extremity rehabilitation system using efficient vision-based action identification techniques. In *Applied Sciences (Switzerland)*. 2018, Vol. 8, no. 7, 1161. Available from: https://doi.org/10.3390/app8071161
80. AYED, Ines. Antoni JAUME-I-CAPÓ. Pau MARTÍNEZ-BUESO. and others. Balance measurement using microsoft kinect v2: Towards remote evaluation of patient with the functional reach test. In *Applied Sciences (Switzerland)*. 2021, Vol. 11, no. 13, 6073. Available from: https://doi.org/10.3390/app11136073
81. SAINI, Rajkumar. Pradeep KUMAR. Barjinder KAUR. and others. Kinect sensor-based interaction monitoring system using the BLSTM neural network in healthcare. In *International Journal of Machine Learning and Cybernetics*. 2019, Vol. 10, no. 9, p. 2529–2540. Available from: https://doi.org/10.1007/s13042-018-0887-5
82. XU, Ganbin. and Jianwei LIN. Automatic motion recognition technology based on fuzzy clustering algorithm and VR video image. In *Alexandria Engineering Journal*. 2025, Vol. 124, p. 462–469. Available from: https://doi.org/10.1016/j.aej.2025.03.083

83. MUNTEAN, Emanuel. and Monica LEBA. Comparative Analysis of Wearable and Computer Vision-Based Methods for Human Torso Posture Estimation. In *9th International Conference on Mathematics and Computers in Sciences and Industry (MCSI)*. 2024, p. 56–61. Available from: https://doi.org/10.1109/MCSI63438.2024.00017
84. DESHPANDE, Uttam U. Veena KANGRALKAR. Rudragoud PATIL. and others. A review of Machine Learning Techniques for Ergonomic Risk Assessment based on Human Pose Estimation. In *Discover Artificial Intelligence*. 2025, Vol. 5, 287.
85. MASKELIŪNAS, Rytis. Robertas DAMAŠEVIČIUS. Tomas BLAŽAUSKAS. and others. BiomacVR: A Virtual Reality-Based System for Precise Human Posture and Motion Analysis in Rehabilitation Exercises Using Depth Sensors. In *Electronics (Switzerland)*. 2023, Vol. 12, no. 2, 339. Available from: https://doi.org/10.3390/electronics12020339
86. NICOLAE-ADRIAN, Jurjiu. Avram CLAUDIU. Vutan ANA-MARIA. and others. A systematic review of integrated machine learning in posture recognition. In *Timisoara Physical Education and Rehabilitation Journal*. 2021, Vol. 14, no. 27, p. 15–20. Available from: https://doi.org/10.2478/tperj-2021-0009
87. THAOKAR, Chetana B. Vinay V. DONGRE. Himanshu H. HIWARE. and others. Body Posture Recognition and Fitness Training. In *OCIT 2023 - 21st International Conference on Information Technology, Proceedings*. Institute of Electrical and Electronics Engineers Inc., 2023, p. 927–932. Available from: https://doi.org/10.1109/OCIT59427.2023.10430680
88. SRIVASTAVA, Nitish. Geoffrey HINTON. Alex KRIZHEVSKY. and others. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research*. 2014, Vol. 15, p. 1929–1958.
89. YANG, Ziqian. Dechuan SONG. Jiachuan NING. and others. A Systematic Review: Advancing Ergonomic Posture Risk Assessment through the Integration of Computer Vision and Machine Learning Techniques. In *IEEE Access*. 2024, Vol. 12, p. 180481–180519. Available from: https://doi.org/10.1109/ACCESS.2024.3509447
90. SARDAR, Suman Kalyan. and Seul Chan LEE. An ergonomic evaluation using a deep learning approach for assessing postural risks in a virtual reality-based smart manufacturing context. In *Ergonomics*. 2024, Vol. 67, no. 11, p. 1715–1728. Available from: https://doi.org/10.1080/00140139.2024.2349757
91. SHEN, Jing. and Ling CHEN. Application of human posture recognition and classification in performing arts education. In *IEEE Access*. 2024, Vol. 12, p. 125906–125919. Available from: https://doi.org/10.1109/ACCESS.2024.3451172
92. LI, Hongyang. Mujun LIU. Yichuan DENG. and others. Automated Kinect-based posture evaluation method for work-related musculoskeletal disorders of construction workers. In *Journal of Asian Architecture and Building Engineering*. 2025, Vol. 24, no. 3, p. 1731–1743. Available from: https://doi.org/10.1080/13467581.2024.2357202
93. XIE, Fangjuan. Intelligent Recognition Method of Multitype Human Posture Based on Deep Learning. In *Academic Journal of Computing & Information Science*. 2022, Vol. 5, no. 11, p. 68–72. Available from: https://doi.org/10.25236/ajcis.2022.051110
94. BASKAR, Ashwin. and D. RAJESWARI. Posture Similarity Index (PSI) for Ergonomics Using Joint Angle and Neural Network. In *International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*. Chennai, India: Institute of Electrical and Electronics Engineers Inc., 2023, p. 1–5. Available from: https://doi.org/10.1109/ACCAI58221.2023.10200969

95. LEE, Juyoung. Sang Chul AHN. and Jae In HWANG. A walking-in-place method for virtual reality using position and orientation tracking. In *Sensors (Switzerland)*. 2018, Vol. 18, no. 9, 2832. Available from: https://doi.org/10.3390/s18092832

96. HE, Songtao. Yunxin LIU. and Hucheng ZHOU. Optimizing Smartphone Power Consumption through Dynamic Resolution Scaling. In *21st Annual International Conference on Mobile Computing and Networking - MobiCom '15*. 2015, p. 27–39. Available from: https://doi.org/10.1145/2789168.2790117

97. PAPSON, Scott. Joseph OAGARO. Robi POLIKAR. and others. A virtual reality environment for multi-sensor data integration. In *ISA/IEEE Sensors for Industry Conference*. 2004, p. 116–122. Available from: https://doi.org/10.1109/SFICON.2004.1287142.

98. STURGES, Herbert A. The Choice of a Class Interval. In *Journal of the American Statistical Association*. 1926, Vol. 21, no. 153, p. 65–66. Available from: https://doi.org/10.1080/01621459.1926.10502161

99. NOROUZI-GHEIDARI, Nahid. Alejandro HERNANDEZ. Philippe S. ARCHAMBAULT. and others. Feasibility, safety and efficacy of a virtual reality exergame system to supplement upper extremity rehabilitation post-stroke: A pilot randomized clinical trial and proof of principle. In *International Journal of Environmental Research and Public Health*. 2020, Vol. 17, no. 1, 113. Available from: https://doi.org/10.3390/ijerph17010113

100. SCHEPERS, V. P. M. M. KETELAAR. I. G. L. Van de PORT. and others. Comparing contents of functional outcome measures in stroke rehabilitation using the International Classification of Functioning, Disability and Health. In *Disability and Rehabilitation*. 2007, Vol. 29, no. 3, p. 221–230. Available from: https://doi.org/10.1080/09638280600756257

101. STINEAR, Cathy M. Catherine E. LANG. Steven ZEILER. and others. Advances and challenges in stroke rehabilitation. In *The Lancet Neurology*. 2020, Vol. 19, no. 4, p. 348–360. Available from: https://doi.org/10.1016/S1474-4422(19)30415-6

102. SHATTE, Adrian B. R. Delyse M. HUTCHINSON. and Samantha J. TEAGUE. Machine learning in mental health: A scoping review of methods and applications. In *Psychological Medicine*. 2019, Vol. 49, no. 9, p. 1426–1448. Available from: https://doi.org/10.1017/S0033291719000151

103. AYAR, Mehdi. and Saeed SABAMONIRI. An ECG-based feature selection and heartbeat classification model using a hybrid heuristic algorithm. In *Informatics in Medicine Unlocked*. 2018, Vol. 13, p. 167–175. Available from: https://doi.org/10.1016/j.imu.2018.06.002

104. LUO, Jake. Min WU. Deepika GOPUKUMAR. and others. Big Data Application in Biomedical Research and Health Care: A Literature Review. In *Biomedical Informatics Insights*. 2016, Vol. 8, p. 1–10. Available from: https://doi.org/10.4137/bii.s31559

105. ZHOU, Yangming. and Guoping QIU. Random forest for label ranking. In *Expert Systems with Applications*. 2018, Vol. 112, p. 99–109. Available from: https://doi.org/10.1016/j.eswa.2018.06.036

106. XU, Tianhan. and Wataru TAKANO. Graph Stacked Hourglass Networks for 3D Human Pose Estimation [online]. arXiv, 2021 [viewed 2025-12-03]. arXiv:2103.16385 [cs]. Available from: https://doi.org/10.48550/arXiv.2103.16385

107. WU, Yiqi. Shichao MA. Dejun ZHANG. and others. An Improved Mixture Density Network for 3D Human Pose Estimation with Ordinal Ranking. In *Sensors*. 2022, Vol. 22, no. 13, 4987. Available from: https://doi.org/10.3390/s22134987

108. KIM, Seung Taek. and Hyo JONG. Lightweight stacked hourglass network for human pose estimation. In *Applied Sciences (Switzerland)*. 2020, Vol. 10, no. 18. Available from: https://doi.org/10.3390/APP10186497

CURRICULUM VITAE AND DESCRIPTION OF CREATIVE ACTIVITIES

Cenker Canbulut

cenker.canbulut@ktu.lt

Education:

2011 – 2015 Bachelor's degree at Kaunas University of Technology in the Department of Multimedia Engineering, Faculty of Informatics

2015 – 2017 Master's degree at Kaunas University of Technology in the Department of Multimedia Engineering, Faculty of Informatics

Professional Experience:

2015 – 2017 Assistant lecturer at Kaunas University of Technology
2017 – 2022 Full-Time lecturer at Kaunas University of Technology
2022 – now Senior Software Engineer at Quadigi UAB

Areas of Research Interest:

Virtual Reality, Control Methods, Data Synchronisation, Motion tracking, Artificial Intelligence.

LIST OF SCIENTIFIC PAPERS AND SCIENTIFIC CONFERENCES

Scientific Papers Related to the Topic of the Dissertation:

1. [S1; CH; OA] Maskeliūnas, Rytis; Damaševičius, Robertas; Blažauskas, Tomas; Canbulut, Cenker; Adomavičienė, Aušra; Griškevičius, Julius. BiomacVR: a virtual reality-based system for precise human posture and motion analysis in rehabilitation exercises using depth sensors // Electronics. Basel: MDPI. ISSN 2079-9292. 2023, vol. 12, iss. 2, art. no. 339, p. 1-31. DOI: 10.3390/electronics12020339. [Science Citation Index Expanded (Web of Science); Scopus; INSPEC] [IF: 2,600; AIF: 4,266; IF/AIF: 0,609; Q2 (2023, InCites JCR SCIE)] [FOR: T 007, T 009, M 001] [Input: 0,166]
2. [S1; RO; OA] Canbulut, C.; Paulauskas, A.; Blazauskas, T. Prediction of pending data using interpolation and extrapolation techniques for virtual rowing // International journal of computers communications & control. Oradea: Agora University. ISSN 1841-9836. eISSN 1841-9844. 2020, vol. 15, iss. 2, art. no. 3778, p. 1-15. DOI: 10.15837/ijccc.2020.2.3778. [Science Citation Index Expanded (Web of Science); Scopus; DOAJ] [IF: 2,293; AIF: 4,101; IF/AIF: 0,559; Q3 (2020, InCites JCR SCIE)] [FOR: T 007] [Input: 0,334]

National Scientific Conferences:

1. [P1b; DE; OA] Augustauskas, Rytis; Kudarauskas, Aurimas; Canbulut, Cenker. Implementation of artificial intelligence methods for virtual reality solutions: a review of the literature // CEUR workshop proceedings: IVUS 2018: proceedings of the international conference on information technologies, Kaunas, Lithuania, April 27, 2018 / edited by G. Capizzi, R. Damaševičius, A. Lopata, T. Krilavičius, Ch. Napoli, M. Woźniak. Aachen: CEUR-WS. ISSN 1613-0073. 2018, vol. 2145, p. 68-74. [Scopus] [FOR: T 007] [Input: 0,333]
2. [P1c; LT; OA] Drasute, Vida; Dzindzeletaite, Gintare; Kelpsaite, Neringa; Drasutis, Sigitas; Canbulut, Cenker. Videogames in education: analysis of videogames and APPS // ALTA'18: Pažangios mokymosi technologijos ir aplikacijos. Žaidybinimas švietime: konferencijos pranešimų medžiaga, 2018 m. gruodžio 5 d. = Advanced learning technologies and applications. Games for education: conference proceedings, 5th of December, 2018 / edited by D. Rutkauskiene and R. Bartkute. Kaunas: Kauno technologijos universitetas. ISSN 2335-2140. 2018, p. 25-29. [Index Copernicuss] [FOR: T 007] [Input: 0,200]
3. [T2; LT] Paulauskas, Andrius; Valatkevičius, Tomas; Canbulut, Cenker. Investigation of interpolation methods for virtual rowing simulator // 9th International workshop on data analysis methods for software systems, DAMSS: Druskininkai, Lithuania, November 30 - December 2, 2017 / Lithuanian Computer Society, Vilnius University, Institute of Data Science and Digital Technologies, Lithuanian Academy of Sciences. Vilnius: Vilnius University, 2017. ISBN 9789986680642. p. 37. DOI: 10.15388/DAMSS.2017. [FOR: T 007] [Input: 0,333]

International Scientific Conferences

1. Presented poster "Tracking Rehabilitation Exercises Using Virtual Reality System" in 4th International Conference on Intelligent Technologies and Applications INTAP 2021 - Norway, Grimstad 13 October 2021
2. [P1d; UA; OA] Canbulut, Cenker; Blažauskas, Tomas. Using virtual reality technologies for full-body tracking in ice hockey training. // Immersive Technologies in Education: Proceedings of the 4th International Scientific and Practical Conference, Kyiv, April 30, 2024 / edited by Nosenko Yu.G. Kyiv: Institute of Digitalization of Education, National Academy of Pedagogical Sciences of Ukraine, 2024. ISBN 9786178330361. p. 4-8. [FOR: T 007] [Input: 0,500]

APPENDICES

Appendix 1. System Deployment Diagram

The system was built using Unreal Engine and employed HTC Vive VR hardware to capture patient movements. It runs on Microsoft Windows and uses the Steam software package. The system relies on the Steam VR subsystem to configure the virtual reality environment and track the sensors. The system comprises four distinct software components. The first component is a VR program for personal computers and VR headsets that enables recording, editing, and exporting patient exercise data in .csv format. Known as the VR session recorder, this part of the system is used by both the doctor and the patient, with the doctor selecting exercises and recording the patient's movements. The second component is the calibration package, which links the patient to a virtual avatar. This package is primarily used during the initial setup phase, as it handles the calibration data required to display the virtual avatar and track the sensor data. The package contains two essential functions: one to calibrate the vectors representing the bones and joints of the 3D avatar, and another to compare reference bones to moving bones to calculate the angles between them. The third component is the processing package, which facilitates the viewing and editing of recorded session data. When a patient’s exercise is captured, some redundant data may be present, such as movements before and after the exercise. This package lets users review the session and trim the data by marking its start and endpoints. The processing package includes three auxiliary functions: the frame calculator, which processes the node information for each frame; the vector calculator, which computes the required vectors; and the angle calculator, which determines the angles between them. The calculated information can be displayed or exported for further use. The fourth component includes export and import functionality, allowing session data to be saved in .csv format for further editing or analysis with a recording program. Finally, the scenario package manages the analysis of recorded data. Specific nodes and vectors representing bone positions must be recorded for each exercise. Upon selecting an exercise, a corresponding script is triggered to capture the necessary data for that particular movement.

The diagram depicts a VR- and AI-assisted system for patient exercise monitoring and disorder detection. This system comprises two primary software packages: one for VR-based data collection and another for AI-based analysis (see Fig. 125).

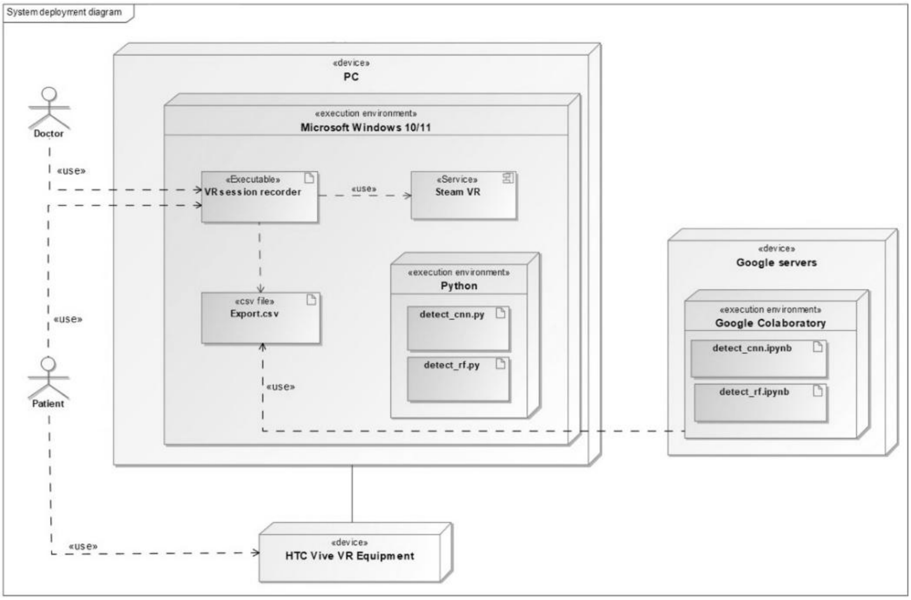


Fig. 125. Deployment diagram of the VR System

The diagram represents a VR and AI-assisted system for patient exercise monitoring and disorder detection. This system comprises two primary software packages: one for VR-based data collection and another for AI-based analysis.

VR Session Recorder

- **Purpose:** Its primary purpose is to record, edit, and export patient exercise data in CSV format (Export.csv), which can later be analysed.
- **User Interaction:**
  - **Doctor: T**he doctor manages the exercise sessions by selecting exercises that are appropriate for the patient. This allows the program to record specific movements as the patient performs them.
  - **Patient:** The patient performs the exercises, with their movements captured by VR sensors.
- **Hardware and Software Requirements**
  - **PC with MS Windows 10/11:** The application runs on a PC that meets the requirements of Microsoft Windows 10 or 11.
  - **HTC Vive VR Equipment:** This VR equipment provides the necessary sensors to capture patient movements in a 3D space, enabling accurate tracking and data collection.

  - **Steam VR Service:** This subsystem, part of the Steam software package, is essential for configuring the VR environment and managing the HTC Vive sensors for motion tracking.

AI-Based Disorder Detection Programs

- The system utilises two AI-based programs (detect_cnn.py and detect_rf.py) to identify potential patient disorders based on recorded exercise data.
- **detect_cnn.py:** This program uses CNN, a type of deep learning model, to detect disorders by analysing movement patterns from the exported CSV data.
- **detect_rf.py:** This program utilises the RF method, a machine learning approach, to identify disorders through pattern recognition in patient movements.
- **Execution Environment:** Both programs currently run in a Python environment on the local system, though plans exist to eventually compile them into standalone executables or transfer them to a server environment for remote deployment.

Google Collaboratory Integration

- For research and development purposes, these AI programs (detect_cnn.ipynb and detect_rf.ipynb) have been adapted to run within the Google Colaboratory environment on Google servers.
- **Features Added:** In addition to running the core algorithms, these Collaboratory-based versions include enhanced charting functions that visualize data and results for research purposes. These visualizations facilitate the understanding and interpretation of AI model outputs.
- **Production Limitation:** These Collaboratory-based versions are intended solely for research and are not included in the final production setup. Only the locally hosted versions (detect_cnn.py and detect_rf.py) will be used in the production environment.

Appendix 2. System Packages Diagram

This section presents the system packages used within the VR system. The system is developed using the Unreal Engine, which provides a unique architectural framework for building applications. Unreal Engine supports two methods for developing system logic: creating blueprints or programming in C++. For the VR system, blueprints were chosen as the primary development approach. Each package in the following diagrams represents a distinct blueprint, with its variables and functions created through visual programming. The packages that make up the system are presented in Fig. 126.

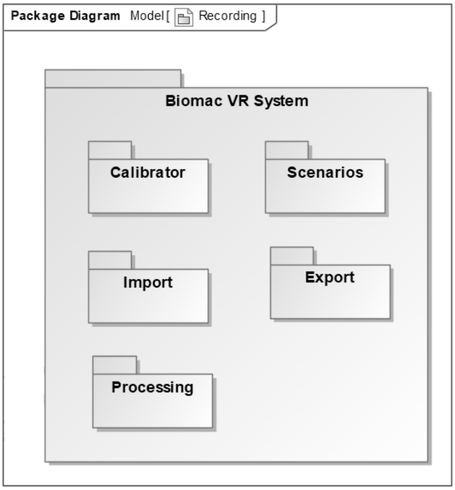


Fig. 126. Packages that make up the system

Calibration Package

The **Calibration Package** (referenced in the package structure) establishes a connection between the patient and their virtual avatar within the VR system. This package performs the essential calibration during the initial setup stage, capturing data that maps the patient's physical movements to the corresponding virtual representation. The collected calibration data is then used to accurately represent the virtual avatar's movements. This functionality ensures that the virtual avatar accurately mirrors the patient's actions. It includes two primary functions for calibrating vectors representing an avatar's bone positions in three-dimensional space. These functions handle two distinct types of bones: moving and reference bones.

- **Moving Bones**: These represent the dynamic, movable parts of the avatar's body, such as arms and legs. Each moving bone is described by three vectors that define its position and orientation. Calibration may be required for these vectors in both the original frame and other frames of the animation data to ensure accurate movement tracking.

- **Reference Bones**: These bones provide static reference points, typically positioned on the avatar's body structure at fixed locations. Each reference bone is calibrated by comparing it with the corresponding moving bone, and the system calculates the angle between them. This comparison ensures that the moving bones align correctly with the avatar's fixed skeletal structure, enhancing the accuracy of movement representation in the virtual environment.

Processing Package

The **Processing Package** is designed for viewing and editing recorded data within the VR system. When a patient's exercise session is recorded, the data often includes unnecessary segments captured before and after the exercise begins. This package allows users to review the entire session and trim these extra portions by marking the start and end of the actual exercise. The refined data is then used in the *Export Package* for further processing or analysis.

The package includes three helper functions to calculate the necessary information:

- **Frame Calculator**: This function processes each frame within the designated session boundaries. It extracts node information from each frame, which forms the basis for vector and angle calculations.
- **Vector Calculator**: This function calculates vectors representing various positional data of the avatar using data from the frame calculator. These vectors are essential for accurately tracking movement.
- **Angle Calculator**: This function calculates the angles between the given vectors. This information is crucial for analysing joint movement and alignment throughout the session.

Together, these functions provide a comprehensive toolkit for refining and analysing session data, ensuring that only relevant and accurate information is exported and used.

Export and Import packages

The Export and Import packages handle the exporting and importing of session records within the VR system. Data is exported in CSV format, which can later be imported back into the program for further editing or used for data analysis.

Each package contains a single function:

- **Data Export**: After processing data through the *Processing Package*, the calculated information is saved into a file. This export function enables the system to output data in a structured format, facilitating further use in analysis or reporting.

- **Data Import**: The import function enables the opening of recorded session files and linking their animation data to the virtual avatar. This capability is essential for reviewing and replaying previously recorded sessions in the VR environment.

Scenarios Package

The Scenarios Package analyses recorded data specific to various exercises. Each exercise requires the system to track particular nodes and vectors that represent the positions of the bones. When an exercise is selected for recording, a corresponding scenario is activated to capture the relevant data. The scenarios in this package closely align with the exercises discussed throughout the report. Therefore, the specific features of each scenario are not covered here.

Appendix 3. System Components Diagram

This section presents the components developed for the VR system within the Unreal Engine environment. In Unreal Engine, a component can consist of multiple packages or sets of blueprints, each serving a distinct purpose. This system organizes components into two primary groups: User Interface (UI) Components and Session Components (responsible for session recording and editing).

User Interface Components

These components manage the system’s user interface. Each UI component is built upon Unreal Engine's Widget class, allowing for the creating of interactive elements and controls within the VR environment. The UI components facilitate user interactions, providing a seamless interface for navigation, exercise selection, and data visualisation (see in Fig. 127).

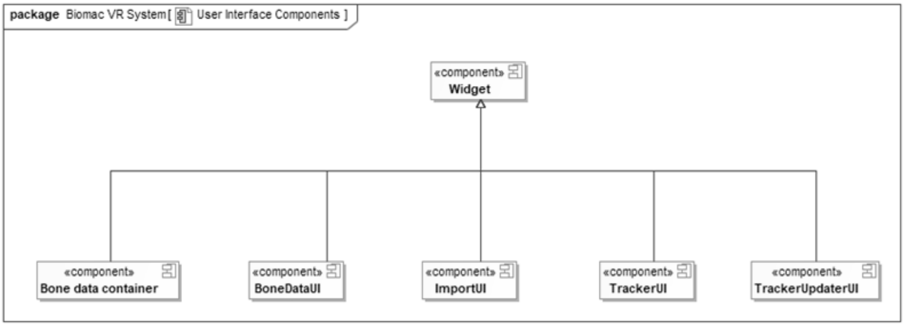

package Biomac VR System[User Interface Components]
«component» Widget
«component» Bone data container
«component» BoneDataUI
«component» ImportUI
«component» TrackerUI
«component» TrackerUpdaterUI

Fig. 127. User Interface components

As depicted in the diagram, the User Interface Components of the VR system inherit from Unreal Engine’s Widget class. These components work together to facilitate interaction with the system, enabling users to manage data, import files, and track sensor input. Here is a breakdown of each component and its functionality:

- **BoneDataContainer:** This component acts as a container for instances of the BoneDataUI component. Its purpose is to provide data to the selected virtual avatar node. When a specific bone is selected, a new BoneDataUI instance is created and added to the BoneDataContainer, allowing users to access and manage data specific to each virtual avatar.
- **BoneDataUI:** The BoneDataUI component displays and interacts with data associated with individual avatar bones. It provides an interface for detailed information about each selected bone, helping to manage attributes and visualize data in real-time regarding the VR environment.
- **ImportUI:** This component serves a simple but essential function in the UI by providing a button that opens a file dialog. Users can use this interface element to browse and import files into the system, facilitating the loading of external data or settings as needed for the avatar or session.

- **TrackerUI:** The TrackerUI component plays a crucial role in receiving and managing data from all detected HTC Vive sensors. It identifies which sensor is assigned to each corresponding bone, allowing it to channel the data directly to the relevant BoneDataUI component. This enables real-time tracking of the patient's movements and ensures that each bone in the virtual avatar aligns with the correct sensor input.
- **TrackerUpdaterUI:** This component updates the tracking data when adding a new bone position to the avatar's animation. Users can refresh the information by pressing the TrackerUpdaterUI button, ensuring that changes in the avatar's bone structure or position are communicated to the TrackerUI component. This ensures accurate and up-to-date alignment between the virtual avatar and sensor data.

Scenario Components

These components handle the core functionality of session recording and editing. They are based on Unreal Engine's Actor class, which enables them to actively participate in the VR world actively, capturing and processing patient movement data in real-time. Session components manage the calibration, tracking, and analysis of recorded exercises, ensuring precise data collection for further processing. The VR system is built upon Unreal Engine's Actor class, which allows it to be included directly in the scene (referred to as the "level" in Unreal Engine). These components handle the main functionalities associated with representing and tracking the virtual avatar, importing animation data, and recording real-time movement (see in Fig. 128).

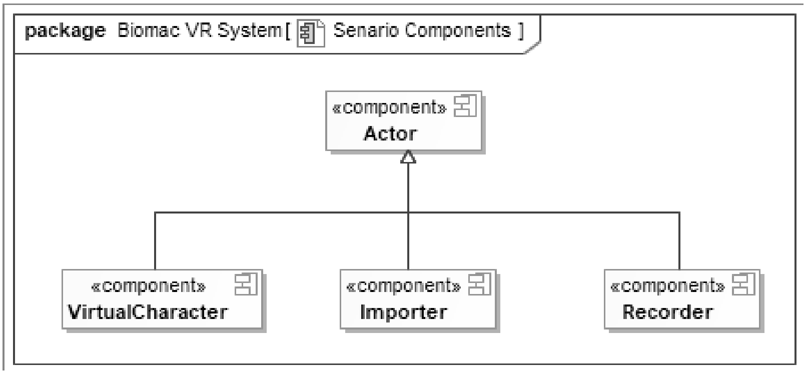


Fig. 128. Stage components

Here is the overview of each component:

- **Actor (Base Class)**: The *Actor* class serves as the foundational class for all scenario components, enabling them to be placed within the VR environment and interact within the scene. The system's primary logic resides in the components inherited from the *Actor*, allowing each to contribute uniquely to the scenario's functionality.

- **VirtualCharacter**: This component represents the visual aspect of the virtual avatar within the VR environment. It is responsible for displaying custom animations, bone positions, bone rotations, and any state changes. These state changes are essential for accurately representing the avatar's real-time movements and adjustments. Additionally, the data export functionality is closely linked to this component, allowing users to export session data based on the avatar's current state.
- **Importer**: The *Importer* component works in conjunction with the *ImportUI* component to receive animation data. Through the *ImportUI*, users can import animation files, which the *Importer* then applies to the virtual avatar. This process involves updating the animation base class and rewriting it to the avatar object, effectively syncing the avatar's animation state with the imported data. This is useful for applying pre-recorded exercises or movement patterns to the avatar.
- **Recorder**: The *Recorder* component captures the patient's movements in real-time and translates them to the virtual avatar. This real-time recording creates an animation dataset from the patient's actions, which can later be imported and analysed to assess the patient's exercise performance. The recorded data provides valuable insights into movement patterns, helping to track and evaluate the patient's progress over time.

Appendix 4. System Activity Diagram

This section outlines the main scenarios for interacting with the system, represented through UML activity diagrams. The primary operational scenarios include configuring the VR environment, sensor calibration, exercise recording, and processing the recorded data.

In the activity diagrams, two leading actors participate:

1. **Operator**: The individual responsible for configuring the VR environment. Although the operator may be a doctor, this role is not restricted to medical professionals, so it’s referred to simply as the “operator”.
2. **Steam VR System**: Steam Corp. provides this software to facilitate seamless interaction with the HTC Vive hardware within the Windows operating system. Its role is crucial in setting up and managing the VR environment for the exercises.
3. **Patient**: The Patient in the system interacts primarily with the VR environment during exercise sessions. Their role involves performing physical movements captured in real-time by VR sensors and reflected in the virtual avatar. The patient’s interactions are primarily passive regarding system operation - they focus on following exercise instructions and performing the required movements while the system records and processes their actions. The data collected from the patient’s movements is later analysed to evaluate exercise performance and progress.

Environment Configuration Activity Diagram

The VR Environment Configuration Script focuses on configuring SteamVR, which is critical for setting up the virtual environment. Although it primarily involves technical interactions with SteamVR, this process is crucial for creating a functional, accurate VR environment in which patients can safely perform exercises. This scenario, represented in Fig. 129 Outlines the steps required to prepare the environment, ensuring that all VR components are correctly calibrated and positioned to capture the patient's movements accurately.

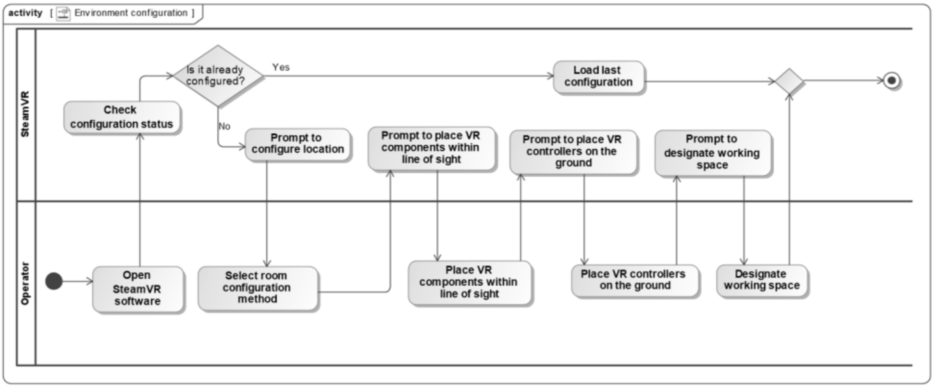

activity [Environment configuration]
SteamVR
Operator
Is it already configured?
Yes
No
Load last configuration
Check configuration status
Prompt to configure location
Prompt to place VR components within line of sight
Prompt to place VR controllers on the ground
Prompt to designate working space
Open SteamVR software
Select room configuration method
Place VR components within line of sight
Place VR controllers on the ground
Designate working space

Fig. 129. Activity diagram to configure the VR environment

When starting the environment configuration, the operator launches the Steam VR application, which checks the current configuration status. The process follows one of two paths:

- **If the environment is already configured**: The system loads the last saved configuration and ready the environment without further adjustments.
- **If the environment is not configured**: The operator must follow a series of steps to set up the VR environment from scratch:
  1. **Select Room Configuration Method:** The operator chooses the configuration approach within the Steam VR application.
  2. **Position VR Components:** Steam VR prompts the operator to place all VR components within the base station's line of sight. This ensures accurate tracking and calibration of the VR space.
  3. **Place VR Controllers on the Ground:** After positioning the VR components, the operator places the controllers on the ground per the application’s instructions.
  4. **Designate the Workspace:** The operator uses the controller to mark the boundary of the free space where the patient will perform exercises. This step defines the area within which movements will be tracked and recorded.
  5. **Save Configuration Data:** Once the workspace is defined, the configuration data is saved, completing the setup process.

VR Sensor Calibration Activity Diagram

This process aims to calibrate VR sensors by associating them with specific patient body parts and synchronizing their positions with the virtual avatar.

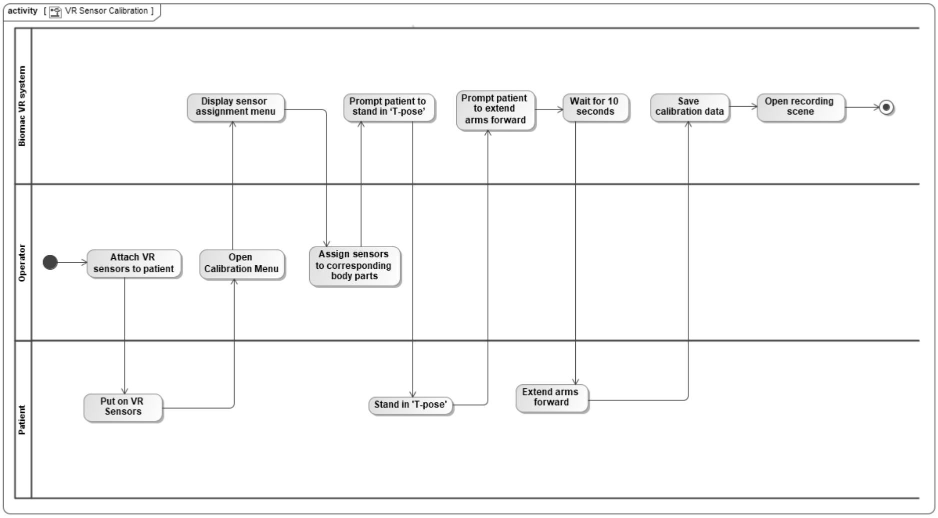

activity [VR Sensor Calibration]
Biomac VR system
Operator
Patient
Display sensor assignment menu
Prompt patient to stand in 'T-pose'
Prompt patient to extend arms forward
Wait for 10 seconds
Save calibration data
Open recording scene
Attach VR sensors to patient
Open Calibration Menu
Assign sensors to corresponding body parts
Put on VR Sensors
Stand in 'T-pose'
Extend arms forward

Fig. 130. Activity diagram showing the calibration of VR sensors on the system

1. **Attach VR Sensors to the Patient**
   - **Patient Action:** The patient is equipped with VR sensors, which are placed on specific body parts such as the head, wrists, ankles, and torso. These sensors will track movement and position in real-time.
2. **Open Calibration Menu**
   - **Operator Action**: After placing the sensors, the operator opens the calibration menu within the VR system. This menu provides options to assign each sensor to its corresponding body part.
3. **Display Sensor Assignment Menu**
   - **System Action**: The VR system displays the sensor assignment menu, showing a list of available sensors and allowing the operator to map each sensor to a specific body part on the virtual avatar.
4. **Assign Sensors to Body Parts**
   - **Operator Action**: In the assignment menu, the operator selects each sensor and assigns it to the correct body part. This step is crucial to ensure that the avatar's movements match those of the patient.
5. **Prompt Patient to Stand in 'T-Pose'**
   - **System Prompt**: The system prompts the patient to stand in a 'T-pose,' where they hold their arms straight out to the sides. This position helps align the sensors and provides a neutral starting point for calibration.
6. **Patient Stands in 'T-Pose'**

- **Patient Action**: The patient assumes the T-pose position, allowing the system to detect the initial positions of the sensors relative to one another.

7. **Prompt Patient to Extend Arms Forward**

   - **System Prompt**: The system then asks patients to extend their arms forward. This movement allows the system to capture dynamic positioning data and adjust for any discrepancies in sensor placement.

8. **Patient Extends Arms Forward**

   - **Patient Action**: The patient stretches their arms out in front of them, enabling the system to validate the tracking accuracy for this new position.

9. **Wait for 10 Seconds**

   - **System Action**: The system pauses for 10 seconds to allow the sensors to stabilize and capture accurate positional data. This pause helps reduce errors and ensures precise calibration.

10. **Save Calibration Data**

    - **System Action**: Once the sensors have been calibrated, the system saves the configuration data, recording the positions and alignments for future reference. This data is essential for maintaining consistency in tracking across sessions.

11. **Open Recording Scene**

    - **System Action**: With calibration complete, the system transitions to the recording scene, where the patient's movements can be tracked accurately in real-time, aligned with the virtual avatar.

Recording an Exercise Activity Diagram

The exercise recording process captures model animation data and sensor position data to represent the patient's movements accurately. This straightforward process is illustrated in Fig. 131.

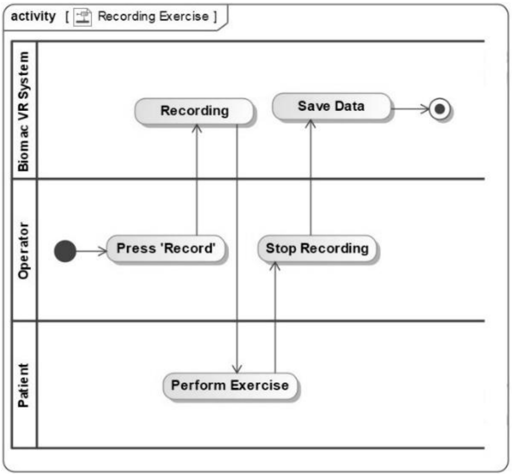


Fig. 131. Activity diagram showing the recording of an exercise

1. **Start**
   - The recording begins after the VR sensors are calibrated and the recording scene (or level) is loaded.
2. **Press 'Record'**
   - **Operator Action**: The operator initiates the recording by pressing the "Record" button. This action triggers the system to capture data, linking the VR sensor inputs to the avatar’s movements in real-time.
3. **Perform Exercise**
   - **Patient Action**: The patient performs the designated exercise. The VR sensors track their positions as they move, and the VR system records these movements.
4. **Recording**
   - **System Action**: During this step, the VR system continuously captures animation data (for the virtual avatar) and positional data from the sensors. This ongoing recording ensures that every aspect of the patient's movements is logged for later analysis.
5. **Stop Recording**
   - **Operator Action**: Once the patient completes the exercise, the operator stops the recording by pressing the "Stop" button. This halts data collection and prepares the system to save the recorded information.
6. **Save Data**

- **System Action**: The VR system saves the collected data to a file, preserving animation and sensor data. This file can be used later for analysis, evaluation, or to track the patient's progress over time.

7. **End**

   - The recording session is complete, and the system returns to an idle state, ready for the next exercise or calibration.

Record Processing Activity Diagram

After an exercise session is recorded, the data must be refined before being used for research or diagnostic purposes. The processing phase involves removing redundant data (recorded before and after the actual exercise) and exporting specific parameters relevant to the exercise. Fig. 132 Illustrates a typical record-processing workflow.

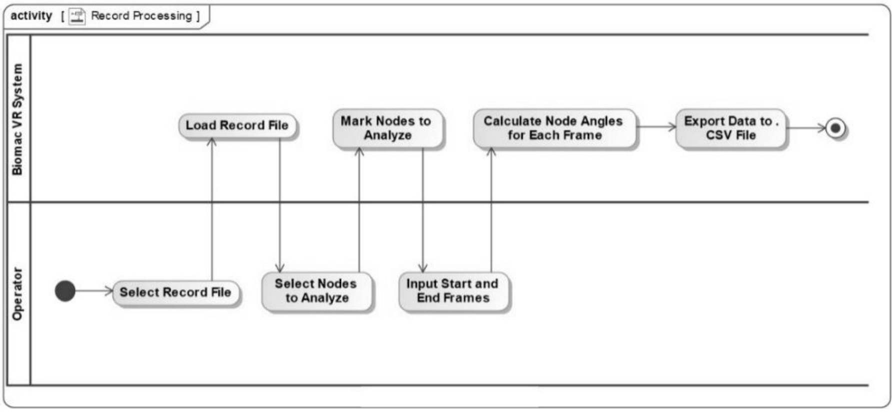

activity [Record Processing]
Biomac VR System
Operator
Load Record File
Mark Nodes to Analyze
Calculate Node Angles for Each Frame
Export Data to . CSV File
Select Record File
Select Nodes to Analyze
Input Start and End Frames

Fig. 132. The Activity diagram illustrates the Processing of the data recorded

1. **Start**

   - The process begins once the recording session has finished and can be initiated immediately or later.

2. **Select Record File**

   - **Operator Action:** The operator selects the file containing the recorded exercise data for the specific patient. This file contains the raw session data collected during the exercise.

3. **Load Record File**

   - **System Action**: The Bioman VR system loads the selected record file and associates the data with the virtual avatar. The system lets the operator view the patient's recorded movements along a timeline, which can be controlled using a slider.

4. **Select Nodes to Analyse**

    - **Operator Action**: The operator chooses the specific nodes (body parts or sensor points) for which data will be exported. These nodes are crucial for evaluating movement quality and determining angles in subsequent stages.

5. **Input Start and End Frames**

    - **Operator Action:** The operator inputs the frames that represent the start and end of the exercise, removing any irrelevant data outside this interval. This step isolates the active exercise data, ensuring that only meaningful movements are included in the analysis.

6. **Calculate Node Angles for Each Frame**

    - **System Action**: For each frame within the specified interval, the system calculates the angles at the selected nodes. These angles are essential for analysing movement and assessing alignment, providing valuable data for further study.

7. **Export Data to .CSV File**

    - **System Action:** Once all relevant frames and nodes have been processed, the system exports the calculated data to a .CSV file. This file includes information on node angles and other parameters, making it ready for further analysis.

8. **End**

    - The processing phase is complete, and the exported file can now be used for data analysis.

UDK 004.358+004.5+004.94](043.3)

SL344. 2025-12-29, 29,75 leidyb. apsk. l. Tiražas 14 egz. Užsakymas 001.
Išleido Kauno technologijos universitetas, K. Donelaičio g. 73, 44249 Kaunas
Spausdino leidyklos „Technologija“ spaustuvė, Studentų g. 54, 51424 Kaunas