

# Detecting Cyber-Attacks Against Cyber-Physical Manufacturing System: A Machining Process Invariant Approach

Zedong Li<sup>✉</sup>, Xin Chen, *Member, IEEE*, Yuqi Chen, Shijie Li, Hangyu Wang, Shichao Lv<sup>✉</sup>, and Limin Sun<sup>✉</sup>

**Abstract**—The era of the Industrial Internet of Things has led to an escalating menace of cyber-physical manufacturing systems (CPMSs) to cyber-attacks. Presently, the field of intrusion detection for CPMS has significant advancements. However, current methodologies require significant costs for collecting historical data to train detection models, which are tailored to specific machining scenarios. Evolving machining scenarios in the real world challenge the adaptability of these methods. In this article, We found that the machining code of the CPMS contains a complete machining process, which is an excellent detection basis. Therefore, we propose MPI-CNC, an intrusion detection approach based on Machining Process Invariant in the machining code. Specifically, MPI-CNC automates the analysis of the machining codes to extract machining process rules and key parameter rules, which serve as essential detection rules. Then, MPI-CNC actively acquires runtime status from the CPMS and matches the detection rules to identify cyber-attacks behavior. MPI-CNC was evaluated using two FANUC computer numerical control (CNC) machine tools across ten real machining scenarios. The experiment demonstrated the exceptional adaptability capability of MPI-CNC. Furthermore, MPI-CNC showed superior accuracy in detecting cyber-attacks against CPMS compared to existing state-of-the-art detection methods while ensuring normal machining operations.

**Index Terms**—Computer numerical control (CNC), cyber attack, cyber-physical manufacturing systems (CPMSs), Industrial Internet of Things, intrusion detection.

## I. INTRODUCTION

MANUFACTURING industry is an important cornerstone of modern industrial development. With the advent of the Industrial Internet of Things and intelligent manufacturing, the global manufacturing industry is rapidly moving toward networked and intelligent development [1]. Computer numerical control (CNC) system is the core of

Manuscript received 23 October 2023; revised 11 January 2024; accepted 22 January 2024. Date of publication 25 January 2024; date of current version 9 May 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 110400ZG21. (*Corresponding author: Limin Sun*.)

Zedong Li, Xin Chen, Shijie Li, Hangyu Wang, Shichao Lv, and Limin Sun are with the Beijing Key Laboratory of IOT Information Security Technology, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100085, China, and also with the School of Cyber Security, University of Chinese Academy of Sciences, Beijing 101408, China (e-mail: lizedong@iie.ac.cn; chenxin1990@iie.ac.cn; lishijie@iie.ac.cn; wanghangyu@iie.ac.cn; lvshichao@iie.ac.cn; sunlimin@iie.ac.cn).

Yuqi Chen is with the School of Information Science and Technology, ShanghaiTech University, Shanghai 201210, China (e-mail: chenyq@shanghaitech.edu.cn).

Digital Object Identifier 10.1109/JIOT.2024.3358798

cyber-physical manufacturing systems (CPMSs) that control the machining process of manufacturing equipment. CNC systems are widely used in important industries, such as the aviation industry, automobile manufacturing, and military industry. Tesla's Giga factory connects CNC systems to the industrial Internet to automatically control production processes, greatly improving production efficiency.

As an increasing number of factories integrate their CNC systems into the Industrial Internet, the security of CPMS has become a paramount requirement and faces formidable challenges. Intrusion detection approaches for CPMS have emerged as a prominent and burgeoning topic. Currently in the field of CPMS security, most researchers focus on training machine learning classification models to detect anomalies by analyzing side channel data, such as current [2], video [3], or audio [4], [5], [6], generated during machining. Some researchers have built digital twin models for CNC systems with a data-driven approach to do consistency checks on the runtime state of CNC systems to detect cyber-attacks [7]. There is also an offline approach to detect whether machining codes have been tampered with, which extracts digital features of machining codes and trains machine learning anomaly classification models [8]. These solutions can effectively detect anomalous processing behaviors for specific machining scenarios.

Regrettably, the absence of adequate security considerations for CNC system manufacturers has resulted in attackers being able to easily launch cyber-attacks by exploiting CNC system vulnerabilities, such as the lack of authentication mechanisms, plain-text transmission, and the existence of unfixed vulnerabilities in the system. Primarily, attackers target the machining code to introduce defects in the product processing. For instance, they may implant a Trojan into the firmware of the CNC system, surreptitiously tamper with the machining code passed into the system, and execute a malicious hole attack [9]. Additionally, attackers have demonstrated the use of steganography to tamper with machining code files in network traffic, diminishing the mechanical strength of the resulting product [10]. In a further form of attack, assailants manipulate key parameters in the memory of the CNC system. For instance, they substitute the processing material by tampering with parameters, integrating smart materials into gas masks to plant physical logic bombs. This causes gas masks to crack and leak during use [11]. Furthermore, attackers have utilized existing open-source tools like C3PO [12] and Industrial

Security Exploitation Framework (ISF) [13] to send malicious instructions to CNC systems disrupting the processing processes. These instances underscore the pressing need for intrusion detection systems for CNC systems to mitigate such threats effectively.

**Motivation:** In practical production processes, the CNC system employs various machining codes to handle different products, leading to diverse machining scenarios. These different scenarios require distinct tool paths, raw materials, and machining tools, resulting in different side-channel features with audio, image, current, and voltage. To develop intrusion detection models using side-channel data across different machining scenarios, researchers typically need to gather side-channel data for each new scenario and repeat the training process, incurring significant time and labor costs. However, once attackers successfully deploy an attack script in a CPMS system, they can easily disrupt the different machining scenarios. Consequently, there is an urgent need for an adaptable intrusion detection approach within the CPMS system that can be readily deployed across a variety of machining scenarios to effectively counter existing attack methods.

**Insight:** Invariant rule-based detection is currently a popular method in the field of industrial control security to effectively detect anomalies due to cyber-attacks. Usually, industrial control devices execute control logic codes to control the normal operation of industrial systems based on the invariant control logic in the control logic codes. Researchers have utilized data-driven [14] or code-driven [15] approaches to extract control logical invariant rules in industrial control systems as the basis for intrusion detection in industrial control systems. They have achieved excellent detection results. Inspired by the invariant rule-based detection, we found that in the field of CPMS, the machining process of the CNC system is invariant, and the machining code contains comprehensive machining process invariant information. Therefore, the complete machining process invariant rules can be extracted by analyzing the machining code. The machining process invariant rules include key elements, such as machining trajectory and machining speed, which can comprehensively describe the machining process of the CNC system and is a reliable basis for intrusion detection.

**Method:** This article addresses the issue of the limited adaptive ability of the CPMS intrusion detection system by proposing MPI-CNC, an intrusion detection method based on Machining Process Invariant. MPI-CNC automatically and rapidly extracts detection rules from the machining code. The method first parses the machining code to extract tool paths, machining sequences, spindle speeds, and other key machining-related parameters as rules for detecting attacks. MPI-CNC then actively collects runtime machining status, and key parameters from the CNC system during machining. Finally, MPI-CNC verifies the consistency of the runtime machining data based on the detection rules to identify cyber-attacks.

**Result:** To verify the feasibility of the approach in this article, a prototype was developed based on the FANUC CNC system. We conducted experiments using real CNC machines, analyzed 10 real machining scenarios and 3 attack

methods, and evaluated the deployment time cost, detection performance, and interference to the CNC system. Experiments demonstrated that MPI-CNC can be quickly applied to new machining scenarios without preprocessing and detect cyber-attacks accurately in runtime without affecting the normal operation of the CNC. MPI-CNC has better detection performance compared to the other state-of-the-art detection methods. The detection accuracy of machining code injection attack and parameter injection attack reaches 98.81% and 100%, respectively, while the best detection results of other methods are 98.38% [8] and 93.25% [7].

This article contributes as follows.

- 1) We propose a novel approach for the automatic extraction of detection rules by analyzing machining codes. It can rapidly generate detection rules for different machining scenarios, thereby improving the adaptability capability of CPMS intrusion detection.
- 2) We conducted a reverse analysis of the FOCAS protocol used in FANUC CNC systems and developed low-interference acquisition request packets that conform to the protocol format. This approach improves the efficiency of data acquisition while reducing interference to the machining process.
- 3) A prototype CPMS IDS was developed based on the FANUC CNC system. Although this prototype was developed for a specific CNC system, based on this idea, it can be modified to expand and adapt to other CNC systems and protocols.
- 4) We evaluated the adaptability capability and detection performance of our proposed approach in 10 real machining scenarios and 3 attack scenarios. Experiments show that this approach can be quickly applied to new machining scenarios without preprocessing and detect cyber-attacks accurately in runtime without affecting the normal operation of the CNC.

**Roadmap:** The remainder of this article is structured as follows. Section II briefly introduces the technical background related to CPMS, and Section III provides an overview of the MPI-CNC. Section IV details the MPI-CNC and the specific implementation. The experimental evaluation is detailed in Section V. Section VI introduces the related work of current CPMS intrusion detection methods. Section VII discusses the limitations of the MPI-CNC. Section VIII is the conclusion.

## II. BACKGROUND

This article is primarily dedicated to proposing an intrusion detection method for CPMS. This chapter serves to provide an overview of the research background, focusing on two essential aspects: 1) the composition and 2) machining process of CPMS, as well as the various forms of attacks encountered by CPMS.

### A. Cyber-Physical Manufacturing Systems

The CPMS generally consists of an engineering station, a distributed numeric control (DNC) server, a machine data collection (MDC) server, and manufacturing equipment connected through an industrial switch [see Fig. 1(a)]. The

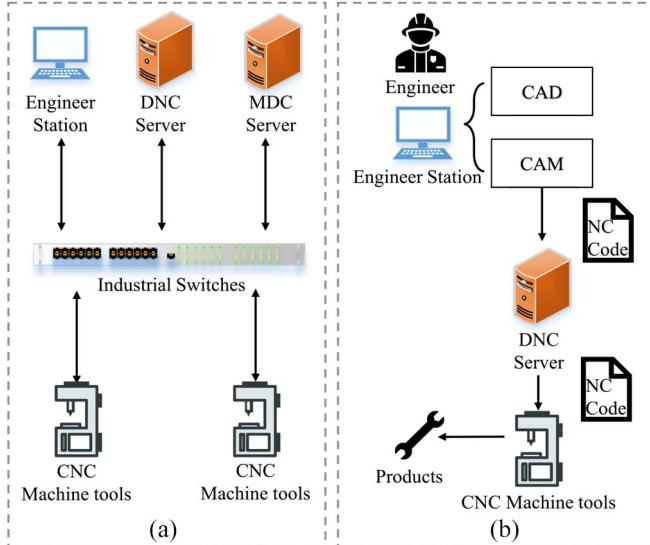


Fig. 1. Network topology and processing process of a CPMS. (a) Network topology of CPMS. (b) Processing process of CPMS.

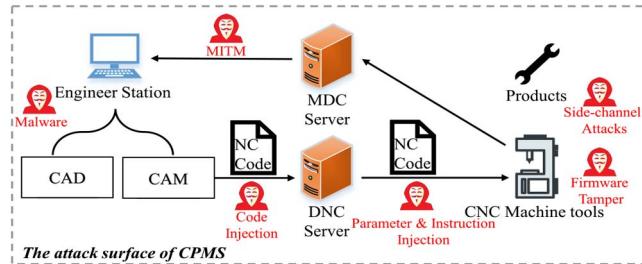


Fig. 2. Attack surface of CPMS.

engineering station is usually an office computer equipped with computer aided design (CAD), and computer aided machining (CAM) programs. The Processing process is shown in Fig. 1(b). Engineers utilize engineering design software to generate machining code (alternative name NC code), which is then uploaded to the DNC server. The DNC server distributes the NC code to the appropriate manufacturing equipment. The CNC system automatically controls the machining process by parsing the NC code. The MDC server interacts with the manufacturing equipment to collect various states of the equipment, including position, speed, temperature, and other information. This data is returned to the monitoring program of the engineering station, allowing engineers to monitor the machining process. Additionally, engineers can send control commands to perform runtime operations during the machining process.

### B. Attack Model

In recent times, scholars have analyzed and categorized cyber-attacks targeting CPMS [16], [17]. This article investigates recent cyber-attacks against CPMS, analyzing the attack surface from the perspective of the production process (see Fig. 2). In the production process, the engineer station is connected to a local area network or even the Internet. Attackers can maliciously target the engineer station using spear-phishing [18], BadUSB [19], and other vectors carrying

malicious code to exploit system and software vulnerabilities, stealing and tampering with CAD models and NC codes. Devices, such as engineer stations and DNC servers, typically communicate with the CNC system via Ethernet, using communication protocols that often lack authentication, encryption, and other security mechanisms. For example, DNC servers may use the FTP protocol to transmit NC code in clear text. Attackers can perform man-in-the-middle attacks [20], tampering with and stealing NC code from network traffic, and replaying network packets to inject malicious commands and parameters. The attacker can also tamper with the CNC system firmware [9], [21] to interfere with normal processing. However, such attacks are more difficult and require a deep understanding of the underlying code structure of the CNC system. As CPMS is a typical cyber-physical system [22], attackers can use side-channel attack methods, such as electrical measurement interference and acoustic resonance, to interfere with normal processing [23], or infer the production state of the machine tool and workpiece geometry information from leaked physical information [24], achieving a steganography attack.

The attacks were classified into three categories: 1) machining code injection; 2) parameter injection; and 3) instruction injection.

**Machining Code Injection:** Machining Code injection attack [9], [10], [25] refers to tampering with or replacing the machining code, the NC code, of the CNC system. By modifying key code segments, such as the machining path, spindle speed, or auxiliary control code, attackers can interfere with and disrupt the CNC machining process.

**Parameter Injection:** Parameter Injection [11], [26] refers to tampering with the parameters of the CNC system. There are many important parameters in the CNC system that affect the machining process, such as the spindle speed ratio value, the rapid feed rate value, and the alarm shielding. Therefore, if attackers can tamper with these key parameters, it will cause serious damage to the CNC system, affecting machining accuracy and potentially damaging the CNC machine.

**Instruction Injection:** Instruction Injection refers to sending malicious control commands to the CNC system, which disrupts the normal machining process. McCormack et al. [12] introduced an open-source tool called C3PO, which analyzes potential vulnerabilities in network services of 3-D printers and uses network vulnerabilities to send malicious commands to attack remote-controlled CNC systems. Attackers can also use the ISF [13] to inject malicious commands by sending attack scripts to disrupt the production process.

Moreover, the CNC systems also face security threats, such as physical cross-domain attacks and side-channel information leakage [27]. Nevertheless, it is pertinent to note that these threats lie outside the purview of this article. Their inclusion is excluded due to factors, such as their diminished feasibility, limited potential for harm, or their susceptibility, to detection by existing IDSs.

### III. OVERVIEW

In this research, we propose an innovative intrusion detection method for CPMS, denoted as MPI-CNC. As illustrated in

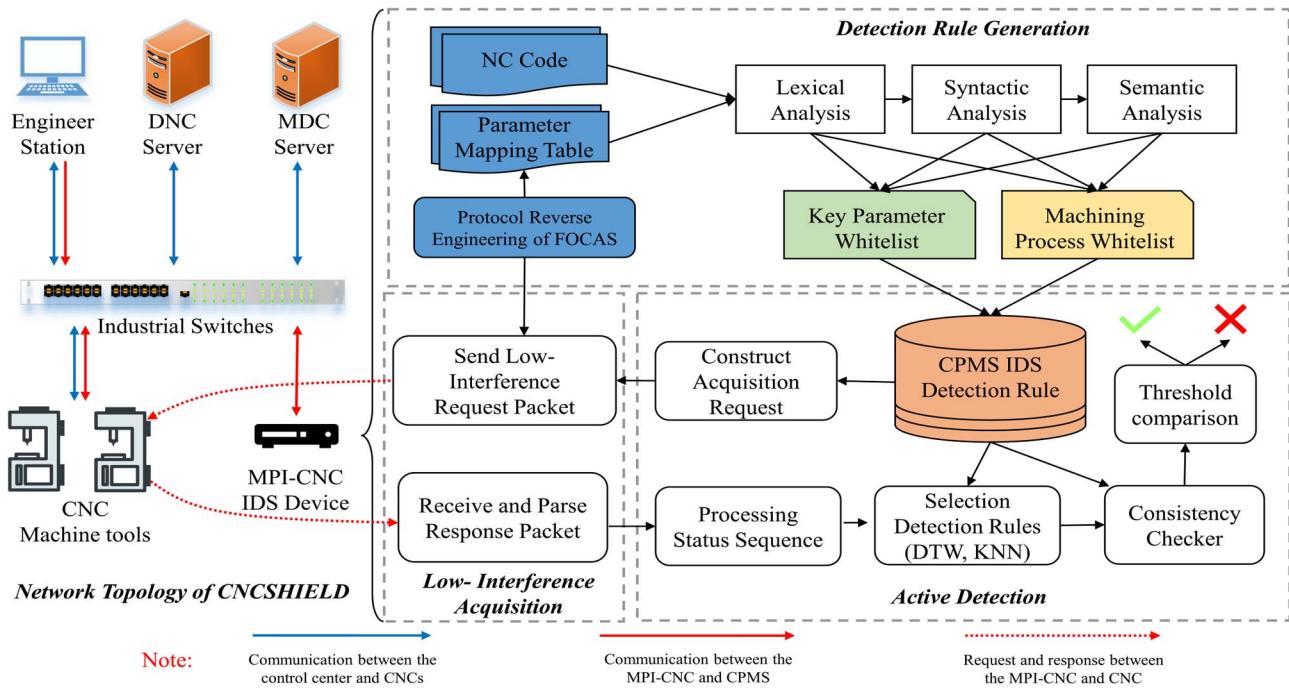


Fig. 3. Systematic approach to build MPI-CNC.

Fig. 3, this method comprises three distinct stages: 1) detection rule generation; 2) low-interference state acquisition; and 3) active detection. In this section, we present a concise overview of the fundamental framework for analyzing intrusion detection methods.

**Detection Rule Generation:** The detection rule generation module employs static analysis to parse the NC code and extract detection rules. The NC code encapsulates complete machining processes, including key parameter rules and machining process rules. The key parameter rules are utilized to monitor and verify vital parameters within the CNC system, ensuring their accuracy, stability, and safeguarding against malevolent tampering that could lead to diminished machining precision or machine malfunctions. On the other hand, the machining process rules establish reference guidelines by analyzing the invariant characteristics of machining processes in the NC code, enabling the detection of malicious attacks, such as tampering with machining trajectories or program substitution. The extracted detection rules from the NC code furnish a comprehensive depiction of the machining process and enable proactive detection of unexpected anomalies and network attacks.

**Low-Interference Acquisition:** The low-interference state acquisition module is responsible for the runtime collection of machining states within the CNC system. Typically, CNC system manufacturers provide monitoring software or development kits for monitoring the system's operational status. For instance, the FANUC Focas 1/2 development component facilitates secondary development. It enables runtime remote monitoring through active communication with the CNC system. The development component provides essential information, such as NC programs, tool positions, and spindle speeds. However, direct use of the original development kit for

high-frequency data collection can increase the network load of the CNC system, negatively impacting normal machining operations and real-time performance. To circumvent this issue, our study employs reverse engineering to analyze proprietary protocols. We also customize data collection requests and eliminate redundant ones. As a result, we achieve low-interference high-frequency acquisition of runtime machining states within the CNC system.

**Active Detection:** The active detection module primarily identifies anomalies in the machining process. It processes the runtime machining state data collected by the low-interference state acquisition module and generates alerts. It verifies whether the machining state of the CNC system adheres to the key parameter rules and the machining process rules. This approach prevents code tampering, manipulation of key parameters, and malicious instruction attacks. The active detection stage necessitates determining two critical monitoring parameters: 1) error threshold and 2) monitoring window size. Initially, we experiment with multiple monitoring window sizes based on the state acquisition frequency to determine the optimal size. Subsequently, under specific window sizes, we calculate the cumulative normal error for each monitoring window and set the error threshold using the maximum observed error.

## IV. APPROACH

### A. Problem Statement

The CNC manufacturing process is a complex industrial control process in which the CNC system performs closed-loop control of the relative motion of the tool and the workpiece based on multiple sensor data. In this article, we use  $u(t)$  in (1) to describe the machining state of the CNC at time

$t$ , where  $P(x, y, z)$  indicates the coordinates of the tool in the xyz three axes,  $S(t)$  indicates the spindle speed,  $F(t)$  indicates the feed rate, and  $T(t)$  indicates the current tool number

$$u(t) = (P(x, y, z), S(t), F(t), T(t)). \quad (1)$$

We define (2) with  $r(n)$  to describe the machining process indicated by the machining code, which represents the invariant characteristics of the CNC machining process. Specifically, we use  $F(x, y, z)$  to represent the curve equation of the machining path, which is commonly straight lines and circular arcs. Start( $x, y, z$ ) and End( $x, y, z$ ) represent the start and end points of the machining path. The combination of machining path, spindle speed, feed rate, and tool number provides a complete description of the machining process

$$r(n) = (F(x, y, z), \text{Start}(x, y, z), \text{End}(x, y, z), S(n), F(n), T(n)). \quad (2)$$

We use  $\mathcal{M}()$  in (3) to describe the CNC machining model, where  $\varepsilon(t)$  represents the internal losses of the CNC and reasonable errors due to natural factors

$$u(t+1) = \mathcal{M}(u(t), r(t), \varepsilon(t)). \quad (3)$$

In the normal machining process, the CNC machining state has reasonable errors  $\varepsilon(t)$  due to machine wear and tear, current and voltage jitter, and other factors. However, when the CNC is under a cyber-attack, it can deviate significantly from the CNC machining model  $\mathcal{M}()$  and violate the current machining process  $r(n)$ . Therefore, in this article, we designed an intrusion detection method based on the invariant characteristics of the CNC machining process. Our approach is divided into a detection rule generation module, a low-interference acquisition module, and an active detection engine [see Fig. 3].

### B. Intrusion Detection Rule

The machining process is a crucial basis for manufacturing and processing workpieces. The NC code contains the most complete and comprehensive machining process information, such as spindle speed, feed rate, and machining path. The CNC system interprets the NC code into executable instructions to control the various components of the CNC machine tool to complete the machining operations. Through the analysis of the NC code, the following intrusion detection rules can be generated: key parameter whitelist rules and machining process whitelist rules. The approach of generating detection rules based on code analysis demonstrates great applicability in the industrial control field [15], [28].

1) *Key Parameter Whitelist Rule:* In CNC systems, there are numerous important parameters that can affect the actual production process. The process of sending commands from the CNC system to control the hardware needs to be adjusted to the specific parameters. For instance, the CNC system adjusts the tool's landing position and movement trajectory during the actual machining process based on parameters, such as tool radius compensation and length compensation, or the CNC system controls the feed acceleration based on parameters related to acceleration and deceleration. These

TABLE I  
FANUC PARAMETERS MAPPING TABLE

Parameter Type	FOCAS Address Mapping	Data Type	Number of parameters
Tool Compensation Parameters	0x000800000001 -0x000800000190	Real	400
Macro Variables	0x001500000001 -0x0015000003e7	Real	633
CNC Parameter	0x008d00000001 -0x008d00006bd9	Bit(axis), Byte(axis), Word(axis), Real(axis)	27609
PMC Parameter	0x800100000000 00000000 -0x80010000bb7 00000009	Bit(axis), Byte(axis), Word(axis), Real(axis)	6572
All Parameters			35214

parameters directly impact the machining accuracy and stability of the machine tool. If the key parameters in the CNC system are maliciously tampered with by attackers, it can result in decreased machining accuracy or even machine tool failure. Typically, key parameters in CNC systems have specific values or value ranges. For instance, specific tool radius compensation and length compensation parameters have fixed values, and the control parameters for rapid feed acceleration and deceleration generally fall within the range of 140–160 ms. Therefore, we analyze the parameters of FANUC CNC in Table I, and establish whitelist rules for key parameters and their value ranges to monitor the correctness of the key parameters in the CNC system.

2) *Machining Process Whitelist Rule:* The International Organization for Standardization (ISO) has established ISO-6983-1 [29] as the international standard for CNC programming languages. This standard delineates the lexical and syntactic rules governing CNC codes, thereby forming a programming language comprising G-codes and M-codes. Numerous CNC system manufacturers have introduced CNC control products adhering to the ISO-6983-1 standard. For instance, SIEMENS CNC systems, such as SINUMERIK 802D and SINUMERIK 840D, as well as FANUC CNC systems like 0i-md and 0i-mf, support NC programming in compliance with this standard. While CNCs from various manufacturers or models may exhibit diverse representations of NC code programming, they share commonalities, and the discrepancies in syntax and programming concepts are essentially minimal. Consequently, leveraging our proposed detection scheme and considering these shared characteristics, we can customize the intrusion detection system to be applicable to different models of CNC systems.

The CNC system controls the machining process through the NC code. The spindle and multiple servo axes in the CNC system work in coordination to control the movement and rotation of the tool and workpiece, completing automated production machining. Therefore, we parse the NC code and extract the invariant relationships of tool motion trajectories, feed rates, spindle speeds, and other parameters for each step of the machining process flow to generate machining process rules. These rules serve as benchmarks for detecting malicious attacks, such as NC code tampering or parameter injection.

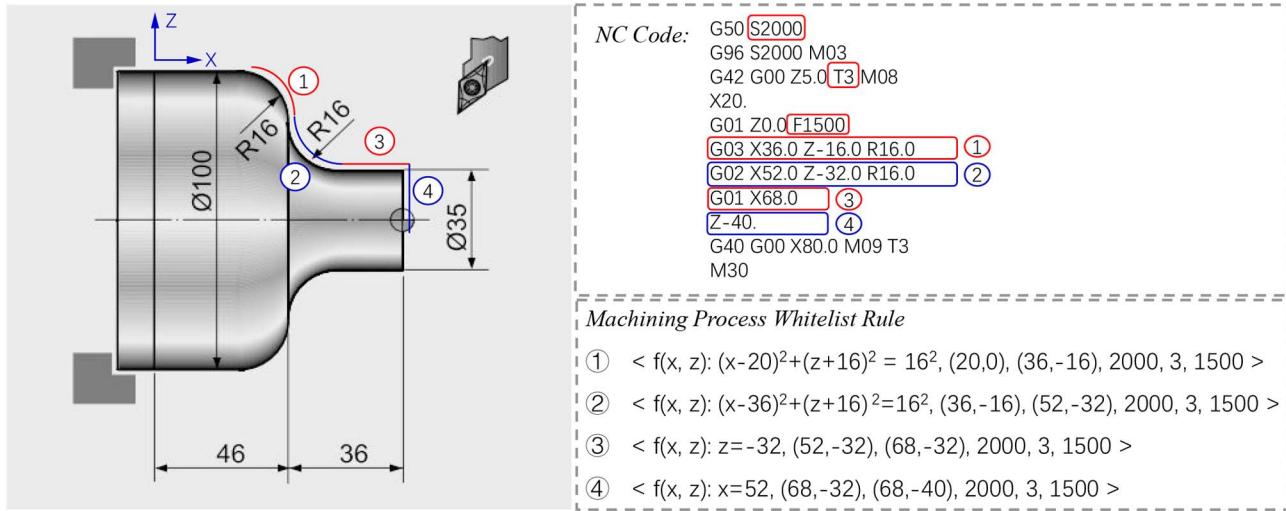


Fig. 4. Case of machining process whitelist rule.

Taking turning machining as an example [see Fig. 4], we demonstrate how to generate detection rules based on the invariance of the machining process using NC codes. The CNC system executes the NC codes to automatically control the machining process during turning. The NC codes specify the spindle speed, feed rate, and tool number for the machining process, and then use G codes to specify the tool's movement trajectory, such as common linear machining(*G01*) and circular machining(*G02*, *G03*). Therefore, we perform lexical, syntactic, and semantic analysis on the NC codes to generate the machining process rules, such as rule ①, which indicates that under the condition of spindle speed  $S = 2000$  and feed rate  $F = 1500$ , tool number 3 moves along the curve  $(x - 20)^2 + (z + 16)^2 = 16^2$ , with a starting point of  $(20, 0)$  and an ending point of  $(36, -16)$ .

### C. Low-Interference Acquisition

In order to collect the runtime status of the CNC, the conventional method is to use the communication interface provided by the CNC manufacturer. However, we found that the acquisition frequency of Focas, the communication interface provided by FANUC, is too low, which leads to an increase in the alarm delay for intrusion detection and affects the accuracy of the detection rule selection. For this reason, we manually reverse analyzed the protocol format of Focas and designed low-interference acquisition packets.

1) *FOCAS Protocol Reverse Engineering*: We capture mirror traffic on an industrial switch and conduct reverse protocol analysis on the proprietary protocol Focas for FANUC CNC systems. Focas protocol is an application-layer protocol based on TCP/IP. During the process of establishing a connection, the Focas protocol requires two rounds of TCP handshake to establish the connection. First, the client uses port A (any available port) to initiate a connection establishment request to port 8139 of the CNC system. Then, the client establishes a second connection to port 8193 of the CNC system using port A + 1 or A + 2. Subsequent request and response operations are performed on port A + 1 or A + 2. Reverse

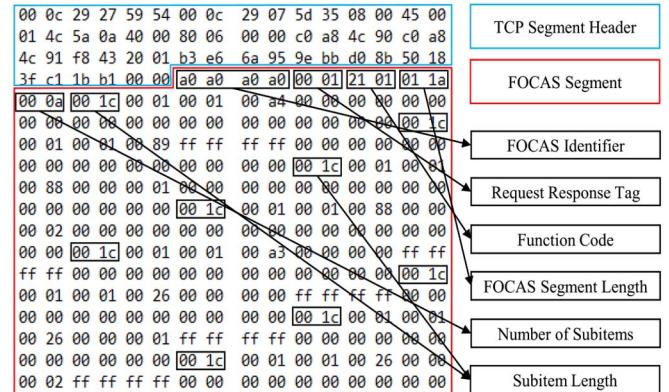


Fig. 5. FOCAS protocol reverse analysis results. This figure shows a binary request packet for collecting the current machining coordinates of a CNC using Focas, which consists of a Focas protocol header and ten subitems.

protocol analysis reveals the frame format of Focas protocol [see Fig. 5]. The first 4 bytes of the payload section are always a0a0a0a0, serving as the identification for Focas protocol. The 5th and 6th bytes represent the request/response flag. The Focas function code is located in bytes 7 and 8. The 9th and 10th bytes represent the length of the payload data. The 11th and 12th bytes indicate the number of subitems. The first 12 bytes form the header of the Focas protocol. The subitems in the Focas protocol include subitem length, fixed padding, and subfunction code. The payload information of the subitem includes the request parameter address and data format, which are not explicitly described in this article to prevent potential misuse by malicious individuals.

2) *Constructing Low-Interference Acquisition Packages*: Based on the results of the reverse protocol analysis mentioned above, we found that when using the API interface functions provided by Focas to collect position coordinates, spindle speed, and feed speed of CNC systems, multiple Focas request packets need to be sent to collect the machining status of the CNC system at the same moment. Moreover, these request packets usually contain irrelevant subitems that are unrelated

to attack detection. Under the high-frequency collection, these irrelevant subitems consume a significant amount of network and CNC system computing resources, which affects the CNC system and reduces detection efficiency. To solve this problem, we extracted the detection-related subitems from multiple request packets and combined them into a single request packet, which is then sent to the CNC system to collect multiple machining statuses at the same moment. Upon receiving the response packet, the status data of the CNC system is extracted based on the Focas protocol frame format.

The low-interference acquisition packages based on reverse engineering of the proprietary protocol greatly reduce the network overhead of status collection and minimize the interference on the CNC system. Furthermore, the analysis shows that the S7comm-nck protocol used by SINUMERIK 828, and 848 CNCs can also be used to construct low-interference request packets using the methods in this article. Detailed experimental data can be found in Section V-D.

#### D. Active Intrusion Detection Method

In this section, we outline the specific methods used for detecting attacks on manufacturing processes based on detection rules [see Fig. 3]. Our approach employs a low-interference, runtime active intrusion detection technique that does not disrupt the normal CNC machining process. During the implementation of this module, we have effectively addressed two key challenges.

- 1) Common phenomena, such as circuit instability, mechanical jitter, and equipment aging, can occur during the machining process. These issues can lead to inconsistencies in the CNC's execution time for each instruction. As a result, it becomes challenging to accurately and promptly match the collected runtime machining state to the detection rules.
- 2) The introduction of jitter and other interference due to regular errors, which can lead to an increased false alarm rate in the detection program, necessitating the need to distinguish between regular errors and cyber attacks.

1) *Selection Detection Rules*: To address challenge 1, We employed the dynamic time warping (DTW) algorithm and the  $k$ -nearest neighbors (KNNs) algorithm.

DTW is a dynamic programming algorithm that measures the similarity between time series [30], particularly those of varying lengths. It is commonly used in the fields of speech recognition, gesture recognition, and information retrieval due to its applicability to temporal data. In our study, we utilized the DTW algorithm to align a reference state sequence with a rule label to a captured runtime processing state sequence, with timestamps arranged in chronological order.

KNN is a nonparametric method used in supervised learning [31]. KNN is based on a simple and intuitive concept: if the majority of the  $k$ -most similar samples in the feature space of a given sample belong to a certain category, then the sample is also classified as belonging to that category. The algorithm makes its decision by considering only the category of the nearest one or more samples. In our study, we employed the KNN algorithm to classify runtime processing state points.

This allowed us to select the appropriate detection rules based on reference state sequences that were labeled with the rules.

2) *Consistency Checker-Based Detection Windows and Thresholds*: To address challenge 2, we implemented a detection window and alarm threshold in our approach. During the detection process, we collect the runtime state of continuous machining from the CNC machine tool for the duration of the window time and then accumulate the error between each runtime machining state and the machining process rules. An alarm is triggered when the accumulated error exceeds the threshold. If the window expires and the cumulative error does not exceed the threshold, the cumulative error is reset to 0 and a new inspection window is initiated. In this article, we employed (4) to conduct a consistency check, which involves accumulating the Euclidean distance between the runtime machining state and the machining process rule within the inspection window and comparing it with the inspection threshold. Specifically, as in (5), the actual error value is obtained by calculating the distance  $D$  between the actual position and the machining trajectory of the machining process rule, and the deviation of the actual feed rate  $F$  and spindle speed  $S$  from the machining process rule. Additionally, we performed dissimilarity verification between the runtime key parameter matrix  $\mathbb{C}_{3 \times n}$  and the key parameter rule  $\mathbb{K}_{3 \times n}$  to ensure key parameter consistency. Equation (3) serves as the theoretical foundation for our machining process consistency verification, enabling us to identify attacks, such as machining code injection, parameter injection, and instruction injection, on the CNC system during the machining process

$$\left\{ \sum_{t=1}^{W \text{ size}} \|y(t) - r(t)\| \leq \delta(t) \right\} \wedge \{\mathbb{C}_{3 \times n} \oplus \mathbb{K}_{3 \times n} = [0]_{3 \times n}\} \quad (4)$$

$$\|y(t) - r(t)\| = \sqrt{D^2 + (F - F_r)^2 + (S - S_r)^2}. \quad (5)$$

The active detection engine detects whether the CNC is under attack in an active and low-interference way, and its core part is shown in Algorithm 1. It first establishes a communication connection with the CNC (line 1); then parses the detection rules to simulate the machining path and constructs the active acquisition packet (lines 2–4) and then the attack is detected (lines 5–16); and, finally, when the detection is complete, the connection is disconnected (line 17). In the attack detection phase, the first step is to initialize the detection window and detection threshold (line 6). Next, a low-interference request packet is sent to the CNC, followed by receiving the response data and parsing the protocol to extract the processing state values (lines 7–10). Then, the DTW algorithm is used to match the detection rules for the data in the current detection window. Finally, a consistency check is done on the machining state and key parameters to see if the detection rules are satisfied (lines 12–14). Line 15 indicates the setting of the required frequency for the CNC to achieve low interference.

## V. EVALUATION

In this section, we focus on answering the following research question.

**Algorithm 1:** Algorithm of Active Detection

---

**Input :** Key Parameter Rule  
Machining Process Rule  
Detection Window Size  
**Output:** Cyber-Attack Alerts

```

1 Connect (cncIP, cncPORT);
2 RuleDB ← LoadRules (KeyParameter, MachiningProcess);
3 PathSim ← PathSimulation (RuleDB);
4 AcquisitionPKG ← PackageConstructor (RuleDB);
5 while ProcessFlag do
6   InitDetectionWindow();
7   for 0 to DetectionWindowSize do
8     Send (AcquisitionPKG);
9     ProcessingStatus ← Receive();
10  end
11  FlaggedStatus ← DTW (ProcessingStatus, PathSim);
12  if ConsistencyChecker (FlaggedStatus, RuleDB) is
13    False then
14    | Alarm ("Illegal Processes: cncIP, RuleNo.");
15  end
16  Sleep (t);
17 Disconnect ();

```

---

*RQ1:* What is the time cost of MPI-CNC to generate rules?

*RQ2:* What is the effectiveness of MPI-CNC detection against cyber attacks on CNC systems?

*RQ3:* Does MPI-CNC affect CNC machine tool machining efficiency?

We used 3589 lines of C code and 2492 lines of Python code to implement MPI-CNC. MPI-CNC is deployed on a ThinkPad P15Gen2 with an 8 cores Intel Core i9-11950H CPU and 64 GB of RAM. MPI-CNC was evaluated using Fanuc 0i-md CNC and Fanuc 0i-tf CNC.

#### A. Experimental Environment

*Experimental Design:* To answer the three research questions above, experiments were conducted using real machining environments with the FANUC CNC system. Due to the lack of real-world cyber-attack data for the FOCAS CNC system, three attack methods discussed in Section II-B were implemented, and the effects of the attacks were demonstrated on real equipment. First, multiple NC programs from realistic machining scenarios that are applicable to the FANUC CNC system were analyzed, and detection rules were generated to verify the accuracy of the automatic parsing of NC code for rule generation. Second, the proposed intrusion detection method was compared with other detection models to evaluate its performance in detecting network attacks in the NC machining process. Finally, in order to demonstrate that our solution has minimal impact on the CNC system's machining process, the variations in machining time were monitored during the detection phase, and the network resource utilization was compared between using low-interference data collection requests and using FOCAS standard interface for collecting CNC system's machining status.

*Experimental Environment:* In this work, we conducted experiments using a FANUC 0i-tf CNC system [Fig. 6(a)] and FANUC 0i-md CNC system [Fig. 6(b)], as shown in Fig. 6,

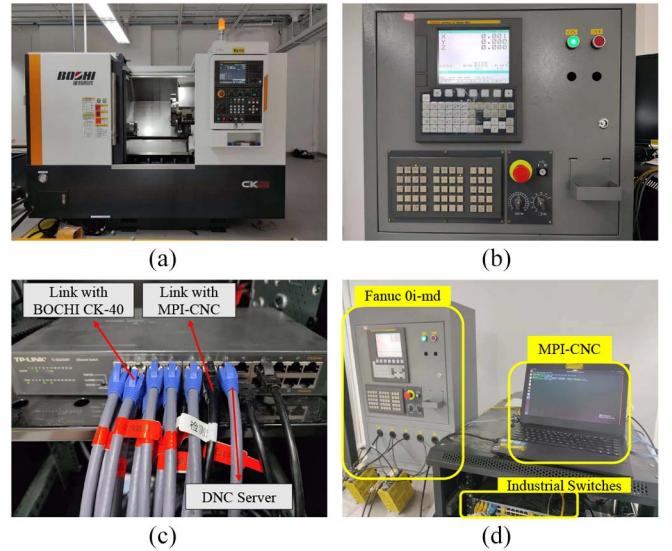


Fig. 6. MPI-CNC experimental environment. (a) BOCHI CK-40 with Fanuc 0i-tf. (b) Fanuc 0i-md. (c) Industrial switches. (d) MPI-CNC deployment environment.

We connect the MPI-CNC to the industrial switch connected to the CNC and configure its IP address to be in the same network segment as the other devices so that it can communicate with the CNC normally. Fig. 6(c) partly shows the connection status of BOCHI CK-40, DNC server, and MPI-CNC to each port of the industrial switch. Fig. 6(d) shows the site layout of the Fanuc 0i-md intrusion detection experimental environment, including the CNC, MPI-CNC, and industrial switch.

*Cyber-Attack Setting:* Specifically, we discuss three attack methods against CNCs in this work, which were implemented on FANUC CNCs due to the lack of available attacks for evaluation purposes. The first attack method is the machining code injection attack, which involves injecting malicious machining instructions and machining paths into the NC code. We introduced 20 tamperings in 10 different machining codes, including creep attacks and trajectory scaling, to evaluate the detection capability of the methods. The second attack method is the remote parameter injection attack, which was implemented by tampering with key CNC parameters through request packets sent to the FANUC CNC based on the Focas protocol inversion results. The third attack method is the malicious command injection attack, which involves tampering with the designated ports of the PMC by sending request packets to the FANUC CNC based on the Focas protocol inversion results and the CNC's interface manual. This allows for malicious commands, such as remote start/stop and on/off coolant, to be injected.

#### B. RQ1—Time Cost and Accuracy of Generating Detection Rules

We collected 57 NC codes applicable to FANUC CNC systems from real machining scenarios and Internet platforms, such as Github and Traceparts. These codes involve turning and milling processes and consist of 8354 instructions, including instructions for linear machining, circular arc machining,

TABLE II  
TIME COST OF GENERATING DETECTION RULES

NC Code	Code Lines	Number of Rules	Times Cost(ms)
O5665-NC	134	90	1.004
O6383	150	93	0.805
NCViewer.nc	5780	5753	0.962
NCtest26.NC	64	61	0.768
7190.3-1A.nc	255	222	0.792
...	...	...	...
Number of NC Code: 57	Total Code Lines: 8354	Total Rules: 7671	Total Time Cost: 53.579ms

tool changing operations, coolant control, and more. By analyzing the machining instructions in these NC programs, we extracted attack detection rules. Unlike other high-level programming languages, NC programs are not as complex, typically consisting of multiple G codes and M codes. We used the number of G codes that control the machining trajectory to represent the size of the program and generated several key parameter rules based on the relationship between G codes and M codes, as well as one machining process rule per G code.

First, we analyzed the FANUC CNC system user manual and interface manual and combined the results of reverse engineering the FOCAS proprietary protocol to establish a mapping table of G codes, M codes, and system parameter addresses. Then, based on the semantic information of G and M codes in the NC codes, we selected important parameters related to machining and generated key parameter rules. Next, we parsed the G codes that control the machining trajectory, abstracted the curve equation of the machining trajectory based on its semantic information, and combined information, such as the starting point and ending point of the machining, spindle speed, feed rate, and tool number to generate machining process rules.

As shown in Table II, we analyzed all the collected NC codes and recorded all the generated key parameter rules and machining process rules. The 57 NC codes we collected totaled 8354 lines of machining instructions. For these collected 57 NC codes, key parameters related to machining are identified and a total of 6056 machining process rules are generated. In the process of generating inspection rules, we recorded the number of lines of machining code, the number of inspection rules generated, and the time cost of generating the rules for each NC code. The time taken to analyze the generation of inspection rules for a single NC code is 0.94 ms on average, and the time taken for a single inspection rule is 0.007 ms [Table II]. To verify the correctness of the generated rules, we selected 10 representative NC codes and manually verified the accuracy of the automatically generated detection rules using all the key parameter rules and machining process rules. The results showed that the accuracy of the detection rules in a limited number of NC program samples was 100%.

*Answer RQ1:* The proposed method in this article automatically analyzes NC code and generates comprehensive detection rules without relying on historical manufacturing data. Each NC code takes 0.94 ms to generate accurate inspection rules.

TABLE III  
COMPARING THE ACCURACY AND TIME COST OF DTW AND KNN ALGORITHMS IN SELECTING DETECTION RULES

Algorithms	KNN				DTW
	k=3	k=5	k=7	kd-tree	
Accuracy %	96.95	97.82	96.83	97.32	99.38
Times Cost(ms)	273	321	326	137	352

The time cost of the method in this article is extremely low and can be quickly used in new machining scenarios.

### C. RQ2—CPMS Cyber-Attack Detection Results

In Section II-B, we provide a comprehensive review of the current state-of-the-art research on attacks against manufacturing processes and a summary of three common attack methods. It is worth noting that publicly available CPMS attack methods or attack data sets are typically tailored to specific machining scenarios, devices, and processing processes, and there are currently no generic CPMS attacks. Therefore, in this work, we evaluate the detection effectiveness of the MPI-CNC by implementing three attack methods.

1) *Selection Detection Rules Results:* We conducted experiments to evaluate the accuracy and time cost of KNN and DTW algorithms in rule selection using 25 283 data points from real machining scenarios. The results are shown in Table III. We tested the performance of the KNN algorithm by setting different values of k in the rule selection experiments and using a kd-tree data structure. The test results showed that when k was set to 5, the rule selection accuracy was 97.82%, which was the optimal parameter for the KNN algorithm in rule selection. It is worth noting that the use of a kd-tree data structure greatly reduced the time cost, with rule selection for 25 283 data points taking only 137 ms. This makes it suitable for complex machining scenarios and real-time detection requirements. When using the DTW algorithm for rule selection, the selection accuracy was as high as 99.38% and the time cost was 352 ms, which falls within an acceptable range and meets real-time alarm requirements. In summary, to improve detection accuracy, MPI-CNC adopts the DTW algorithm as its rule selection algorithm. For complex machining scenarios with high-real-time detection requirements, the KNN algorithm based on a kd-tree data structure should be used.

2) *Cyber-Attack Detection Results:* We conducted experiments to evaluate the performance of different detection windows in attack detection, and the results are presented in Table IV. The active detection engine successfully detected the machining code injection attack, the key parameter injection attack, and the malicious instruction injection attack. These attacks interfere with the normal machining process and result in changes to the machining trajectory and machining state, which can be directly reflected in the machining process and the key parameters of the CNC system. The active detection engine actively communicates with the FANUC CNC to map its actual machining status and key parameters. This active detection approach makes it difficult for the attacks to be

TABLE IV  
COMPARISON OF CYBER-ATTACK DETECTION RESULTS IN DIFFERENT DETECTION WINDOWS AND DETECTION THRESHOLD CASES

Window size /Threshold	Detection accuracy	False alarm rate	Missing alarm rate	Alarm delay(s)
10/0.1	99.15%	2.89%	1.83%	1.45
<b>50/0.5</b>	<b>98.81%</b>	<b>1.09%</b>	<b>2.42%</b>	<b>2.45</b>
100/1	98.68%	0.75%	6.17%	3.71
200/2	96.88%	2.61%	7.15%	6.32
500/5	94.87%	4.44%	6.25%	13.75

hidden, as it requires the attacker to gain insight into the real machining process and manipulate the CNC firmware, tamper with the network communication module, or employ other sophisticated methods to feedback network data that conforms to the machining process rules and key parameter rules.

As shown in Table IV, we set different detection window sizes of 10, 50, 100, 200, and 500 in the active detection experiments. We set the alarm threshold in millimeters based on the machining accuracy of the CNC machine tool of 0.01 mm multiplied by the window size. The test results show that the detection accuracy of the method proposed in this article is 99.15% with a detection window of 10, and the accuracy decreases with the increase of the detection window, down to 94.87%. The reason for this phenomenon is that as the detection window increases, the detection threshold increases, increasing the missing alarm rate and a decrease in detection accuracy. In Table IV, the missing alarm rate is 1.83% when the detection window is 10. The larger the detection window, the larger the missing alarm rate, and when the detection window is 500, the missing alarm rate is 6.25%. It is worth noting that the false alarm rate becomes larger when the detection window is too large and too small. The alarm delay increases as the detection window increases, mainly because the DTW algorithm takes more time to match more data. Considering the above, we conclude that the optimal detection window size of this method is 50, the detection accuracy is 98.81%, the false alarm rate is 1.09%, the missing alarm rate is 2.42%, and the alarm delay is 2.45 s.

3) *Comparison With Other CPMS IDS*: In order to demonstrate the effectiveness of our detection scheme, MPI-CNC was compared to its performance with other CPMS ICS models. Specifically, we compared with representative models that are commonly used for detecting manufacturing process attacks, namely, digital twin-based intrusion detection [7], side-channel analysis-based intrusion detection (KCAD [4], LTDT [3], LSTM-AE [6]), and machining code analysis-based intrusion detection [8]. The digital twin-based intrusion detection models require historical processing state data for fitting digital twin models. KCAD and LSTM-AE collect audio data generated by manufacturing equipment during processing to learn anomaly classification models. LTDT analysis of processing video classification anomalies. The machining code analysis-based intrusion detection extracts features of NC codes to train SVM models for offline classification of anomaly codes.

We launched 200 machining code injection attacks against 10 different machining scenarios (machining codes), each introducing 20 tampering points (such as replacement of G02

TABLE V  
COMPARISON WITH OTHER CPMS IDS

Attack Type	Machining Code Injection	Parameter Injection	Instruction Injection
Digital Twins[7]	N/A	93.25%	93.25%
KCAD[4]	81.39%	81.39%	N/A
LTDT[3]	95.55%	95.55%	N/A
LSTM-AE[6]	94.79%	94.79%	N/A
Machining Code Analysis[8]	98.38%	N/A	N/A
<b>MPI-CNC</b>	<b>98.81%</b>	<b>100.00%</b>	<b>100.00%</b>

and G03 with G01; insertion of protrusions or depressions; and modification of endpoints to cause deformation). In addition, 100 parameter injection attacks and command injection attacks were used to test the detection performance of the MPI-CNC. Detection results and attack logs are collected and used to calculate detection accuracy, miss rate and false alarm rate of MPI-CNC, as in

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (6)$$

$$\text{Missing Alarm} = \frac{\text{FN}}{\text{TP} + \text{FN}} \quad (7)$$

$$\text{False Alarm} = \frac{\text{FP}}{\text{TN} + \text{FP}}. \quad (8)$$

The methods have achieved high-detection accuracy using current state-of-the-art techniques, as shown in Table V. The detection accuracy of the digital twin-based detection method against parameter injection attacks and instruction injection attacks is 93.25% [7]. The detection accuracy of the KCAD against processing code tampering and instruction injection attacks is 81.39% [4]. In addition, the LTDT and LSTM-AE models have high-detection accuracies of 95.55% [3] and 94.79% [6]. However, the side channel data features behave differently in different processing scenarios, which makes it difficult to apply to new scenarios quickly. The code analysis-based intrusion detection directly analyzes processing code for offline detection with 98.38% detection accuracy [8], but it lacks runtime detection capability during processing. Compared with the above methods, MPI-CNC can cope with a wider range of attack scenarios. Since we analyze the key parameters of the CNC system and actively detect the key parameter information during the machining process, we can detect parameter injection attacks and instruction injection attacks by 100%. Also, the method in this article analyzes the NC code to generate comprehensive detection rules, so it has a higher detection accuracy similar to the machining code analysis method, with a detection accuracy of 98.81%.

*Answer RQ2:* Experiments have proved that MPI-CNC can cope with three attack methods, which is more comprehensive than the traditional detection model based on historical data. Moreover, MPI-CNC is significantly better than other detection methods in terms of accuracy of 98.81%.

#### D. RQ3—Low-Interference Experimental Results

The computational performance of CNC systems is not as high as that of traditional PCs. Therefore, we need to be cautious about whether active detection methods will affect the normal operation of NC systems. In this article, we adopt a low-interference, polling-based approach to collect the

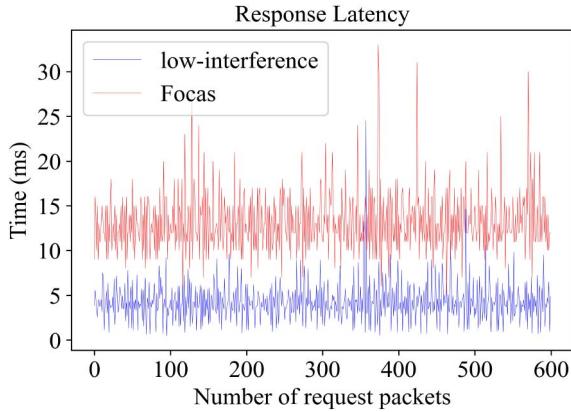


Fig. 7. Comparison of response latency for low interference and Focas.

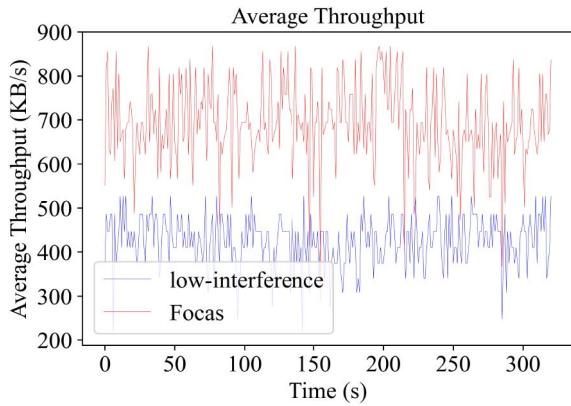


Fig. 8. Comparison of average throughput for low interference and Focas.

machining status of NC systems, and then process the data and detect attack behaviors remotely. In this section, we compare the acquisition delay, network resource utilization, and impact on machine tool processing times between the low-interference collection method proposed in this article and the FOCAS standard interface for collecting the machining status of CNC systems.

*1) Acquisition Latency and Network Throughput:* As shown in Fig. 7, the average response time for a single collection using the conventional FOCAS collection method is 13.240 ms, with a collection frequency of 75.53 times per second. Using the low-interference collection method proposed in this article, the average response time for a single collection is 4.368 ms, with a collection frequency of 228.94 times per second. The response delay is reduced by 67.00%, and the sampling frequency is increased by 203.12%. Meanwhile, in Fig. 8 the average throughput of the CNC system using the conventional FOCAS collection method is 692.04 KB/s, while the average throughput of the NC system using the low-interference collection method proposed in this article is 426.23 KB/s, resulting in a decrease in network throughput of 38.41%. Experiments have shown that using the self-assembled packet method based on private protocol reverse engineering proposed in this article for collecting the machining status of CNC systems reduces interference to the CNC systems while improving the collection frequency.

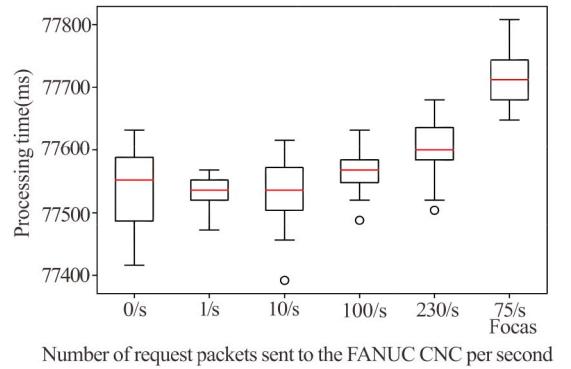


Fig. 9. Influence of network traffic on FANUC CNC.

*2) Processing Times Under CPMS IDS:* In order to demonstrate that our detection scheme has minimal impact on the machining time of CNC systems, we conducted experiments with different request frequencies during the active detection process to observe changes in machining time. To avoid the forwarding delay of industrial switches, we directly connected the CNC system with Ethernet cables and sent 1, 10, 100, and 230 low-interference request packets per second, as well as 75 Focas standard interface request packets per second. As shown in Fig. 9, under normal conditions without any external interference, the machining time of the CNC system was approximately 1 min and 17.545 s. When the CNC system was subjected to varying degrees of external interference, we found that its machining time was minimally affected. In the case of sending 230 packets per second, the machining time of the CNC system increased by only 0.076% and no packet loss was observed. This indicates that the FANUC CNC system has the capability to process at least 230 packets per second without affecting its normal operation. However, when using Focas standard interface request packets with a maximum rate of 75 requests per second, the machining time of the CNC system increased by 0.217%, which was significantly higher than the low-interference data collection method used in this article. Moreover, the FANUC system can handle a maximum of 75 requests per second. The experiment results show that our detection scheme has minimal impact on the machining time of CNC systems.

*Answer RQ3:* MPI-CNC significantly improves acquisition frequency and efficiency by reverse engineering dedicated acquisition protocols and customized packet acquisition, reducing CNC network resource usage and increasing machining time by only 0.076%. MPI-CNC does not affect normal machining.

## VI. RELATED WORK

Cyber-attacks against CPMS directly affect production efficiency and even threaten the safety of human life. Therefore, IDS research in the field of CPMS has become an academic hotspot. Cyber-attacks on CPMS mainly focus on controlling and disrupting the manufacturing process, which is the key concern of CPMS IDS. By analyzing research results in this

field over the past few years, we have classified the state-of-the-art CPMS IDS into three categories based on detection methods.

**CPMS IDS Based on Digital Twins:** Digital twin technology utilizes historical data to fit a control model that simulates physical processes [32]. Balta et al. [7] collected and analyzed historical machining data from a 3-D printer to construct controller digital twin models for the 3-D printer's CNC system. By comparing the consistency between the simulated machining state of the controller digital twin model and the actual machining state, they detected cyber-attacks that tampered with the temperature parameters of the 3-D printer's nozzle heaters.

**CPMS IDS Based on Side-Channel Analysis:** Manufacturing equipment generates a large amount of measurement channel data during the machining process, which can indirectly reflect the machining state. Detection methods based on side-channel analysis are a popular approach in the CPMS IDS field. Chhetri et al. [4] proposed for the first time the use of audio data around manufacturing equipment to train detection models for detecting machining path tampering attacks. Bayens et al. [33] combined analysis of acoustic features of machining equipment, machining location features, and production waste features to verify product consistency. Belikovetsky et al. [5] analyzed audio data from 3-D printer stepper motors and evaluated the similarity between their audio features and audio fingerprints to detect the 3-D printing process. Mamun et al. [3] detecting 3-D printer processing trajectory changes using video stream analysis. Yoginath et al. [2] analyzed the current values of 3-D printer power lines using the Bayesian model to detect creep attacks. Shi et al. [6] extracted features from side-channel data collected by vibration sensors based on the LSTM-autoencoder algorithm and later used the OCSVM classification algorithm for anomaly detection.

**CPMS IDS Based on Machining Code Analysis:** The ISO has developed the ISO-6983-1 [29] standard as an international standard for NC programming languages. This standard specifies the lexical and syntax rules of NC code, forming a programming language composed of G codes and M codes. NC code contains the most complete and comprehensive control information of the machining process, and the CNC system automatically controls the machining process according to the instructions in the NC code. By analyzing the NC code, anomalies can be effectively detected. Beckwith et al. [8] extracted statistical features from NC code, including the number of G codes and M codes, as well as the frequency of XYZ values. They trained a machine learning anomaly classification model and conducted an offline analysis of NC code to identify anomalies. Tsoutsos et al. [34] reverse-engineered NC code to generate 3-D models, and then simulated pressure tests on these models. They discovered vulnerabilities in the NC code during this process.

## VII. DISCUSSION

The intrusion detection method proposed in this article has the following limitations.

- 1) The method may have difficulty in dealing with man-in-the-middle attacks implemented through tampering with the firmware of the CNC system. Such attacks require high-technical skills from the attackers and can effectively bypass the intrusion detection method proposed in this article.
- 2) The method is effective for application in 2-axis and 3-axis CNC machines, but it may not be able to generate detection rules specifically for 5-axis machine centers.
- 3) The active detection approach proposed in this article may not be applicable to CNC systems with interface authentication mechanisms. However, it should be noted that currently, most CNC systems do not restrict remote access to machining status information.
- 4) Low-interference acquisition methods can be applied to CNCs from different vendors, but protocol reversal demands high-technical skill. Automated protocol reversal is a meaningful task that needs to be addressed.

## VIII. CONCLUSION

This article proposes a novel approach for runtime detection of CPMS cyber-attacks, denoted as MPI-CNC. We implement a prototype system on the FANUC CNC machine tools as an example. Specifically, MPI-CNC automatically analyzes NC programs, extracts machining process invariants, and generates attack detection rules, including machining process rules and key parameter rules. Then, using low-interference request packets, MPI-CNC actively communicates with the CNC system to collect process status and key parameters, while setting detection windows and thresholds to detect attack behaviors. In the end, we evaluate MPI-CNC in real machining scenarios using a FANUC CNC machine tool. Experimental results demonstrate that MPI-CNC exhibits excellent adaptability performance, being able to accurately detect various cyber-attacks without affecting the normal operation of the CNC system. Compared with other state-of-the-art detection models, our approach shows superior adaptability performance and detection performance.

## REFERENCES

- [1] A. Kusiak, "Smart manufacturing," *Int. J. Prod. Res.*, vol. 56, nos. 1-2, pp. 508–517, 2018.
- [2] S. Yoginath et al., "Stealthy Cyber anomaly detection on large noisy multi-material 3-D printer datasets using probabilistic models," in *Proc. ACM CCS Workshop Addit. Manuf. Secur.*, New York, NY, USA, 2022, pp. 25–38.
- [3] A. A. Mamun, C. Liu, C. Kan, and W. Tian, "Securing cyber-physical additive manufacturing systems by in-situ process authentication using streamline video analysis," *J. Manuf. Syst.*, vol. 62, pp. 429–440, Jan. 2022.
- [4] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, "KCAD: Kinetic Cyber-attack detection method for cyber-physical additive manufacturing systems," in *Proc. IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, 2016, pp. 1–8.
- [5] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici, "Digital audio signature for 3-D printing integrity," *IEEE Trans. Inf. Forensics Security*, vol. 14, pp. 1127–1141, 2019.
- [6] Z. Shi, A. A. Mamun, C. Kan, W. Tian, and C. Liu, "An LSTM-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing," *J. Intell. Manuf.*, vol. 34, no. 4, pp. 1815–1831, Apr. 2023.

- [7] E. C. Balta, M. Pease, J. Moyne, K. Barton, and D. M. Tilbury, "Digital twin-based cyber-attack detection framework for cyber–physical manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, early access, May 25, 2023, doi: 10.1109/TASE.2023.3243147.
- [8] C. Beckwith et al., "Needle in a haystack: Detecting subtle malicious edits to additive manufacturing G-code files," *IEEE Embed. Syst. Lett.*, vol. 14, no. 3, pp. 111–114, Sep. 2022.
- [9] H. Pearce, K. Yanamandra, N. Gupta, and R. Karri, "FLAW3D: A trojan-based cyber attack on the physical outcomes of additive manufacturing," *IEEE/ASME Trans. Mechatron.*, vol. 27, no. 6, pp. 5361–5370, Dec. 2022.
- [10] M. Yampolskiy, L. Graves, J. Gatlin, J. T. McDonald, and M. Yung, "Crypto-steganographic validity for additive manufacturing (3D printing) design files," in *Proc. Int. Conf. Inf. Secur.*, 2022, pp. 40–52.
- [11] T. Le et al., "Physical logic bombs in 3-D printers via emerging 4-D techniques," in *Proc. 37th Annu. Comput. Security Appl. Conf.*, New York, NY, USA, 2021, pp. 732–747.
- [12] M. McCormack, S. Chandrasekaran, G. Liu, T. Yu, S. DeVincent Wolf, and V. Sekar, "Security analysis of networked 3-D printers," in *Proc. IEEE Security Privacy Workshops (SPW)*, 2020, pp. 118–125.
- [13] B. Shadow, "Industrial security exploitation framework." 2020. [Online]. Available: <https://github.com/w3h/isf>
- [14] R. R. Maiti, C. H. Yoong, V. R. Palletti, A. Silva, and C. M. Poskitt, "Mitigating adversarial attacks on data-driven invariant checkers for cyber–physical systems," *IEEE Trans. Depend. Secure Comput.*, vol. 20, no. 4, pp. 3378–3391, Jul./Aug. 2023.
- [15] J. Liu et al., "ShadowPLCs: A novel scheme for remote detection of industrial process control attacks," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 3, pp. 2054–2069, Jun. 2022.
- [16] P. Mahesh et al., "A survey of cybersecurity of digital manufacturing," *Proc. IEEE*, vol. 109, no. 4, pp. 495–516, Apr. 2021.
- [17] Y. Pan et al., "Taxonomies for reasoning about cyber–physical attacks in IoT-based manufacturing systems," *Int. J. Interact. Multimedia Artif. Intell.*, vol. 4, no. 3, pp. 45–54, Jul. 2017.
- [18] T. Lin et al., "Susceptibility to spear-Phishing emails: Effects of Internet user demographics and email content," *ACM Trans. Comput. Human Interact.*, vol. 26, no. 5, pp. 1–28, Jul. 2019.
- [19] N. Karsten and L. Jakob, "BadUSB—On accessories that turn evil," presented at Blackhat Conf., Las Vegas, NV, USA, 2014, pp. 1–28.
- [20] M. Conti, N. Dragoni, and V. Lesyk, "A survey of man in the middle attacks," *IEEE Commun. Surveys Tuts.*, vol. 18, no. 3, pp. 2027–2051, Mar. 2016.
- [21] S. B. Moore, W. B. Glisson, and M. Yampolskiy, "Implications of malicious 3-D printer firmware," in *Proc. 50th Hawaii Int. Conf. Syst. Sci.*, 2017, pp. 1–10.
- [22] Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, "Review on cyber–physical systems," *IEEE/CAA J. Automatica Sinica*, vol. 4, no. 1, pp. 27–40, Jan. 2017.
- [23] G. Y. Dayanikli, S. Sinha, D. Muniraj, R. M. Gerdes, M. Farhood, and M. Mina, "Physical-layer attacks against pulse width modulation-controlled actuators," in *Proc. 31st USENIX Secur. Symp.*, Boston, MA, USA, 2022, pp. 953–970.
- [24] J. Gatlin et al., "Encryption is futile: Reconstructing 3D-printed models using the power side-channel," in *Proc. 24th Int. Symp. Res. Attacks, Intrusions Defenses*, New York, NY, USA, 2021, pp. 135–147.
- [25] L. J. Wells, J. A. Camelio, C. B. Williams, and J. White, "Cyber–physical security challenges in manufacturing systems," *Manuf. Lett.*, vol. 2, no. 2, pp. 74–77, 2014.
- [26] T. Zinner, G. Parker, N. Shamsaei, W. King, and M. Yampolskiy, "Spooky manufacturing: Probabilistic sabotage attack in metal AM using shielding gas flow control," in *Proc. ACM CCS Workshop Additive Manuf. (3D Printing) Security*, New York, NY, USA, 2022, pp. 15–24.
- [27] S. R. Chhetri, A. Barua, S. Faezi, F. Regazzoni, A. Canedo, and M. A. Al Faruque, "Tool of spies: Leaking your IP by altering the 3-D printer compiler," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 2, pp. 667–678, Apr. 2021.
- [28] H. Choi et al., "Detecting attacks against robotic vehicles: A control invariant approach," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, New York, NY, USA, 2018, pp. 801–816.
- [29] *Automation Systems and Integration—Numerical Control of Machines—Program Format and Definitions of Address Words—Part-1: Data Format for Positioning, Line Motion and Contouring Control Systems*, International Organization for Standardization, ISO Standard 6983-1:2009, 2009.
- [30] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 26, no. 1, pp. 43–49, Feb. 1978.
- [31] E. Fix and J. L. Hodges, "Discriminatory analysis. nonparametric discrimination: Consistency properties," *Int. Statist. Rev./Revue Internationale de Statistique*, vol. 57, no. 3, pp. 238–247, 1989.
- [32] R. Minerva, G. M. Lee, and N. Crespi, "Digital twin in the IoT context: A survey on technical features, scenarios, and architectural models," *Proc. IEEE*, vol. 108, no. 10, pp. 1785–1824, Oct. 2020.
- [33] C. Bayens, T. Le, L. Garcia, R. Beyah, M. Javanmard, and S. Zonouz, "See no evil, hear no evil, feel no evil, print no evil? malicious fill patterns detection in additive manufacturing," in *Proc. 26th USENIX Secur. Symp.*, Vancouver, BC, Canada, 2017, pp. 1181–1198.
- [34] N. G. Tsoutsos, H. Gamil, and M. Maniatakos, "Secure 3-D printing: Reconstructing and validating solid geometries using Toolpath reverse engineering," in *Proc. 3rd ACM Workshop Cyber Phys. Syst. Secur.*, New York, NY, USA, 2017, pp. 15–20.

**Zedong Li** is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

His research interests include cyber–physical manufacturing systems security and intrusion detection.

**Xin Chen** (Member, IEEE) is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

His research interests include cyber–physical manufacturing systems security and intrusion detection.

**Yuqi Chen** received the B.Sc. degree in computer science from South China University of Technology, Guangzhou, China, in 2015, and the Ph.D. degree from Singapore University of Technology and Design, Singapore, in 2019.

He is an Assistant Professor with the School of Information Science and Technology, ShanghaiTech University, Shanghai, China. Before joining ShanghaiTech, he was a Research Scientist with the System Analysis and Verification Group, Singapore Management University, Singapore. He employs a range of techniques, including testing, reverse engineering, program analysis, and formal methods, to develop practical solutions for securing critical cyber–physical systems. His research interests lie at the intersection of software engineering and security.

**Shijie Li** is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

His research interests include industrial control system security and intrusion detection.

**Hangyu Wang** is currently pursuing the Ph.D. degree with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China.

His research interests include industrial control system security and access control.

**Shichao Lv** received the M.S. degree in cryptography from the University of Electronic Science and Technology of China, Chengdu, China, in 2012, and the Ph.D. degree in information security from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is a Ph.D. Professorate Senior Engineer and an M.S. Supervisor from the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His main research interests include Internet of Things security and industrial control system security.

**Limin Sun** received the Ph.D. degree from the National University of Defense Technology, Changsha, China.

He is currently a Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China. His main research interests include Internet of Things security and industrial control system security.

Prof. Sun is also the Secretary General of the Select Committee of CWSN and the Director of the Beijing Key Laboratory of IoT Information Security Technology. He is an Editor of the *Journal of Computer Science* and the *Journal of Computer Applications*.