Integration of LLMs for Multitasking Workload Prediction in Mixed Reality Environments

Safanah Abbas[1], Heejin Jeong[2], and David He[3]

[1,3]*University of Illinois Chicago, Chicago, Illinois, 60607, USA*
sabbas28@uic.edu
davidhe@uic.edu

[2]*Arizona State University, Mesa, AZ 85212, USA*
heejin.jeong@asu.edu

ABSTRACT

Multitasking in mixed reality (MR) environments introduces unique cognitive demands, particularly in workload management. Accurate workload prediction is critical for optimizing user experience, safety, and performance in such settings. This study proposes a novel framework that integrates large language models (LLMs) with traditional workload assessment tools to enhance prediction accuracy in MR multitasking scenarios. A multitasking experiment involving 36 participants was conducted, combining real-world and virtual tasks, with workload evaluated using NASA-TLX. To address limited sample sizes, synthetic data was generated using generative adversarial networks (GANs), enabling robust model training. We employed a hybrid deep learning model that integrates LLM-generated text embeddings with numerical features in a feedforward neural network (FNN). Results show that integrating LLMs, specifically BERT and GPT-2, significantly improves workload prediction accuracy, with a root mean square error (RMSE) reduction from 6.82 (FNN-only) to 0.95 (BERT-integrated model). The findings underscore the potential of LLMs to augment cognitive workload assessment, supporting more adaptive and scalable human-machine collaboration in MR environments.

1. INTRODUCTION

Multitasking has become increasingly valuable in today's complex digital landscape (Spink et al., 2008). The rapid evolution of computing technologies has not only facilitated multitasking but also redefined how humans interact with digital content. Mixed reality (MR), a spectrum that blends real and virtual environments, is at the forefront of this transformation, enabling seamless interactions in hybrid digital-physical worlds. As Abrash et al. (2021) note, these emerging platforms are expected to shape human-computer interaction for decades to come.

While MR environments offer significant advantages for enhancing multitasking capabilities, they also introduce new cognitive and perceptual challenges. Users may experience limitations such as motion sickness, visual strain, divided attention, reduced performance, and increased cognitive workload (Rokhsaritalemi et al., 2020). Among these, workload, defined as the total mental, physical, or combined demands placed on an individual or system to complete tasks (Matthews et al., 2015), plays a crucial role in maintaining system usability, user well-being, and operational safety.

Accurately assessing and predicting workload in MR environments is essential for optimizing user experiences and ensuring effective human-machine collaboration. One of the most established subjective methods for workload evaluation is the NASA Task Load Index (NASA-TLX), introduced by Hart and Staveland (1988), which measures six key dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. While NASA-TLX provides structured insight into perceived workload, it may not fully capture the nuance and context-dependency of individual user experiences.

To address this limitation, we explore the integration of large language models (LLMs) as a novel approach to enhancing workload prediction. LLMs such as BERT and GPT-2 have demonstrated strong capabilities in natural language understanding, sentiment analysis, and context-aware reasoning. These models are particularly well-suited for analyzing unstructured user inputs, such as verbal feedback or written reflections, which often contain implicit indicators of cognitive and emotional strain. By leveraging their capacity to extract meaning from complex textual data, LLMs offer the potential to infer user workload more accurately and

responsively than traditional models. When used in conjunction with structured tools like NASA-TLX, LLMs can provide a richer, more adaptive framework for workload assessment in multitasking MR scenarios.

Thus, we hypothesize that integrating a pre-trained LLM into the workload evaluation process will enhance the precision and contextual sensitivity of workload prediction in mixed-reality multitasking environments.

Multitasking has become valuable as our environment becomes increasingly complex (Spink et al., 2008). Furthermore, technological advancements have integrated multitasking into our daily lives, with new computing platforms poised to shape our digital interactions for the next 50 years, as explained by Abrash et al. (2021). These technological advancements allow humans to engage seamlessly in digital-physical worlds, known as mixed reality.

Despite the advantages of multitasking in mixed reality environments, which enhance user interaction in the real world compared to other technologies (Rokhsaritalemi et al., 2020), users may encounter challenges stemming from human limitations, such as discomfort, motion sickness, visual impairment, reduced focus, increased workload, and more. Workload, defined as the total mental, physical, or combined effort and demands placed on an individual or a system to complete tasks (Matthews et al., 2015), is a critical factor in human-machine collaboration and is essential for promoting overall well-being and safety within health management.

In this context, integrating advanced technologies like LLMs offers innovative approaches to analyzing and predicting perceived workload from user interactions and feedback. A standard method for assessing perceived workload is the NASA-TLX, the most established and widely used subjective method for detailed workload analysis (Bousdekis et al., 2022). Hart and Staveland (1988) introduced the development of the NASA-TLX to evaluate workload across six dimensions: mental demand, physical demand, temporal demand, performance, effort, and frustration. By combining LLMs with established tools like the NASA-TLX, we can gain deeper insights into the workload, enhancing decision-making processes and improving the reliability and efficiency of human interactions with mixed reality applications. Thus, we hypothesize that integrating a pre-trained LLM model will enhance the workload prediction in mixed-reality multitasking.

2. RELATED WORK

The challenges of multitasking often surface through increased cognitive demands, such as elevated mental workload, greater attention requirements, and limited working memory capacity (Dzubak, 2008; Kudesia et al., 2022). These effects are particularly pronounced because workload, in particular, is believed to increase during multitasking due to decreased available resources for each task (Strayer et al., 2022). In our technology-driven society, shaped by tools designed to support human activity (Carroll, 2017), multitasking has evolved significantly, especially within MR environments, where it has become a fundamental aspect of user interaction (North et al., 2021). In MR settings, multitasking involves simultaneously engaging with both physical and virtual elements (Speicher et al., 2019).

Empirical studies have consistently confirmed that multitasking contributes to increased workload, particularly in MR contexts. For instance, Li et al. (2022) examined mental workload during a simulated flight multitasking scenario, finding that higher workload conditions corresponded with elevated NASA-TLX scores. Similarly, Fick et al. (2023) conducted a study involving medical students and neurosurgeons who performed a virtual tumor detection task using mixed reality, traditional MRI, and a 3D viewer. While MR yielded the best task performance, it also resulted in higher reported mental and temporal workloads compared to the other methods. In another study, Criollo et al. (2024) investigated the cognitive demands of immersive technologies in higher education. Their findings revealed that students experienced a moderate level of mental workload when using virtual and mixed reality tools, significantly higher than when immersive technologies were not used in the learning process, as measured by NASA-TLX.

2.1. Applications of LLMs in Workload Assessment

Exploring the integration of LLMs presents a promising direction for addressing workload challenges in mixed-reality environments. LLMs, a class of advanced artificial intelligence systems, have shown exceptional capabilities in natural language processing, machine translation, and question-answering tasks (Hadi et al., 2023). Their strength lies in their ability to manage complex, context-rich information through extensive pretraining on large-scale datasets and the use of deep neural network architectures (Liu et al., 2024). When combined with established workload assessment tools, LLMs offer the potential to improve our understanding of user experiences and support more informed decision-making processes.

Several recent studies have explored the application of LLMs in workload detection and management. Gao et al. (2024) introduced WorkloadGPT, a language model designed to classify pilot workload into low, medium, and high categories to enhance aviation safety. Their model utilized eye-tracking metrics from 20 pilots, such as gaze fixations, average gaze duration, blink frequency, and pupil diameter, collected during flight simulations of varying difficulty levels. These physiological features were serialized into a natural language format to create input data for the LLM. In addition to physiological data, participants completed the NASA-TLX questionnaire to provide subjective workload assessments.

The researchers employed the pre-trained ChatGLM3-6B as the model backbone, fine-tuned using Low-Rank Adaptation (LoRA), and expanded the dataset with Generative Adversarial Networks (GANs) for data augmentation. The resulting model achieved a classification accuracy of 87.3%, with less than 2% standard deviation across participants, significantly outperforming traditional machine learning models such as random forest (69.2%), support vector machine (62.4%), and k-nearest neighbor (60.2%). While WorkloadGPT demonstrated high classification performance, it did not address real-time or continuous workload prediction.

In another study, Colabianchi et al. (2024) examined the use of Digital Intelligent Assistants (DIAs) powered by LLMs to support manufacturing assembly tasks. Thirty participants were divided into two groups: one followed traditional instructions, while the other used the DIA-enhanced system. Results showed that the DIA group experienced reduced workload across all NASA-TLX dimensions, including notable improvements in mental demand (14.67%), temporal demand (21.34%), and effort (10.67%).

Similarly, Sonawani et al. (2024) introduced the SiSCo (Signal Synthesis for Effective Human-Robot Communication) framework, which integrates LLMs to generate intuitive visual cues in a mixed-reality assembly task. The system synthesized visual signals based on contextual task information using hierarchical LLM queries. These signals were projected into the environment to assist 21 participants during task execution. NASA-TLX results revealed a 46% reduction in reported cognitive load compared to conventional language-based guidance, demonstrating the effectiveness of LLM-driven visual communication.

In contrast, Nam et al. (2024) found a limited impact when evaluating an LLM-augmented tool designed to assist programmers within integrated development environments (IDEs). Although 32 participants reported that the tool improved ease of use and reduced perceived time pressure, most NASA-TLX dimensions, particularly mental demand, did not show statistically significant differences compared to the baseline condition without LLM assistance.

Despite the mixed outcomes, these studies collectively suggest that the integration of LLMs holds substantial promise for improving workload assessment and management. From enhancing classification accuracy to enabling more intuitive human-machine interactions, LLMs have the potential to overcome limitations in traditional approaches. As Gao et al. (2024) suggest, these models may ultimately transform workload detection systems, providing scalable and adaptive solutions across a wide range of applications.

2.2. Limitations in Leveraging LLMs for Effective Workload Assessment

Explicit workload modeling plays a critical role in enabling system designers to anticipate users' cognitive demands during the early stages of system development (Xie & Salvendy, 2000). Despite the availability of such models, a persistent gap remains in translating them effectively into real-world applications. As our understanding of human cognitive processes continues to evolve, there is a significant opportunity to bridge the divide between theoretical research and practical implementation (Card et al., 2018).

In the domain of LLMs, much of the existing research has focused on evaluating mental workload using tools like the NASA-TLX, particularly in tasks that involve language comprehension or generation. However, relatively little work has explored the use of LLMs to build predictive models that assess or classify workload directly. This represents a key limitation, as the structured, tabular data derived from NASA-TLX scores lacks the rich, localized features that LLMs are optimized to process in text or image-based formats. Furthermore, the integration of multiple data modalities—such as physiological signals, numerical indicators, and subjective assessments—poses a challenge for standard LLM architectures, which are inherently more effective with natural language inputs.

Another major obstacle is the difficulty of collecting sufficiently large datasets for training complex models in human-machine interaction scenarios, such as those found in mixed-reality environments. Small sample sizes can lead to overfitting and reduced generalization performance, underscoring the importance of data augmentation techniques (Ru et al., 2024). However, synthetic data generation introduces its own challenges. As noted by Rashid et al. (2019), augmented samples often suffer from low fidelity and may not accurately reflect real-world data distributions, thereby introducing uncertainty and reducing model reliability.

Overcoming these limitations offers a compelling opportunity to advance the use of LLMs in workload assessment, particularly within the context of mixed-reality human-machine collaboration. Improved methods for integrating multimodal data and generating high-quality, diverse training samples could enhance the robustness of workload classification and prediction models. For industries leveraging mixed reality technologies, such advancements would provide critical insights into workers' cognitive demands, which are essential for effective task management, user acceptance, and overall system performance (Widiastuti et al., 2020).

3. THE METHODOLOGY

The framework of the integrated LLM model for workload prediction in a mixed environment is provided in Figure 1.

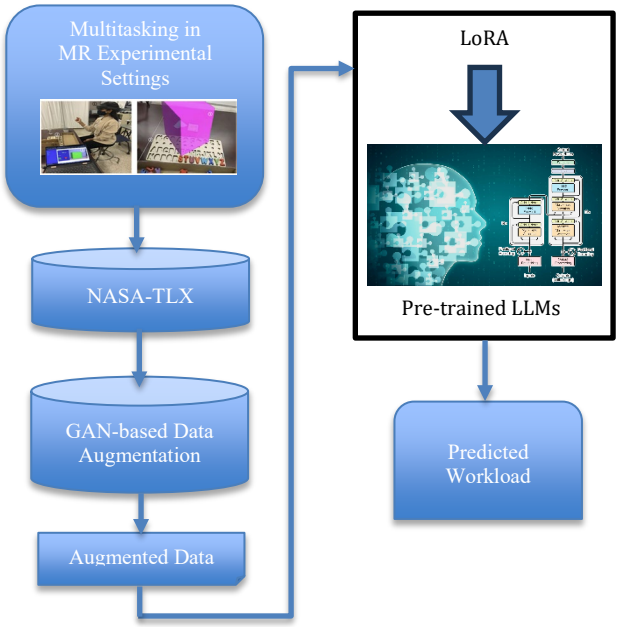


Figure 1. The Integrated Mixed Reality Workload Prediction Framework.

As shown in Figure 1, the integrated MR workload prediction framework consists of two key components: GAN-based data augmentation and LLM-based workload prediction. These two key components are explained in the following sections.

3.1. GAN-based Data Augmentation

Applying deep learning and LLMs to workload prediction normally requires a large amount of data. Generating such a huge amount of data from a mixed reality experiment could be expensive and infeasible. Generating a large amount of synthetic data from a small set of experimental data using data augmentation represents an attractive approach for meeting the challenge. One effective data augmentation method is GANs. GANs are a type of deep learning model composed of two components, a generator (G) and a discriminator (D), that are trained simultaneously in an adversarial manner. The generator G attempts to produce realistic data, while the discriminator D learns to distinguish real from synthetic data, ultimately enhancing the quality of the generated outputs (as shown in Figure 2).

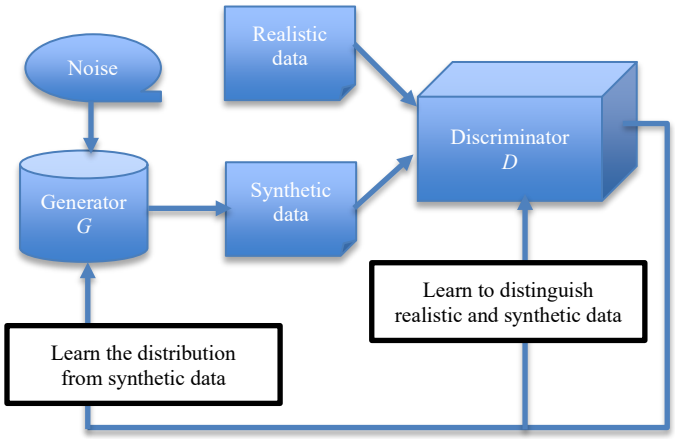


Figure 2. The process of GANs.

As shown in Figure 2, after the generator and the discriminator are trained individually, the GAN is trained on a newly generated batch of synthetic samples labeled as realistic for testing. Then, after each epoch, three losses are calculated: generator loss (L_G), discriminator loss (L_D), and GAN loss (L_{GAN}). The GAN loss is the sum of both losses to measure the overall performance of the GAN model. Equations (1), (2), and (3) represent the mathematical expressions of each loss.

$$L_G = E_{z\sim P(z)}\{logD[G(z)]\} \quad (1)$$

$$L_D = E_{x\sim P(x)}\{logD(x)\} + E_{z\sim P(z)}\{\log(1 - D[G(z)])\} \quad (2)$$

$$L_{GAN} = L_G + L_G \quad (3)$$

In Equation (1), z is a noise vector of synthetic data. $E_{z\sim P(z)}$represents the expectation over the latent variable z sampled from a prior distribution. This expectation calculates the average value of $logD[G(z)]$, where $G(z)$ is the generator's output, and $D[G(z)]$ is the discriminator's probability that the generated sample is real. The generator aims to maximize this quantity, meaning it tries to generate samples that the discriminator classifies as real with high confidence (i.e., $D[G(z)]$ close to 1). The goal is to minimize the L_G, which drives the generator to produce more realistic samples that the discriminator is more likely to classify as real.

In Equation (2), $E_{x\sim P(x)}\{logD(x)\}$ calculates the expected value of $logD(x)$ for real samples x drawn from the true distribution $P(x)$. This encourages the discriminator to classify real samples as real. $E_{z\sim P(z)}\{\log(1 - D[G(z)])\}$ computes the expected value of $\log(1 - D[G(z)])$ for generated samples, encouraging the

discriminator to classify generated samples as fake (i.e., $D[G(z)]$ close to 0). Together, these two terms push the discriminator to maximize its ability to distinguish real from fake samples.

The limitation of GANs, much like the other generative models, lies in performing best when dealing with image data. This is because when dealing with images, there is a structure that can be utilized to produce additional artificial photos. This becomes more complicated when dealing with tabular data, an area where GANs are not widely used because of the absence of a structural advantage, like in images. Jordon et al. (2022) indicated that the field of tabular data generation still needs to address such limitations.

3.2. Workload Prediction using LLMs

In general, two types of data are generated from the mixed reality multitasking experiments: text data and numerical data. To integrate the LLMs into the application of workload prediction, the LLMs are used to convert the text input data into numerical embeddings. These numerical embeddings of the text data, along with the numerical data, are input into a feedforward neural network (FNN) to predict the workload, as shown in Figure 3. In this strategy, a combined loss function is used to train the model. The combined loss function is represented by the following equation:

$$Loss = (\alpha)Loss_{LLM} + (1-\alpha)Loss_{FNN} \quad (4)$$

In (4), α is the weight, $0 \leq \alpha \leq 1$.

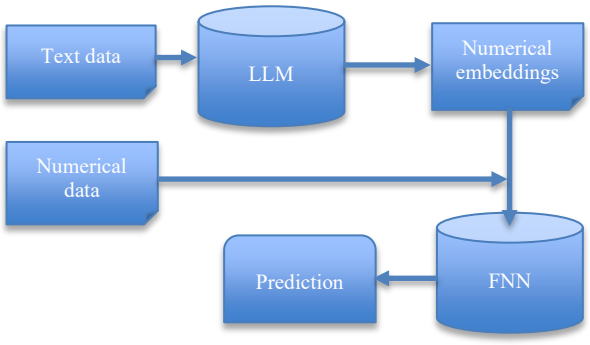


Figure 3. LLM integration strategy.

4. THE EXPERIMENTAL SETUP AND DATA

In this paper, the effectiveness of the presented approach is demonstrated with an experimental study. In this study, 36 participants performed a mixed-reality multitasking experiment and measured their corresponding workload via NASA-TLX.

4.1. The Multitasking Mixed Reality Experiment Setting

For our experiment, we recruited 36 eligible participants of an equal number of males and females, with an average age of 23.9 and a standard deviation of 4.22. All participants had normal vision and no hearing impairments. However, they reported varying levels of experience with AR/VR devices. The majority of the participants were right-handed. The study obtained approval from the Institutional Review Board (IRB) at the University of Illinois Chicago (IRB# 2020-0466).

A combination of real-world (RW) and virtual-world (VW) tasks was assigned for multitasking. A block-matching task was assigned for the physical world task. It assesses participants' workload using Getianlai toys educational material, including English letter blocks and a board for pairing them with their hands within a 90-second time frame. This real-world task was selected because of its simplicity and ability to provide data for the dependent variable workload. This task includes a visual search, which affects human workload, as is evident in previous studies such as Dang et al. (2020). An *N*-back task application was developed and augmented into the HoloLens2 device for the VW task. This task assesses working memory, where the '*N*' parameter is the number of steps required to recall information from memory for a given stimulus (Chen et al., 2008). The virtual *N*-back task used in our experiment had two N values: $N = 1$ and $N = 2$. This dynamic measure affects the working memory, which affects presence in virtual environments (Rawlinson et al., 2012). In this experiment, the *N*-back involved matching the colors of a virtual cube for one step and two steps through hand gestures in a given 90-second frame that shut off automatically from the application. A total of 34 stimuli were recorded for each participant. For more details of the experiment design, please refer to Abbas and Jeong (2024a, 2024b).

The apparatus used in the experiment was a video camera to record participants' performance, a HoloLens 2 headset for the virtual task, a board for the physical task, and a laptop for recording the collected data. Figure 4 shows the experimental setup.

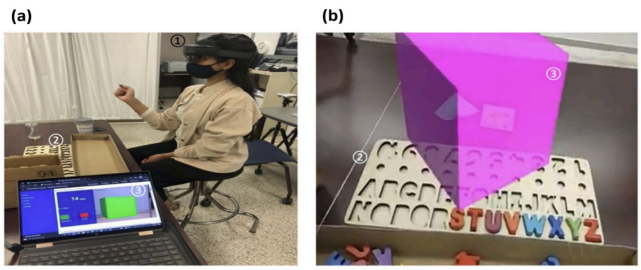


Figure 4. Experimental setup: (a) a perspective from the experimenter's viewpoint showing a participant wearing

HoloLens and engaging with a virtual cube that is displayed on the experimenter's laptop screen, and (b) a view captured from the participant's point of view while interacting with a virtual cube and pairing English letters to a wooden board concurrently.

Four multitasking conditions were assigned per participant, as outlined in Table 1.

Table 1. Description of multitasking conditions

Condition	Definition
Easy RW and Easy VW	No order is given for the RW task and $N = 1$ for the VW task.
Easy RW and Hard VW	No order is given for the RW task and $N = 2$ for the VW task.
Hard RW and Easy VW	An order is given for the RW task and $N = 1$ for the VW task.
Hard RW and Hard VW	An order is given for the RW task and $N = 2$ for the VW task.

Four independent variables were selected; the task difficulty level variable is categorical with four levels, gender is also categorical with two levels, and familiarity and experience with AR/VR handheld and wearable devices are numerical and expressed in percentages. Age and dominant hand variables were excluded because of their limited variability in the collected data. The dependent variable, i.e., the output variable, is the weighted workload average from the six NASA-TLX dimensions.

4.2. Data Collection

Each participant performed two practice trials for each world task before the experiment took place to acquaint themselves with the nature of the experiment. After each practice, they completed a pairwise comparison survey to evaluate subjective workload using the NASA-TLX measure. In this survey, they selected the dimensions they believed had a higher impact in each of the 15 pairs. After that, the four multitasking conditions were assigned to participants randomly. After performing each condition of task difficulty, they completed the post-task NASA-TLX to give a rating for the demand of each condition on a scale from 0 to 100% for each workload dimension based on how they perceived it. All surveys were filled out using the Qualtrics survey tool.

Our study gathered data from 144 observations from the 36 participants, each performing four multitasking conditions, and no missing data was present. The descriptive statistics for each multitasking condition are shown in Table 2.

Table 2. Summary of descriptive statistics of the NASA-TLX dimensions across the multitasking conditions

NASA-TLX Dimensions	Multitasking Conditions			
	Easy RW–Easy VW	Easy RW–Hard VW	Hard RW–Easy VW	Hard RW–Hard VW
Mental Demand Mean (SD)	41.11 (26.57)	44.75 (26.48)	41.06 (27.48)	44.11 (27.64)
Physical Demand Mean (SD)	63.14 (23.25)	75.39 (19.29)	72.42 (26.25)	82.19 (23.03)
Temporal Demand Mean (SD)	55.17 (23.33)	64.56 (25.41)	67.67 (27.66)	66.47 (28.56)
Performance Mean (SD)	51.67 (24.14)	50.83 (22.85)	47.53 (22.60)	52.86 (27.55)
Effort Mean (SD)	66.42 (22.50)	72.89 (19.79)	75.25 (20.34)	82.11 (17.80)
Frustration Mean (SD)	36.53 (28.14)	44.28 (27.44)	49 (32.78)	53.14 (33.44)

The six NASA-TLX dimensions results were combined into a weighted average for the output overall workload. This weighted rating was calculated by taking the recorded responses to each NASA-TLX six dimensions for each participant in the pairwise comparison survey, which assessed 15 pairs. The ultimate weighted rating score was calculated by dividing the sum-product of the inputs of each NASA-TLX dimension (0-100%) from the post-task survey and the corresponding counts from the pairwise survey by 15. Equation (5) is the formula for calculating the weighted average.

$$Workload = \frac{\sum_{i=1}^{6}(Rating\ Response \times count)_i}{15} \tag{5}$$

5. THE RESULTS

5.1. GAN Generated Synthetic Data

The GAN model was implemented on the dataset in a Google Colab notebook. The observations from each condition were treated as individual datasets, as each observation corresponds to a distinct participant. To account for the balanced gender variable, the four datasets were further divided by gender, resulting in a total of eight datasets with

18 observations each. This ensured equal representation of males and females in the synthesized data, consistent with the original dataset. This implies that both the task difficulty and gender variables were omitted during the training of the GAN and reintroduced afterward.

The traditional GAN is applied with an 80/20 split rule, where approximately 80% of the original dataset is used for training and 20% for testing after installing the necessary libraries, including NumPy, Pandas, and TensorFlow.Keras, and Scikit-learn. Three neural networks were built: the generator, the discriminator, and the GAN, which combines both the generator and the discriminator to compete with each other. The 'build_model' function is used to construct the three models implemented as sequential deep-learning models. The generator model is built with a total of four dense layers, while the discriminator model is constructed with three dense layers. The 'LeakyReLU' activation function is employed in the hidden layers for both the generator and the discriminator models with a small positive value of 0.2 for the alpha parameter. This parameter controls the slope of the function for the negative values. The choice of using the 'LeakyReLU' function instead of the 'ReLU' function is because the 'ReLU' function can result in inactive neurons during training. This occurs because the 'ReLU' assigns a value of 0 to all negative inputs. In contrast, the 'LeakyReLU' function resolves this problem by introducing a small slope for the negative inputs instead of setting them to 0, ensuring that neurons stay active during training and avoiding the issue of inactive neurons that could occur when using 'ReLU' (Salam et al., 2021). However, both functions were experimented with individually, and the 'LeakyReLU' achieved better results.

The sigmoid function is utilized in both the generator and discriminator output layers. Using the 'sigmoid' activation function in the generator ensures that the generated values match the percentage range of the original samples. It transforms the input into a scale from 0 to 1, making it ideal for representing probabilities or values within a range. Although other activation functions like 'tanh' and 'linear' were tested for experimentation, they did not produce better outcomes. The discriminator's 'sigmoid' activation function assigns values closer to 0 to indicate fake samples and values nearer to 1 to indicate actual samples. Then, the GAN model is trained as a binary classification problem, aiming to classify samples as real or fake. The combined GAN model uses the 'binary cross entropy' loss function. The GAN model is optimized using the Adaptive Moment Estimation (Adam) optimizer, a commonly used stochastic gradient descent optimization algorithm in deep learning (Zaheer & Shziya, 2019).

After the generator and the discriminator were trained individually, the GAN was trained on a newly generated batch of fake samples labeled as real for testing. Then, after each epoch, from equations (1) - (3), three losses were calculated: L_G, L_D, and L_{GAN}.

A total of 1000 epochs were utilized to train the model. However, to prevent overfitting, an early stopping rule is implemented to monitor the model's performance. It stops the training once there are no more improvements in the GAN loss.

Throughout each epoch, the model assesses the current GAN loss compared to a variable called 'best_loss'. This variable keeps track of the best GAN losses achieved, with a threshold set at 0.001 called 'min_delta'. This threshold determines what qualifies as an improvement in the GAN loss. If the current loss surpasses the 'best_loss' by at least 'min_delta,' the 'best_loss' gets updated with the current loss value, and the counter variable 'wait' resets to 0. Otherwise, the 'wait' is incremented by 1. If there is no improvement in the GAN loss for a specific number of epochs based on a parameter known as 'patience', then early termination is triggered. The 'patience' parameter is set to 20. Multiple runs are performed within a loop using different batch sizes to identify the optimal batch size that yields the best performance for the model. Since each dataset consists of 18 samples, batch sizes of 2, 3, 6, and 9 were used for experimentation. The optimal batch size is selected as 2 based on comparing the best GAN loss achieved in each trial. Table 3 summarizes the best GAN loss achieved for every batch size, categorized by each categorical variable.

Table 3. Summary of the best GAN loss achieved for each batch size per condition.

Condition	Batch Size	Best L_{GAN} (Male)	Best L_{GAN} (Female)
Easy RW – Easy VW	2	0.628	0.590
	3	0.83	0.609
	6	0.784	0.621
	9	0.817	0.631
Easy RW – Hard VW	2	0.663	0.598
	3	0.851	0.691
	6	0.878	0.686
	9	0.844	0.644
Hard RW – Easy VW	2	0.628	0.534
	3	1.005	0.786
	6	1.013	0.774
	9	0.996	0.756
Hard RW – Hard VW	2	0.519	0.604
	3	0.943	0.731
	6	0.981	0.737
	9	0.997	0.662

Multiple trials were carried out using different values of 'latent_dim,' such as 10, 100, and 500, to determine if adjusting this parameter would affect output diversity, as it influences the variety of the generated output. However, no significant differences were observed; therefore, the 'latent_dim' was ultimately set to 500 because the larger the latent_dim, the more varied the data can be. After training the

model with the optimal batch size, 625 fake samples are generated per dataset from the 'generate_samples' function, employing the generator network. This results in a combined total of 5,000 generated fake samples. Several experiments were conducted to determine the appropriate number of fake samples to generate from our original data. While it is generally desirable to have more data, it is crucial to strike a balance between data quality and diversity. The resulting fake samples are then stored in a data frame named 'fake_data'. Lastly, the 'fake_data' data frame is converted into a CSV file to be called for evaluation. The results of the assessment are presented in the results section. Eventually, the real (original) and the fake (GAN-generated) are combined after evaluation. This combined dataset is then used to build a prediction model for the weighted rating variable.

The synthesized data by GAN was evaluated. Since all variables were continuous, the performance of the model on the unseen data (testing set) was evaluated by measuring the distribution matching evaluation using the Kolmogorov-Smirnov (KS) test between these two sets as a part of the evaluation process (Kolmogorov & Smirnov, 1933). The assessment of the synthesized data compared to the original data was measured by the overall quality score by calling the 'evaluate_quality' function from the 'sdv.evaluation' module, in addition to the KS distribution comparison statistical test. The overall quality score was best achieved at around 65%, and the *p*-values from the KS test were less than 0.001, indicating matching distributions. Table 4 shows the overall quality scores per category for the synthesized data generated by GAN.

Table 4. Summary of the overall quality scores of the synthesized data by GAN

Difficulty Level	Male Quality Score (%)	Female Quality Score (%)
Easy RW – Easy VW	64.25	63.98
Easy RW – Hard VW	62.23	65.71
Hard RW – Easy VW	61.65	65.85
Hard RW – Hard VW	62.73	64.63

5.2. LLM-based Workload Prediction Results

An LLM-based model was applied to the synthetic data generated by GAN to predict the workload. This model used a pre-trained large language model like BERT to process textual data related to the difficulty level variables. The remaining numerical variables were fed into a feedforward neural network at the same time. The gender variable was encoded using '0' for males and '1' for females. The model was developed using the Python programming language in a Google Colab notebook.

After importing the necessary libraries like transformers, Sklearn.model_selection, TensorFlow, Numpy, and pandas, the BERT model was called for custom configuration for fine-tuning according to our specified task. The 'attention_probs_dropout_prob' and 'hidden_dropout_prob' are set to 0.1, which means 10% of attention probabilities in the hidden layers will be dropped during training to prevent overfitting. The activation function used in the hidden layer is 'gelu', as it is commonly used in natural language processing tasks. The 'hidden size' is set to 768, which defines the number of units in each hidden layer, and the 'intermediate_size' is set to 3072, which represents the size of the intermediate feedforward layer. The model can process sequences up to 512 tokens long. It uses 12 attention heads, and 12 transformer layers, allowing for complex representations of the input sequence. The 'vocab_size' used by BERT is set to 30522. These configurations optimize the model to offer a balance between performance and efficiency.

After that, a utility function called 'tokenize_text' is defined. This function tokenizes the text data using BERT's tokenizer, transforming raw text into a structured, numerical format that BERT can use to understand and process text effectively. After uploading our dataset from a CSV file, the tokenized output of the difficulty level variable is stored in a new column called 'text_tokenized.' The numerical variables are normalized using 'StandardScaler', which scales the data to have a mean of zero and a standard deviation of one, preparing it for input into the model.

After that, the dataset is split using the 80/20 split rule, and the input for the BERT model 'bert_model' is prepared using TensorFlow/Keras. An input layer named 'input_ids' is defined, which expects sequences of integers with a fixed length of 32; these integers represent tokenized words from the text. The input is then processed through the BERT model. The numerical inputs are set up for a basic feedforward neural network, also using TensorFlow/Keras. An input layer called 'num_input' is created to receive the numerical data for processing, and a dense layer named 'num_dense' is added with 16 neurons and a ReLU activation function. This layer takes the numerical input and applies the ReLU function, which helps introduce non-linearity into the model.

Instead of concatenating the output of the BERT with the numerical inputs altogether, the BERT output is passed as part of the dense layer input to the neural network 'bert_dense =tf.keras.layers.Dense(16, activation='relu')(bert_output),' and then the model combines the numerical features with the processed BERT output 'combined = tf.keras.layers.concatenate([bert_dense, num_dense])'. Figure 4 shows an illustration of this explanation.

After this combination, a dense layer named 'dense' is created with 32 neurons and a ReLU activation function, and another dense layer named 'output' is defined, which has a single neuron where this layer is used for regression tasks to

output a continuous value based on the input data. Lastly, the model is defined using 'tf.keras.Model', which specifies the entire network's input and output to make predictions.

The model's training was evaluated based on the comparison of two parts, based on the Root Mean Square Error (RMSE), because of the numerical nature of the weighted workload target variable and based on a combined loss function from the BERT and the neural network. For evaluation based on the RMSE value, the model is compiled using the 'model.compile' method, where the Adam optimizer is chosen, and the loss function is set to be the RMSE. The learning rate in compiling the model was set to 0.0001, which allows the model to learn slowly and more precisely than setting a higher learning rate. The model was trained using the 'model.fit' function, which takes the training data, where 'X_text_train' contains the tokenized text inputs, and 'X_num_train' holds the numerical inputs. The target values the model aims to predict are represented by 'y_train'. Additionally, the model evaluates its performance on a separate validation dataset consisting of 'X_text_test' and 'X_num_test' along with their corresponding target values 'y_test'. The training process runs for 20 epochs to balance learning efficiency and computational constraints, such as CPU capacity. More training epochs were experimented with for testing, and the results did not significantly differ. Thus, the number of epochs was set to 20. Different common batch sizes were tested during this process, including 16, 32, and 64, to determine the optimal size for training given the available resources. The RMSE equation for this part is represented by the standard RMSE equation as:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \quad (6)$$

In (6), y_i represents the actual values (training data), and $\hat{y}_i$ is the predicted value (the output of the model). The difference between them is the residual or the error for each data point. By squaring the errors, we ensure that both positive and negative errors contribute positively to the total error, preventing any cancellations. Summing out the squared errors for all the n data points, and then taking the square root returns the error to the original units.

Another modification to the model was made to enhance its prediction accuracy. Instead of keeping the BERT constant by only taking it as an input to the NN, where in this case, its weights were not being updated during training, a training loop of the NN was created by integrating the BERT and fine-tuning it inside the loop for the NN. In each iteration of the loop, the tokenized text input is passed through the pre-trained BERT, which generates the CLS token embeddings. These embeddings are then processed by additional layers of the NN, combined with numerical features and passed through more layers to produce the final output. In this method, the loss of the BERT is updated each time rather than remaining constant in case it is not integrated into the neural network loop. A total of 20 epochs were used to train this model as well. Increasing the number of epochs did not provide much of a difference in the results. The same batch sizes of 16, 32, and 64 were used for comparison. This modified model also uses an Adam optimizer.

Thus, before training, an individual loss function was defined based on RMSE, one for the BERT model called 'loss_bert,' and the other one for the NN called 'loss_nn.' Then, a custom loss function called 'Weighted_rmse_loss' was defined, which combines the two losses. This function calculates the losses for both 'y_true', representing the true values, and 'y_pred', representing the predicted values. Then, it creates a weighted sum using the parameter α 'alpha'. The loss for BERT is multiplied by 'alpha', while the loss of the NN is multiplied by '(1-alpha)'. This method allows control to give how much importance each model has in the final loss. So, the 'alpha' here, by default, is set to 0.5.

For the simple neural network evaluation, the best performance was achieved with a batch size of 16, yielding an RMSE value of 6.82. Table 6 shows the results of each batch size. When comparing the performance of the proposed hybrid model and the simple neural network, it is noted that the proposed hybrid model achieved better results in predicting the workload, especially when integrating the BERT into the training of the FNN, and was kept constant. Therefore, the results of this study supported our hypothesis.

Table 5. Comparison of RMSE values of different workload prediction methods

Workload Prediction Method	RMSE		
	batch size = 16	batch size = 32	batch size = 64
FNN	6.82	6.88	6.93
BERT no Integration	2.66	2.68	2.67
BERT with Integration	1.03	0.95	0.99
GPT-2 no Integration	2.70	2.71	2.68
GPT-2 with Integration	0.99	0.98	1.01

As shown in Table 5, in comparison, the methods with LLM integration give better results than those without LLM integration. Between the two methods integrated with different LLMs, the prediction accuracy of BERT is comparable to that of GPT-2.

Figure 5 and Figure 6 show the predicted workload vs. the actual workload for a simple FNN and an integrated LLM model, respectively. The results presented by Figures 5 and 6 are consistent with the results presented in Table 5.

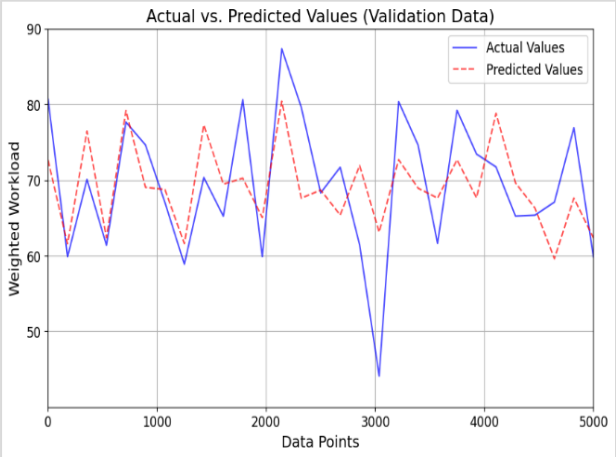


Figure 5. Predicted workload vs. actual workload for the simple FNN model.

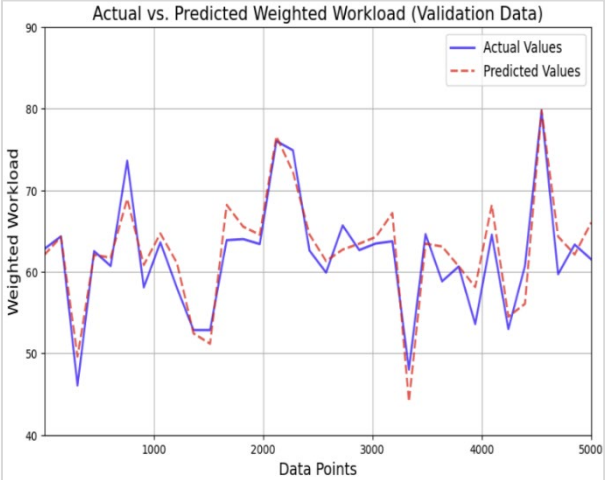


Figure 6. Predicted workload vs. actual workload for the integrated BERT model.

6. CONCLUSION

This paper presents a novel framework for predicting user workload in mixed reality (MR) multitasking environments by integrating large language models (LLMs) with generative adversarial networks (GANs) and traditional workload metrics. By leveraging both structured numerical data and unstructured textual inputs, our approach enhances the precision and contextual relevance of workload predictions, key for improving human-machine interaction in immersive environments. Unlike conventional methods that rely solely on subjective post-task evaluations such as NASA-TLX, our model utilizes BERT and GPT-based embeddings to interpret task complexity and user experience in real-time or near real-time, enabling more proactive system adaptations.

To overcome data scarcity, a common barrier in experimental MR research, we implemented GAN-based data augmentation, which effectively synthesized diverse and realistic training samples. The integration of these synthetic datasets enabled more robust model generalization and predictive accuracy. Our results clearly demonstrate that LLM integration significantly outperforms baseline feedforward neural network models, reducing root mean square error (RMSE) from 6.82 to as low as 0.95 when BERT is fine-tuned and jointly trained with the FNN.

The proposed methodology holds promise for a wide range of applications in fields such as training simulation, healthcare, aerospace, and manufacturing, where understanding cognitive workload is essential to system design and operational safety. Additionally, the framework can serve as a foundation for further research in adaptive MR systems that dynamically adjust task difficulty or interface elements based on predicted cognitive load.

Future work should explore multimodal data fusion by incorporating physiological signals (e.g., EEG, eye tracking) alongside textual and numerical inputs, and investigate real-time workload prediction in live MR environments. Moreover, transfer learning and federated learning approaches could improve scalability across different user populations and hardware platforms. This research lays the groundwork for intelligent, user-aware MR systems that are both scalable and responsive to individual cognitive demands.

ACKNOWLEDGEMENT

The authors acknowledge the support of the first author's Saudi Arabian Cultural Mission (SACM) fellowship and the College of Engineering at the University of Illinois Chicago. The authors also acknowledge the use of ChatGPT, an AI language model developed by OpenAI, for assistance in refining the writing and improving the clarity of the manuscript.

REFERENCES

Abbas, S., & Jeong, H. (2024a). Task difficulty impact on multitasking in mixed reality environments. *Computers & Education: X Reality, 4, 100065.* https://doi.org/10.1016/j.cexr.2024.100065

Abbas, S., & Jeong, H. (2024b). Unveiling gender differences: A mixed reality multitasking exploration. *Frontiers in Virtual Reality, 4.* https://doi.org/10.3389/frvir.2023.1308133

Abrash, M. (2021). Creating the future: Augmented reality, the next human-machine interface. In *2021 IEEE International Electron Devices Meeting (IEDM)* (pp. 1–11). https://doi.org/10.1109/iedm19574.2021.9720526

Bousdekis, A., Mentzas, G., Apostolou, D., & Wellsandt, S. (2022). Evaluation of AI-based digital assistants in smart manufacturing. In *Proceedings of the IFIP Advances in Information and Communication Technology* (pp. 503–510). Springer. https://doi.org/10.1007/978-3-031-16411-8_58

Card, S. K., Moran, T. P., & Newell, A. (2018). Applying psychology to design. In *The psychology of human-computer interaction* (pp. 403–424). https://doi.org/10.1201/9780203736166-12

Carroll, L. (2017). A comprehensive definition of technology from an ethological perspective. *Social Sciences, 6*(4), 126. https://doi.org/10.3390/socsci6040126

Chen, Y.-N., Mitra, S., & Schlaghecken, F. (2008). Sub-processes of working memory in the N-back task: An investigation using erps. *Clinical Neurophysiology, 119*(7), 1546–1559. https://doi.org/10.1016/j.clinph.2008.03.003

Colabianchi, S., Costantino, F., & Sabetta, N. (2024). Assessment of a large language model-based digital intelligent assistant in assembly manufacturing. *Computers in Industry, 162*, 104129. https://doi.org/10.1016/j.compind.2024.104129

Criollo-C, S., Enrique Cerezo Uzcátegui, J., Guerrero-Arias, A., Dwinggo Samala, A., Rawas, S., & Luján-Mora, S. (2024). Analysis of the mental workload associated with the use of virtual reality technology as support in the higher educational model. *IEEE Access, 12*, 114370–114381. https://doi.org/10.1109/access.2024.3445301

Dang, Y. M., Zhang, Y., Brown, S. A., & Chen, H. (2020). Examining the impacts of mental workload and task-technology fit on user acceptance of the Social Media Search System. *Information Systems Frontiers, 22*(3), 697–718. https://doi.org/10.1007/s10796-018-9879-y

Dzubak, C. M. (2008). Multitasking: The good, the bad, and the unknown. *Journal of the Association for Tutoring Professionals, 1*(2), 1–12.

Fick, T., Meulstee, J. W., Köllen, M. H., Van Doormaal, J. A., Van Doormaal, T. P., & Hoving, E. W. (2023). Comparing the influence of mixed reality, a 3D viewer, and MRI on the spatial understanding of brain tumours. *Frontiers in Virtual Reality, 4*. https://doi.org/10.3389/frvir.2023.1214520

Gao, Y., Yue, L., Sun, J., Shan, X., Liu, Y., & Wu, X. (2024). WORKLOADGPT: A large language model approach to real-time detection of pilot workload. *Applied Sciences, 14*(18), 8274. https://doi.org/10.3390/app14188274

Hadi, M. U., Tashi, Q. A., Qureshi, R., Shah, A., Muneer, A., Irfan, M., Zafar, A., Shaikh, M. B., Akhtar, N., Wu, J., & Mirjalili, S. (2023). A survey on large language models: Applications, challenges, limitations, and practical usage. *TechRxiv*. https://doi.org/10.36227/techrxiv.23589741.v1

Hart, S. G., & Staveland, L. E. (1988). Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In P. A. Hancock & N. Meshkati (Eds.), *Advances in Psychology* (pp. 139–183). North-Holland. https://doi.org/10.1016/s0166-4115(08)62386-9

Jordon, J., Szpruch, L., Houssiau, F., Bottarelli, M., Cherubin, G., Maple, C., ... & Weller, A. (2022). Synthetic Data--what, why and how?. *arXiv preprint arXiv:2205.03257*.

Kolmogorov, A. N., & Smirnov, N. V. (1933). On the empirical determination of a distribution law. *Sankhya, 4*(2), 116–119.

Kudesia, R. S., Pandey, A., & Reina, C. S. (2022). Doing more with less: Interactive effects of cognitive resources and mindfulness training in coping with mental fatigue from multitasking. *Journal of Management, 48*(2), 410–439. https://doi.org/10.1177/0149206320964570

Li, W., Li, R., Xie, X., & Chang, Y. (2022). Evaluating mental workload during multitasking in simulated flight. *Brain and Behavior, 12*(4). https://doi.org/10.1002/brb3.2489

Liu, Y., He, H., Han, T., Zhang, X., Liu, M., Tian, J., Zhang, Y., Wang, J., Gao, X., Zhong, T., Pan, Y., Xu, S., Wu, Z., Liu, Z., Zhang, X., Zhang, S., Hu, X., Zhang, T., Qiang, N., & Ge, B. (2024). Understanding LLMs: A comprehensive overview from training to inference. *SSRN*. https://doi.org/10.2139/ssrn.4706201

Matthews, G., Reinerman-Jones, L., Wohleber, R., Lin, J., Mercado, J., & Abich, J. (2015). Workload is multidimensional, not unitary: What now? In D. Schmorrow & C. Fidopiastis (Eds.), *Proceedings of the 7th International Conference on Augmented Cognition* (pp. 44–55). Springer. https://doi.org/10.1007/978-3-319-20816-9_5

Nam, D., Macvean, A., Hellendoorn, V., Vasilescu, B., & Myers, B. (2024). Using an LLM to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering* (pp. 1–13). IEEE. https://doi.org/10.1145/3597503.3639187

North, S., North, M., Garofalo, D., & Parajapati, D. (2021). The effects of mixed reality immersion on users' performance and perception of multitasking while performing concurrent real-world tasks. *Journal of Computer Science in Colleges, 36*(8), 73–88.

Rashid, H., Tanveer, M. A., & Aqeel Khan, H. (2019). Skin lesion classification using GAN-based data augmentation. In *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)* (pp. 916–919). IEEE. https://doi.org/10.1109/embc.2019.8857905

Rawlinson, T. G., Lu, S., & Coleman, P. (2012). Individual differences in working memory capacity and presence in virtual environments. *Advances in Brain Inspired Cognitive Systems*, 22–30. https://doi.org/10.1007/978-3-642-31561-9_3

Rokhsaritalemi, S., Sadeghi-Niaraki, A., & Choi, S. M. (2020). A review on mixed reality: Current trends,

challenges, and prospects. *Applied Sciences, 10*(2), 636. https://doi.org/10.3390/app10020636

Ru, Y., Wei, Z., An, G., & Chen, H. (2024). Combining data augmentation and deep learning for improved epilepsy detection. *Frontiers in Neurology, 15*, 1378076. https://doi.org/10.3389/fneur.2024.1378076

Salam, A., Hibaoui, A. E., & Saif, A. (2021). A comparison of activation functions in multilayer neural network for predicting the production and consumption of electricity power. *International Journal of Electrical and Computer Engineering (IJECE), 11*(1), 163–170. https://doi.org/10.11591/ijece.v11i1.pp163-170

Sonawani, S., Weigend, F., & Amor, H. B. (2024). SiSCo: Signal synthesis for effective human-robot communication via large language models. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 7107–7114). IEEE.

Speicher, M., Hall, B. D., & Nebeling, M. (2019). What is mixed reality? In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (pp. 1–15). ACM. https://doi.org/10.1145/3290605.3300767

Spink, A., Cole, C., & Waller, M. (2008). Multitasking behavior. *Annual Review of Information Science and Technology, 42*(1), 93–118. https://doi.org/10.1002/aris.2008.1440420110

Strayer, D. L., Castro, S. C., & McDonnell, A. S. (2022). The multitasking motorist. In *Handbook of Human Multitasking* (pp. 399–430). Springer. https://doi.org/10.1007/978-3-031-04760-2_10

Widiastuti, R., Nurhayati, E., Wardani, D. P., & Sutanta, E. (2020). Workload measurement of batik workers at UKM Batik Jumputan Yogyakarta using RULA and NASA-TLX. *Journal of Physics: Conference Series, 1456*(1), 012032. https://doi.org/10.1088/1742-6596/1456/1/012032

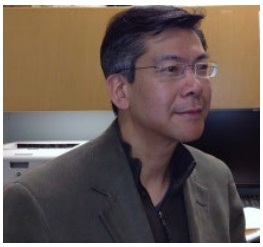
Xie, B., & Salvendy, G. (2000). Review and reappraisal of modelling and predicting mental workload in single- and multi-task environments. *Work & Stress, 14*(1), 74–99. https://doi.org/10.1080/026783700417249

Zaheer, R., & Shaziya, H. (2019). A study of the optimization algorithms in deep learning. In *2019 Third International Conference on Inventive Systems and Control (ICISC)* (pp. 1–6). IEEE. https://doi.org/10.1109/icisc44355.2019.9036442

BIOGRAPHIES

Safanah Abbas is currently completing a PhD in Industrial Engineering at the University of Illinois Chicago, where she is in the final stages of her dissertation. Her research focuses on human performance modeling and system optimization, with an emphasis on improving the efficiency, safety, and usability of complex systems. Safanah is particularly interested in understanding how humans interact with technology and how these interactions can be optimized through data-driven design and engineering principles. In addition to her academic work, she is passionate about applying research insights to solve real-world problems. She aims to contribute to the development of smarter, more adaptive systems that support human decision-making and performance.

Heejin Jeong, **PhD**, is an Assistant Professor of Human Systems Engineering in the Polytechnic School at the Ira A. Fulton Schools of Engineering, Arizona State University. He also serves as Graduate Faculty in Biomedical Engineering in the School of Biological and Health Systems Engineering. He received his PhD in Industrial and Operations Engineering from the University of Michigan in 2018. Dr. Jeong leads the Human-in-Mind Engineering Research (HiMER) Lab, with support from organizations such as the National Science Foundation (NSF) and the National Institute for Occupational Safety and Health (NIOSH). He has received several prestigious honors, including the NSF CAREER Award, the 2023 Applied Ergonomics Conference/Texas A&M Ergo Center Young Investigator Award, and the ASU Faculty Women's Association Outstanding Faculty Mentor – Early Career Award. He is also the co-editor of the book Human-Centered Metaverse: Concepts, Methods, and Applications.



David He received a Ph.D. in Industrial Engineering from The University of Iowa. Dr. He is a Professor and Director of the Industrial AI and PHM Laboratory in the Department of Mechanical and Industrial Engineering at The University of Illinois Chicago. Dr. He is a Fellow of the Prognostics and Health Management (PHM) Society and serves as Associate Editor for *Journal of Intelligent Manufacturing*. Dr. He's research areas include

PHM, Industrial AI, smart manufacturing systems modeling and analysis, and quality and reliability engineering.