ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

Wenhao Yang
Department of Industrial and Systems Engineering,
Rochester Institute of Technology,
Rochester, NY 14623
e-mail: wy7711@rit.edu

Xiwen Dengxiong
Department of Computing and Information Sciences,
Rochester Institute of Technology,
Rochester, NY 14623
e-mail: sd6384@rit.edu

Xueting Wang
Department of Industrial and Systems Engineering,
Rochester Institute of Technology,
Rochester, NY 14623
e-mail: xw4860@rit.edu

Yidan Hu
Department of Computing Security,
Rochester Institute of Technology,
Rochester, NY 14623
e-mail: yidan.hu@rit.edu

Yunbo Zhang[1]
Department of Industrial and Systems Engineering,
Rochester Institute of Technology,
Rochester, NY 14623;
School of Information,
Rochester Institute of Technology,
Rochester, NY 14623
e-mail: ywzeie@rit.edu

"I Can See Your Password": A Case Study About Cybersecurity Risks in Mid-Air Interactions of Mixed Reality-Based Smart Manufacturing Applications

This paper aims to present a potential cybersecurity risk existing in mixed reality (MR)-based smart manufacturing applications that decipher digital passwords through a single RGB camera to capture the user's mid-air gestures. We first created a test bed, which is an MR-based smart factory management system consisting of mid-air gesture-based user interfaces (UIs) on a video see-through MR head-mounted display. To interact with UIs and input information, the user's hand movements and gestures are tracked by the MR system. We setup the experiment to be the estimation of the password input by users through mid-air hand gestures on a virtual numeric keypad. To achieve this goal, we developed a lightweight machine learning-based hand position tracking and gesture recognition method. This method takes either video streaming or recorded video clips (taken by a single RGB camera in front of the user) as input, where the videos record the users' hand movements and gestures but not the virtual UIs. With the assumption of the known size, position, and layout of the keypad, the machine learning method estimates the password through hand gesture recognition and finger position detection. The evaluation result indicates the effectiveness of the proposed method, with a high accuracy of 97.03%, 94.06%, and 83.83% for 2-digit, 4-digit, and 6-digit passwords, respectively, using real-time video streaming as input with known length condition. Under the unknown length condition, the proposed method reaches 85.50%, 76.15%, and 77.89% accuracy for 2-digit, 4-digit, and 6-digit passwords, respectively. [DOI: 10.1115/1.4062658]

Keywords: cyber-physical security for factories, human computer interfaces/interactions, machine learning for engineering applications, virtual and augmented reality environments

1 Introduction

The industrial world is right now undergoing a transformation through the fourth revolution, also known as Industry 4.0 or smart manufacturing. Industry 4.0 advances manufacturing towards a highly flexible production model of customized and digital products or services, with real-time interactions between people, products, and devices during the production process [1]. In the implementation of Industry 4.0, cyber-physical systems (CPS) are incorporated with the advanced manufacturing systems to increase autonomous adaptability, autonomy, and flexibility [2]. In spite of the benefit of Industry 4.0, there are also challenges arising from this rapid transition [3]. One of the biggest problems is that the increased complexity of manufacturing systems makes it hard for people to interact with them because they do not have the appropriate interfaces. To address this problem, the extended reality (XR) techniques, consisting of virtual reality (VR), augmented reality (AR), and mixed reality (MR), are considered the key to bridging human operators and the manufacturing systems [4]. While VR techniques display the virtual objects in a fully immersive environment, AR and MR techniques overlay context-aware virtual information on the display of real objects. XR's new affordances in visualization and interaction make it well-suited to seamlessly connect humans and complex hardware and software systems, such as the internet of things (IoT), autonomous robots, advanced simulations, digital twins, artificial intelligence, etc.

Recently, XR techniques have been extensively investigated in the context of manufacturing applications, such as design [5], training [6], robot programming [7,8], maintenance [9], and assembly [10]. XR is expected to be widely deployed in manufacturing [11] within the next decade; nevertheless, concern about the cybersecurity risk of XR has been raised, which may prevent its adoption

[1]Corresponding author.

Manuscript received February 21, 2023; final manuscript received May 10, 2023; published online October 10, 2023. Assoc. Editor: William Bernstein.

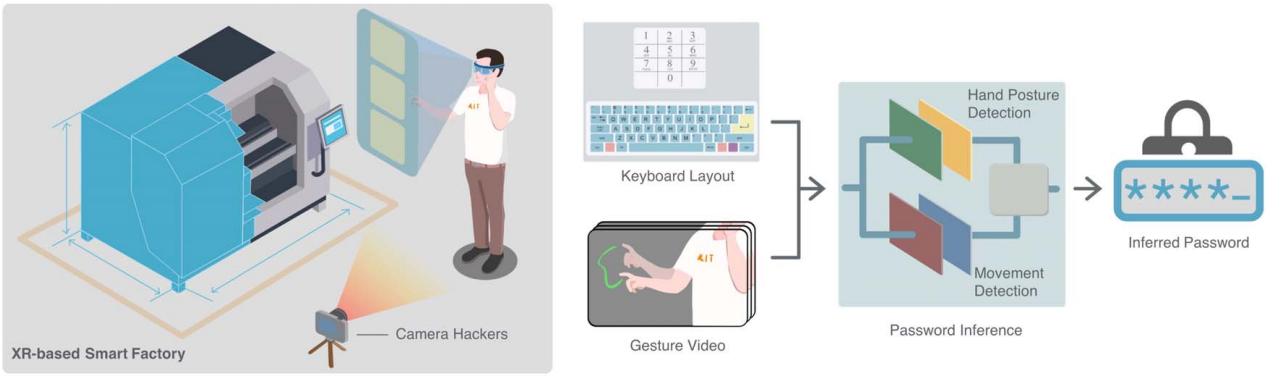

Fig. 1 A cyberattack case based on mid-air interactions in MR-based smart factory environment

in manufacturing applications. In fact, recent articles pointed out the potential cybersecurity risks in XR [12,13], including ransomware, malware, stealing network credentials, man-in-the-middle-attacks, social engineering, etc. Thus, it is critical to identify potential cybersecurity risks associated with XR-based systems in manufacturing applications and to educate both the developer and user of these systems on how to eliminate or avoid risks.

Like other connected devices in Industry 4.0, XR devices are vulnerable to similar attacks as other IoTs [14], such as these attacks against sensors [15], networks [16], middleware [17], and the software [18]. Apart from these threats, we pay attention to the unique challenge that XR devices present: the potential privacy/security issue in mid-air interactions with the intangible XR interfaces. There are several reasons for us to decide to investigate this risk. First of all, mid-air interactions may contain information that can be potentially used for inferring privacy/security-sensitive information. Compared with the conventional hand-held controller, the mid-air interaction has become a more trendy interaction metaphor in mainstream XR devices (e.g., Microsoft Hololens MR headset, Lenovo ThinkReality AR headset, Magic leap AR headset, and Meta Oculus VR headset), as it frees the user's hands and provides an intuitive way to interact. Mid-air interaction is especially preferable in manufacturing applications, where people constantly have to touch machines or tools with their hands. XR systems in industrial applications usually require the user to input privacy-sensitive information (e.g., password, production information, etc.) through users' interactions with the intangible virtual keyboard. As a result, if the user's mid-air interactions are recorded and recognized, it is possible to retain the user's privacy/security-sensitive information. Second, the privacy/security issue in mid-air interactions has not attracted enough attention from both the user and the developer. The mid-air interaction has no tangible interfaces, and the risk of cyberattacks would be easily ignored by the user. On the other hand, the developer usually relies on the XR device manufacturers' software development kits (SDKs) to develop applications, which may cause their interface layouts to be similar. The layout information can be utilized by attackers to infer privacy- and security-sensitive information, whereas both XR device manufacturers and developers have no awareness of it. Last but not least, there are few papers talking about inferring privacy/security-sensitive information through the user's interactions using XR devices. The existing works focus on either physical touchscreen-based inferences [19] or the head motion-based [20]. Some recent papers discussed the password estimation through the user's interactions with the XR interfaces, either based on a human's observation [21] or an algorithm requiring the user's head location and orientation information from the VR headset [22]. Our proposed method, on the other hand, takes only 2D video streams and automatically estimates the input information through the user's interactions.

To deepen our understanding of the cybersecurity risk of mid-air interactions in XR and to attract enough attention from the user, the developer, the researcher, and the practitioner, we conduct a case study to investigate the following research question: Is it possible to infer the password information using a camera to record the user's mid-air interactions in the physical world while the password input is completed with the intangible interfaces in virtual space? In order to answer this question, we created a prototyping XR-based smart factory management system and developed a password input interface based on a virtual numerical pad. We then proposed and developed a machine learning framework to identify hand gestures and motions based on single-view images input extracted from a video stream, and then estimate the password. Figure 1 illustrates the pipeline of the proposed work. We perform two types of attacks using this test bed and password estimation algorithm: one with known password length and one without. The experimental results indicate the effectiveness of the proposed password inference method, and therefore, prove the necessity and urgency of paying attention to the cybersecurity risks in XR systems' mid-air interactions. Please refer to an illustration video about our proposed method through the YouTube link.[2]

The rest of the paper is organized as follows: Sec. 2 presents related works of XR's applications in Industry 4.0 and associated cybersecurity risks. The design and development of a prototyping MR system are described in Sec. 3, and a password inference method is proposed and presented in Sec. 4. Section 5 reports the experiment's details and results. In the last section, the conclusion is summarized and the future work is pointed out.

2 Related Works

2.1 Extended Reality Techniques and Applications in Industry 4.0. XR devices, based on their fundamental working principles, can be generally categorized into three classes [10,23]: on the user's head (head-mounted), in the user's hand (hand-held), or installed in the environment (spatial). Among these devices, the head-mounted display (HMD) becomes the primary one to support Industry 4.0 applications. Different interaction metaphors are proposed around the HMDs, such as voice interaction [24], gaze interaction [25], eye tracking [26], mid-air gesture [27], physical interactive widgets [28], and haptic interaction based on gloves [29]. Bailenson [30] and JofréPasinetti et al. [31] mentioned that non-verbal communication with computers such as eye movement, gaze, and gesture-operated in XR would be of value in the future. In fact, the mid-air gesture, which includes all techniques by means of tracking the bare hands or finger gestures as inputs, is an increasingly popular interaction method to manipulate 3D objects in current HMD-based XR applications [32].

XR techniques have shown their potential in Industry 4.0 applications as they enable the integration of humans with other complex

[2]https://youtu.be/gTZPE8S1-0M

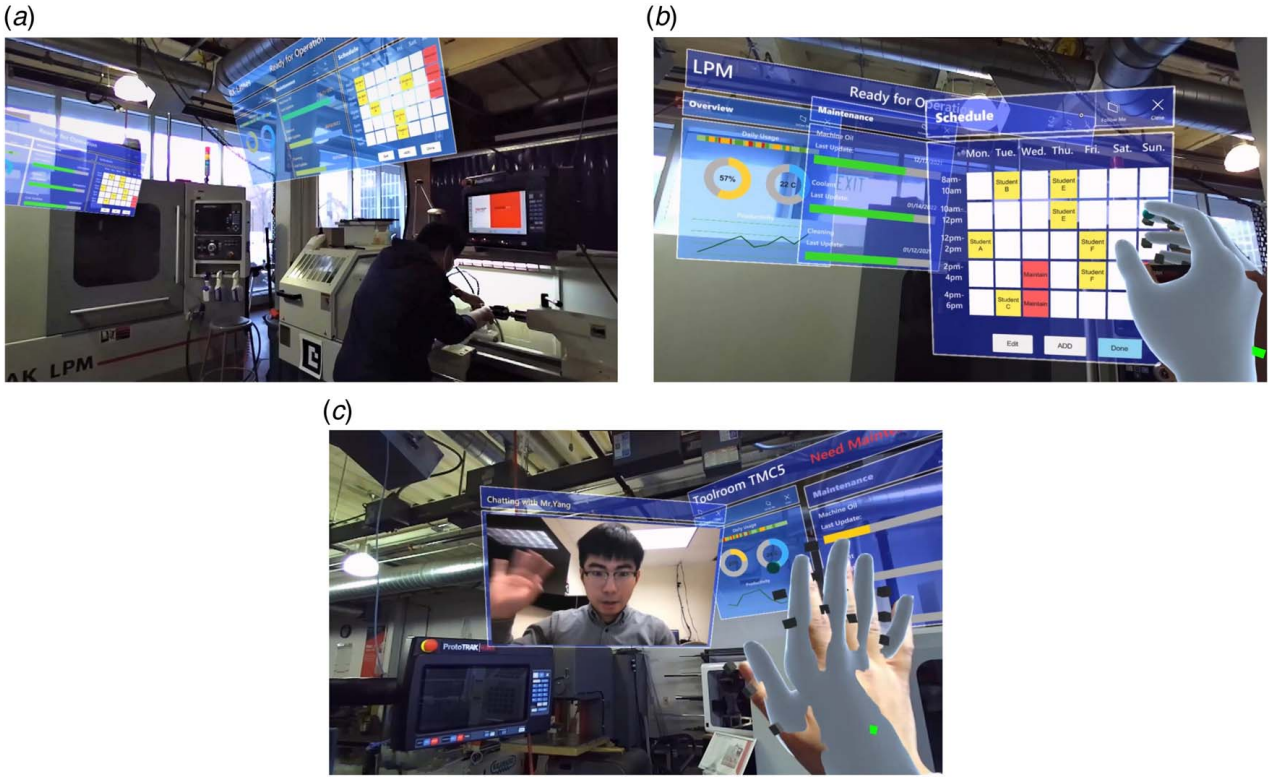

Fig. 2 The prototyping MR system for smart factory management is displayed in the first-person view of the operator wearing the MR headset in the machine shop: (*a*) Interface layout, (*b*) machine maintenance interface, and (*c*) live support interface

systems [33]. The specific applications include design [34], training [35], maintenance [36], assembly [10], and human–robot interaction [37]. Alongside the exploration of XR techniques in industrial applications, different interaction metaphors have also been studied. Gattullo et al. [38] summarized the list of visual assets in industrial AR applications, namely, text, signs, photographs, videos, drawings, technical drawings, product models, and auxiliary models. Yew et al. [39] present bare-hand interaction applications for smart 3D objects and smart machining objects via a griddable distributed manufacturing system. Maharjan et al. [40] also applied hand gestures in an AR-enabled human–infrastructure interface for inspection and monitoring structures to achieve hands-free operation. Wang et al. [41] evaluated the improvement of a combination of 3D gestures and computer aided design models in remote collaboration on an assembly task with respect to the performance time and user experience.

2.2 Cybersecurity Risks in Extended Reality Systems. Although XR techniques have shown tremendous potential benefits, they could also introduce new cybersecurity risks. Casey et al. [42] have shown that immersion attacks could incur physical harm and disrupt the user experience in VR systems. Examples of immersion attacks include chaperone attacks to tamper with the virtual environment boundaries, overlay attacks for overlaying unwanted content on VR users' view, joystick attacks to mislead the physical movement of VR users, and disorientation attacks to cause dizziness and confusion in VR users (as known as cybersickness in Ref. [43]). The XR platform is also vulnerable to various specific side-channel attacks [44–46] and other general security threats, e.g., DoS attacks and man-in-the-middle attacks [47]. Apart from the security threat, data privacy is another serious concern in XR systems, such as the potential leakage of users' physical locations in AR systems [48], leakage of personal information [49], and de-anonymization of users via leveraging their unique movement patterns [50]. The detailed security, privacy challenges, and ethical issues in XR systems have been well summarized in Refs. [47,51].

Keystroke inference attack is one of the most critical potential security threats in XR systems. In keystroke inference attacks, when a user presses a key on the keyboard, the user's hand coverage and finger motions could be captured and featured by motion sensors, videos, or the fluctuation of the wireless signals, which could then be used to infer users' inputs. The existing keystroke inference attack could be mainly divided into three categories: WiFi signal-based, video-based, and sensor-based. A WiFi-based keystroke inference framework could accurately infer users' passwords via analyzing channel state information of WiFi signals [45,52]. Wang et al. [53] proposed a GazeRevealer to infer the users' inputted passwords based on the smartphone's front camera. Sensor-based keystroke inferences attacks exploit different sensor data, e.g., electromyography sensor data and motion sensors data, to infer the inputted password [19,46]. Recently, there have been some papers focusing on estimating passwords through the user's mid-air interactions. Compared with Kreider [21], which is mainly based on human observation input, our work proposes an efficient hand-tracking algorithm and a password estimation algorithm that only needs 2D video frames for inferring the password by recognizing the user's mid-air interactions. The keystroke inputs in HoloLogger [22] requires a data stream of head location and orientation information from the VR headset, while our work does not need to get information from the headset. However, most existing keystroke inference attacks were originally designed for physical keystrokes and were well-evaluated in the real world. Their performances in virtual environments, especially for the XR applications in Industry 4.0, are still under-explored.

3 The Development of the Mixed Reality System

We assume the attack scenario where the user is entering a password with mid-air gestures in MR to log in to a smart factory management system. We consider that the smart factory management system contains sensitive information, including production processes, machine conditions, and schedules. In this section, we

Fig. 3 Augmented reality head-mounted display device implemented with Oculus Quest first generation and ZED mini in this work

present how we design and implement the interfaces and mid-air interactions of the MR system.

3.1 Hardware and Software. We setup the video see-through MR system following Yang et al.'s work [8]. An Oculus Quest headset and a ZED mini-stereo camera are integrated as shown in Fig. 3. ZED mini is a USB 3.0 stereo camera with 720p resolution, 30 Hz frame rate, and a wide field of view (Vertical 54 deg and horizontal 85 deg). The latency of the MR system is expected to be under 100 ms. Both the HMD and the camera are connected to a laptop via two cables.

The entire system is developed based on the unity platform. There are several SDKs utilized to develop the system, including Oculus integration [54] for supporting Oculus HMD development, mixed reality toolkit [55] for supporting user interfaces and interactions development, and OpenCV for unity plugin [56] for detecting ArUco markers in the physical world and localizing the MR interface.

3.2 Interfaces and Interactions. This prototyping system is designed for the management and maintenance of a smart factory consisting of manufacturing systems such as computerized numerical control (CNC) machines, metal 3D printers, etc. The user who wears the MR device is able to interact with manufacturing systems using mid-air interactions to maintain the machines, manage the schedule, and receive live support through the visualization and communication modules. Please refer to Fig. 2 for detailed information about the interface or access the demonstration video through the link.[3] Using such an MR interface, essential information such as production plans, machine status, and schedules will be accessed. Therefore, it is necessary to setup an authorization or identity verification step when the user starts to use the interface. We developed a login interface which is a numeric keypad for the user to input passwords, and the cybersecurity attack is set to be inferring the password through the user's mid-air interactions. The details about the interfaces and interactions are presented as follows.

MR Interfaces. The MR interfaces access each machine's information through a background cloud server, and the information visualized in the MR environment includes the daily status of the machine and schedules (see Figs. 2(*a*) and 2(*b*)). Moreover, a live video support interface is available to allow the user to chat with the support engineer of the machine's manufacturer from a distance (see Fig. 2(*c*)). The support engineer can share the user's first-person view to see the situation on-site and can also annotate the shared view.

Login Interface. We designed a virtual interactive number pad for the login interface, which is illustrated in Fig. 4. Virtual keyboards are the current standard in most XR headsets and applications. The layout of the number pad is similar to the current physical and touchscreen numerical keyboards, so the users can start using the method without any friction. The bottom left is the "Delete" button, and the bottom right is the "Enter" button (see Fig. 4(*a*)). This virtual numerical pad is interactive with hand tracking. Users use the tap gesture to physically press the virtual buttons in mid-air so that the input is recognized. Note that, from a third-person view, this virtual numerical pad is invisible.

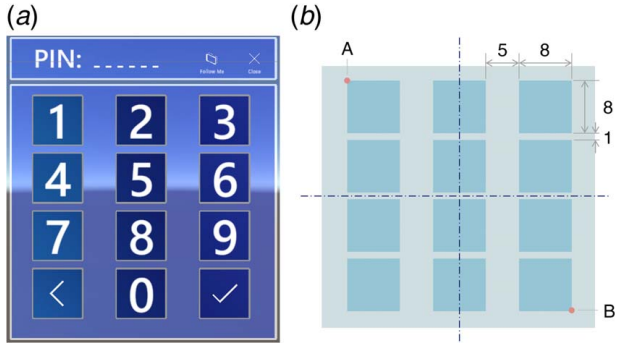

Fig. 4 The login interface is a virtual numerical keypad displayed in the MR system. (*a*) The login keypad has ten numerical keys, a "Delete" button (left bottom corner), and an "Enter" button (right bottom corner). (*b*) The layout design of the main 12 buttons with two calibration points is illustrated.

User Interactions. Instead of directly displaying information, all MR interfaces are interactive with hand tracking. The hand-tacking is proven to increase immersion and presence for the virtual elements [57]. The interface panels and subpanels are moveable with a mid-air gesture control and can be put anywhere in the space. In XR interactions, there are two methods enabled by hand tracking: near-interaction and cursor-pointer. We adopted both methods to support the user in our prototyping system. Near interaction is similar to the human's interaction with physical objects, like grabbing, pressing, and tapping, while a laser cursor-pointer is driven by the hand's pose and behaves like a standard controller cursor to highlight, select, click, or any other customized trigger. Both of these two methods make use of mid-air hand gestures such as point, pinch, unpinch, scroll, and palm pinch. For example, the pinch is used for selecting and a combination of pick and drag to move the panels. For entering characters, near interaction is used because it is faster than typing with a cursor or physical controller.

4 The Password Inference Method

This section presents the hand-tracking algorithm and the password estimation algorithms for inferring the password by recognizing the user's mid-air interactions. We first describe the hypothetical attack scenarios and key assumptions. Then, we present the proposed inference method, consisting of hand tracking and password estimation algorithms. Hand tracking is used for detecting hand key points and hand gestures. The password estimation procedure uses the hand gesture detection results and the hand tracking results to estimate the password entered in the virtual panel.

4.1 Hypothetical Attack Scenario and Assumptions. An experiment of potential password inference attacks is developed based on the login interface shown in Fig. 4 with ten numerical keys, a "Delete" button, and an "Enter" button. There are several key assumptions as follows.

ASSUMPTION 1. *The attacker can neither physically touch the HMD nor get access to its data.*

ASSUMPTION 2. *The attacker knows the layout information and the position of the numerical keypad (as illustrated in Fig. 4(b)).*

ASSUMPTION 3. *The input data for the attack are a 2D video clip containing the user's mid-air interactions, taken from the first-person view by a hidden camera in front of the user.*

[3] https://youtu.be/6m-VTKPbyMc

Fig. 5 The setup of password inference attacks in the MR system is illustrated

Our research focus determines Assumption 1 that the mid-air interactions information is the only information for the password inference. The reason for adopting Assumption 2 is that the virtual interactive interfaces are usually developed with the support of XR device manufacturers' SDKs and are therefore universal and well known. Assumption 3 again comes from the setup of the proposed cybersecurity attack, which is the password inference based on mid-air interactions. We also want to minimize the required input data, therefore, only a first-person view video clip is assumed to be sufficient for the attack. Because the password length could vary (e.g., 4 or 6 digits), we devise two attack scenarios: *Attack 1* (known password length) and *Attack 2* (unknown password length).

4.2 Hand Gesture Tracking. Unlike typing on the physical keyboard, the password input mechanism in the prototyping system is different from traditional computers and phones. Users are required to click the virtual panel displayed in the headset through a particular hand gesture. The login interface is shown in Fig. 5. Normally, the hand gesture is combined with a series of hand poses in the time period, and the virtual panel is only visible to the headset wearer. However, the hand gesture in the password input video contains spatial–temporal information about the password and the virtual panel. Spatial–temporal information refers to the information having connections between time and spatial movement. In this section, we propose to use the hand-tracking algorithm to reveal the location of the hand and the gesture classification method to detect the action of the user.

Palm Detection Model and Hand Kinematic Skeleton Model. Due to the computing device constraint in the password estimation case, we use a lightweight hand-tracking algorithm to track the palm movement and the skeleton key points of the hand. There are two steps for tracking the hand. First, the tracking algorithm will detect the palm. We use a single-shot detector to detect a palm, and palms are modeled using square bounding boxes. To reduce the influence of a bigger scene context, encoder–decoder features are used to train the detector to enhance the attention on the palm. The second step is to reconstruct the skeleton's key points of the hand. Based on the palm location, we use a hand landmark model to learn a consistent hand pose representation. Then we select 21 skeleton key points of the hand and use the regression method to get the coordinates of the key points (Fig. 6(*a*)). Details of the palm detection model and the skeleton key points reconstruction model are based on Mediapipe [58].

Hand Pose Estimation. The hand pose estimation is designed for capturing hand pose and enabling additional functions after recording the hand skeleton key points from the hand-tracking algorithm [59]. In this section, we design four common hand poses, including close, open, pointer, and pick, to simulate the user's hand pose in the MR system (Fig. 6(*b*)). The pick hand pose is used to adjust the location and size of the element's virtual button area in the captured scene. The pointer hand pose is designed for collecting spatial–temporal data from the hand poses. Open and close hand poses are basic poses that indicate no additional functions are activated. Then we use a neural network with two hidden layers to estimate the hand poses. The input of the neural network is 21 hand skeletons' key point coordinates. The output of the neural network is hand pose labels.

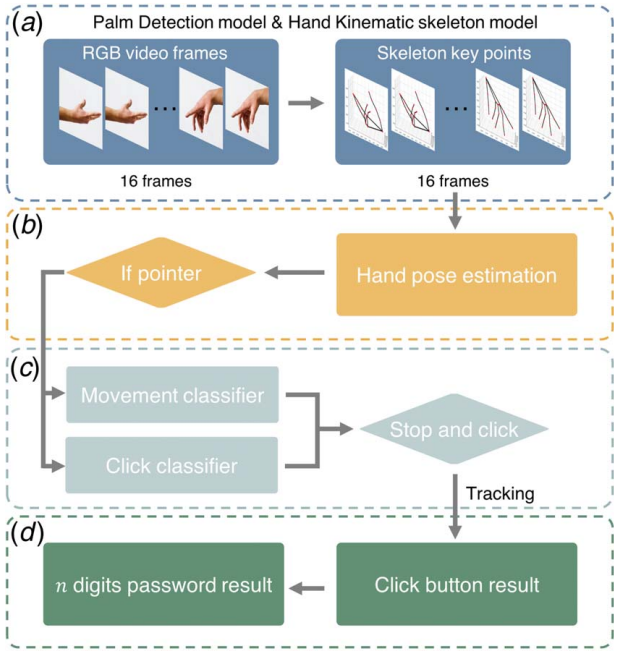

Fig. 6 The password inference workflow. The palm detection model and the hand kinematic skeleton model are used to transform 16 RGB video frames into a series of skeleton key points (*a*). Then the hand pose estimation (*b*) classifies the hand into four hand pose categories. If the hand pose is a pointer, movement and click classifier results (*c*) will determine the tracking result and output the corresponding passwords (*d*).

Hand Movement Classification. After collecting spatial–temporal data on the hand poses, we need to analyze the hand gesture sequence to get the movement of the hand and identify the action. In this case study, we design two classifiers to recognize the meaning of hand movements (Fig. 6(*c*)). The first neural network, the movement classifier, is to separate the directional movement from the stop. The inputs of the movement classifier are the 48 skeleton key points of the index finger in the last 16 frames. The movement classifier produces the probabilities for eight directions and the stop. The click classifier is a neural network that identifies the click and the unclick. The inputs for the click classifier are identical to the movement classifier, and the output is the probability of the two classes. To minimize the impact of unrelated hand frames, we use a neural network with long-short-term-memory (LSTM) [60]. Both the movement classifier and the click classifier use a neural network with two hidden layers and the LSTM. Since the input dimension of the framework is small, a network with two layers and one LSTM layer is sufficient for the classifiers, which is shown in Fig. 7. The first and second hidden layers have 48 and 10 nodes, respectively, and the LSTM layer contains 16 nodes.

4.3 Password Estimation. Based on the hand skeleton detection model, hand pose estimation model, and hand movement model, we propose to estimate the password using the RGB webcam (Fig. 6(*d*)). There are two types of attacks: *Attack 1* and *Attack 2*. *Attack 1* will provide the length of the password, while

the length of the password in *Attack 2* is not given. The pseudo-code of *Attack 1* is presented in Algorithm 1. Figure 6 illustrates the overall framework for the unknown length password estimation. The inputs of the framework are RGB video frames from the webcam. We first detect the hand and skeleton key points for each frame and identify the hand pose. When the hand pose is a pointer, the movement classifier and the click classifier identify the movement and the click separately. After that, the password estimation model estimates the digit based on the results of the movement classifier and the click classifier.

Attack 1. In *Attack 1* (see Algorithm 1), the attacker can get the coordinates of the virtual interface (x, y, width, height), recorded video V, and the length of the password n. Since the length of the password is available, we roughly separate the recorded information into n clips according to the finger area a, where n represents the digit number of the password. If the number of lasting frames for the index fingertip on the digit button is greater than a threshold t_l, digits are selected as the potential password p_p. When the number of potential digits n_{p_p} is equal to n, output the selected digit as the password. When the number of selected digits is greater than n, we select the top-n common digits in p_p as the output password P. If $n_{p_p} < n$, we use the click and movement information to determine whether some of the digits need to be counted multiple times. Following that, we add or remove digits to generate the password P with a length of n. Please refer to Algorithm 1 for more details.

Algorithm 1 Attack 1

Require: Coordinate of virtual interface (x, y, w, h); recorded video V; length of password n

for each frame $f_i \in V$ **do**
  Detect click c_i, click confidence c_i^c, movement m_i, and movement confidence m_i^c, finger area a
end for
Separate video clips using finger area a
Delete non-digit frames using t_l, get potential password p_p
if length of potential password $n_{p_p} = n$ **then**
  Password $P = p_p$
else if $n_{p_p} > n$ **then**
  Select n_n non-consecutive digits in detected order
  Shorten top common consecutive digits until $n_{p_p} = n$
else
  if Multiple $c_i^c \in c_V$ **then**
    Add digits to p_p and separate digit frequency
    if $n_{p_p} < n$ **then**
      Add random numbers until $n_{p_p} = n$
    else if $n_{p_p} = n$ **then**
      Password $P = p_p$
    else
      Select n_n non-consecutive digits in detected order
      Shorten top common consecutive digits until $n_{p_p} = n$
    end if
  else if Multiple $c_i^c \notin c_V$ **then**
    Add random numbers until $n_{p_p} = n$
  end if
end if

Attack 2. In *Attack 2*, the attacker can only get the coordinates of the virtual interface and the layout of the keyboard to estimate the password in a real-time situation. We use the click result, the movement results, and the index finger location to determine the password. Since the length of the password is not known, we use a sliding set s_s to record the effective movement and the click results. The effective movement results are selected when its movement confidence is greater than the threshold t_m, and the click results are recorded when the click confidence is greater than the threshold t_c. After that, the potential password digit is the most common number in the sliding window. When there is no movement, the potential password digit is selected for the final password. We also set a click parameter p_c to dynamically control the waiting frames, and a movement parameter p_m to determine which potential password digit can be selected. Specifically, the number of waiting frames $w_i = p_c * w_c$, where w_c is the number of the most common numbers in the sliding set.

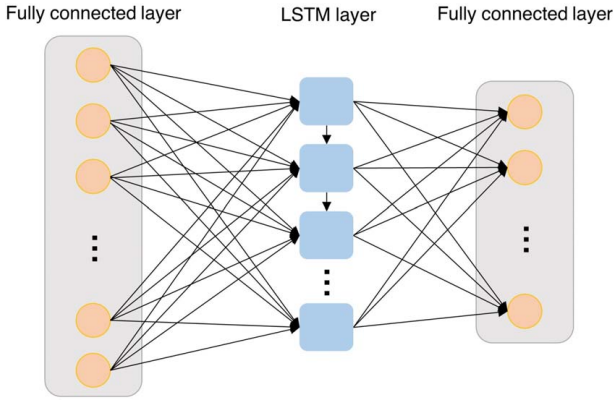

Fig. 7 The neural network of the movement classifier and the click classifier, consisting of two fully connected hidden layers and an LSTM layer in between

5 Inference Performance Evaluation

5.1 Experimental Setup. We setup experiments to evaluate the real-time performance of the proposed method. First, we collect the hand gesture dataset to train the hand pose detector and the hand movement classifier, as discussed in Sec. 4.2. Then we setup the attack scenarios mentioned in Sec. 4.1, which are demonstrated in Fig. 5.

We compare the proposed password estimation method with a clustering method, which is based on one neural network with two hidden layers. The output of the clustering method is eight directions, click, and not move. To capture videos for the proposed method and the clustering method, we put two hidden cameras in front of the user. In order to eliminate the difference in both intrinsic and extrinsic parameters of the two cameras, calibration is conducted. The button "1" at the upper left corner and button "Enter" at the lower right corner of the virtual interfaces (see Fig. 4(*b*)) are picked and compared to obtain the horizontal and vertical distance difference (Δu, Δv) in the two videos captured by two cameras.

$$\Delta u = 3 \cdot l + 2 \cdot s_u \tag{1}$$

$$\Delta v = 4 \cdot l + 3 \cdot s_v \tag{2}$$

$$l : s_u : s_v = 8 : 5 : 1 \tag{3}$$

where l is the length of the button square, s_u and s_v mean the space horizontally and vertically among buttons. Their ratio relationships are directly obtained from the interface design process, see Fig. 4(*b*). We first use Eqs. (1) and (3) to obtain l and s_u, then apply Eq. (2) to get s_v.

Details. The user first conducts the 500 sets of single-key button tests. Then the user inputs three groups of passwords (i.e., 100 sets of 2-digit, 4-digit, and 6-digit, respectively, for each group) for *Attack 1* and *Attack 2*, respectively. For *Attack 1*, separated finger area can be combined if the non-digit frames are less than threshold $t_l = 7$. When multiple clicks happen, the lasting time of one finger area will be separated evenly. For *Attack 2*, we set the movement parameter p_m and the click parameter p_c as 1.0 and 0.75 to

control the sensitivity of detecting movement and click, and the confidence thresholds $t_m = 0.5$ and $t_c = 0.5$. In the clustering method, the clustering sensitivity index i_c is 0.95. The average frame rate of the experiment for the proposed method and the clustering baseline method is 8.1 fps. For hand detection and tracking, both the minimum detection threshold and tracking threshold are 0.5. The confidence range for click button c_V is (0.7, 1).

5.2 Dataset. We collected 4797 hand pose images to train the hand pose detector. Each hand pose image is labeled by one of the four poses (*open*, *close*, *pointer*, and *pick*), where *pointer* indicates the hand gesture during clicking buttons in the virtual panel, and *pick* is used to adjust the layout of the virtual keyboard. Other 5800 hand gestures were collected to train the hand movement classifier. Each hand gesture data are in the form of $\langle name, click \rangle$, where *name* indicates the direction of movement (e.g., eight direction or no move), and the *click* label indicates whether the gesture has the click action or not, i.e., $click \in \{click, non-click\}$. The hand pose detection accuracy is 93.0% and the hand movement detection accuracy is 94.3%.

To evaluate the proposed method, we also collected the following testing datasets. A 1-digit test dataset includes 500 samples, where each digit has 50 samples. A 2-digit test dataset consists of samples from "00" to "99" and 100 in total. A 4-digit dataset includes 100 samples randomly generated from "0000" to "9999" and a 6-digit dataset also includes 100 samples that are randomly generated from "000000" to "999999."

5.3 Metrics. We use the inference accuracy to evaluate the performance of the two types of attacks, which is defined by

$$\text{Inference} \quad \text{Acc.} = \frac{N_m}{\max(L_{\text{psd}}, N_t)} \times 100\% \tag{4}$$

where N_m is the size of the correct inferred digits where each digit has a right relative position with the targeted password. L_{psd} is the length of the targeted password, and N_t is the size of the inferred password. Note that for *Attack 1* with knowledge of the password length, i.e., $L_{\text{psd}} = N_t$, the inference accuracy could be simplified as N_m/N_t.

5.4 Results. *Impact of Different Key Numbers.* We use the 1-digital test dataset to evaluate the inference accuracy of the proposed method and the clustering method. Since the inference of *Attack 1* and clustering with known password length both high and comparable accuracy (i.e., 99.01% and 92.08%, respectively), we focus on the comparison of *Attack 2* and clustering without known password length. As we can see from Fig. 8, the average inference accuracy of our proposed method, i.e., *Attack 2*, is 80.97%, which is much higher than the one of clustering (60.35%). An interesting observation we have is that *Attack 2* has different inference accuracies for different key numbers. As shown in Table 1, key numbers "1," "4," and "7" have a low accuracy compared with others (see Table 1). The main reason is that these three number buttons are placed in the left column of the interface panel, as shown in Fig. 4(*a*). Since the dataset is collected from a right-handed person, the "Click" operation with hand tracking affects the gesture recognition of the left column in pixel space resulting in low inference accuracy. In contrast, the key numbers in the right column of the pixel space, e.g., "3," "6," and "9," have a higher inference accuracy due to the same reason (shown in Table 1).

Impact of Password Length. We also evaluate the impact of password length on inference accuracy. From Fig. 9, the inference accuracy of *Attack 1*, *Attack 2*, clustering with known password length, and clustering without password known length all decrease as the password length increases. This result is anticipated as the longer the password length, the smaller size of the correct inferred digits, e.g., smaller N_m, resulting in lower inference accuracy. Moreover,

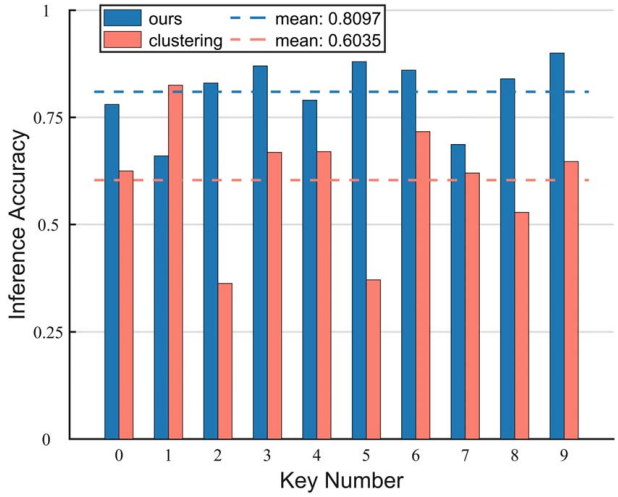

Fig. 8 Inference accuracy of each key number using our proposed method and clustering method

Attack 1 and the clustering with known password length have higher inference accuracy. This is expected as the more preknowledge the attacker has, the higher the inference accuracy is. *Attack 1* achieves the highest inference accuracy among different digits cases (i.e., 97.03%, 94.06%, and 83.83% for 2-digit, 4-digit, and 6-digit, respectively). In addition, both *Attack 1* and *Attack 2* outperform the clustering method under the same conditions, respectively (i.e., with or without known password length). In fact, the inference accuracy of *Attack 2* is much higher than the one of clustering without known password length for all cases. Even compared with clustering with known password length, the inference accuracy of *Attack 2* is very close, which further confirms the superiority of our proposed method. Please refer to Table 2 for detailed statistics. During our investigation, we observed an unexpected increase in accuracy from 4-digit to 6-digit passwords using the proposed method during attack 2. Our findings indicated that, on average, the inference accuracy was slightly higher for 6-digit passwords than for 4-digit passwords based on 100 test cases. However, we conducted a two-tailed *t*-test on the two datasets and obtained a *p*-value of 0.3943, which is greater than the standard significance level of 0.05. This suggests that there is no significant difference between the accuracy of the two datasets. Additionally, when considering only the total number of corrected estimations from both datasets, we found that the 4-digit group outperformed the 6-digit group with 26 corrected estimations showing higher accuracy, compared to only six for the 6-digit group. We attribute the observed increase in accuracy for 6-digit passwords to the calculation metric employed. The input of a 6-digit password requires more gesture movements by the operator, resulting in a longer length of output under the unknown length condition, which can increase the accuracy of some test cases. These results indicate that the password length would significantly affect the performance of the attack, and the knowledge about password length increases the inference accuracy as well. Therefore, increasing the password length and varying the password length are expected to be effective ways to improve security.

Table 1 Key number accuracy according to the column distribution

Column	Key numbers	Inference accuracy	Standard deviation
1	1,4,7	71.01%	0.4418
2	2,5,8,0	83.17%	0.3510
3	3,6,9	87.65%	0.2883

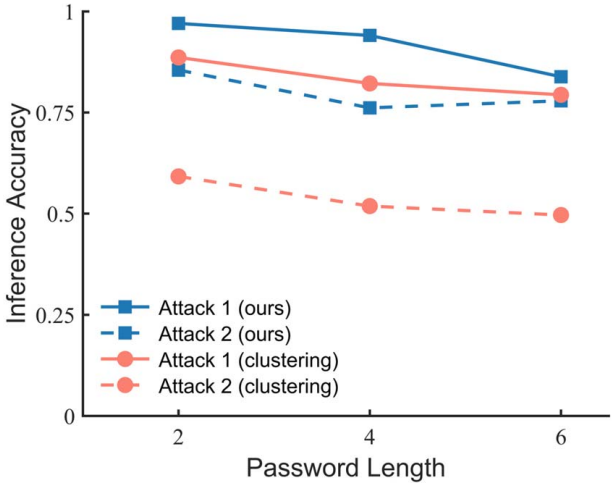

Fig. 9 Inference accuracy of the password according to different lengths using the proposed method (ours) and clustering method

Table 2 Top 1 password inference accuracy under two attacks

Attack	Method	2-digit Acc.	4-digit Acc.	6-digit Acc.
1	Clustering	88.61%	82.18%	79.37%
	Ours	97.03%	94.06%	83.83%
2	Clustering	59.18%	51.85%	49.64%
	Ours	85.50%	76.15%	77.89%

6 Conclusion and Future Works

With the adoption of XR techniques in Industry 4.0, security and privacy issues are seldom discussed in previous research. Mid-air interactions, supported by XR techniques, bring a new affordance of intuitive and efficient interactions to the user but also pose a potential risk for privacy and security information inference. To study this potential risk, we developed a prototyping system for smart factory management based on a video see-through MR head-mounted display. This system allows the user to interact with the MR interfaces through mid-air interactions, including the input of a password on a login interface to access security/privacy-sensitive information. The proposed attack targets inferring the user's password based on the known size, layout, and relative position of the XR interface and a video stream or recording from a hidden camera in front of the user. We built up a machine learning-empowered password inference framework consisting of hand gesture tracking and password estimation modules. To evaluate the performance of the inference, we tested passwords with different lengths in two scenarios: known and unknown password lengths. The result indicates that the proposed method achieves a high accuracy of 97.03%, 94.06%, and 83.83% for 2-digit, 4-digit, and 6-digit passwords, respectively.

Based on the result of this research, the potential preventive actions for cybersecurity attack during password input with hand gesture include using more complex characters, applying remote interaction methods, or inputting the password while moving. In the future, practicability and accuracy are expected to improve with more sophisticated inference algorithms. We plan to extend the proposed method to other application scenarios, for example, the QWERTY keyboard. Furthermore, more challenging attacks, such as those without knowing the positions and layouts of the numerical keypad, are worthy of investigation. A more sophisticated algorithm is also expected to be developed to deal with the current problem of having lower accuracy in estimating the right column in the pixel space.

Acknowledgment

This material is based upon work partially supported by the National Science Foundation under Grant Nos. 2222853 and 2125362. Any opinions, findings, conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation. Yidan Hu is partially supported by the Grant Writing Bootcamp Funding 2021 from the Rochester Institute of Technology, and she is also partially supported by the 2022 Meta Research Awards for Privacy-Enhancing Technologies.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The datasets generated and supporting the findings of this article are obtainable from the corresponding author upon reasonable request.

References

[1] Zhou, K., Liu, T., and Zhou, L., 2015, "Industry 4.0: Towards Future Industrial Opportunities and Challenges," 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Zhangjiajie, China, Aug. 15–17, IEEE, pp. 2147–2152.

[2] Egger, J., and Masood, T., 2020, "Augmented Reality in Support of Intelligent Manufacturing—A Systematic Literature Review," Comput. Ind. Eng., **140**, p. 106195.

[3] Thomas, T., 2020, "Top 4 U.S. Manufacturing Challenges and How to Overcome Them," Online, https://blog.thomasnet.com/top-manufacturing-challenges, Accessed November 9, 2022.

[4] Liu, R., Peng, C., Zhang, Y., Husarek, H., and Yu, Q., 2021, "A Survey of Immersive Technologies and Applications for Industrial Product Development," Comput. Graph., **100**, pp. 137–151.

[5] Malik, A. A., Masood, T., and Bilberg, A., 2020, "Virtual Reality in Manufacturing: Immersive and Collaborative Artificial-Reality in Design of Human–Robot Workspace," Int. J. Comput. Integr. Manuf., **33**(1), pp. 22–37.

[6] Salah, B., Abidi, M. H., Mian, S. H., Krid, M., Alkhalefah, H., and Abdo, A., 2019, "Virtual Reality-Based Engineering Education to Enhance Manufacturing Sustainability in Industry 4.0," Sustainability, **11**(5), p. 1477.

[7] Makhataeva, Z., and Varol, H. A., 2020, "Augmented Reality for Robotics: A Review," Robotics, **9**(2), p. 21.

[8] Yang, W., Xiao, Q., and Zhang, Y., 2021, "An Augmented-Reality Based Human–Robot Interface for Robotics Programming in the Complex Environment," International Manufacturing Science and Engineering Conference, Cincinnati, OH, June 21–25, Vol. 85079, American Society of Mechanical Engineers, p. V002T07A003.

[9] Guo, Z., Zhou, D., Zhou, Q., Zhang, X., Geng, J., Zeng, S., Lv, C., and Hao, A., 2020, "Applications of Virtual Reality in Maintenance During the Industrial Product Lifecycle: A Systematic Review," J. Manuf. Syst., **56**, pp. 525–538.

[10] Danielsson, O., Holm, M., and Syberfeldt, A., 2020, "Augmented Reality Smart Glasses in Industrial Assembly: Current Status and Future Challenges," J. Ind. Inf. Integr., **20**, p. 100175.

[11] xrtoday, 2022, "The State of XR in Manufacturing and Industrial 2022," Online, https://www.xrtoday.com/mixed-reality/the-state-of-xr-in-manufacturing-and-industrial-2022/, Accessed November 1, 2022.

[12] Alismail, A., Altulaihan, E., Rahman, M. H., and Sufian, A., 2022, "A Systematic Literature Review on Cybersecurity Threats of Virtual Reality (VR) and Augmented Reality (AR)," Data Intelligence and Cognitive Informatics: Proceedings of ICDICI 2022, Tirunelveli, India, July 6–7, pp. 761–774.

[13] usa.kaspersky.com, 2023, "What Are the Security and Privacy Risks of VR and AR," Online, https://usa.kaspersky.com/resource-center/threats/security-and-privacy-risks-of-ar-and-vr, Accessed October 6, 2022.

[14] Lu, Y., and Da Xu, L., 2018, "Internet of Things (IoT) Cybersecurity Research: A Review of Current Research Topics," IEEE Internet Things J., **6**(2), pp. 2103–2115.

[15] Pacheco, J., and Hariri, S., 2018, "Anomaly Behavior Analysis for IoT Sensors," Trans. Emerg. Telecommun. Technol., **29**(4), p. e3188.

[16] Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K.-K. R., and Parizi, R. M., 2020, "An Ensemble of Deep Recurrent Neural Networks for Detecting IoT Cyber Attacks Using Network Traffic," IEEE Internet Things J., **7**(9), pp. 8852–8859.

[17] Ayoade, G., El-Ghamry, A., Karande, V., Khan, L., Alrahmawy, M., and Rashad, M. Z., 2019, "Secure Data Processing for IoT Middleware Systems," J. Supercomput., **75**(8), pp. 4684–4709.

[18] Niu, W., Zhang, X., Du, X., Zhao, L., Cao, R., and Guizani, M., 2020, "A Deep Learning Based Static Taint Analysis Approach for IoT Software Vulnerability Location," Measurement, **152**, p. 107139.

[19] Zhang, R., Zhang, N., Du, C., Lou, W., Hou, Y. T., and Kawamoto, Y., 2017, "From Electromyogram to Password: Exploring the Privacy Impact of Wearables in Augmented Reality," ACM Trans. Intell. Syst. Technol., **9**(1), pp. 1–20.
[20] Lehman, S. M., Alrumayh, A. S., Kolhe, K., Ling, H., and Tan, C. C., 2022, "Hidden in Plain Sight: Exploring Privacy Risks of Mobile Augmented Reality Applications," ACM Trans. Privacy Security, **25**(4), p. 26.
[21] Kreider, C., 2018, "The Discoverability of Password Entry Using Virtual Keyboards in an Augmented Reality Wearable: An Initial Proof of Concept," https://aisel.aisnet.org/sais2018/23.
[22] Luo, S., Hu, X., and Yan, Z., 2022, "Holologger: Keystroke Inference on Mixed Reality Head Mounted Displays," 2022 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Christchurch, New Zealand, Mar. 12–16, IEEE, pp. 445–454.
[23] Bimber, O., and Raskar, R., 2006, "Modern Approaches to Augmented Reality," ACM SIGGRAPH 2006 Courses, Boston, MA, July 30–Aug. 3.
[24] Hoppenstedt, B., Kammerer, K., Reichert, M., Spiliopoulou, M., and Pryss, R., 2019, "Convolutional Neural Networks for Image Recognition in Mixed Reality Using Voice Command Labeling," International Conference on Augmented Reality, Virtual Reality and Computer Graphics, Santa Maria al Bagno, Italy, June 24–27, Springer, pp. 63–70.
[25] Park, H. M., Lee, S. H., and Choi, J. S., 2008, "Wearable Augmented Reality System Using Gaze Interaction," 2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality, Cambridge, UK, Sept. 15–18, IEEE, pp. 175–176.
[26] Kytö, M., Ens, B., Piumsomboon, T., Lee, G. A., and Billinghurst, M., 2018, "Pinpointing: Precise Head- and Eye-Based Target Selection for Augmented Reality," Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems, Montréal, Canada, Apr. 21–26, pp. 1–14.
[27] Shen, Y., Ong, S.-K., and Nee, A. Y., 2011, "Vision-Based Hand Interaction in Augmented Reality Environment," Int. J. Human–Computer Interact., **27**(6), pp. 523–544.
[28] Gugenheimer, J., Dobbelstein, D., Winkler, C., Haas, G., and Rukzio, E., 2016, "Facetouch: Touch Interaction for Mobile Virtual Reality," Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, San Jose, CA, May 7–12, pp. 3679–3682.
[29] Yang, T.-H., Kim, J. R., Jin, H., Gil, H., Koo, J.-H., and Kim, H. J., 2021, "Recent Advances and Opportunities of Active Materials for Haptic Technologies in Virtual and Augmented Reality," Adv. Funct. Mater., **31**(39), p. 2008831.
[30] Bailenson, J., 2018, "Protecting Nonverbal Data Tracked in Virtual Reality," JAMA Pediatrics, **172**(10), pp. 905–906.
[31] JofréPasinetti, N., Rodríguez, G., Alvarado, Y., Fernández, J., and Guerrero, R. A., 2016, "Non-verbal Communication for a Virtual Reality Interface," XXII Congreso Argentino de Ciencias de la Computación (CACIC 2016), Provincia de San Luis, Argentina, Oct. 3–7.
[32] Goh, E. S., Sunar, M. S., and Ismail, A. W., 2019, "3d Object Manipulation Techniques in Handheld Mobile Augmented Reality Interface: A Review," IEEE Access, **7**, pp. 40581–40601.
[33] Oliveira, M., Arica, E., Pinzone, M., Fantini, P., and Taisch, M., 2019, "Human-Centered Manufacturing Challenges Affecting European Industry 4.0 Enabling Technologies," International Conference on Human–Computer Interaction, Orlando, FL, July 26–31, Springer, pp. 507–517.
[34] Lawson, G., Herriotts, P., Malcolm, L., Gabrecht, K., and Hermawati, S., 2015, "The Use of Virtual Reality and Physical Tools in the Development and Validation of Ease of Entry and Exit in Passenger Vehicles," Appl. Ergon., **48**, pp. 240–251.
[35] Matsas, E., and Vosniakos, G.-C., 2017, "Design of a Virtual Reality Training System for Human–Robot Collaboration in Manufacturing Tasks," Int. J. Interactive Des. Manuf., **11**(2), pp. 139–153.
[36] Palmarini, R., Erkoyuncu, J. A., Roy, R., and Torabmostaedi, H., 2018, "A Systematic Review of Augmented Reality Applications in Maintenance," Robot. Comput.-Integr. Manuf., **49**, pp. 215–228.
[37] Ong, S.-K., Yew, A., Thanigaivel, N. K., and Nee, A. Y., 2020, "Augmented Reality-Assisted Robot Programming System for Industrial Applications," Robot. Comput.-Integr. Manuf., **61**, p. 101820.
[38] Gattullo, M., Scurati, G. W., Evangelista, A., Ferrise, F., Fiorentino, M., and Uva, A. E., 2019, "Informing the Use of Visual Assets in Industrial Augmented Reality," International Conference of the Italian Association of Design Methods and Tools for Industrial Engineering, Modena, Italy, Sept. 9–10, Springer, pp. 106–117.
[39] Yew, A., Ong, S., and Nee, A. Y., 2016, "Towards a Griddable Distributed Manufacturing System With Augmented Reality Interfaces," Robot. Comput.-Integr. Manuf., **39**, pp. 43–55.
[40] Maharjan, D., Agüero, M., Mascarenas, D., Fierro, R., and Moreu, F., 2021, "Enabling Human-Infrastructure Interfaces for Inspection Using Augmented Reality," Struct. Health Monit., **20**(4), pp. 1980–1996.
[41] Wang, P., Bai, X., Billinghurst, M., Zhang, S., Wei, S., Xu, G., He, W., Zhang, X., and Zhang, J., 2021, "3dgam: Using 3d Gesture and CAD Models for Training on Mixed Reality Remote Collaboration," Multimedia Tools Appl., **80**(20), pp. 31059–31084.
[42] Casey, P., Baggili, I., and Yarramreddy, A., 2021, "Immersive Virtual Reality Attacks and the Human Joystick," IEEE Trans. Dependable Secure Comput., **18**(2), pp. 550–562.
[43] Vallurupally, S., Gulhane, A., Hoque, K. A., and Calyam, P., 2022, "Modeling and Defense of Social Virtual Reality Attacks Inducing Cybersickness," IEEE Trans. Dependable Secure Comput., **19**(6), pp. 4127–4144.
[44] Meyer-Lee, G., Shang, J., and Wu, J., 2018, "Location-Leaking Through Network Traffic in Mobile Augmented Reality Applications," IEEE 37th International Performance Computing and Communications Conference (IPCCC), Orlando, FL, Nov. 17–19, pp. 1–8.
[45] Al Arafat, A., Guo, Z., and Awad, A., 2021, "Vr-spy: A Side-Channel Attack on Virtual Key-Logging in VR Headsets," 2021 IEEE Virtual Reality and 3D User Interfaces (VR), Lisbon, Portugal, Mar. 27–Apr. 2, IEEE, pp. 564–572.
[46] Ling, Z., Li, Z., Chen, C., Luo, J., Yu, W., and Fu, X., 2019, "I Know What You Enter on Gear VR," IEEE Conference on Communications and Network Security (CNS), Washington, DC, June 10–12, pp. 241–249.
[47] Giaretta, A., 2022, "Security and Privacy in Virtual Reality—A Literature Survey," preprint arXiv:2205.00208.
[48] Shang, J., Chen, S., Wu, J., and Yin, S., 2022, "Arspy: Breaking Location-Based Multi-player Augmented Reality Application for User Location Tracking," IEEE Trans. Mobile Comput., **21**(2), pp. 433–447.
[49] Maloney, D., Zamanifard, S., and Freeman, G., 2020, "Anonymity Vs. Familiarity: Self-disclosure and Privacy in Social Virtual Reality," 26th ACM Symposium on Virtual Reality Software and Technology, VRST '20, Ottawa, Canada, Nov. 1–4.
[50] Falk, B., Meng, Y., Zhan, Y., and Zhu, H., 2021, "Poster: Reavatar: Virtual Reality De-anonymization Attack Through Correlating Movement Signatures," Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security, CCS '21, Seoul, South Korea, Nov. 15–19, pp. 2405–2407.
[51] De Guzman, J. A., Thilakarathna, K., and Seneviratne, A., 2019, "Security and Privacy Approaches in Mixed Reality: A Literature Survey," ACM Comput. Surveys, **52**(6), pp. 1–37.
[52] Meng, Y., Li, J., Zhu, H., Liang, X., Liu, Y., and Ruan, N., 2019, "Revealing Your Mobile Password Via Wifi Signals: Attacks and Countermeasures," IEEE Trans. Mobile Comput., **19**(2), pp. 432–449.
[53] Wang, Y., Cai, W., Gu, T., and Shao, W., 2019, "Your Eyes Reveal Your Secrets: An Eye Movement Based Password Inference on Smartphone," IEEE Trans. Mobile Comput., **19**(11), pp. 2714–2730.
[54] Oculus, 2020, "Oculus Integration: Integration," Online, https://assetstore.unity.com/packages/tools/integration/oculus-integration-82022#description, Accessed October 6, 2022.
[55] Microsoft, 2020, "Unity: Mixed Reality Toolkit (MRTK)," Online, https://github.com/microsoft/MixedRealityToolkit-Unity, Accessed October 6, 2022.
[56] Enox Software, 2014. "Opencv for Unity," Online, https://assetstore.unity.com/packages/tools/integration/opencv-for-unity-21088#releases, Accessed October 6, 2022.
[57] Voigt-Antons, J.-N., Kojic, T., Ali, D., and Möller, S., 2020, "Influence of Hand Tracking as a Way of Interaction in Virtual Reality on User Experience," 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), Athlone, Ireland, May 26–28, IEEE, pp. 1–4.
[58] Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., Zhang, F., Chang, C.-L., Yong, M. G., Lee, J., Chang, W., Hua, W., Georg, M., and Grundmann, M., 2019, "Mediapipe: A Framework for Building Perception Pipelines," Third Workshop on Computer Vision for AR/VR at IEEE Computer Vision and Pattern Recognition (CVPR) 2019, Long Beach, CA, June 17.
[59] Shanthakumar, V. A., Peng, C., Hansberger, J., Cao, L., Meacham, S., and Blakely, V., 2020, "Design and Evaluation of a Hand Gesture Recognition Approach for Real-Time Interactions," Multimedia Tools Appl., **79**(25), pp. 17707–17730.
[60] Hochreiter, S., and Schmidhuber, J., 1997, "Long Short-Term Memory," Neural Comput., **9**(8), pp. 1735–1780.