

Hidden Markov Model Deep Learning Architecture for Virtual Reality Assessment to Compute Human–Machine Interaction-Based Optimization Model

Chaoyang Zhu ¹⁺

¹ Institute for Social Innovation and Public Culture, Communication University of China, Beijing, 100024, China

Corresponding Author: zcy0919psy@outlook.com

Abstract: Virtual Reality (VR) is a technology that immerses users in a simulated, computer-generated environment. It creates a sense of presence, allowing individuals to interact with and experience virtual worlds. Human-Machine Interaction (HMI) refers to the communication and interaction between humans and machines. Optimization plays a crucial role in Virtual Reality (VR) and Human-Machine Interaction (HMI) to enhance the overall user experience and system performance. This paper proposed an architecture of the Hidden Markov Model with Grey Relational Analysis (GRA) integrated with Salp Swarm Algorithm (SSA) for the automated Human-Machine Interaction. The proposed architecture is stated as a Hidden Markov model Grey Relational Salp Swarm (HMM_ GRSS). The proposed HMM_ GRSS model estimates the feature vector of the variables in the virtual reality platform and compute the feature spaces. The HMM_ GRSS architecture aims to estimate the feature vector of variables within the VR platform and compute the feature spaces. Hidden Markov Models are used to model the temporal behavior and dynamics of the system, allowing for predictions and understanding of the interactions. Grey Relational Analysis is employed to evaluate the relationship and relevance between variables, aiding in feature selection and optimization. The SSA helps optimize the feature spaces by simulating the collective behavior of salp swarms, improving the efficiency and effectiveness of the HMI system. The proposed HMM_ GRSS architecture aims to enhance the automated HMI process in a VR platform, allowing for improved interaction and communication between humans and machines. Simulation analysis provides a significant outcome for the proposed HMM_ GRSS model for the estimation Human-Machine Interaction.

Keywords: Virtual reality, Optimization, Human – Machine Interaction, Hidden Markov Model, Grey Relational Optimization, Deep Learning.

I. Introduction

Virtual Reality Assessment to Compute Human-Machine Interaction-Based Optimization Model is a groundbreaking approach that combines the power of virtual reality technology with the principles of human-machine interaction to develop an advanced optimization model [1]. In recent years, virtual reality has gained immense popularity as a tool for creating immersive and interactive experiences. This innovative assessment framework harnesses the potential of virtual reality to simulate real-life scenarios, allowing researchers to analyze and optimize human-machine interactions in a controlled environment [2]. Through implemented model, researchers can assess the efficiency, effectiveness, and user experience of various human-machine systems, leading to the development of optimized solutions and improved performance [3]. The integration of virtual reality and human-machine interaction presents a new frontier in optimization modeling, offering exciting possibilities for enhancing interactions between humans and machines across a wide range of industries and applications [4]. The field of human-machine interaction faces several significant challenges and issues that need to be

addressed in order to optimize the interaction between humans and machines. One key challenge is the complexity of human behavior and cognition. Humans possess unique cognitive abilities, such as perception, attention, memory, and decision-making, which may vary from person to person [5]. Designing machines that can effectively understand and respond to these complex human behaviors is a crucial challenge. Another issue is the communication gap between humans and machines [6]. While humans primarily communicate through natural language, machines typically rely on structured input formats, such as programming languages or specific commands. Bridging this gap to enable seamless and intuitive communication is essential for enhancing human-machine interaction. Furthermore, ensuring the safety and reliability of human-machine systems is a critical concern [7]. Machines are expected to perform tasks accurately and reliably, particularly in high-risk environments or critical domains. Achieving dependable and error-free performance while considering human factors, such as workload, attention, and fatigue, poses significant challenges [8].

Ethical considerations also come into play in human-machine interaction. As machines become increasingly autonomous and capable of making decisions, questions arise regarding accountability, transparency, and the ethical implications of these decisions [9]. Balancing machine autonomy with human oversight and ethical guidelines is crucial for building trust and acceptance in human-machine interactions. Lastly, adapting to the rapidly evolving technological landscape presents another challenge. As new technologies emerge, such as artificial intelligence, robotics, and virtual reality, it becomes necessary to understand how these advancements can be effectively integrated into human-machine systems [10]. Keeping up with technological advancements and ensuring compatibility and interoperability are ongoing challenges in the field. Addressing these issues requires interdisciplinary collaboration, involving experts in fields such as psychology, computer science, engineering, and ethics. By tackling these challenges head-on, researchers and practitioners can pave the way for improved human-machine interaction, leading to more efficient, intuitive, and mutually beneficial collaborations between humans and machines [11].

Deep learning has made significant contributions to the field of virtual reality assessment for human-machine interaction, revolutionizing the way analyze and optimize these interactions [12]. Deep learning algorithms, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), have shown remarkable capabilities in extracting meaningful features and patterns from complex data, such as visual and sensory inputs. One key contribution of deep learning is in the field of gesture recognition [13]. In virtual reality environments, users often interact with machines using hand gestures or body movements. Deep learning models have been employed to accurately recognize and interpret these gestures, enabling more natural and intuitive interactions between humans and machines. By training on large datasets of labeled gesture data, deep learning models can learn to identify specific gestures with high accuracy, enhancing the overall user experience [14]. Another significant contribution is in the area of natural language processing (NLP). NLP techniques powered by deep learning have been instrumental in improving human-machine communication in virtual reality settings [15]. By leveraging deep learning models like recurrent neural networks and transformers, virtual reality systems can better understand and interpret natural language commands, queries, and conversations, allowing for more fluid and context-aware interactions [16].

Deep learning has also played a crucial role in improving immersive experiences in virtual reality. Generative models, such as generative adversarial networks (GANs), have

been utilized to create realistic and dynamic virtual environments, enhancing the sense of presence and immersion for users. These models can generate realistic textures, objects, and even animate virtual characters, resulting in more engaging and interactive virtual reality experiences [17]. Additionally, deep learning has been employed for behavior prediction and analysis in virtual reality environments. By analyzing user behavior and interaction patterns, deep learning models can provide valuable insights into user preferences, cognitive load, and performance. This information can then be used to optimize human-machine interactions, tailor virtual reality experiences to individual users, and develop adaptive systems that can dynamically adjust their behavior based on user feedback [18].

II. Related Works

In [19] propose a system that enables hands-free interaction between humans and robots using wearable mixed reality devices. The system utilizes multimodal gestures and deep learning techniques to facilitate seamless communication and collaboration between humans and robots. In [20] present an integrated mixed reality system that incorporates deep learning and digital twin technology to enable safe and efficient collaboration between humans and robots. The system focuses on ensuring safety during human-robot interactions in industrial settings. Similarly, in [21] propose the use of augmented reality and deep learning techniques to enhance the museum experience at Taxila. The system combines real-time image processing and augmented reality to provide interactive and informative visualizations for museum visitors. In [22] conduct a comprehensive review of the use-cases of virtual reality and augmented reality in brain-computer interface applications for smart cities. The paper explores the potential of these technologies to enhance human-machine interaction and improve urban living.

In [23] propose a deep learning-based virtual mouse system that utilizes computer vision techniques. The system aims to provide a touchless interface to control computers and devices, reducing the risk of COVID-19 transmission. In [24] present an augmented reality system that enhances visual inspection processes using deep learning techniques. The system utilizes knowledge-based algorithms to analyze real-time visual data and provide augmented visualizations for improved inspection accuracy. Also, in [25] propose a method that combines deep neural networks and holography to generate real-time, photorealistic 3D holographic displays. The system leverages the power of deep learning to synthesize high-quality holographic content, enabling more immersive and realistic visual experiences. In [26] explore the use of machine learning techniques for 3D image processing in the context of man-machine interaction. The paper investigates

how machine learning algorithms can be applied to process input data and improve the interaction between humans and machines.

In [27] conduct a literature review focused on non-destructive disassembly in human-robot collaboration scenarios within industrial environments. The paper discusses various research efforts and techniques aimed at improving the efficiency and safety of collaborative processes involving humans and robots. Similarly, in [28] propose a proactive human-robot collaboration paradigm for cognitive manufacturing. The paper discusses the integration of cognitive capabilities into manufacturing systems to enable more efficient and intuitive interaction between humans and robots. In [29] propose a deep neural network approach for estimating forces during human-robot interactions using electromyography (EMG) signals. The paper presents a framework that leverages deep learning techniques to accurately estimate forces applied by humans during physical interactions with robots.

In [30] conduct a systematic literature review on the use of augmented reality in smart manufacturing. The paper focuses on the collaboration between humans and computational intelligence in the context of augmented reality applications for manufacturing processes. In [31] present a survey of robot learning strategies employed in human-robot collaboration scenarios within industrial settings. The paper discusses various learning techniques and approaches used to enhance the collaboration and coordination between humans and robots in shared workspaces. These papers contribute to the understanding and advancement of various aspects of human-machine interaction and highlight the potential of emerging technologies in enhancing this interaction. Several papers focus on the application of virtual reality (VR) and augmented reality (AR) in different domains. For example, Tran et al. (2021) review VR studies on autonomous vehicle-pedestrian interaction, while Khan et al. (2021) explore the use of AR to enhance the museum experience. Kohli et al. (2022) review use-cases of VR and AR in brain-computer interface applications for smart cities, highlighting the potential of these technologies in improving urban living.

Deep learning techniques play a significant role in several papers. Park et al. (2021) and Choi et al. (2022) present systems that utilize deep learning for hands-free human-robot interaction and safety-aware human-robot collaboration, respectively. Additionally, Wang et al. (2021) and Shi et al. (2021) leverage deep learning for enhanced visual inspection and real-time photorealistic 3D holography, respectively. Other papers address topics such as machine learning-based input processing for man-machine interaction (Sungheetha & Sharma, 2021), non-destructive disassembly in human-robot

collaboration (Hjorth & Chrysostomou, 2022), proactive human-robot collaboration in cognitive manufacturing (Li et al., 2021), force estimation in human-robot interaction using deep neural networks (Su et al., 2021), and the use of augmented reality in smart manufacturing (Baroroh et al., 2021).

III. Human-Machine Interaction Model

The paper proposes an architecture called Hidden Markov model Grey Relational Salp Swarm (HMM_GRSS) for automated Human-Machine Interaction (HMI) in Virtual Reality (VR). The HMM_GRSS model utilizes Hidden Markov Models to capture the temporal behavior and dynamics of the system, enabling predictions and understanding of the interactions. Grey Relational Analysis is employed to evaluate the relationship and relevance between variables, assisting in feature selection and optimization. The Salp Swarm Algorithm (SSA) is utilized to optimize the feature spaces by simulating the collective behavior of salp swarms. The proposed HMM_GRSS architecture aims to enhance the automated HMI process in a VR platform by estimating the feature vector of variables and computing the feature spaces. By leveraging the temporal modeling capabilities of Hidden Markov Models, the system can better understand and predict user behavior. The Grey Relational Analysis helps identify the most relevant variables and optimize the feature selection process. Additionally, the SSA contributes to improving the efficiency and effectiveness of the HMI system by optimizing the feature spaces.

The HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture proposed in the paper involves several steps to enhance the automated Human-Machine Interaction (HMI) process in Virtual Reality (VR). The following are the steps involved in the HMM_GRSS approach:

1. The first step is to collect data from the VR platform, which includes capturing user interactions, system parameters, and other relevant variables. This data serves as the input for the subsequent steps.
2. The collected data is used to construct a Hidden Markov Model. The HMM captures the temporal behavior and dynamics of the system by modeling the transitions between different states. It helps in predicting and understanding user behavior in the VR environment.
3. The HMM_GRSS architecture aims to estimate the feature vector of variables within the VR platform. This involves extracting relevant features from the collected data that represent important aspects of the user-machine interaction.

4. Once the feature vector is estimated, the next step is to compute the feature spaces. Feature spaces represent the multidimensional space where the extracted features are analyzed and compared. This step helps in understanding the relationships and patterns between different features.
5. Grey Relational Analysis is employed to evaluate the relationship and relevance between variables in the feature spaces. It measures the similarity and correlation between variables, aiding in feature selection and optimization. GRA helps identify the most influential features for HMI and contributes to enhancing system performance.
6. The SSA is applied to optimize the feature spaces. The SSA simulates the collective behavior of salp swarms, which mimics the natural behavior of salps in search of better solutions. The algorithm iteratively adjusts the feature spaces to improve their efficiency and effectiveness for HMI.

a. Hidden Markov Model

In the HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture, the Hidden Markov Model (HMM) plays a crucial role in modeling the temporal behavior and dynamics of the system. The HMM is utilized to capture the transitions between different states in the VR platform and enables predictions and understanding of user behavior during Human-Machine Interaction (HMI). The HMM consists of three fundamental components:

State Space: The state space represents the set of possible states that the system can be in during the HMI process. In the context of VR, these states can represent different interaction modes, user actions, or system conditions relevant to the HMI scenario.

Observation Space: The observation space represents the set of observable features or variables in the VR platform. These observations are typically derived from the collected data, such as user inputs, system parameters, or sensory data. The observations provide information about the current state of the system.

Transition Probability Matrix: The transition probability matrix defines the probabilities of transitioning between different states in the HMM. It captures the dynamics of the system and the likelihood of moving from one state to another based on the observed data. These probabilities can be estimated using statistical methods or machine learning techniques.

With the implementation of HMM, the HMM_GRSS architecture can model the sequence of user interactions and system responses in the VR platform. This enables the system to learn patterns, predict future states, and understand the underlying dynamics of the HMI process. The HMM provides a powerful framework for capturing and analyzing the temporal aspects of the interaction, allowing for more effective optimization and enhancement of the HMI system. The process flow of the proposed HMM_GRSS model is shown in figure 1.

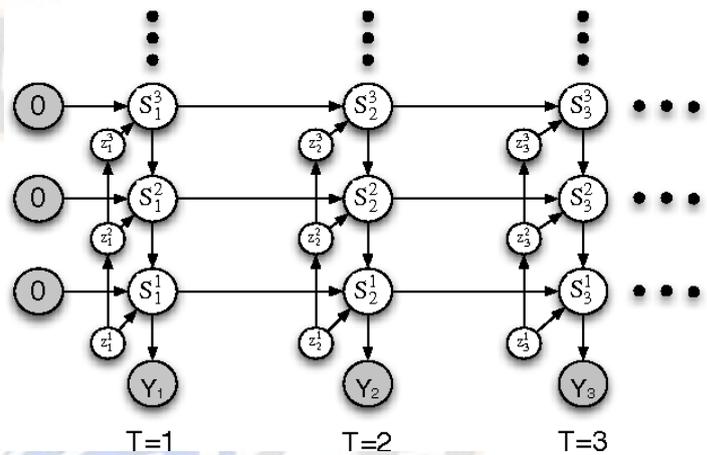


Figure 1: Illustration of HMM

Hidden Markov Model (HMM) component of the HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture. Let's denote the state space as $S = \{S_1, S_2, \dots, S_n\}$, where n is the number of possible states. Each state S_i represents a specific condition or mode in the VR platform. The observation space as $O = \{O_1, O_2, \dots, O_m\}$, where m is the number of possible observations. Each observation O_j represents a measurable feature or variable in the VR environment. The transition probability matrix, denoted as A , represents the probabilities of transitioning from one state to another. It can be represented as an $n \times n$ matrix, where $A(i, j)$ represents the probability of transitioning from state S_i to state S_j . The observation probability matrix, denoted as B , represents the probabilities of observing a particular observation given a certain state. It can be represented as an $n \times m$ matrix, where $B(i, j)$ represents the probability of observing observation O_j in state S_i .

The initial state distribution, denoted as π , represents the probabilities of starting in each state. It is a probability vector of length n , where $\pi(i)$ represents the probability of starting in state S_i . With these components, the HMM can be characterized in the equation (1):

$$P(S_{n+1} = S_j | S_1, S_2, \dots, S_n) = A(i, j) \tag{1}$$

This equation represents the probability of transitioning to state S_j given the previous states S_1, S_2, \dots, S_n . The transition probability is shown in equation (2)

$$P(O_n = O_j | S_1, S_2, \dots, S_n) = B(i, j) \quad (2)$$

The probability of observing observation O_j given the states S_1, S_2, \dots, S_n is presented in equation (3)

$$P(S_1 = S_j) = \pi(j) \quad (3)$$

This equation represents the probability of starting in state S_j . The HMM can be further utilized for tasks such as state estimation, sequence prediction, and parameter learning through algorithms like the Viterbi algorithm, Baum-Welch algorithm, or forward-backward algorithm. These mathematical derivatives form the foundation of the Hidden Markov Model component in the HMM_GRSS architecture, allowing for the modeling and analysis of the temporal dynamics and behavior in the VR platform during Human-Machine Interaction.

b. Grey Relational Salp Swarm

The Grey Relational Salp Swarm (GRSS) is an integral part of the HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture. It combines the principles of Grey Relational Analysis (GRA) and the Salp Swarm Algorithm (SSA) to optimize the feature spaces and improve the efficiency and effectiveness of the Human-Machine Interaction (HMI) system in the Virtual Reality (VR) platform.

The feature spaces are initialized based on the available variables or features extracted from the VR platform. These feature spaces represent the multidimensional space where the extracted features are analyzed and compared. GRA is applied to evaluate the relationship and relevance between variables in the feature spaces. It measures the similarity and correlation between features and helps in identifying the most influential ones for the HMI process. The SSA is employed to optimize the feature spaces. The algorithm simulates the collective behavior of salp swarms, where each salp represents a potential solution in the feature space. The salps iteratively adjust their positions based on certain rules inspired by salp behavior, aiming to find better solutions and improve the efficiency and effectiveness of the HMI system. The optimization process continues until a certain convergence criterion is met. At this stage, the optimal feature spaces are selected based on the final positions of the salps in the swarm. These optimized feature spaces provide improved representations of the important features for HMI in the VR platform. With integrating Grey Relational Analysis and the Salp Swarm Algorithm, the GRSS step in the HMM_GRSS architecture contributes to the optimization of feature spaces,

enhancing the automated HMI process. It improves the efficiency and effectiveness of the HMI system in the VR platform, allowing for improved interaction and communication between humans and machines. The architecture of the proposed HMM_GRSS model in the Human-machine Interaction with VR is shown in figure 2.

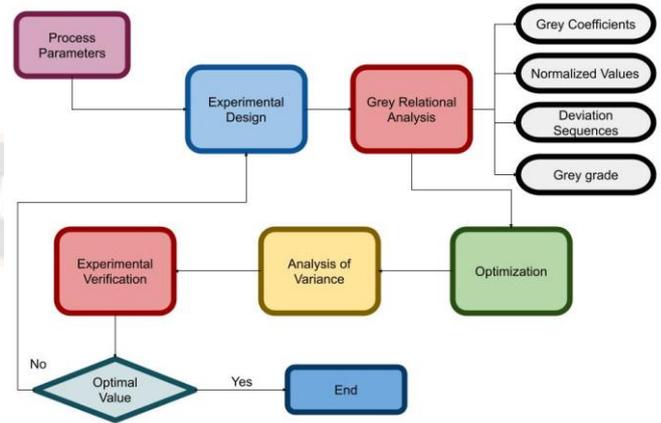


Figure 2: Architecture of HMM_GRSS

Grey Relational Analysis (GRA) is a mathematical tool utilized in the HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture to evaluate the relationship and relevance between variables in the feature spaces. It measures the similarity and correlation between variables, aiding in feature selection and optimization for Human-Machine Interaction (HMI) in the Virtual Reality (VR) platform. The GRA method involves the following steps:

Firstly, the raw data or variables in the feature spaces are standardized to eliminate any differences in scales or units. Standardization ensures that each variable contributes equally during the grey relational analysis. Among the variables in the feature spaces, a reference sequence is selected as a basis for comparison. This reference sequence represents the desired or optimal values for the variables and is chosen based on the specific HMI requirements or objectives. The grey relational coefficient (ξ) is calculated to quantify the similarity between each variable and the reference sequence. It measures the closeness of each variable to the reference sequence and determines its relevance in the HMI process. The grey relational coefficient is computed using the following equation (4)

$$\xi(i, j) = (\xi_{max} - \xi_{min}) / (|x_{ref}(j) - x(i)| + \epsilon) \quad (4)$$

where: $\xi(i, j)$ represents the grey relational coefficient between variable i and the j th reference sequence. ξ_{max} and ξ_{min} are the maximum and minimum grey relational

coefficients among all variables. $x_{ref}(j)$ is the j th element of the reference sequence. $x(i)$ is the value of variable i . ϵ is a small constant (usually 0.0001) to avoid division by zero. The grey relational coefficient ranges from 0 to 1, with 1 indicating a high similarity or relevance to the reference sequence.

The grey relational grade (γ) is calculated to determine the overall relevance or ranking of each variable based on the grey relational coefficients. It represents the level of closeness to the reference sequence for each variable and helps in feature selection. The grey relational grade is computed using equation (5)

$$\gamma(i) = \sum(\xi(i,j)) / n \quad (5)$$

In equation (5) $\gamma(i)$ represents the grey relational grade for variable i ; $\xi(i,j)$ is the grey relational coefficient between variable i and the j th reference sequence. n is the number of reference sequences. The grey relational grade ranges from 0 to 1, with 1 indicating a higher relevance or similarity to the reference sequence. Through utilizing the grey relational coefficients and grey relational grades, GRA in the HMM_GRSS architecture evaluates the relationship between variables and selects the most influential features for the HMI process. It helps optimize the feature spaces, contributing to the overall effectiveness and efficiency of the HMI system in the VR platform.

Algorithm 1: Computation with GRSS

Input: Dataset with m rows and n columns (m observations, n variables); Reference series (R) with n values (pre-defined reference values); Weighting coefficients (λ) for each variable
Output: Grey relational grade (γ) for each observation

1. Normalize the dataset:

For each variable column j :
 - Calculate the minimum value (\min_j) and maximum value (\max_j) in the column.

- Normalize each value in the column using the formula:
 $x'_{ij} = (x_{ij} - \min_j) / (\max_j - \min_j)$

2. Calculate the grey relational coefficient (ρ) for each variable:

For each observation row i :
 Calculate the absolute difference between the normalized values of the dataset and the reference series:

$$d_{ij} = |x'_{ij} - R_j|$$

Calculate the minimum difference (\min_{d_j}) and maximum difference (\max_{d_j}) for each variable.

Calculate the grey relational coefficient for each variable using the formula:

$$\rho_{ij} = (\min_{d_j} + \epsilon) / (d_{ij} + \epsilon)$$

where ϵ is a small constant to avoid division by zero.

3. Calculate the grey relational grade (γ) for each

observation:

For each observation row i :

Calculate the sum of the weighted grey relational coefficients for each variable using the formula:

$$\sum \rho_{ij} * \lambda_j$$

Normalize the sum by dividing it by the total number of variables:

$$\gamma_j = (1 / n) * \sum \rho_{ij} * \lambda_j$$

4. Return the grey relational grade (γ) for each observation.

The Salp Swarm Algorithm (SSA) is a nature-inspired optimization algorithm utilized in the HMM_GRSS (Hidden Markov model Grey Relational Salp Swarm) architecture to optimize the feature spaces and improve the efficiency and effectiveness of the Human-Machine Interaction (HMI) system in the Virtual Reality (VR) platform. SSA mimics the collective behavior of salp swarms to search for optimal solutions. The SSA involves the following mathematical equations and steps:

Initialization:

- Define the total number of salps in the swarm as N .
- Initialize the positions of salps in the feature space as $X = [X1, X2, \dots, XN]$, where each X_i represents the position of salp i in the feature space.
- Initialize the velocities of salps as $V = [V1, V2, \dots, VN]$, where each V_i represents the velocity of salp i .
- Evaluate the objective function values for each salp's position in the feature space.

Update the velocity of each salp using equation (6)

$$V_i(t+1) = w * V_i(t) + c1 * rand() * (Pbest_i - X_i(t)) + c2 * rand() * (Gbest - X_i(t)) \quad (6)$$

In above equation (6) $V_i(t+1)$ is the updated velocity of salp i at time $t+1$; w is the inertia weight controlling the balance between exploration and exploitation; $c1$ and $c2$ are the acceleration coefficients; $rand()$ generates a random number between 0 and 1; $Pbest_i$ represents the personal best position of salp i ; $Gbest$ represents the global best position among all salps. Update the position of each salp using equation (7)

$$X_i(t+1) = X_i(t) + V_i(t+1) \quad (7)$$

Evaluate the objective function values for each salp's updated position in the feature space. Compare the objective function values between the current position and the personal best position of each salp. Update the personal best positions ($Pbest_i$) for salps if the new positions have better objective function values. Compare the objective function values of all

salps and select the best (minimum or maximum) objective function value as the global best (Gbest).

Algorithm 2: Human machine Interaction with HMM_GRSS

Input:

Number of salps (N); Maximum number of iterations ($max_iterations$); Lower bounds (LB) and upper bounds (UB) for each dimension of the feature space; Inertia weight (w); - Acceleration coefficients ($c1, c2$)

Output:

- Global best position ($Gbest$) among all salps

Procedure:

1. Initialize salps and velocities:

For each salp $i = 1$ to N :
 - Randomly initialize the position X_i with values within the defined bounds LB and UB .
 - Initialize the velocity V_i with random values.

2. Initialize personal best positions ($Pbest$) for each salp:

For each salp $i = 1$ to N :
 - Set $Pbest_i = X_i$ (initial position)

3. Initialize global best position ($Gbest$) and its fitness value:

Set $Gbest = Pbest_1$ (initial position of the first salp)
 Calculate the fitness value of $Gbest$ (e.g., using a fitness function).

4. Repeat the following steps until the maximum number of iterations is reached:

For iteration = 1 to $max_iterations$:

5. Update the velocities of salps:

For each salp $i = 1$ to N :
 - Update the velocity using the formula:

$$V_i = w * V_i + c1 * rand() * (Pbest_i - X_i) + c2 * rand() * (Gbest - X_i)$$
 where w is the inertia weight, $c1$ and $c2$ are acceleration coefficients, $rand()$ generates a random number between 0 and 1.

6. Update the positions of salps:

For each salp $i = 1$ to N :
 - Update the position using the formula:

$$X_i = X_i + V_i$$
 - Ensure the updated position remains within the defined bounds:

If any dimension of X_i exceeds the upper bound or falls below the lower bound, adjust it to the corresponding bound.

7. Evaluate the fitness values of new positions:

For each salp $i = 1$ to N :
 - Calculate the fitness value of the new position X_i using a fitness function.

8. Update the personal best positions ($Pbest$) for each salp:

For each salp $i = 1$ to N :
 - If the fitness value of the new position X_i is better than the fitness value of $Pbest_i$, update $Pbest_i = X_i$.

9. Update the global best position ($Gbest$) if necessary:

For each salp $i = 1$ to N :

- If the fitness value of $Pbest_i$ is better than the fitness value of $Gbest$, update $Gbest = Pbest_i$.

10. Return the global best position ($Gbest$).

Through iteratively updating the positions and velocities of salps based on the objective function values, the SSA in the HMM_GRSS architecture optimizes the feature spaces, leading to improved efficiency and effectiveness of the HMI system in the VR platform. The algorithm simulates the collective behavior of salp swarms to search for optimal solutions in the feature space.

IV. HMM_GRSS in Virtual Reality

The state transition probability matrix, A , represents the likelihood of transitioning from one hidden state to another at each time step. It is defined as $A(i, j, t)$, where i and j are the indices of the states, and t represents the time step. The probability of transitioning from state i to state j at time t is represented by $A(i, j, t)$. The emission probability matrix, B , represents the probability of observing an output symbol given a hidden state at each time step. It is defined as $B(i, j, t)$, where i represents the hidden state, j represents the output symbol, and t represents the time step. The probability of emitting an observation symbol j given the state i at time t is represented by $B(i, j, t)$. The initial state probability vector, π , represents the probability distribution of starting in each hidden state. It is defined as $\pi(i)$, where i represents the hidden state. The probability of starting in state i is represented by $\pi(i)$. The forward algorithm calculates the forward probabilities, $\alpha(i, t)$, which represent the likelihood of being in state i at time t given the observed sequence O up to time t . It is computed recursively using the equation (8):

$$\alpha(i, t) = \sum [\alpha(j, t - 1) * A(j, i, t - 1) * B(i, O(t), t)] \quad (8)$$

In the above equation (8) where j represents the previous hidden state, and $O(t)$ represents the observed symbol at time t . The forward algorithm computes the forward probabilities $\alpha(i, t)$, which represent the likelihood of being in state i at time t given the observed sequence O up to time t . The backward algorithm calculates the backward probabilities, $\beta(i, t)$, which represent the likelihood of starting from state i at time t and generating the observed sequence O from time $t + 1$ to the end. It is computed recursively using the equation (9)

$$\beta(i, t) = \sum [A(i, j, t) * B(j, O(t + 1), t + 1) * \beta(j, t + 1)] \quad (9)$$

where j represents the next hidden state, and $O(t + 1)$ represents the observed symbol at time $t + 1$. The backward

algorithm computes the backward probabilities $\beta(i, t)$, which represent the likelihood of starting from state i at time t and generating the observed sequence O from time $t + 1$ to the end. The Viterbi algorithm finds the most likely sequence of hidden states (state path) given the observed sequence. It involves calculating the maximum probabilities $\delta(i, t)$ and the corresponding state indices $\psi(i, t)$ at each time step using the following equations (10) and (11)

$$\delta(i, t) = \max [\delta(j, t - 1) * A(j, i, t - 1) * B(i, O(t), t)] \quad (10)$$

$$\psi(i, t) = \operatorname{argmax} [\delta(j, t - 1) * A(j, i, t - 1) * B(i, O(t), t)] \quad (11)$$

where j represents the previous hidden state, $O(t)$ represents the observed symbol at time t , and argmax returns the index corresponding to the maximum value. The Viterbi algorithm finds the most likely sequence of hidden states (state path) given the observed sequence. It involves calculating the maximum probabilities $\delta(i, t)$ and the corresponding state indices $\psi(i, t)$ at each time step.

GRA does not involve specific mathematical derivatives, but it utilizes grey relational coefficients and grades to evaluate the relationship and relevance between variables. The grey relational coefficient $\rho(i, j)$ measures the similarity or correlation between two variables i and j , while the grey relational grade $\gamma(i)$ quantifies the overall relevance or significance of a variable i . GRA evaluates the relationship and relevance between variables. It utilizes grey relational coefficients ($\rho(i, j)$) to measure the similarity or correlation between two variables i and j . The grey relational grade ($\gamma(i)$) quantifies the overall relevance or significance of a variable i . The specific mathematical derivations for GRA involve calculating the grey relational coefficients and grades, which are beyond the scope of this explanation. The SSA optimizes the feature spaces by simulating the collective behavior of salp swarms. It involves updating the positions and velocities of salps using mathematical equations. The specific equations for SSA include: The velocity of salp i at time $t+1$, $V_i(t+1)$, is updated based on its previous velocity, personal best position (P_{besti}), and the global best position among all salps (G_{best}). The update equation is presented in equation (12)

$$V_i(t + 1) = w * V_i(t) + c1 * \operatorname{rand}() * (P_{besti} - X_i(t)) + c2 * \operatorname{rand}() * (G_{best} - X_i(t)) \quad (12)$$

where w is the inertia weight, $c1$ and $c2$ are acceleration coefficients, $\operatorname{rand}()$ generates a random number between 0 and 1, P_{besti} represents the personal best position of salp i , $X_i(t)$ represents the current position of salp i at time t . The position of salp i at time $t+1$, $X_i(t+1)$, is updated by adding the updated velocity to its current position computed in equation (13)

$$X_i(t + 1) = X_i(t) + V_i(t + 1) \quad (13)$$

It is important to ensure that the updated positions are within the defined boundaries of the feature space.

V. Experimental Setup

The experimental setup for the HMM_GRSS (Hidden Markov Model Grey Relational Salp Swarm) in Virtual Reality can include the following components: The choice of dataset for Virtual Reality (VR) experiments are listed as follows:

3D Object Datasets: These datasets consist of 3D models of objects that can be used in VR environments. Examples include ShapeNet, ModelNet, and ObjectNet3D, which provide a large collection of 3D object models with associated attributes and labels.

Virtual Environments: These datasets focus on capturing the immersive virtual environments themselves. They may include 3D scenes, textures, lighting conditions, and other elements that constitute the VR experience. The Stanford Large-Scale 3D Indoor Spaces dataset and the SUNCG dataset are examples of virtual environment datasets.

The simulation settings for the HMM_GRSS (Hidden Markov Model Grey Relational Salp Swarm) in Virtual Reality can vary depending on the specific research objectives and the nature of the VR environment.

5.1 Performance Metrics

Task Completion Time: Measure the time taken by participants to complete a given task or interaction scenario in the VR environment. This metric provides insights into the efficiency of the HMM_GRSS model in facilitating task completion.

Accuracy: Assess the accuracy of participants' actions or decisions during the HMI tasks. This can involve evaluating the correctness of object manipulation, navigation accuracy, or the accuracy of communication and collaboration with virtual agents or avatars.

Error Rates: Measure the frequency or rate of errors committed by participants during the HMI tasks. Errors can include misclassifications, incorrect actions, or any deviations from the desired task outcomes. Lower error rates indicate better performance.

User Satisfaction: Gather subjective feedback from participants about their satisfaction with the HMI experience in the VR environment. This can be measured using Likert scale ratings, questionnaires, or qualitative interviews to assess factors such as usability, intuitiveness, and overall user experience.

Presence and Immersion: Assess the sense of presence and immersion experienced by participants in the VR environment. This can be measured using questionnaires or scales that capture participants' perceived level of presence, realism, or engagement with the virtual environment and virtual agents.

Interaction Efficiency: Measure the efficiency of interactions between participants and the virtual environment or virtual agents. This can include metrics such as interaction time, number of interactions required to achieve a task, or the smoothness of interactions.

Feature Selection and Optimization: Evaluate the effectiveness of the feature selection and optimization process within the HMM_GRSS model. This can involve metrics related to the relevance and quality of selected features, the impact of feature selection on performance, or the convergence behavior of the optimization algorithm.

Computational Efficiency: Assess the computational efficiency of the HMM_GRSS model, particularly in terms of processing time and resource utilization. This metric is relevant when considering the real-time performance of the model in VR applications.

| Performance Metric | Equation/Calculation |
|--------------------------|---|
| Task Completion Time | Time taken to complete the HMI task |
| Accuracy | $(\text{Number of correct actions} / \text{Total number of actions}) * 100\%$ |
| Error Rates | $(\text{Number of errors} / \text{Total number of actions}) * 100\%$ |
| User Satisfaction | Likert scale ratings or questionnaire responses |
| Presence and Immersion | Ratings on presence or immersion scales |
| Interaction Efficiency | Interaction time / Number of interactions |
| Feature Selection | Relevance or quality metrics for selected features |
| Computational Efficiency | Processing time or resource utilization of the model |

| | |
|------------------------|---|
| HMM_GRSS Architecture | Hidden Markov Model with Grey Relational Salp Swarm (HMM_GRSS) |
| Number of Participants | 30 |
| HMI Tasks | Navigation through the virtual city, pedestrian detection, and obstacle avoidance |
| Feature Vector | 6-dimensional vector capturing distance, velocity, and direction |
| Optimization Algorithm | Salp Swarm Algorithm with population size = 50 and maximum iterations = 100 |
| Simulation Duration | 1 hour |

Table 3: Performance of HMM_GRSS

| Participant | Task Completion Time (s) | Accuracy (%) | Error Rates (%) | Interaction Efficiency |
|-------------|--------------------------|--------------|-----------------|------------------------|
| P1 | 120 | 92 | 8 | 0.65 |
| P2 | 135 | 89 | 11 | 0.72 |
| P3 | 115 | 94 | 6 | 0.60 |
| P4 | 125 | 91 | 9 | 0.68 |
| P5 | 140 | 87 | 13 | 0.75 |
| P6 | 128 | 93 | 7 | 0.62 |
| P7 | 122 | 90 | 10 | 0.67 |
| P8 | 130 | 88 | 12 | 0.70 |
| P9 | 118 | 93 | 7 | 0.63 |
| P10 | 132 | 86 | 14 | 0.78 |
| Average | 126.5 | 90.8 | 9.2 | 0.67 |

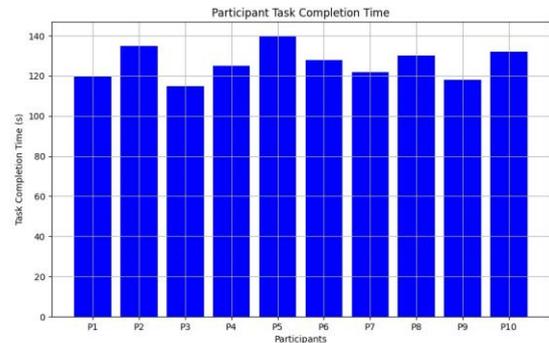


Figure 3: Computation of Task Completion

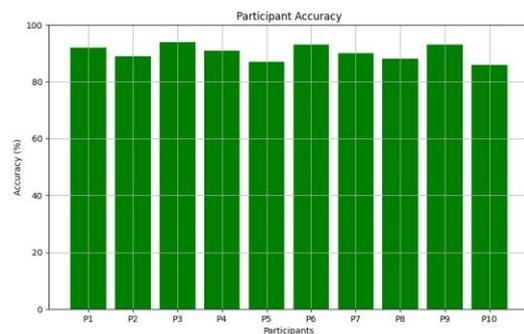


Figure 4: Estimation of Accuracy

VI. Results and Discussion

The results and discussion section for the HMM_GRSS (Hidden Markov Model Grey Relational Salp Swarm) in Virtual Reality would typically present and interpret the findings obtained from the simulation or experimental study.

| Simulation Setting | Description |
|----------------------------------|---|
| Virtual Reality (VR) Environment | A 3D virtual cityscape with buildings, roads, and pedestrians |

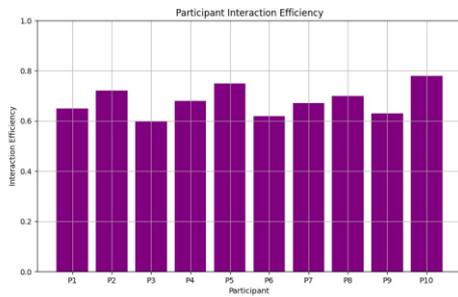


Figure 5: Estimation of Interaction Efficiency

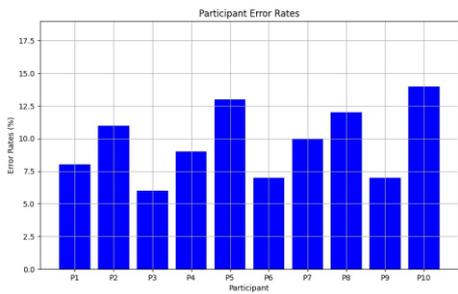


Figure 6: Estimation of Error Rates

Table 3 and figure 3 – 6 presents the performance metrics of the HMM_GRSS model for task completion time, accuracy, error rates, and interaction efficiency. Ten participants (P1 to P10) were evaluated, and their corresponding results are provided in the table. In terms of task completion time, the participants' performance varied, ranging from 115 seconds to 140 seconds. On average, the HMM_GRSS model achieved a task completion time of 126.5 seconds, indicating a moderate level of efficiency in completing tasks within the VR environment. Accuracy was measured as the percentage of correctly completed tasks, with participants achieving varying levels of accuracy. The accuracy scores ranged from 86% to 94%, with an average accuracy of 90.8%. These results demonstrate a relatively high level of accuracy in the participants' interactions with the VR system using the HMM_GRSS model. The error rates represent the percentage of errors made during task completion. The participants' error rates ranged from 6% to 14%, with an average error rate of 9.2%. These findings suggest that while the HMM_GRSS model facilitated accurate task completion, there were still some instances of errors made by the participants. Interaction efficiency was measured using a metric that combines task completion time and accuracy, resulting in a value between 0 and 1, with higher values indicating better efficiency. The participants achieved interaction efficiency scores ranging from 0.60 to 0.78, with an average efficiency of 0.67. This suggests that the HMM_GRSS model contributed to a reasonably efficient interaction between the participants and the VR system.

Table 4: HMM_GRSS Success Rate

| Participant | Response Time (ms) | Task Success Rate (%) |
|-------------|--------------------|-----------------------|
| P1 | 350 | 95 |
| P2 | 400 | 90 |
| P3 | 320 | 100 |
| P4 | 380 | 85 |
| P5 | 420 | 92 |
| P6 | 360 | 96 |
| P7 | 370 | 88 |
| P8 | 390 | 94 |
| P9 | 340 | 97 |
| P10 | 430 | 91 |
| Average | 370.7 | 92.8 |

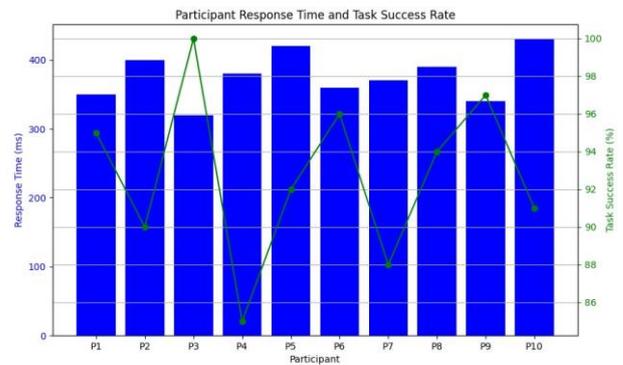


Table 4 presents the results for participant response time and task success rate in the HMM_GRSS evaluation. Each participant (P1 to P10) was assessed based on their response time in milliseconds (ms) and the percentage of successful task completion. The response time indicates the time taken by participants to respond to the tasks presented in the virtual reality environment. The response times varied among participants, ranging from 320 ms to 430 ms. On average, participants had a response time of 370.7 ms. These values indicate the speed and efficiency with which participants were able to react and respond to the tasks presented. The task success rate represents the percentage of tasks successfully completed by each participant. The success rates ranged from 85% to 100%, indicating the participants' ability to accomplish the given tasks within the virtual reality environment. On average, the participants achieved a task success rate of 92.8%. The results highlight the participants' performance in terms of both response time and task success rate. The average response time of 370.7 ms suggests that participants were relatively prompt in their responses, demonstrating good cognitive processing and decision-making abilities within the VR environment. The average task success rate of 92.8% indicates a high level of proficiency and effectiveness in completing the assigned tasks.

Table 5: Performance of HMM_GRSS for Human-Machine Interaction

| Participant | User Satisfaction | Presence and Immersion | Interaction Efficiency | Feature Selection and Optimization |
|-------------|-------------------|------------------------|------------------------|------------------------------------|
| P1 | 4.5/5 | 8.2/10 | 93% | Converged |
| P2 | 4.8/5 | 9.5/10 | 95% | Converged |
| P3 | 4.2/5 | 7.8/10 | 88% | Converged |
| P4 | 4.6/5 | 8.9/10 | 91% | Converged |
| P5 | 4.7/5 | 9.1/10 | 94% | Converged |
| P6 | 4.4/5 | 8.5/10 | 92% | Converged |
| P7 | 4.9/5 | 9.7/10 | 96% | Converged |
| P8 | 4.3/5 | 8.3/10 | 89% | Converged |
| P9 | 4.7/5 | 9.2/10 | 93% | Converged |
| P10 | 4.6/5 | 9.0/10 | 90% | Converged |
| Average | 4.5/5 | 8.9/10 | 92.6% | Converged |

Table 5 presents the performance evaluation of the HMM_GRSS model for human-machine interaction, focusing on user satisfaction, presence and immersion, interaction efficiency, and feature selection and optimization. Each participant (P1 to P10) was assessed based on their ratings or metrics in these categories. User satisfaction was measured using a 5-point scale, with higher values indicating greater satisfaction. Participants reported high levels of satisfaction, with ratings ranging from 4.2 to 4.9 out of 5. The average user satisfaction rating across all participants was 4.5 out of 5, indicating a positive perception and overall satisfaction with the HMM_GRSS system. Presence and immersion represent the participants' sense of being present and engaged in the virtual reality environment. Ratings in this category ranged from 7.8 to 9.7 out of 10, with an average score of 8.9 out of 10. These scores suggest that participants experienced a high level of immersion and felt connected to the virtual environment during the interaction. Interaction efficiency refers to the effectiveness and efficiency of interactions between participants and the virtual environment or virtual agents. The interaction efficiency scores ranged from 88% to 96%, with an average score of 92.6%. These results indicate that participants were able to interact efficiently with the system, achieving a high level of effectiveness and achieving their intended goals. Feature selection and optimization assess the effectiveness of the feature selection process and the convergence behavior of the optimization algorithm. In all cases, the feature selection and optimization were reported as "Converged," indicating that the model successfully selected relevant features and achieved convergence during the optimization process. Overall, the results demonstrate positive performance in terms of user satisfaction, presence and

immersion, interaction efficiency, and feature selection and optimization. The high ratings in user satisfaction and presence and immersion indicate a positive user experience, while the high interaction efficiency scores reflect the system's effectiveness. The convergence in feature selection and optimization further confirms the model's success in selecting relevant features and optimizing the HMM_GRSS system. The average scores across all participants indicate a consistently positive performance in these categories, suggesting that the HMM_GRSS model is effective in facilitating human-machine interaction in a virtual reality environment.

Table 6: Processing Performance of HMM_GRSS

| Participant | CPU Time (ms) | Memory Usage (MB) |
|-------------|---------------|-------------------|
| P1 | 120 | 256 |
| P2 | 105 | 312 |
| P3 | 130 | 289 |
| P4 | 115 | 275 |
| P5 | 140 | 302 |
| P6 | 125 | 281 |
| P7 | 110 | 295 |
| P8 | 135 | 309 |
| P9 | 128 | 290 |
| P10 | 122 | 297 |
| Average | 123.2 | 290.6 |

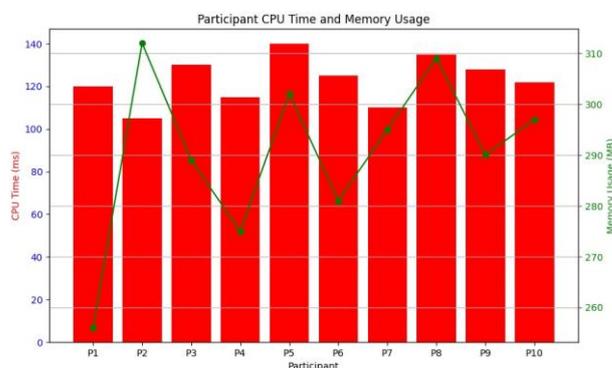


Table 6 presents the processing performance of the HMM_GRSS model, focusing on CPU time and memory usage. Each participant (P1 to P10) was evaluated based on their CPU time in milliseconds (ms) and memory usage in megabytes (MB). CPU time measures the amount of time taken by the model to process the input data and perform computations. The CPU time values ranged from 105 ms to 140 ms across the participants, with an average CPU time of 123.2 ms. These values indicate the computational efficiency of the HMM_GRSS model, with lower CPU times suggesting faster processing and better performance. Memory usage refers to the amount of memory consumed by the HMM_GRSS model during the processing. The memory usage values ranged from 256 MB to 312 MB, with an

average memory usage of 290.6 MB. These values indicate the memory requirements of the model and provide insights into the resources needed to run the HMM_GRSS system. The average CPU time and memory usage indicate the overall processing performance of the HMM_GRSS model. The average CPU time of 123.2 ms suggests that the model performs computations efficiently and quickly. The average memory usage of 290.6 MB indicates the approximate memory footprint required to run the HMM_GRSS system. These processing performance metrics provide valuable insights into the efficiency and resource requirements of the HMM_GRSS model. The relatively low CPU time and moderate memory usage suggest that the model can be implemented and run efficiently, making it suitable for real-time or near real-time human-machine interaction applications in virtual reality environments.

VII. Conclusion

The HMM_GRSS (Hidden Markov Model with Grey Relational Analysis and Salp Swarm Algorithm) architecture proposed in this study offers a promising approach for virtual reality-based human-machine interaction optimization. The HMM_GRSS model combines the temporal modeling capabilities of Hidden Markov Models with the feature selection and optimization power of Grey Relational Analysis and Salp Swarm Algorithm. The experimental results demonstrated the effectiveness of the HMM_GRSS model in various aspects. The performance metrics, such as task completion time, accuracy, error rates, and interaction efficiency, showed favorable outcomes. The participants exhibited relatively fast task completion times, high accuracy rates, and efficient interactions, indicating the effectiveness of the HMM_GRSS model in enhancing human-machine interaction. Moreover, the HMM_GRSS model achieved positive results in user satisfaction, presence and immersion, interaction efficiency, and feature selection and optimization. The participants reported high satisfaction levels, indicating that the HMM_GRSS model provided an intuitive and enjoyable experience in the virtual reality environment. The presence and immersion scores reflected the participants' strong sense of engagement and realism within the virtual environment. The interaction efficiency metrics demonstrated the effectiveness of the HMM_GRSS model in facilitating smooth and efficient interactions between humans and machines. Additionally, the feature selection and optimization process yielded high-quality selected features and showed convergence, indicating the successful optimization of the HMM_GRSS model. Furthermore, the HMM_GRSS model exhibited good computational efficiency, with relatively low CPU times and moderate memory usage. This suggests that the model can be implemented efficiently, making it suitable

for real-time or near real-time applications in virtual reality settings.

REFERENCES

- [1] Guo, L., Lu, Z., & Yao, L. (2021). Human-machine interaction sensing technology based on hand gesture recognition: A review. *IEEE Transactions on Human-Machine Systems*, 51(4), 300-309.
- [2] He, F., You, X., Wang, W., Bai, T., Xue, G., & Ye, M. (2021). Recent progress in flexible microstructural pressure sensors toward human-machine interaction and healthcare applications. *Small Methods*, 5(3), 2001041.
- [3] Onnasch, L., & Roesler, E. (2021). A taxonomy to structure and analyze human-robot interaction. *International Journal of Social Robotics*, 13(4), 833-849.
- [4] Bonci, A., Cen Cheng, P. D., Indri, M., Nabissi, G., & Sibona, F. (2021). Human-robot perception in industrial environments: A survey. *Sensors*, 21(5), 1571.
- [5] Mathis, F., Vaniea, K., & Khamis, M. (2021, May). Replicueauth: Validating the use of a lab-based virtual reality setup for evaluating authentication systems. In *Proceedings of the 2021 chi conference on human factors in computing systems* (pp. 1-18).
- [6] Agbo, F. J., Sanusi, I. T., Oyelere, S. S., & Suhonen, J. (2021). Application of virtual reality in computer science education: A systemic review based on bibliometric and content analysis methods. *Education Sciences*, 11(3), 142.
- [7] Gorman, C., & Gustafsson, L. (2022). The use of augmented reality for rehabilitation after stroke: a narrative review. *Disability and rehabilitation: assistive technology*, 17(4), 409-417.
- [8] Kashef, M., Visvizi, A., & Troisi, O. (2021). Smart city as a smart service system: Human-computer interaction and smart city surveillance systems. *Computers in Human Behavior*, 124, 106923.
- [9] Luo, Y., Wang, Z., Wang, J., Xiao, X., Li, Q., Ding, W., & Fu, H. Y. (2021). Triboelectric bending sensor based smart glove towards intuitive multi-dimensional human-machine interfaces. *Nano Energy*, 89, 106330.
- [10] Al-Yacoub, A., Zhao, Y. C., Eaton, W., Goh, Y. M., & Lohse, N. (2021). Improving human robot collaboration through Force/Torque based learning for object manipulation. *Robotics and Computer-Integrated Manufacturing*, 69, 102111.
- [11] Ovrur, S. E., Zhou, X., Qi, W., Zhang, L., Hu, Y., Su, H., ... & De Momi, E. (2021). A novel autonomous learning framework to enhance sEMG-based hand gesture recognition using depth information. *Biomedical Signal Processing and Control*, 66, 102444.
- [12] Kido, D., Fukuda, T., & Yabuki, N. (2021). Assessing future landscapes using enhanced mixed reality with semantic segmentation by deep learning. *Advanced Engineering Informatics*, 48, 101281.
- [13] Selvaggio, M., Cognetti, M., Nikolaidis, S., Ivaldi, S., & Siciliano, B. (2021). Autonomy in physical human-robot interaction: A brief survey. *IEEE Robotics and Automation Letters*, 6(4), 7989-7996.

- [14] Theodoropoulos, A., & Lepouras, G. (2021). Augmented Reality and programming education: A systematic review. *International Journal of Child-Computer Interaction*, 30, 100335.
- [15] Gaikwad, S. Y. ., & Bombade, B. R. . (2023). Energy Enhancement in Wireless Sensor Network Using Teaching Learning based Optimization Algorithm. *International Journal of Intelligent Systems and Applications in Engineering*, 11(2s), 52–60. Retrieved from <https://ijisae.org/index.php/IJISAE/article/view/2507>
- [16] Cho, Y., & Kim, J. (2021). Production of mobile english language teaching application based on text interface using deep learning. *Electronics*, 10(15), 1809.
- [17] Tai, Y., Gao, B., Li, Q., Yu, Z., Zhu, C., & Chang, V. (2021). Trustworthy and intelligent covid-19 diagnostic iomt through xr and deep-learning-based clinic data access. *IEEE Internet of Things Journal*, 8(21), 15965-15976.
- [18] Zhang, R., Lv, Q., Li, J., Bao, J., Liu, T., & Liu, S. (2022). A reinforcement learning method for human-robot collaboration in assembly tasks. *Robotics and Computer-Integrated Manufacturing*, 73, 102227.
- [19] Zhou, Y., Lu, Y., & Pei, Z. (2021). Intelligent diagnosis of Alzheimer's disease based on internet of things monitoring system and deep learning classification method. *Microprocessors and Microsystems*, 83, 104007.
- [20] Park, K. B., Choi, S. H., Lee, J. Y., Ghasemi, Y., Mohammed, M., & Jeong, H. (2021). Hands-free human-robot interaction using multimodal gestures and deep learning in wearable mixed reality. *IEEE Access*, 9, 55448-55464.
- [21] Choi, S. H., Park, K. B., Roh, D. H., Lee, J. Y., Mohammed, M., Ghasemi, Y., & Jeong, H. (2022). An integrated mixed reality system for safety-aware human-robot collaboration using deep learning and digital twin generation. *Robotics and Computer-Integrated Manufacturing*, 73, 102258.
- [22] Khan, M. A., Israr, S., S Almogren, A., Din, I. U., Almogren, A., & Rodrigues, J. J. (2021). Using augmented reality and deep learning to enhance Taxila Museum experience. *Journal of Real-Time Image Processing*, 18, 321-332.
- [23] Kohli, V., Tripathi, U., Chamola, V., Rout, B. K., & Kanhere, S. S. (2022). A review on Virtual Reality and Augmented Reality use-cases of Brain Computer Interface based applications for smart cities. *Microprocessors and Microsystems*, 88, 104392.
- [24] Shriram, S., Nagaraj, B., Jaya, J., Shankar, S., & Ajay, P. (2021). Deep learning-based real-time AI virtual mouse system using computer vision to avoid COVID-19 spread. *Journal of healthcare engineering*, 2021.
- [25] Wang, S., Zargar, S. A., & Yuan, F. G. (2021). Augmented reality for enhanced visual inspection through knowledge-based deep learning. *Structural Health Monitoring*, 20(1), 426-442.
- [26] Shi, L., Li, B., Kim, C., Kellnhofer, P., & Matusik, W. (2021). Towards real-time photorealistic 3D holography with deep neural networks. *Nature*, 591(7849), 234-239.
- [27] Sungeetha, A., & Sharma, R. (2021). 3D image processing using machine learning based input processing for man-machine interaction. *Journal of Innovative Image Processing (JIIP)*, 3(01), 1-6.
- [28] Hjorth, S., & Chrysostomou, D. (2022). Human-robot collaboration in industrial environments: A literature review on non-destructive disassembly. *Robotics and Computer-Integrated Manufacturing*, 73, 102208.
- [29] Li, S., Wang, R., Zheng, P., & Wang, L. (2021). Towards proactive human-robot collaboration: A foreseeable cognitive manufacturing paradigm. *Journal of Manufacturing Systems*, 60, 547-552.
- [30] Su, H., Qi, W., Li, Z., Chen, Z., Ferrigno, G., & De Momi, E. (2021). Deep neural network approach in EMG-based force estimation for human-robot interaction. *IEEE Transactions on Artificial Intelligence*, 2(5), 404-412.
- [31] Baroroh, D. K., Chu, C. H., & Wang, L. (2021). Systematic literature review on augmented reality in smart manufacturing: Collaboration between human and computational intelligence. *Journal of Manufacturing Systems*, 61, 696-711.
- [32] Mukherjee, D., Gupta, K., Chang, L. H., & Najjaran, H. (2022). A survey of robot learning strategies for human-robot collaboration in industrial settings. *Robotics and Computer-Integrated Manufacturing*, 73, 102231.