www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

A Machining Process Invariant Approach to Cyber-Attack Detection in Cyber-Physical Manufacturing Systems

[1] Mr.K.Lakshmi Narayana,[2] Sai Kiran Yenneti,

[1] Assistant Professor,Dept.of Master of Computer Applications, Rajamahendri Institute of Engineering & Technology, Bhoopalapatnam, Near Pidimgoyyi,Rajahmundry,E.G.Dist.A.P. 533107.

[2] Students,Dept.of Master of Computer Applications, Rajamahendri Institute of Engineering & Technology, Bhoopalapatnam, Near Pidimgoyyi,Rajahmundry,E.G.Dist.A.P. 533107.

Abstract—

With the advent of the Industrial Internet of Things (IIoT), cyber-physical manufacturing systems (CPMSs) have become an increasingly serious threat due to cyber-attacks. Intrusion detection for content management systems is now a rapidly developing topic. Nevertheless, the present approaches incur high expenses in gathering past data for the purpose of training detection models that are customized to individual machining situations. Their flexibility is put to the test by the ever-changing nature of real-world machining settings. According to this article, a comprehensive machining process is included in the CPMS machining code, providing a solid foundation for detection. In light of this, we provide MPI-CNC, an intrusion detection method that relies on the machining code's Machining Process Invariant. For crucial detection rules, MPI-CNC automates the examination of the machining codes to extract machining process rules and key parameter rules. Then, in order to identify cyber-attack activity, MPI-CNC actively gets the runtime status from the CPMS and compares it with the detection criteria. Over 10 real-world machining situations, MPI-CNC was tested with two FANUC computer numerical control (CNC) machines. In this experiment, we saw that MPI-CNC is quite versatile. In addition, while maintaining regular machining processes, MPI-CNC demonstrated better accuracy in identifying cyber-attacks on CPMS when compared to current state-of-the-art detection approaches. Cyber assault, cyber-physical manufacturing systems, intrusion detection, industrial internet of things, computer numerical control (CNC) are all terms that may be found in an index.

INTRODUCTION

An essential lynchpin of contemporary industrialization is the manufacturing sector. Towards networked and intelligent development, the global manufacturing industry is quickly progressing with the introduction of the Industrial Internet of Things and intelligent manufacturing [1]. Cyber-physical manufacturing systems (CPMSs) rely on computer numerical control (CNC) systems to regulate the machining operations of production machinery. The aviation, automotive, and military industries are just a few of the significant ones that employ CNC equipment extensively. The Gigafactory, built by Tesla, automates and optimizes manufacturing by connecting CNC equipment to the industrial Internet. With more and more manufacturers connecting their CNC systems to the Industrial Internet, ensuring the security of CPMS is crucial, but it is challenging. Methods for detecting intrusions in CPMS have recently grown in popularity. At now, the majority of CPMS security researchers are devoted on teaching ML classification models to spot outliers in side channel data, which may be anything from current[2] and video[3] to audio[4,5], and 6] produced by machining. In order to identify cyber-attacks, some researchers have constructed data-driven digital twin models of CNC systems and used them to undertake consistency checks on the runtime state of the systems [7]. An alternative method that uses machine learning anomaly classification models and extracts digital properties from machining codes is an offline way to detecting tampering with these codes [8]. With these technologies, abnormal processing behaviors may be efficiently detected in certain machining scenarios. Unfortunately, hackers have been able to easily launch cyber-attacks by taking advantage of vulnerabilities in CNC systems, such as the absence of authentication mechanisms, plain-text transmission, and unfixed vulnerabilities. This is all because manufacturers of CNC systems did not adequately consider security. In order to cause processing errors in the product, attackers mostly aim at the machining code. For example, they may insert a Trojan into the CNC system's software, covertly alter the machining code that is input into the system, and carry out a harmful hole attack [9]. Attackers have also shown that they may use steganography to compromise machining code files in network communication, which weakens the final output mechanically [10]. The second kind of attack involves the hacker accessing the CNC system's memory and changing important settings. For example, they insert smart materials inside gas masks to plant physical logic bombs, swap out the processing material by

manipulating settings. Because of this, gas masks can break and let gas out while used [11]. Beyond that, criminals have made use of pre-existing open-source technologies like as C3PO[12] and Industrial To deliver malicious instructions to CNC systems and interrupt processing, one can use the Security Exploitation Framework (ISF) [13]. These incidents highlight the critical requirement of intrusion detection systems for CNC systems in order to properly combat these types of attacks. Driving force: The CNC system uses a wide variety of machining codes to process a wide variety of goods in real-world production processes. The audio, picture, current, and voltage side-channel properties are determined by the unique tool paths, raw materials, and machining equipment needed for each of these cases. Researchers usually have to spend a lot of time and energy gathering side-channel data for each new machining situation and starting the training process all over again in order to build intrusion detection models that work for multiple scenarios utilizing side-channel data. The various machining scenarios are readily disruptible once an attacker successfully deploys an attack script in a CPMS system. Therefore, a flexible intrusion detection solution inside the CPMS system is urgently required to successfully defeat current attack tactics in a range of machining scenarios. Observation: Invariant rule-based detection is now a common approach in industrial control security for successfully detecting cyber-attack abnormalities. Industrial control devices often use the invariant control logic embedded in control logic codes to regulate the typical operation of industrial systems. For the purpose of intrusion detection in industrial control systems, researchers have employed data-driven [14]and code-driven [15]approaches to derive control logical invariant principles. They have got great results with detection. Our discovery in the field of CPMS is based on the invariant rule-based detection. We discovered that the machining process of the CNC system is invariant, and the machining code provides extensive information on the machining process invariant. Consequently, by examining the machining code, the invariant rules for the full machining process may be retrieved. Key components of the machining process variables, such as machining speed and machining trajectory, provide a solid foundation for intrusion detection and allow for a thorough description of the CNC system's machining process. A approach based on Machining Process Invariant (MPI-CNC) is proposed in this article as a solution to the problem of the CPMS intrusion detection system's poor adaptive capabilities. Detection criteria may be quickly and easily extracted from machining code using MPI-CNC. As a first step in identifying assaults, the approach parses the machining code to get important machining-related information such as tool routes, machining sequences, spindle speeds, and more. Afterwards, MPI-CNC actively gathers critical parameters and realtime machining status from the CNC system while milling. In the end, MPI-CNC uses detection criteria to check if the runtime machining data is consistent in order to detect cyber-attacks. Conclusion: A FANUCCNC-based prototype was created to confirm the approach's practicability. Using actual CNC machines, we tested out ten different machining situations and three different attack vectors. strategies, and assessed the costs, deployment times, detection efficiencies, and potential interference with the CNC system. The results of the experiments proved that MPI-CNC can be easily implemented into new machining scenarios without preprocessing and can reliably identify cyber threats while running, without impacting the CNC's normal operation. Among the most cutting-edge detection methods, MPI-CNC outperforms the competition. Machine tool code injection attack has a detection accuracy of 98.81% and parameter injection attack of 100%, whilst other approaches get the best detection results of 93.25% and 98.38%, respectively. What follows is an outline of the article's contributions. 1) Our algorithm for automatically extracting detection criteria from machine codes is new and innovative. Rapid rule generation for various machining conditions enhances the versatility of CPMS intrusion detection. 2) We created low-interference acquisition request packets that follow the structure of the FOCAS protocol and performed an examination of its usage in FANUC CNC systems. With this method, data collecting becomes more efficient with less disturbance to the machining process. Thirdly, the FANUC CNC system served as the basis for the development of a prototype CPMS IDS. It is possible to expand and adapt this prototype to different CNC systems and protocols, even though it was designed for a specific CNC system based on this. 4) In three assault scenarios and ten real-world machining scenarios, we tested our suggested approach's flexibility, detection, and overall performance. Using this method to identify cyber-attacks precisely in runtime without impacting the regular functioning of the CNC, experiments demonstrate that it can be implemented rapidly to new machining settings without preprocessing. Strategic Plan: What follows is the outline for the rest of the article. The technological basis of CPMS is briefly introduced in Section II, while the MPI-CNC is described in Section III. Part IV describes the MPI-CNC and how it is put into practice. In Section V, we discuss the experimental evaluation. The relevant work of existing CPMS intrusion detection algorithms is introduced in Section VI. In Section VII, we look at the MPI-CNC's shortcomings. The final section is Section VIII.

BACKGROUND

This article is primarily dedicated to proposing an intrusion detection method for CPMS. This chapter serves to providean overview of the research background, focusing on two essentialaspects:1)thecompositionand2)machiningprocess of CPMS, as well as the various forms of attacks encountered by CPMS.

Cyber–PhysicalManufacturingSystems

TheCPMSgenerallyconsistsofanengineeringstation, a distributed numeric control (DNC) server, a machine data collection (MDC) server, and manufacturing equipment con- nectedthroughanindustrialswitch[seeFig.1(a)].The

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

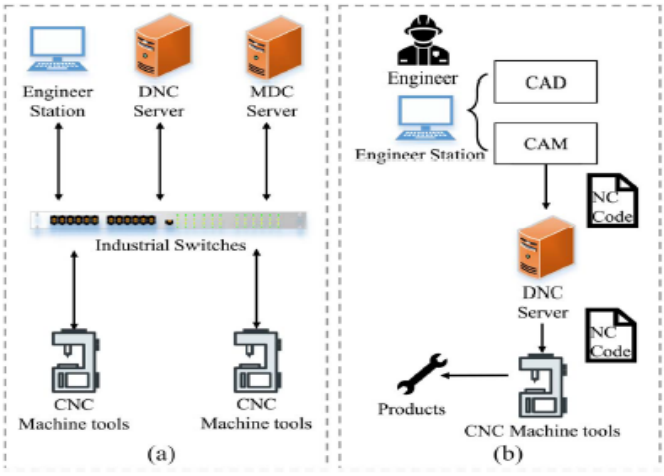


Fig. 1. Network topology and processing process of a CPMS. (a) Network topology of CPMS. (b) Processing process of CPMS.

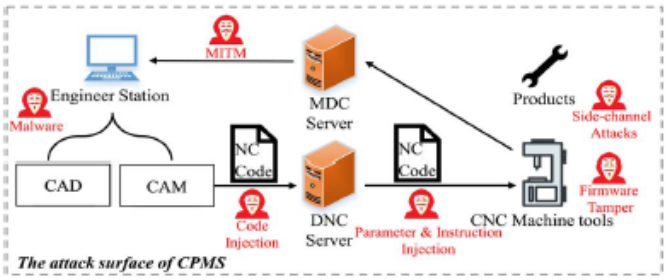


Fig. 2. Attack surface of CPMS.

In most cases, an engineering station will be an office PC that has CAD and CAM software installed. Figure 1(b) depicts the processing procedure. The machining code, also known as NC code, is created by engineers in engineering design software and subsequently sent to the DNC server. The DNC server is responsible for distributing the NC code to the machinery that needs it during production. By reading the NCcode, the CNC system automatically controls the machining process. Position, speed, temperature, and other data are among the many states that the manufacturing equipment provides to the MDCserver through its interactions with the server. So that engineers can keep tabs on the machining process, this data is sent back to the engineering station's monitoring program. Engineers also have the option to do runtime operations during milling by sending control commands. Model B. Attack Cyberattacks on CPMS have recently been considered and classified by academics [16], [17]. Using a production-process viewpoint (see Fig. 2), this research investigates recent cyber-attacks on CPMS and analyzes the threat surface. The engineer's workstation is linked to a local area network (LAN) or potentially the Internet during production. The engineer station is vulnerable to attacks such as spear phishing [18], BadUSB [19], and others that use vectors conveying harmful malware that steals and tampers with CAD models and NCcodes, as well as software and system vulnerabilities. Authentication, encryption, and other security features are commonly missing from the communication protocols used by devices like DNC servers and engineer stations when they connect to the CNC system over Ethernet. For instance, DNC servers can send NC code in clear text via the FTP protocol. By inserting harmful instructions and parameters into replayed packets, attackers can execute man-in-the-middle attacks [20], modify or steal NC code from network traffic, or both. Additionally, the firmware of the CNC system is vulnerable to manipulation by the attacker, who can disrupt regular operations [9], [21]. It takes an in-depth familiarity with the CNC system's code structure to launch such an assault, though, and it's more challenging. Since CPMS is a typical cyber-physical system, there are a number of ways that attackers can disrupt normal processing [23] or achieve a steganographic attack [24] by obtaining leaked physical information, or by using side-channel attack methods like electrical measurement interference and acoustic resonance. Three groups were established to classify the attacks: 1) Injection of machine code; 2) Injection of parameters; and 3)

Index in Cosmos
MAY 2025, Volume 15, ISSUE 2
UGC Approved Journal

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Injection of instructions.Injection of Machining Code: Modifying or altering the CNC system's machining code, also known as the NC code, is known as a MachiningCodeinjectionattack [9], [10], [25]. Attackers can interrupt and disrupt the CNC machining process by changing important code segments, such as the machining route, spindlespeed, or auxiliarycontrolcode. Modifying the CNC system's parameters is known as parameter injection [11], [26]. Spindle speed ratio, quick feed rate, and alarm shielding are just a few of the numerous critical CNC system factors that impact the machining process. As a result, machining precision and the CNC machine itself are both compromised if attackers get access to these critical parameters and interfere with the CNC system. The term "Instruction Injection" describes the practice of interfering with a CNC system's regular machining process by delivering harmful control commands to it. C3PO, developed and released as open-source software by McCormack et al.[12], scans 3D printers' network services for security holes and exploits those holes to launch attacks against CNC systems managed remotely. Additionally, attackers can interrupt production by delivering attack scripts and injecting malicious commands through the ISF [13]. Physical cross-domain attacks and side-channel information leakage are additional security risks that CNC systems encounter [27]. But these dangers aren't covered in this essay, so keep that in mind. Reasons such as their low risk, infeasibility, or vulnerability to detection by current IDSs prevent their inclusion.

OVERVIEW

An unique intrusion detection approach for CPMS, called MPI-CNC, is proposed in this paper. Presented in

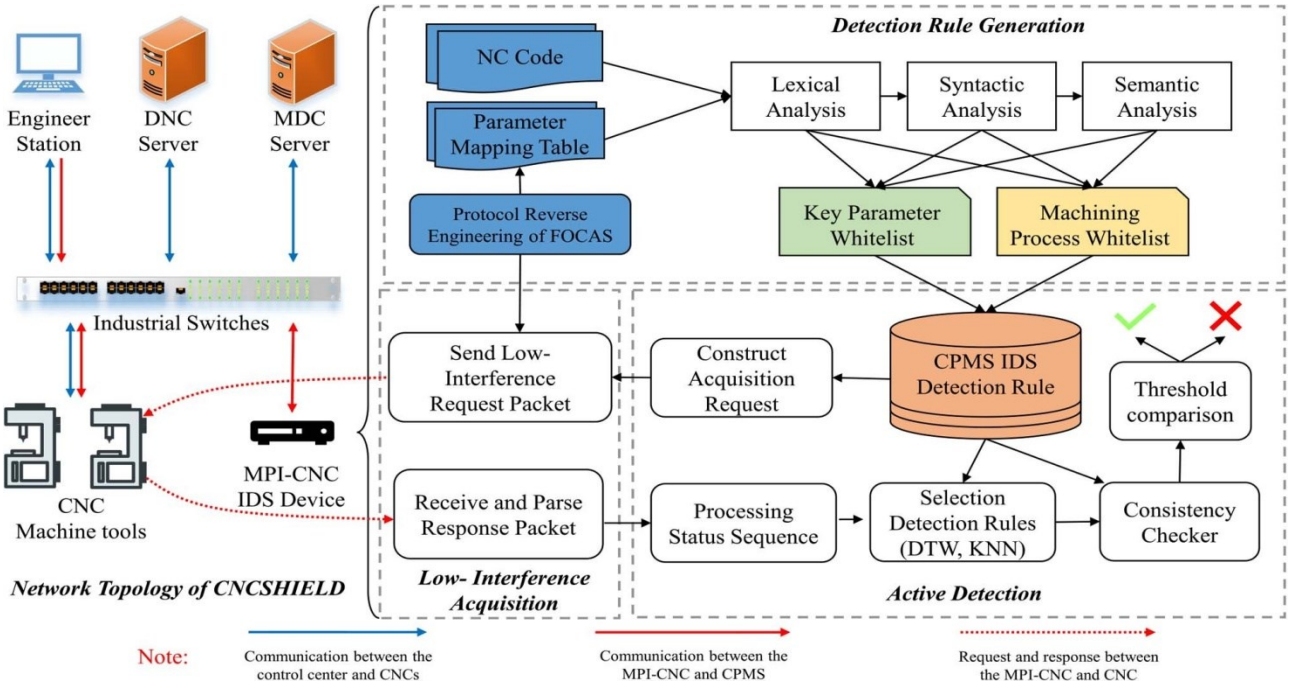


Fig.3.SystematicapproachtobuildMPI-CNC.

Figure 3 shows the three separate steps that make up this method: 1) generating detection rules, 2) acquiring low-interference states, and 3) actively detecting. Here we give a brief outline of the basic structure for evaluating intrusion detection technologies. The Creation of Detection Rules: Using static analysis, the detection rule generation module parses the NC code and extracts detection rules. Complete machining operations, including rules for critical parameters and the machining process itself, are encapsulated in the NC code. By using the key parameter rules, crucial parameters inside the CNC system may be monitored and verified to make sure they are accurate, stable, and protected from malicious interference that could cause machine malfunctions or reduced cutting precision. In contrast, machining process rules identify malicious attempts such program replacement or tampering with machining trajectories by examining the NC code's machining processes' invariant properties and therefore establishing reference recommendations. The NC code's derived detection rules provide an exhaustive picture of the machining process and allow for the proactive identification of network assaults and unforeseen abnormalities. The CNC system's low-interference state acquisition module gathers machining states while the machine is

Index in Cosmos
MAY 2025, Volume 15, ISSUE 2
UGC Approved Journal

running. In order to keep tabs on how well a CNC system is running, most manufacturers include development kits or software for monitoring. As an example, secondary development is made easier by the FANUC Focas 1/2 development component. It communicates actively with the CNC system, allowing for remote monitoring during runtime. Important details including NC programs, tool locations, and spindle velocities are supplied by the development component. Nevertheless, using the original development kit for

Normal machining operations and real-time performance might be significantly affected by an increase in the network load of the CNC system caused by high-frequency data collecting. Our research uses reverse engineering to examine proprietary protocols in order to get around this problem. Furthermore, we personalize data gathering requests while removing unnecessary ones. Consequently, we are able to get runtime machining states within the CNC system at high frequencies with minimal interference. Detection in Action: Anomalies in the machining process are mainly detected by the active detection module. This module analyzes the data gathered by the low-interference state collection module regarding the machining state during runtime and then provides alerts. The CNC system's machining status is checked to see if it follows the regulations for key parameters and machining process. Malicious instruction assaults, code tampering, and key parameter manipulation are all prevented by this strategy. Error threshold and monitoring window size are two crucial monitoring parameters that must be determined during the active detection stage. We start by trying out different monitoring window widths based on the state acquisition frequency to find the one that works best. Then, for certain window widths, we determine the total normal error for every monitoring window and use the maximum observed error to define the error threshold.

APPROACH

Problem Statement

The computer numerical control (CNC) system controls the relative movements of the tool and the workpiece based on input from various sensors in a complicated industrial control process known as CNC production. u(t) is defined in (1) to represent the CNC's machining status at time t in this article. in where P(x,y,z) denotes the tool's x, y, and z coordinates, S(t) denotes the spindle speed, F(t) denotes the feed rate, and T(t) is the current tool number

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

The machining code indicates the machining process, which is the constant characteristic of the CNC machining process. To explain this process, we define (2) with r(n). Particularly, the machining path curve equation—typically straightlines and circular arcs—is represented by F(x,y,z). The machining route begins at Start(x,y,z) and ends at End(x,y,z). The machining process may be fully described by combining the cutting route, spindle speed, feed rate, and tool number.

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

We use *M()* in (3)to describe the CNC machining model, where *ε(t)* represents the internal losses of the CNC and reasonable errors due to natural factors

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

Errors ε(t) in the CNC machining state are realistic in a typical machining process because of factors such as machine wear and tear, current and voltage jitter, and others. Cyberattacks on CNCs, however, might cause them to violate the present machining process r(n) and veer far from the CNC machining model M(). Hence, we developed an intrusion detection approach in this work that relies on the CNC machining process's invariant characteristics. Our method is structured as follows: an active detection engine, a low-interference acquisition module, and a module for generating detection rules [see Fig.3]. Rule for Detecting Intrusions (B) The foundation of producing and processing workpieces is the machining process. Details about the machining process, including the spindle speed, feedrate, and cutting route, are best found in the NC code. As the machining processes progress, the CNC system converts the NCcode into executable instructions that drive the CNC machine tool's many components. Key parameter whitelist rules and machining process whitelist rules can be established by the examination of the NC code. Industrial control is a promising area for the method of creating detection rules from code analysis [15], [28]. rule for key parameters: Numerous critical factors in CNC systems influence the real manufacturing process. There has to be fine-tuning of the procedure for the parameters in order for the CNC system to control the hardware. For example, the CNC system takes into account parameters like tool radius compensation and length compensation when adjusting the tool's landing position and movement trajectory during the machining process. It also uses acceleration and deceleration parameters to control the feed acceleration.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

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 -0x800100000bb7 00000009	Bit(axis), Byte(axis), Word(axis), Real(axis)	6572
All Parameters			35214

These factors have an immediate effect on the machine tool's stability and precision during machining. Machine tool failure or poor machining precision could occur if attackers intentionally alter the critical settings in the CNC system. Important parameters in CNC systems often have predetermined values or ranges of values. For example, the control parameters for quick feed acceleration and deceleration typically lie within the range of 140-160 ms, and the particular tool radius compensation and length compensation parameters have set values. Hence, we examine the FANUC CNC parameters shown in Table I and set up whitelist rules for important parameters along with their ranges of values so that we can check if the CNC system is running correctly. Rule No. 2 for the Machining Process Checklist: The worldwide standard for CNC programming languages is ISO-6983-1 [29], which was produced by the worldwide Organization for Standardization (ISO). A programming language consisting of G-codes and M-codes is formed by outlining the lexical and syntactic rules regulating CNC codes in this standard. The ISO-6983-1 standard has been adopted by a number of CNC control product manufacturers. Some examples of CNC systems that allow NC programming in accordance with this standard include SIEMENS's SINUMERIK 802D and SINUMERIK840D, and FANUCCNC's 0i-md and 0i-mf. Although CNCs from different brands or models may display different NCcode programming representations, they all share similarities and there are only minor differences in syntax and programming ideas. Therefore, we may adapt the intrusion detection system to various CNC system architectures by utilizing our suggested detection technique and taking these commonalities into account. By use of the NCcode, the CNC system directs the machining operation. To complete automated production machining, the spindle and several servoaxes in the CNC system coordinate the motion and rotation of the tool and the workpiece. To create machining process rules, we first analyze the NC code and identify the constant correlations between tool motion trajectories, feed rates, spindle speeds, and other parameters at each stage of the process flow. Malicious attempts, such as parameter injection or manipulation with the NC code, can be detected using these criteria as benchmarks.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

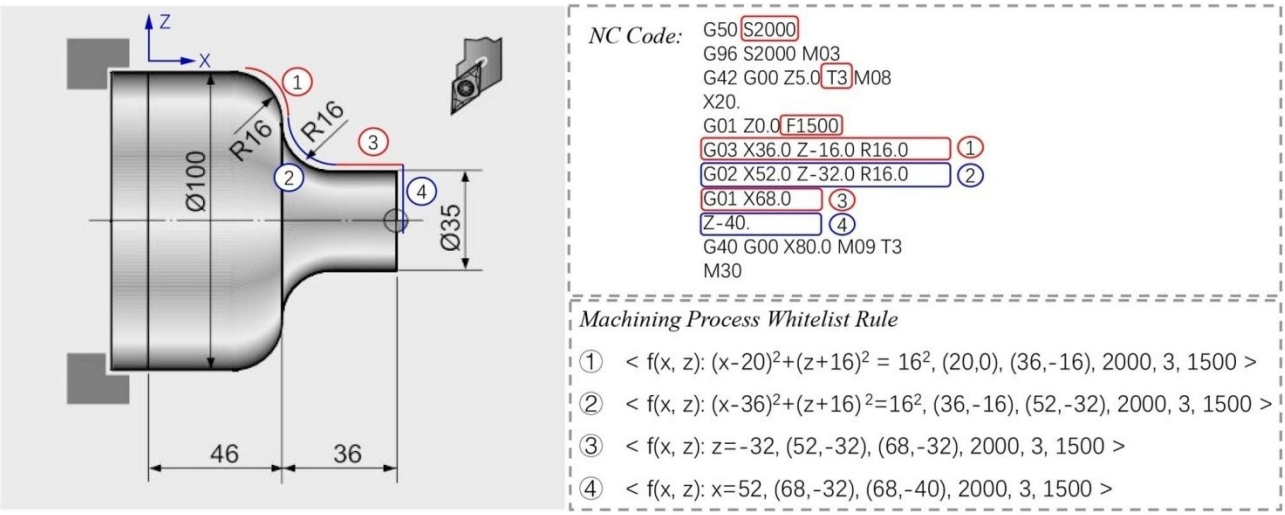
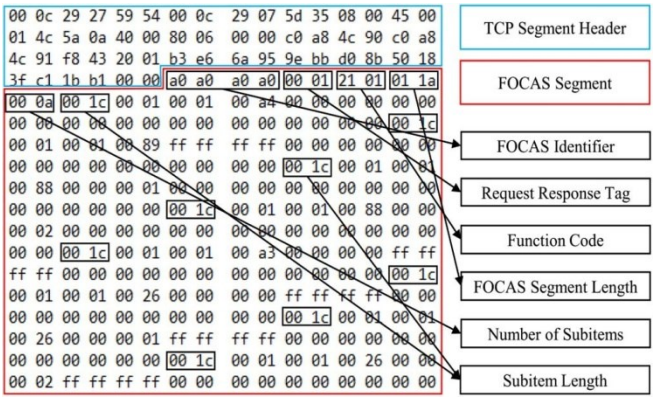


Fig.4.Caseofmachiningprocesswhitelistrule.

By utilizing NC codes, we show how to develop detection rules based on the invariance of the machining process, as demonstrated in turning machining (see Fig. 4). In order to automatically regulate the machining process during turning, the NC codes are executed by the CNC system. For machining, the NC codes define the spindle speed, feed rate, and tool number; for example, G01 is for typical linear machining, while G02 and G03 are for circular machining. The tool's movement trajectory is then specified using G codes. Thus, we conduct lexical, syntactic, and semantic analysis on the NC codes in order to produce rules for the machining process. For example, rule Ï1 states that when the spindle speed S is 2000 and the feed rate F is 1500, tool number 3 travels along the curve (x−20)2+(z +16)2= 162, beginning at (20,0) and ending at (36,−16). The Acquisition of Low-Interference The traditional approach to gathering CNC runtime status data is to make advantage of the manufacturer-provided communication interface. But we discovered that FANUC's communication interface, Focas, has an acquisition frequency that is too low. This causes the intrusion detection alarm latency to grow and impacts the accuracy of the detection rule selection. This is why we built low-interference acquisition packets by hand-analyzing the Focas protocol format. Our team does reverse protocol analysis on the proprietary protocol Focas for FANUC CNC systems by capturing mirror traffic on an industrial switch. Focas is an application-layer protocol that uses the TCP/IP protocol suite. Two rounds of the Transmission Control Protocol (TCP) handshake are necessary for the Focas protocol to establish a connection. To begin, the client sends a connection setup request to port 8139 of the CNC system using port A (or another accessible port). Afterwards, the client uses either port A + 1 or port A + 2 to create a second connection to port 8193 of the CNC system. Ports A+1 and A+2 are used for further request and response activities. Turn it backwards



Findings from the FOCAS protocol's reverse analysis (Fig. 5). You can see a Focas protocol header and 10 subitems in this picture, which make up a binary request packet for gathering CNC current machining coordinates using Focas. The format of the Focas protocol's frames

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

is revealed through protocol analysis [see Fig. 5]. As the Focas protocol identifier, the first four bytes of the payload segment are always a0a0a0a0. Bytes 5 and 6 stand for the request/response flag. You may find the code for the Focas function between bytes 7 and 8. The length of the payload data is represented by the 9th and 10th bytes. Number of subitems is indicated by the 11th and 12th bytes. The first twelve bytes are the Focas protocol's header. The Focas protocol includes the following subitems: subfunction code, subitem length, and fixed padding. The data format and request parameter address are part of the item's payload information, which is not disclosed in this article to avoid possible abuse by bad actors. Making Low-Interference Acquisition Packets: According to the results of the reverse protocol analysis mentioned earlier, we discovered that in order to gather the CNC system's machining status, position coordinates, spindle speed, and feed speed using the API interface functions provided by Focas, it is necessary to send multiple Focas request packets simultaneously. In addition, these request packets often include unrelated and irrelevant subitems.

to identify attacks. These unimportant subitems use up a lot of resources in the network and CNC system during high-frequency collection, which impacts the CNC system and decreases detection efficiency. We solved this problem by merging all the request packets containing detection-related information into one. Then, we transmitted it to the CNC system to gather various machining statuses at the same time. The CNC system's status data is derived from the response packet using the format of the Focas protocol frames. The network overhead of status collecting is considerably reduced and interference on the CNC system is minimized by using low-interference acquisition packages that are based on reverse engineering of the proprietary protocol. In addition, the research demonstrates that the S7comm-nck protocol might be utilized to create low-interference request packets using the methods described in this article, just like SINUMERIK 828 and 848 CNCs. Sections V–D contain the detailed experimental data. Detecting Active Intruders Using detection criteria [see Fig. 3], we describe the precise procedures utilized to identify assaults on production processes in this section. Using a runtime active intrusion detection technology with little interference, our solution ensures that the normal CNC machining process is not disrupted. Two major obstacles were successfully overcome throughout the execution of this module. Circuit instability, mechanical jitter, and equipment aging are common phenomena that can occur throughout Thanks to the reference state sequences that were tagged with the rules, we were able to choose the proper detection rules. Windows and Thresholds for Consistency Checker-Based Detection: Our strategy included a detection window and an alert threshold to deal with difficulty 2. As part of the detection procedure, we obtain the CNC machine tool's runtime state for continuous machining throughout the window period, and then we add up the discrepancy between each runtime machining state and the machining process regulations. When the total mistake goes beyond a certain point, an alert is sent out. When the timer runs out and the total error is less than the threshold, the total error is reset to zero and a fresh inspection window is started. Here, we used (4) to perform a consistency check, which entails measuring the inspection window's accumulation of the Euclidean distance between the runtime machining state and the machining process rule and comparing it to the inspection threshold. To be more precise, the real error value is determined by subtracting the actual feed rate F and spindle speed S from the machining process rule, as well as the distance D between the actual location and the machining trajectory of the process rule (as per equation (5)). Also, to make sure the key parameters were consistent, we checked for dissimilarities between the runtime key parameter matrix C3×n and the key parameter rule K3×n. For our machining process consistency verification, which allows us to spot assaults on the CNC system during machining, such as machining code injection, parameter injection, and instruction injection, we use Equation (3) as our theoretical foundation. machined parts. The inconsistent execution time of the CNC during eachin construction might be caused by several difficulties. Unfortunately, this makes it very difficult to precisely and

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

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

2)Determining the difference between routine mistakes and cyber assaults is important because normal errors can introduce jitter and other interference, which can raise the false alarm rate in the detection algorithm. 1) Rules for Detecting Selections: Task 1 was tackled by utilizing the dynamic temporal warping (DTW) technique in conjunction with the k-nearest neighbors (KNNs) approach. DTW is a method for dynamic programming that compares time series, especially ones with different durations, to find out how similar they are [30]. Its suitability to temporal data makes it a popular choice in speech recognition, gesture recognition, and information retrieval. Our investigation involved using the DTWalgorithm to align a reference state sequence with a rule label to a recorded runtime processing state sequence with chronologically ordered timestamps. When it comes to supervised learning, KNN is a nonparametric

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

approach [31]. KNN operates on a straightforward and basic principle: a sample is categorized as belonging to a specific category if the majority of the k-most comparable samples in its feature space also belong to that category. The algorithm just takes the category of there are stone or more samples into account while making its choice. As part of our research, we used the KNN algorithm to categorize processing state points during runtime. Algorithm 1 shows the essential components of the active detection engine, which determines if the CNC is being attacked in a low-interference and active manner. The program begins by connecting to the CNC (line 1), then uses the rules for detection to model the machining path and build the active acquisition packet (lines 2-4). Lines 5-16 indicate the detection of an attack, and line 17 indicates the completion of the detection process. Setting up the detection window and threshold is the initial step in the assault detection phase (line 6). Sending a low-interference request packet to the CNC is the next step, after which you get the response data and extract the processing state values by parsing the protocol (lines 7–10). The current detection window's data is compared to the detection criteria using the DTWalgorithm. Lastly, lines 12–14 are used to check if the detection rules are satisfied by doing a consistency check on the machining state and key parameters. The frequency that the CNC has to be tuned to ensure minimum interference is shown on line 15.

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
   |   False then
13 |   |   Alarm("Illegal Processes: cncIP, RuleNo.");
14 |   end
15 |   Sleep(t);
16 end
17 Disconnect();
```

Design of Experiment: The FANUC CNC system was used in real-world machining settings to solve the three study objectives mentioned above. The FOCAS CNC system was subjected to three attack techniques described in Section II-B, and their effects were shown on genuine equipment, in order to address the paucity of real-world cyber-attack data. To ensure the automatic parsing of NC code for rule creation is accurate, we first examined several NC programs from real-world machining scenarios that are appropriate to the FANUC CNC system. Then, we created detection rules. As a second step, we tested the suggested intrusion detection approach in NC machining against other detection models to see how well it worked. Last but not least, we compared the network resource utilization between collecting CNC system machining status through the standard interface and low-interference data collection requests to show that our solution has minimal impact on the machining process. We also monitored variations in machining time during the detection phase.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

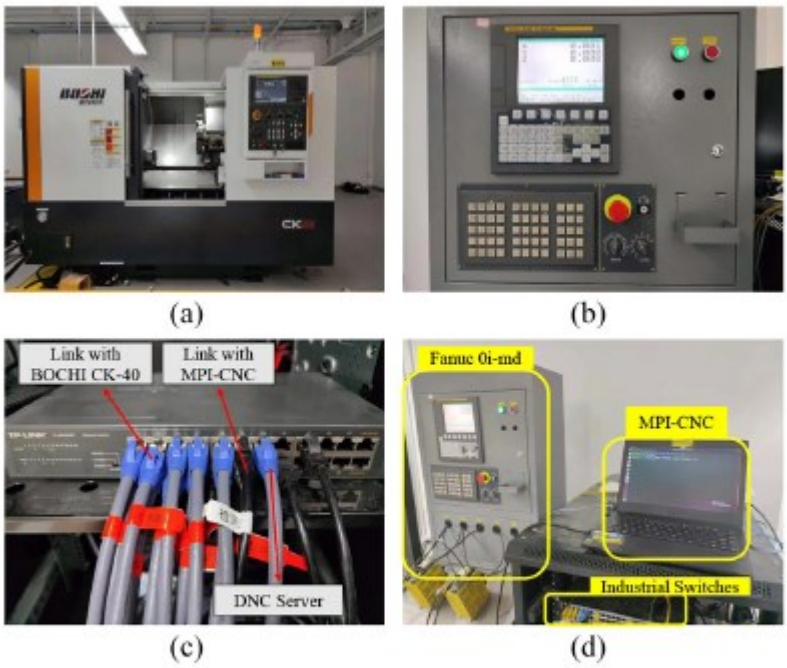


Fig. 6. MPI-CNC experimental environment. (a) BOCHI CK-40 with Fanuc 0i-tf. (b) Fanuc 0i-md. (c) Industrial switches. (d) MPI-CNC deployment environment.

To ensure regular communication between the MPI-CNC and the CNC, we install it into the same network segment as the other devices and link it to the industrial switch that is linked to the CNC. Figure 6(c) illustrates a partial view of the industrial switch's port state as it pertains to BOCHI CK-40, DNC server, and MPI-CNC. You can see the CNC, MPI-CNC, and industrial switch in Fig. 6(d), which also illustrates the site configuration of the Fanuc 0i-md intrusion detection testing environment. Situation of Cyber-Attack: In this study, we address three attack methods against CNCs. These approaches were applied to FANUC CNCs because there were no existing assaults that could be used for assessment. Machining code injection is the initial attack approach. It entails inserting malicious machining instructions and pathways into the NC code.

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	Totle Time Cost: 53.579ms

To test the approaches' detection capabilities, we created 20 tamperings in 10 distinct machining codes. These tamperings included creep assaults and trajectory scaling, among others. Second, using the findings of the Focas protocol inversion, an attacker might execute a remote parameter injection attack by manipulating critical CNC parameters in request packets provided to the FANUC CNC. Bad command injection is the third kind of attack. It entails manipulating the PMC's assigned ports and sending request packets to the FANUC CNC using the results of the Focas protocol inversion and the CNC's interface manual. This opens the door for the injection of harmful directives like remote start/stop and on/off coolant.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

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

In Section II-B, we summarize three typical attack techniques and present a thorough analysis of the present cutting-edge research on assaults against industrial processes.
It should be mentioned that there are no general CPMS assaults accessible at the moment; instead, the publicly available attack data sets and methodologies are usually designed for use with specific devices, processing situations, and machining scenarios. Since this is an evaluation of the MPI-CNC's detection efficacy, we employ three attack techniques to test it.
1) Rules for Detecting Selection End result: To compare the efficiency and precision of KNN and DTW algorithms for rule selection, we ran trials with 25,283 data points derived from actual machining situations. Table III displays the outcomes.

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
50/0.5	**98.81%**	**1.09%**	**2.42%**	**2.45**
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

In the rule selection tests, we used a kd-tree data structure and varied the values of k to evaluate the KNN algorithm's performance. The results of the test demonstrated that the ideal value for the k parameter in the KNN algorithm for rule selection was 5, with a result of 97.82% accuracy. Importantly, selecting rules for 25,283 data points took just 137 ms when a kd-tree data structure was used, significantly reducing the time cost. This means it can handle real-time detection needs and complicated machining settings with ease. Rule selection using the DTW algorithm achieved an accuracy of 99.38% with a time cost of 352 ms, which is acceptable and satisfies the requirements for real-time alarms. To summarize, the DTW method is used as the rule selection algorithm by MPI-CNC to enhance detection accuracy. Using the KNN algorithm with a kd-tree data structure is the way to go for complicated machining scenarios that demand high-real-time detection.

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
MPI-CNC	**98.81%**	**100.00%**	**100.00%**

Result Set for Cyber-Attack Detection: Table IV displays the findings of our research that compared the efficacy of various detection windows in assault detection. The three types of injection attacks—machining code, key parameters, and malicious instructions—were all

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

identified by the active detection engine. The machining process and the critical parameters of the CNC system can be directly affected by these assaults, which disrupt the normal machining process and alter the machining trajectory and state. In order to create a map of the FANUC CNC's real machining state and critical parameters, the active detection engine constantly communicates with it.

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

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

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

According to Table V, the approaches have successfully attained high detection accuracy by utilizing the most recent state-of-the-art methodologies. With a detection accuracy of 93.25 percent, the digital twin-based detection system is able to ward off assaults that inject parameters or instructions [7]. When it comes to detecting attacks including instruction injection and processing code modification, the KCAD has an accuracy of 81.39% [4]. Further, with high-detection accuracies of 95.55% [3] and 94.79% [6], respectively, the LTDT and LSTMAE models are quite effective. Nevertheless, it is challenging to swiftly adapt to new situations since the side channel data properties react differently in diverse processing circumstances. Offline, code analysis-based intrusion detection has a detection accuracy of 98.38% [8], but it can't detect anything while processing is happening.

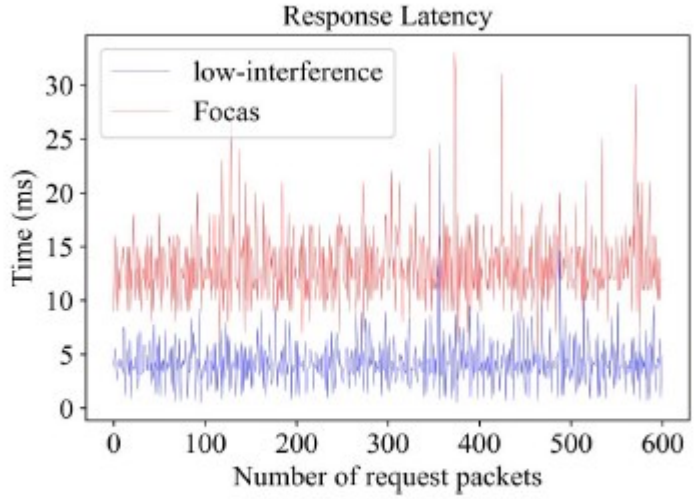


Fig. 7. Comparison of response latency for low interference and Focase.

www.ijbar.org
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

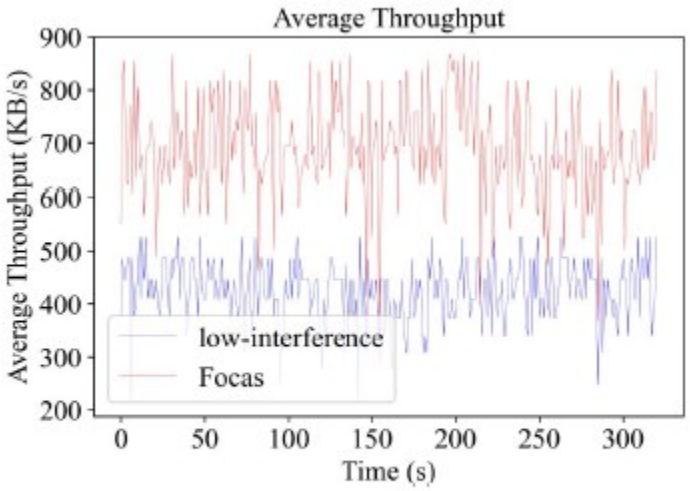


Fig. 8. Comparison of average throughput for low interference and Focase.

The attack situations that MPI-CNC can handle are more diverse than those of the other approaches. We can identify parameter injection attacks and instruction injection attacks by 100% because we actively monitor the key parameter information throughout the machining process and evaluate the critical parameters of the CNC system. In addition, the article's technique generates complete detection criteria by analyzing the NC code; as a result, it achieves a greater detection accuracy—98.81%—similar to the machining code analysis method.

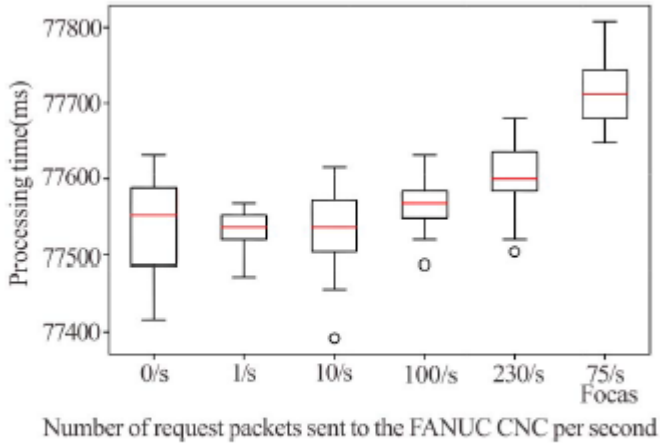


Fig. 9. Influence of network traffic on FANUC CNC.

2) Processing Times Under CPMS IDS: We ran studies with varying request frequency throughout the active detection process to see how machining time changed. This showed that our detection strategy had no effect on CNC systems' machining time. We bypassed industrial switches' forwarding delays by connecting the CNC system directly to the network via Ethernet cables; at 1, 10, 100, and 230 bps, we transmitted low-interference request packets; and at 75 bps, we sent Focas standard interface request packets. Figure 9 shows that the CNC system's machining time was around 1 minute and 17.545 seconds under typical conditions with no outside interference. We discovered that the machining time of the CNC system was little impacted when it was exposed to different levels of external interference. There was no packet loss and the CNC system's machining time increased by just 0.076% while delivering 230 packets per second. Which means the FANUC CNC system can handle 230 packets/second without bogging down its functioning. The CNC system's machining time rose by 0.217 percent while utilizing Focas standard interface request packets at a maximum rate of 75 requests per second; this approach of low-interference data collecting was much lower. A further 75 queries per second are the limit for the FANUC system. Our detection approach significantly reduces the machining time of CNC systems, according on the experimental findings.

VI. RELATED WORK

Cyberattacks on CPMS could jeopardize human lives and have a direct impact on manufacturing efficiency. Consequently, CPMS-related IDS research has become a major focus in academia. The major goal of CPMS IDS cyber-attacks is managing and interrupting the manufacturing process. Our analysis of recent studies in this area allowed us to group the most cutting-edge CPMS IDS into three distinct types according to the detection techniques they employ.

CPMS IDS Built on Digital Twins: With the use of historical data, control models that mimic physical processes may be fitted using digital twin technology [32]. In order to build controller digital twin models for the CNC system of a 3D printer, Balta et al. [7] gathered and analyzed machining data from the printer's history. The cyber-attacks that altered the temperature settings of the 3-D printer's nozzle heaters were identified by comparing the consistency between the real machining state and the simulated machining state of the controller digital twin model.

Manufacturing equipment produces a mountain of measurement channel data when machining, which can provide an indirect reflection of the machining condition; this data is used by CPMS IDS based on side-channel analysis. In the CPMS IDS domain, side-channel analysis-based detection approaches are widely used. To identify machining path manipulation attempts, Chhetri et al. [4] initially suggested using auditory data around production machines to train detection models. For the purpose of ensuring product consistency, Bayens et al. [33] integrated analysis of machining equipment acoustic features, features pertaining to the location of the machining, and features pertaining to production waste.

By comparing the 3-D printer stepper motors' auditory characteristics to audio fingerprints, Belikovetsky et al. [5] were able to identify the printing process. Using video stream analysis, Mamun et al. [3] were able to identify changes in the processing trajectory of 3-D printers. In order to identify creep assaults, Yoginath et al. [2] used the Bayesian model to examine the current values of the power lines of 3-D printers. Using the LSTM-autoencoder technique, Shi et al. [6] retrieved features from vibration sensors' side-channel data. They then applied the OCSVM classification algorithm to detect anomalies.

IDS CPMS Drawing from the analysis of machining codes: As a global standard for NC programming languages, the ISO-6983-1 [29] standard was established. A computer language made up of G codes and M codes, this standard covers the syntax and lexical rules of NC code.

CNC systems automatically regulate the machining process in accordance with the instructions in the NC code, which provides the most extensive and detailed control information of the machining process. Anomalies can be successfully recognized by studying the NC code. Statistical characteristics were retrieved from NC code by Beckwith et al. [8]. These features included the frequency of XYZ values, the quantity of G codes and M codes, and more. By analyzing NC code offline and training a machine learning anomaly classification model, they were able to detect abnormalities. After creating 3-D models by reverse-engineering NC code, Tsoutsos et al. [34] ran simulated pressure tests on them. While doing so, they found security holes in the NC code.

VII. DISCUSSION

These are some of the drawbacks of the article's suggested intrusion detection approach. 1) Man-in-the-middle attacks, when the CNC system's software is tampered with, may be difficult for the approach to handle. Such assaults can successfully evade the intrusion detection approach suggested in this paper and necessitate highly sophisticated attackers. 2) Whilst the approach works well with CNC machines with two or three axes, it might not be able to come up with detection criteria tailored to 5-axis machine centers. It's possible that CNC systems that use authentication procedures at the interface won't work with the active detection method suggested in this article. It should be mentioned, nonetheless, that the majority of CNC systems do not limit remote access to machining status information at this time. 4) While low-interference acquisition methods are applicable to CNCs from many suppliers, reversing protocols requires a high level of technical expertise. An important and pressing matter that requires attention is automated protocol reversal.

VIII. CONCLUSION

In order to identify CPMS cyberattacks in real time, this paper suggests a new method called MPI-CNC. We use the FANUC CNC machine tools to demonstrate how to create a prototype system. In particular, MPI-CNC does automated analysis of NC programs, derives invariants from the machining process, and creates rules for attack detection, including rules for the machining process and rules for important parameters. After that, MPI-CNC establishes detection windows and thresholds to identify attack behaviors, and it actively connects with the CNC system to gather process status and critical parameters via low-interference request packets. Last but not least, we put MPI-CNC through its paces on a FANUC CNC machine. The experimental findings show that MPI-CNC is very adaptable; it can identify different types of cyberattacks with high accuracy and doesn't interfere with the CNC system's regular functioning. Our method outperforms existing cutting-edge detection methods in terms of both flexibility and detection accuracy.

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 noisymulti-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–physicaladditive manufacturing systems by in-situ process authentication usingstreamline 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 manufacturingsystems,” 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 LSTMautoencoderbased 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 maliciousedits 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: Atrojan-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-Dtechniques,” in *Proc. 37th Annu. Comput. Security Appl. Conf.*, NewYork, 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. Palleti, A. Silva, and C. M. Poskitt,“Mitigating adversarial attacks on data-driven invariant checkers forcyber–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 detectionof industrial process control attacks,” *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 3, pp. 2054–2069, Jun. 2022.