Available online at www.sciencedirect.com

Manufacturing Letters

Manufacturing Letters 35 (2023) 1009-1018

51st SME North American Manufacturing Research Conference (NAMRC 51, 2023)

Systems and Methods for Authenticating Manufacturing Machines Through an Unobservable Fingerprinting System

Pavel Koprov[a]*, Shyam Gadhwala[a], Aniket Walimbe[a], Xiaolei Fang[a], Binil Starly[b]

[a] *Edward P. Fitts Department of Industrial and Systems Engineering, North Carolina State University, Raleigh, NC 27695, USA*
[b] *School of Manufacturing Systems & Networks, Ira A. Fulton Schools of Engineering, Arizona State University, Tempe, AZ 85281, USA*

* Corresponding author. Tel.: +1757-933-3235. *E-mail address:* pkoprov@ncsu.edu

Abstract

Digital transformation leads to the inevitable change in the security paradigm for machines on a factory production floor. A unified namespace for machines in an Industrial Internet of Things (IIoT) network is only reliable when machine assets can trust and verify the identity of assets connected to the IIoT system. Current methods of asset authentication do not consider physical unclonable functions (PUFs) and can easily be spoofed or misused. Our work proposes using PUFs for industrial equipment such as CNC machines, robots, and 3D printers for identifying machines on a network and providing authentication procedures. In this work, we chose to use the vibration associated with machines and its embedded moving parts as a means to identify machine assets on a network. It is hypothesized that the vibrations associated with specific machine movements will be unique to each machine even when machines look exactly the same. The moving parts within a machine may produce a unique vibration pattern that can be used for machine identification throughout the working cycle. Our method requires light computing and relatively cheap measuring devices to capture the 'fingerprints' of machines and verify the signal's integrity. An adequate number of equipment has been tested for the worst-case scenario, i.e. when two machines look exactly the same with the same moving parts and produce exactly similar motion to generate the vibration signal. Data preprocessing and standard machine learning techniques like RF, LASSO, and SVM show great performance on raw time series data, enabling 100% TPR and more than 94% TNR in detecting the false class of the machines.

Peer-review under responsibility of the Scientific Committee of the NAMRI/SME.
Keywords: Cybersecurity; Connected Manufacturing, Authentication; Physical Unclonable Function; Digital Twin; Vibration

1. Introduction

Manufacturing systems have been moving towards digital transformation or Industry 4.0 for more than a decade [1], and the security threats that were once hypothetical are now commonplace [2]–[4]. Some companies are even moving towards Digital Twins and Manufacturing in Metaverse [5]. The core of this paradigm is access to data and decision-making (control) of physical assets according to changing conditions. However, access to data, virtual assets, and management of physical assets can lead to potential disasters that were not a concern a couple of decades ago.

One significant challenge for cybersecurity in manufacturing is the lack of a reliable method to identify machines on a network. While machine serial numbers and MAC addresses exist to identify machines, they are easily spoofed and not reliable for unique identification. This creates problems with the traceability of data generated by specific machines. To address this issue, this paper proposes a method to extract a digital "fingerprint" of a manufacturing machine to uniquely identify itself on a connected network of machines.

Peer-review under responsibility of the Scientific Committee of the NAMRI/SME.

Currently, there are two methods of authentication: hardware keys and operator login processes through employee IDs associated directly with computers that control production cells or directly on the machines themselves. However, these methods authenticate the operator and not the machine itself, which can be a problem if a compromised machine is sending falsified data to higher-order manufacturing IT systems. Additionally, existing systems are vulnerable to insider threats and modification of data sources, which makes traceability to specific machine assets difficult.

Machines need to identify themselves through fingerprints to ensure that only authorized machines and devices are allowed to access sensitive resources or perform specific actions. This is important in preventing insider attacks, which are security breaches caused by someone with authorized access to the system or network.

An insider attacker may be a disgruntled employee, a contractor, or a partner who has access to sensitive information or systems. They can exploit this access to steal data, install malware, or cause other types of damage. By implementing machine identification, the system can verify the identity of each machine or device that tries to access the network or system, which helps to prevent unauthorized access and insider attacks.

Another important use case for authentication requirements is in the context of Digital Twins. It is crucial to ensure that the data being fed to a Digital Twin is actually coming from its corresponding physical twin. This is particularly important for training machine learning models in cyberspace, where it is essential to have accurate and reliable data from the correct source. Ensuring the authenticity of the data used in Digital Twins is paramount to their success and can have a significant impact on the accuracy and usefulness of their output.

While several authors have approached this problem to some extent, no research has been found on real industrial equipment. Related work involves the use of sound, electromagnetic, and acceleration data for the authentication of drones [6], 3D printers [7], and computing devices [8]. However, physical fingerprinting is particularly important for industrial cybersecurity since authentication mechanisms are crucial for identifying nodes on a network. Machine serial numbers or machine pseudo-names are currently used for identification, but they can be easily modified at the IT level and are not tied to specific data streams produced the from machines. These are akin to humans using usernames and passwords to login, while a more secure manner would be using fingerprints or eye-scans to login to a system.

To address these challenges, this paper proposes a method of digital fingerprinting that can uniquely identify a manufacturing machine on a network. The aim is to determine whether unique signatures exist for machine assets on a production floor and how we might easily obtain them. By providing a reliable and secure method for machine authentication, this approach could help improve the traceability of manufacturing data and enhance cybersecurity in the manufacturing industry.

In summary, this paragraph provides an overview of the challenges in machine authentication in the context of Industry 4.0, introduces a proposed method of digital fingerprinting, and highlights the importance of physical fingerprinting for industrial cybersecurity.

2. Project Goal and Contributions

Physically unclonable functions (PUFs) utilize the natural variations within a hardware device to generate a distinct and unrepeatable response to a specific input, making it an effective method for hardware security. At a broader level, PUFs can be compared to biometrics for humans, serving as inherent and one-of-a-kind identifiers for each equipment. [9]. The idea behind PUFs is to leverage the unique physical properties of the device that cannot be easily cloned or predicted to create a digital fingerprint, which can be used to generate secure keys for encryption or authentication purposes. For example, PUFs can use random variations in a device's manufacturing process, such as differences in capacitance, resistance, or delay, to generate a unique response. This response can then be used to authenticate a device or to generate a secret key that can be used to encrypt or decrypt data.

This project aims to investigate the feasibility of using vibration PUFs [10] as an authentication method for machine floor assets, including CNC machines, robots, and 3D printers. PUFs generate unique signatures specific to each individual device, even those of the same brand and model. We hypothesize that the vibration signal produced by the acceleration of these machines can be captured as a fingerprint, which can be used as an authentication token for accessing IT systems. This approach is insensitive to external noise and repeatable, making it suitable for generating consistent signatures over time. It is also expected to be resistant to repeated challenge attacks, such as those that use machine learning or side-channel attacks. In addition, this approach can be easily integrated into existing or legacy machines without disrupting their operation and requires minimal human intervention and lightweight computation to perform registration, authentication, and certification.

Our work contributes to the smart manufacturing field in the following ways:

- We test the feasibility of using vibration data associated with moving components in a machine as physically unclonable functions (PUFs) for industrial equipment authentication.
- We identify the optimal machine learning model for feature selection and decision-making to support vibration-based PUFs for manufacturing machine assets.

3. Literature Review

Previous research has explored the use of PUFs for authenticating various devices, and several studies have investigated their use in equipment such as 3D printers and CNC machines. For instance, Mehdi and Starly [11]

developed a "Witness Box Protocol (WBP)" to identify machines on a network. The protocol was tested on a 5-Axis tabletop CNC machine and six similar small-scale tabletop 3D printers, using data collected from a 3-axis accelerometer, magnetometer, and microphone. The machine's "password" move signals were recorded and divided into specific time snippets, and K-means clustering was used to differentiate between devices. The authors reported 100% and 95% authentication accuracy for the 5-Axis CNC machine and the six 3D printers, respectively. However, the study also found that even slight changes in the environment could result in significant changes in classification accuracy.

Additive manufacturing (AM) has gained popularity among hobbyists and professionals, but it also poses security risks. Do et al. [12] demonstrated the vulnerability of 3D printers, which can be impersonated and have their history of printed models extracted. Belikovetsky et al. propose detecting deviations in individual G-Code execution through audio signal analysis, as in [6] and [11]. They found that the physical movement of the machine has a significant impact on the PUF, and modifying the G-code to perform the same move did not affect the detection. Replacing G00 with G01 with the same feed rate had no change in the PUF. This is mainly because the physical movement of the machines had not changed.

In a similar vein, Ramesh et al. [6] explored the challenge of authenticating drones for last-mile delivery services using the audio signals generated by the drones' 54 motors. To test their approach, they assembled 11 quadcopters and extracted features from the audio signals using Discrete Fourier Transform and Discrete Cosine Transform. The team then used a Support Vector Machines (SVM) based machine learning approach to achieve an impressive 99.5% accuracy in differentiating the drones.

Gascon et al. [8] investigated a continuous authentication method for mobile devices that can detect malicious attacks even if the initial access to the device was mistakenly granted. To achieve this, the authors developed a digital keyboard for Android devices and asked over 300 participants to type the same text (160 characters) for machine learning training. While typing, the system collected acceleration data and signal length per character, which were represented as a 2376-dimensional vector. SVMs were used for the classification of intruders versus authentic users. Gascon et al. found that some users exhibit characteristic typing behaviors that can be identified with a high True Positive Rate (TPR) of 92% and a low False Positive Rate (FPR) of 1%. However, some users have similar typing behaviors to attackers, making it difficult to distinguish them from intruders.

Continuous authentication using various methods is a hot research topic in cybersecurity. Espin Lopez et al. [13] explored biometric parameters for continuous authentication using sensor vectors (gyroscope and accelerometer), statistics vectors (apps), and speaker vectors. They compared supervised vs. novelty detection ML methods to analyze the performance as, in most cases, only the target class data is available. Several ML models were applied for the novelty detection approach: One-Class SVM (OCSVM), Isolation Forest (IF), and k-Nearest Neighbors Detector (kNND). They used SVM, Random Forest (RF), and k-Nearest Neighbors Classifiers (KNNC) for the supervised approach. The results show that supervised techniques have higher accuracy in all the vectors. The highest accuracy for sensors and statistics vectors was shown with RF, while the voice vector was with SVM.

Devices that don't have moving parts are less likely to be authenticated with accelerometers. Cheng et al. [14] performed a study where they used electromagnetic interference (EMI) to authenticate CPUs. They demonstrate that it is almost impossible to produce an EMI pattern close to the target device. The authors experimented with 90 mobile devices, including 70 laptops and 20 smartphones. Ensemble classification approach ExtraTrees was employed using 15 features in time and frequency domains. Their method achieved an average of 99.1% precision and recall. In addition to authentication purposes, Pham et al. [15] used EMI to identify and classify malware types. They also used FFT to preprocess the data and LDA as a supervised ML model. The accuracy of type classification reached 98%. This study shows the granularity to which PUFs can be used in cybersecurity.

Gu et al. [16] used their system's time to act, input power, and load as fingerprinting. Every device takes a slightly different time to react to the PLC signal. It thus can be used as an identification feature. They showed a significant difference in operation time for the same action between the same types of devices from different vendors. ML models tested are Decision trees, Naive Bayes, and KNNC. Decision trees' top classifier in this study achieved a 0.89 precision score and 0.89 recall score. Ahmed et al. [17] pointed out that sensor fingerprints can be implemented as a function of noise in sensor measurements. Eight statistic variables were used from the signals to train the model. In this work, an OCSVM classification is used for attack detection in contrast to preliminary analysis, where a multi-class classifier was used. TPR varied for different sensors from 73% to 93.5%, averaging 82%. TNR (attack-free data declared normal) ranges from 86.3% to 94.2%, with a mean of 89.8%.

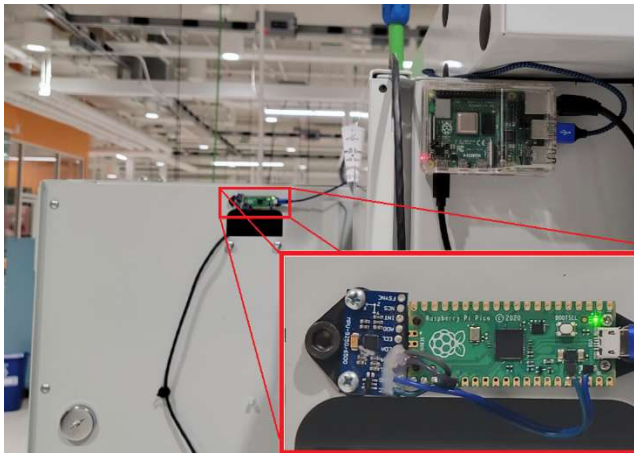

Fig. 1. Hardware setup on VF-2 CNC milling machine

As for other popular methods of ML, Ypma et al. [18] showed how Support Vector Data Description (SVDD) could be used for anomaly detection. Still, this study is related to monotonously working assets that do not have varying patterns in their day-to-day activity.

4. Methods

4.1. Hardware and Software Setup

In this work, we selected three different types of assets, and for each one, we used a specific model. The first two assets were CNC milling machines - a Haas VF-2 with a New Generation Control (NGC) unit and an umbrella tool changer. The second assets were five collaborative robots UR, which were installed on the Vention platform with no end-effectors mounted. The third assets were eight 3D printers, specifically Original Prusa MINI+. To measure the motion of these assets, we used a nine Degrees of Freedom (DoF) internal measuring unit (IMU) MPU-9250. This IMU is equipped with a three-axis accelerometer, gyroscope, and magnetometer and is capable of transmitting data via I2C and SPI protocols, which are specifications for synchronous serial communication interfaces used for short-distance communication.

We decided to use the I2C protocol due to its easy implementation and the availability of libraries for microcontrollers. Unfortunately, we could not find any previous projects that had used the MPU-9250 with Raspberry Pi and Python language. Therefore, we opted to use a Raspberry Pi Pico microcontroller and MicroPython language for implementing the IMU. The ADC of IMU allowed us to sample the data at a rate of 1 kHz. With our setup, we were able to accurately capture the motion data of the assets for further analysis and use in our experimentation.

The Pico device includes a micro-USB port and can be easily connected to a variety of computing devices. In our laboratory, we have connected a Raspberry Pi 4B to each CNC machine to collect data, as described in [19]. To install the sensor on the CNC machines, we used existing holes on the back side of the sheet metal cover (see Figure 1). This location was chosen because it resonates more when every axis moves and is standard for all VF-2 machines. We tested two control methods for password movement: simultaneous XYZ movement and change in spindle RPM versus one-at-a-time axis movement and spindle rotation. We selected the latter for proof of concept because it allows us to extract vibration patterns generated by the single-axis activity. To facilitate data collection and ensure real-life conditions, we mounted two different MPUs on the CNC machines. We tested both measuring devices in one setup and found no significant difference in their measurement signals.

In contrast to CNC machines and 3D printers, 6DOF UR cobots cannot perform the cartesian movement with only one axis. For this experiment, we created a four-waypoint program in which the robot's tool end changes position in XYZ but keeps orientation in roll, pitch, and yaw (RPY). The IMU was placed on the Vention platform and was not removed from any of the tests. We swapped out the robots for each set of measurements to ensure that the location or mounting of the sensors did not affect our results. The Raspberry Pi 4B was positioned on the table near the robot. For the Original Prusa MINI+ 3D printers, we mounted the IMU on the top of the Z-axis gantry. This printer design allows for the largest vibrations on this tower, making it an

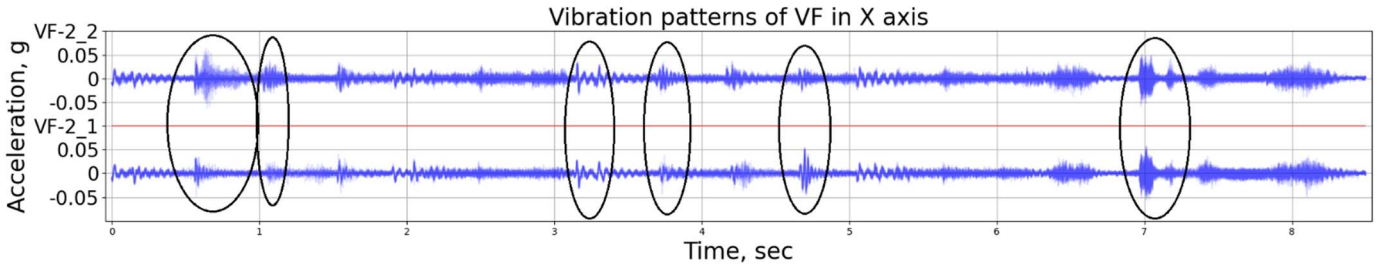

Fig. 2. Vibration patterns of VF-2 CNC machines. Ovals highlight the differences in signals.

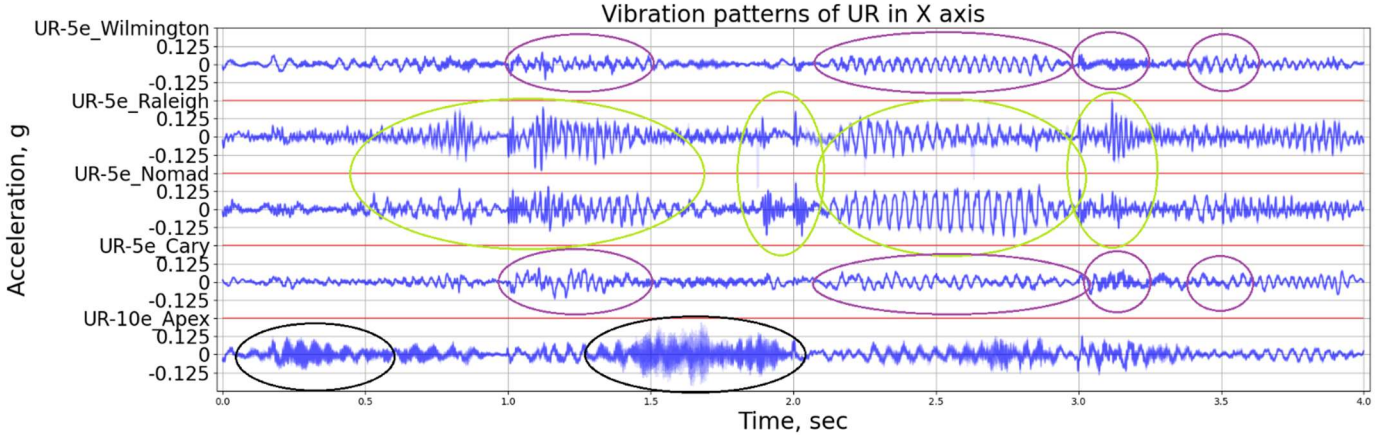

Fig. 3. Vibration patterns of UR cobots. Ovals highlight the differences in signals. Black ovals show the difference for UR-10e. Green ovals highlight the difference between 2 similar UR-5e patterns. Purple ovals highlight the difference between the other two similar UR-5e patterns.

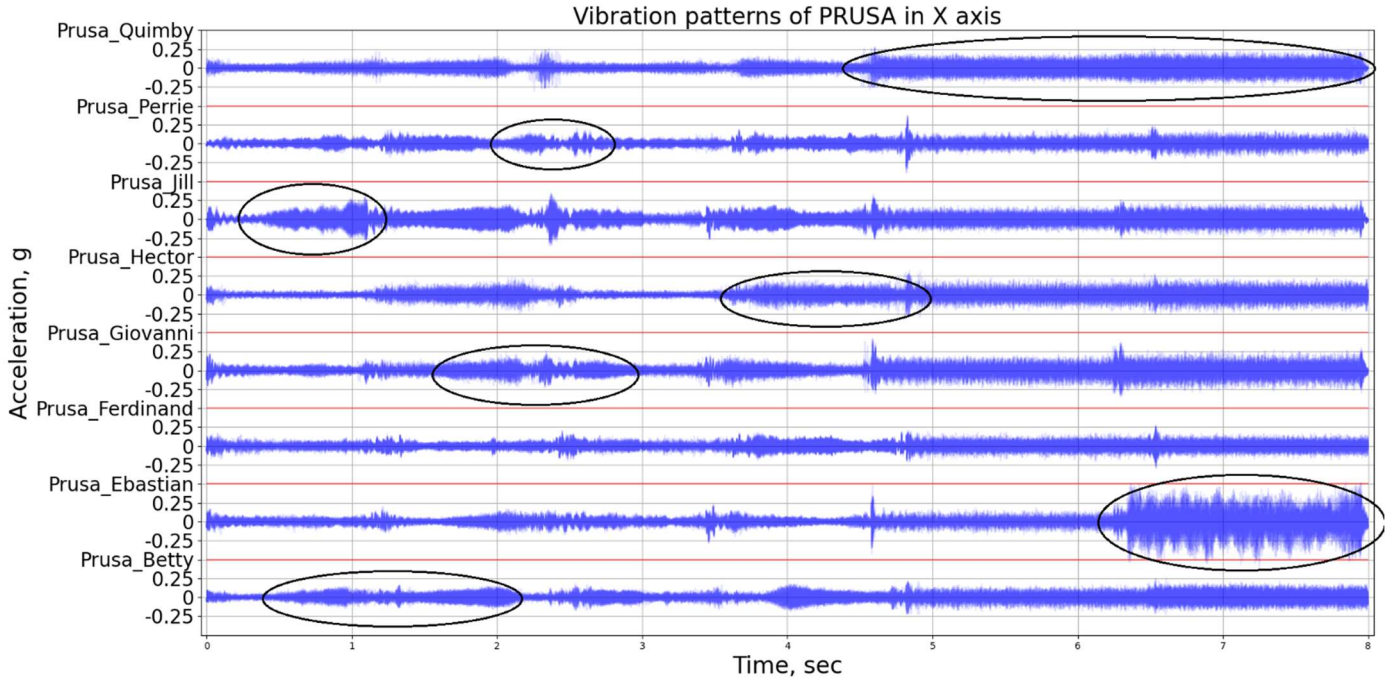

Fig. 4. Vibration patterns of Prusa 3D printers. Ovals highlight the differences in signals. Each oval circumscribes part of the signal specific to only one printer.

ideal location for capturing all X, Y, and Z accelerations produced by the three servo motors. In all hardware setups we used 3D-printed brackets to mount the IMU and Pico, using ABS-black material on a Stratasys F170 printer.

4.2. Password moves

Password moves need to be long enough to catch crucial acceleration patterns that the moving parts of the machinery can produce. If the asset contains several moving axes, it is better to move them all, as it would add more to the total variance of the sample. 3-axis CNC Machines and 3D printers can move one axis at a time to change the tool's location linearly. Such movement involves only one of the servomotors moving at a time. 6DOF robots cannot supply linear movement using only one of the servomotors at a time due to each axis's polar degree of freedom. In this case, the vibration pattern will include the PUF of all the motors at each point of time of the password move.

In the case of material removal by CNC machines, the password move may be implemented at the time of tool changing (but may not be long enough) for the in-process authentication or before the machine starts to run the program.

The G-code program is loaded to the machine before it starts and cannot be modified while running. Hence, this one-time authentication may happen before the program changes. We perform the password move before the program starts for proof of concept. A password move can automatically be added to the G-code in a real-life implementation.

VF-2 milling machines can move up to 30 inches on X-axis, 16 inches on Y-axis, and 20 on Z-axis. To get enough data, the password movement is described as follows:

1. Move 30 inches in X negative direction;
2. Move 15 inches in Y negative direction
3. Move 30 inches in X positive direction;
4. Move 15 inches in Y positive direction;
5. Move 4 inches in Z negative direction;
6. Turn the spindle 8000 RPM clockwise for 1 second;
7. Move 4 inches in Z positive direction;

The same principle can be applied to 3D printers. For example, some industrial-grade FDM 3D printers, such as the Stratasys F123 series, perform material purging between each layer, which can be used as a verification process and treated as a password move. However, controlling the movement of these printer's gantries with G-code or other open-source methods is not possible, which poses a challenge for data collection and verification purposes. This issue can be resolved if manufacturers include the option of password movement in their software.

Original Prusa MINI+ printers can be controlled by G-code and easily follow the patterns and feed. The password move uses the max federate of 1 m/s and the max travel in the X and Y axes. Z axis feed rate is much slower and was limited to 20 mm for time purposes. The program starts from the X0 Y0 Z10 position and waits for 2 seconds before each run. The password move is very similar to that of the CNC machine:

1. Move 180 mm in X positive direction;
2. Move 180 mm in Y positive direction
3. Move 180 mm in X negative direction;
4. Move 180 mm in Y positive direction;
5. Move 20 mm in Z positive direction;
6. Move 20 mm in Z negative direction;

There are many possible scenarios for robotics password moves. Cobots are mostly used for material handling and perform return movement from one fixed waypoint to another without an active payload. This period can be used as a password move and allows continuous authentication between operations. We chose to perform a simple

rectangular path in one plane as a worst-case scenario to study if unique signatures are being created from robot models that look fairly same.

UR robots can perform movements in 3 modes: MoveJ, MoveP, and MoveL [20]. MoveJ mode moves the robot in joint space, creating a smooth path. In this move, we chose to move the Tool Center Point (TCP) in the XY plane, resulting in all joints committing rotations. There were four waypoints created with the following coordinates in mm:

1. 0, -400, 250
2. 300, -400, 250
3. 300, 400, 250
4. 0, 400, 250

All waypoints kept the same RPY 90°, 0°, -90°. Every waypoint has 1 sec to be reached, followed by 5 seconds of dwelling after waypoint 1. The vibration patterns produced by each robot are depicted in figure 3.

Each machine has run this procedure 30 times after warming up the axis motors and the spindle. Every run has been separated by 5 seconds of dwell.

The program for Pico uses the MicroPython programming language and the mpu9250 library written by Mika Tuupola [21]. The code is divided into two scripts: boot.py and main.py. In the boot script, the MPU is initialized, and all libraries are called. The main script contains calibration offset creation for ambient noise/vibration (measured within 1 sec after booting), then triggering the signal that exceeds four standard deviations (SD) above the offset and sending those values to a serial port (USB). If 500 measurements fall below the threshold, then data transmission stops. This is made to collect acceleration within 4 SD within the movement cycle.

The code for collecting vibrations on the PC is written in Python and exploits libraries pandas, serial, and matplotlib. The user is prompted to input the number of runs in this code. After the number of runs is set, the serial port is flushed to erase all previous data sent to it. Data is read from the port; if it contains less than 1000 values, the run is considered erroneous, and the port is flushed again. This is made to avoid recording some accidental vibrations from the assets.

5. Data processing

We can see that each asset produces a different vibration pattern performing the same password movement (figures 2, 3, and 4). A computational method of identifying differences is needed.

One of the common ways for signal processing is applying convolutions on time series data. Ing et al. [22] demonstrated simple low-computing touch localization using accelerometers and convolutions. They show that the signal can mathematically be expressed as a convolution product between the emitted signal $e(t)$ from the touched point P, and the impulse response to the sensor S, $h_{PS}(t)$:

$$S(t) = e(t) \otimes h_{PS}(t)$$

This methodology compares each recorded signal to a benchmark and calculates the normalized score. The signal with the identical score is the convolution of the benchmark signal on itself. If we compare unknown scores – the one with the highest value has a high probability of being received from the same device. In our experiment, this method worked for CNC machines but not for robots. There were cases when the signal was very close in time-acceleration pattern but higher in amplitude, resulting in scores of more than one, which could yield an incorrect prediction.

The SD of the stationary noise for MPU-9250 doesn't exceed 0.03% (0.0024G) of the range in the mode of ±4G. The program starts registering accelerations only if the reading exceeds 4 SDs in any direction. The recording stops if the signal lasts less than 0.5 seconds after that point. The stationary nose lasts less than the 5th percentile of the 3D printer, the 8th percentile of cobots, and the 30th percentile of CNC machine signals. The main principle for differentiating the signals between assets is their temporal pattern and not the actual values from the sensor. The relation between values in consecutive is of main importance. Moreover, we collected 30 samples per asset to include most of the variance in our models.

Each password is generated from 4 to 8.5 thousand time steps with acceleration values in the X, Y, and Z axes. This is time series data, and every timestep represents a feature (variable). All data was fed to the ML models in a pure form with all three axes data. Every run is a row with 12 to 25.5 thousand columns. Non-linear models better fit such data, so we used flexible non-linear tree-type models. In our work, we tested supervised and novelty detection ML methods.

Supervised ML may be applicable when someone wants to identify the asset or authenticate based on binary classification (target and non-target asset) in manufacturing facilities where several identical machine assets are presented. We tested our data on Decision Trees, Random Forests, SVM, and LASSO models. Supervised methods are impossible for cases where the machinery is present only in a single unit, as the non-target class data is not available for training. In the case of anomaly detection or "fingerprinting," one-class classifiers are the better fit. Our work used two novelty detection ML methods: Isolated Forests and One-Class SVM.

We collected 30 samples per machine asset and randomly resampled them with a 3/1 ratio for training and testing purposes. The analysis of data and modeling was performed in RStudio with libraries glmnet (LASSO), rpart (Decision Tree for classification), randomForest, e1071 (SVM), kernlab (OCSVM), and isotree (Isolated Forest).

For supervised methods, data were coded into two classes: the name of the asset and "false." After training, a model made predictions on a testing sample that contained 25% of target class data and all other runs of not included false class assets. The decision for classification in novelty detection models was made by choosing the threshold of the predicted scores. In the base scenario, a 95% score cutoff was chosen.

There are various metrics for evaluating the performance of machine learning models, like the accuracy, precision, recall, Sensitivity, Specificity, and F1 score. All those metrics can be calculated using the confusion matrix. The data was labeled as the asset's name and "false," with the

Table 1. Sensitivity and specificity of tested ML models for every asset.

Sensitivity/Specificity/ F1 score	UR-10e_A	UR-5e_C	UR-5e_N	UR-5e_R	UR-5e_W	VF-2_1	VF-2_2
Decision tree	0.96/1/0.98	1/1/1	1/1/1	0.99/1/0.99	1/0.83/0.97	1/1/1	1/1/1
Random Forest	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1
LASSO	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1
SVM	1/1/1	1/1/1	1/1/1	0.99/1/0.99	1/1/1	1/1/1	1/1/1
OCSVM (95% threshold)	1/0.93/0.99	1/0.93/0.99	1/0.93/0.99	1/0.93/0.99	1/0.93/0.99	1/0.93/0.97	1/0.93/0.97
Isolated Forest (95% threshold)	1/0.93/0.99	1/0.9/0.99	1/0.9/0.99	1/0.9/0.99	1/0.89/0.98	1/0.9/0.95	1/0.93/0.97

continued

Sensitivity/Specificity/ F1 score	Prusa_B	Prusa_E	Prusa_F	Prusa_G	Prusa_H	Prusa_J	Prusa_P	Prusa_Q
Decision tree	0.85/0.78/0.92	0.83/0.88/0.9	0.89/1/0.94	0.99/0.75/0.99	0.91/1/0.95	0.99/0.83/0.99	0.92/1/0.96	0.94/0.89/096
Random Forest	1/1/1	1/1/1	1/1/1	1/0.88/0.99	1/1/1	1/1/1	1/1/1	1/0.89/0.99
LASSO	0.97/1/0.99	0.94/1/0.97	1/1/1	1/0.88/0.99	1/1/1	1/1/1	1/1/1	0.99/0.89/0.99
SVM	1/1/1	0.97/1/0.99	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1	1/1/1
OCSVM (95% threshold)	1/0.94/0.99	1/0.93/0.99	1/0.94/0.99	1/0.93/0.99	1/0.94/0.99	1/0.93/0.99	1/0.94/0.99	1/0.94/0.99
Isolated Forest (95% threshold)	1/0.9/0.99	0.52/0.93/0.67	1/0.93/0.99	1/0.93/0.99	1/0.94/0.99	1/0.89/0.99	1/0.94/0.99	1/0.9/0.99

latter as a "positive" class. This makes sense as cyber security applications triggering the false signal are of prime importance. We chose sensitivity, specificity, and F1 score as performance metrics in our work.

Sensitivity is a statistical metric that measures the proportion of true positive results in a population with the condition of interest. Sensitivity is also known as the true positive rate (TPR) or recall.

In binary classification problems, sensitivity is calculated as the number of true positive (TP) results divided by the sum of true positives and false negatives (FN):

$$Sensitivity = TP / (TP + FN)$$

In other words, sensitivity measures the proportion of actual positives that are correctly identified as such by a binary classifier. A high sensitivity means that the classifier correctly identifies a high proportion of the positive cases and has a low false-negative rate. Sensitivity is particularly important when a false-negative result is more concerning than a false-positive result, such as cybersecurity threat detection.

Specificity is a statistical metric that measures the proportion of true negative results in a population that does not have the condition of interest. Specificity is also known as the true negative rate (TNR). In binary classification problems, specificity is calculated as the number of true negative (TN) results divided by the sum of true negatives and false positives (FP):

$$Specificity = TN / (TN + FP)$$

In other words, specificity measures the proportion of actual negatives that are correctly identified as such by a binary classifier. High specificity means that the classifier correctly identifies a high proportion of the negative cases and has a low false-positive rate. Specificity is particularly important in situations where a false-positive result is more concerning than a false-negative result, such as cybersecurity threat detection or spam filtering.

Sensitivity and specificity are both important metrics for evaluating the performance of binary classifiers. They provide complementary information about the classifier's ability to identify positive and negative cases correctly.

The F1 score is a statistical metric that balances precision and recall in a binary classification problem. It is the harmonic mean of precision and recall and is calculated as follows:

$$F1\ score = 2 * (precision * recall) / (precision + recall)$$

The F1 score ranges from 0 to 1, with a higher value indicating better performance. It is a useful metric in situations where both precision and recall are both important, and it provides a way to compare classifiers with different trade-offs.

Especially in a cybersecurity application, a high F1 score may indicate that a classifier has a good balance between detecting true threats (high recall) and minimizing false positives (high precision). For our scenario detecting the threat should be as close to 100% as possible, whereas rejecting target signals is acceptable if it stays low level. We are looking at sensitivity (recall or TPR), specificity (TNR), and F1 score to give an adequate metric for an imbalanced test set.

6. Results

The performance of ML models is depicted in Table 1. The Decision Tree model for VF-2 machines always reduced the decision to only one variable, as the difference in this variable was stable and large enough for classification. The Random Forest model primarily uses 4 to 6 variables to decide, and the error reduces to 0 after 50 trees. LASSO reduces the model from 5 to 25 variables depending on the training split (random seed). Unlike other methods, Random

Forests take a significantly longer time for training: 34 seconds against 4 seconds for SVM and 0.9 seconds for Isolation Forests. Decision boundaries for novelty detection models IF and OCSVM were chosen in such a way as to reject at max two target class test samples (figures 5 and 6). As shown in figures 5 and 6, VF-2_2 has a very tight decision region in the novelty detection models. However, the separation is very distinct, unlike in the case of IF. Supervised ML models for VF-2 CNC machines performed with an outstanding result of 100% accuracy in predicting the classes. This can be explained by the fact that even though they were purchased at the same time and have identical equipment, they were used for different hours and experienced other conditions while cutting. VF-2_1 machine has experienced crashes several times while training sessions with students, and VF-2_2 did not have such events throughout its lifecycle. These conditions predetermined the variety in vibration patterns performing the same password move (figure 3). Of course, the smaller sample size for only two machines made it easier for supervised ML models to identify the class. On the other hand, novelty detection methods do not have information on other classes in the training process and also showed outstanding results, with the F1 score being more than 95%.

UR robots in the advanced manufacturing lab were used only for research purposes and didn't have many work hours. Neither of them experienced hard crashes. Likewise, in the case of CNC machines, Decision Tree models made decisions by only one variable at the beginning of the signal. Random Forests had a different number of the most important features for the decision process depending on the asset: from 1 to 4. The LASSO model for robots has reduced the model to 1..5 important features. As shown in figure 5, robot R has a tighter decision region in the OCSVM method than other robots. One of the test samples can even be falsely classified as non-target. In the case of IF, the separation is very distinct for every robot (figure 6).

Decision Tree models for Prusa 3D printers chose 1 or 2 parameters to make decisions, and their location was all over the signal length. The number of variables important for the decision-making in the RF model was from 2 to 8, with a median of 4.5. LASSO reduced models to have 9 to 29 variables with a median of 18. OCSVM model had shown a wide decision region for all printers except Prusa E. Printer Prusa G also had one test sample that was very close to non-target sample scores (figure 5). IF models had a worse performance than OCSVM, where scores for printers E, G, and Q were too close to non-target scores. Some test sample scores for printers E and G intersect with false scores. The decision boundary for the Prusa Q printer had to be manually set because one of the train scores was in the false region.

Overall, SVM and OCSVM performed the best, with the highest sensitivity, specificity, and F1 scores. Additionally, SVM had the lowest training time compared to other supervised models. It was eight times faster on average. OCSVM was slower than IF in training but performed better and more consistently. A supervised model can be chosen when you have more than four units of identical machinery and they do not differ much. OCSVM is preferred in the case when only one of the asset types is available, or gathering the data from all the assets of the class is not feasible (stopping the conveyor).

7. Conclusion

In this work, we demonstrated that PUFs could be used not only in electronics industries but in large machinery using such attributes as vibration. Supervised machine learning models can be very precise in detecting the classes of the assets but fail when no additional equipment is available for training. Some platforms like CESMII can potentially be used for training purposes if the owners of the machinery are willing to share such data.

On the other hand, novelty detection ML approaches show great performance. They can be easily tuned for the specific asset to allow sensitivity to be 100% allowing specificity to be as low as 93%. These approaches run faster and do not require additional data except their own.

PUFs of heavy machinery can serve as a fingerprint for cyber-physical systems allowing only authenticated assets to share the data with the Digital Twins. It is a lightweight, non-intrusive method that fits brownfield and greenfield devices.

This method is a great method against repeated challenge attacks. Side-channel attacks on machine learning models that use time series vibration data as input can be challenging for several reasons.

First, time series data is high-dimensional and complex, which can make it difficult for an attacker to identify specific features or patterns in the data that are relevant to the underlying model. Unlike simpler types of input data (e.g., images or text), time series data can be difficult to interpret and visualize, which makes it harder to design effective attacks. Second, time series vibration data is often noisy and contains a lot of irrelevant information, which can make it difficult for an attacker to construct effective attacks that can reliably exploit the model's vulnerabilities.

Further work would include encryption for signal or decision transfer and a programmable user interface for an easy Plug-n-Play solution. Future work will require continued testing our approach on a day-to-day basis to see how the performance of the systems deteriorates when the machine continues to operate and accumulate machine operational time. As mentioned in Mehdi & Starty [5], a re-registration of the PUF associated with the password motion maybe needed if the signal deviates away from the original PUF signal. This is akin to human users being asked to change their password after a specified length of time.

As an extension against side attacks, opaque model architectures, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), can be used. Machine learning models that use time series vibration data as input on these NNs can make it difficult for an attacker to understand how the model makes decisions based on the input data. This can make it harder for attackers to construct effective attacks or extract sensitive information from the model.

As a means of continuous authentication, some normal machine movement can be used as the password move for

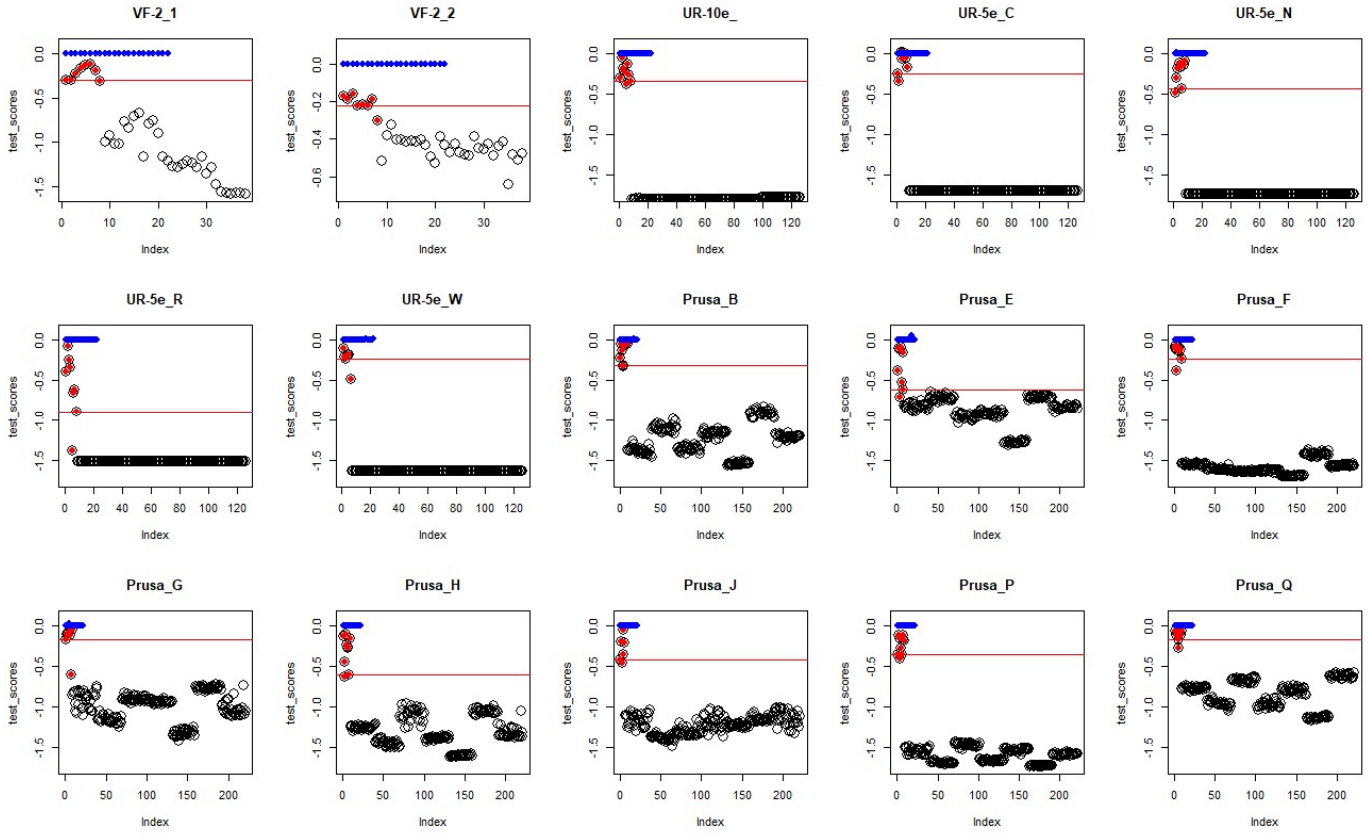

Fig. 5. Decision boundaries for OCSVM. Blue circles represent training scores. Red circles represent test scores for target data, and empty circles represent non-target data. The red line is the decision boundary. The test score is a raw scoring function of the samples returned by the OCSVM model

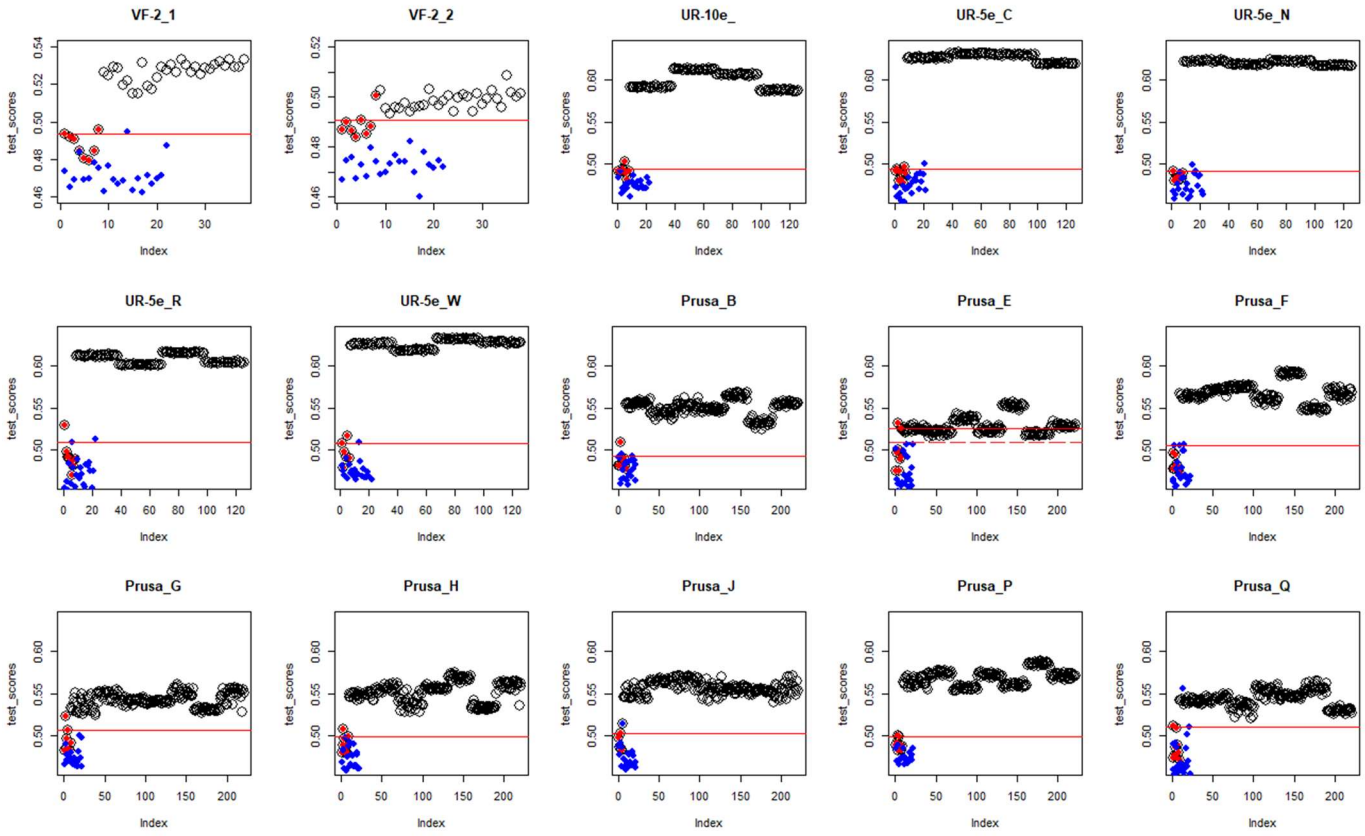

Fig. 6. Decision boundaries for IF. Blue circles represent training scores. Red circles represent test scores for target data, and empty circles represent non-target data. The red line is the decision boundary. The red dashed line is the boundary chosen manually. The test score is a raw scoring function of the samples returned by the IF model.

continuous authentication. 3D printers like Stratasys F123 perform nozzle cleaning between each layer. It can be used to authenticate the asset and prove the integrity of the process. VF-2 CNC machines change tools or perform a warm-up program every day. This routine can be used as the password move every time the operator changes or the program changes the tool. Robots are used to perform routine, mundane work, which is the best case for the continuous authentication of the asset. They perform the empty movement to the starting waypoint allowing authentication between every working cycle.

By demonstrating the potential of vibration data as a means of authenticating industrial equipment and providing insights into the most effective machine learning approaches for this purpose, our research provides a valuable contribution to the field.

Acknowledgments

The work conducted in this project was performed with partial support from CISCO.

References

[1] H. Kagermann and W. Wahlster, "Ten Years of Industrie 4.0," *Sci*, vol. 4, no. 3, Art. no. 3, Sep. 2022, doi: 10.3390/sci4030026.

[2] "H1 2022 – a brief overview of the main incidents in industrial cybersecurity | Kaspersky ICS CERT," Sep. 08, 2022. https://ics-cert.kaspersky.com/publications/h1-2022-a-brief-overview-of-the-main-incidents-in-industrial-cybersecurity/ (accessed Nov. 14, 2022).

[3] "Hackers Breached Colonial Pipeline Using Compromised Password," *Bloomberg.com*, Jun. 04, 2021. Accessed: Nov. 14, 2022. [Online]. Available: https://www.bloomberg.com/news/articles/2021-06-04/hackers-breached-colonial-pipeline-using-compromised-password

[4] "IBM Report: Manufacturing Felt Brunt of Cyberattacks in 2021 as Supply Chain Woes Grew," *IBM Newsroom*. https://newsroom.ibm.com/2022-02-23-IBM-Report-Manufacturing-Felt-Brunt-of-Cyberattacks-in-2021-as-Supply-Chain-Woes-Grew (accessed Nov. 14, 2022).

[5] "Boeing wants to build its next airplane in the 'metaverse' | Reuters." https://www.reuters.com/technology/boeing-wants-build-its-next-airplane-metaverse-2021-12-17/ (accessed Oct. 19, 2022).

[6] "Ramesh et al_2019_iSoundUAV-i.pdf." Accessed: Oct. 16, 2021. [Online]. Available: https://www.comp.nus.edu.sg/~junhan/papers/SoundUAV-Dronet19-CameraReady.pdf

[7] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici, "Digital Audio Signature for 3D Printing Integrity," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1127–1141, May 2019, doi: 10.1109/TIFS.2018.2851584.

[8] H. Gascon, S. Uellenbeck, C. Wolf, and K. Rieck, "Continuous Authentication on Mobile Devices by Analysis of Typing Motion Behavior," p. 12.

[9] "An Introduction to Physically Unclonable Functions - Technical Articles." https://www.allaboutcircuits.com/technical-articles/an-introduction-to-physically-unclonable-functions/ (accessed Feb. 17, 2023).

[10] C. Herder, M.-D. Yu, F. Koushanfar, and S. Devadas, "Physical Unclonable Functions and Applications: A Tutorial," *Proceedings of the IEEE*, vol. 102, no. 8, pp. 1126–1141, Aug. 2014, doi: 10.1109/JPROC.2014.2320516.

[11] N. Mehdi and B. Starly, "Witness Box Protocol: Automatic machine identification and authentication in industry 4.0 | Elsevier Enhanced Reader," 2020. doi: 10.1016/j.compind.2020.103340.

[12] Q. Do, B. Martini, and K.-K. R. Choo, "A Data Exfiltration and Remote Exploitation Attack on Consumer 3D Printers," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 10, pp. 2174–2186, Oct. 2016, doi: 10.1109/TIFS.2016.2578285.

[13] J. M. Espin Lopez, A. Huertas Celdran, F. Esquembre, G. Martinez, and J. G. Marin-Blazquez, "A supervised ML Biometric Continuous Authentication System for Industry 4.0," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2022, doi: 10.1109/TII.2022.3171321.

[14] Y. Cheng, X. Ji, J. Zhang, W. Xu, and Y.-C. Chen, "DeMiCPU: Device Fingerprinting with Magnetic Signals Radiated by CPU," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, London United Kingdom, Nov. 2019, pp. 1149–1170. doi: 10.1145/3319535.3339810.

[15] D.-P. Pham, D. Marion, M. Mastio, and A. Heuser, "Obfuscation Revealed: Leveraging Electromagnetic Signals for Obfuscated Malware Classification," in *Annual Computer Security Applications Conference*, Virtual Event USA, Dec. 2021, pp. 706–719. doi: 10.1145/3485832.3485894.

[16] Q. Gu, D. Formby, S. Ji, H. Cam, and R. Beyah, "Fingerprinting for Cyber-Physical System Security: Device Physics Matters Too," *IEEE Security & Privacy*, vol. 16, no. 5, pp. 49–59, Sep. 2018, doi: 10.1109/MSP.2018.3761722.

[17] C. M. Ahmed, J. Zhou, and A. P. Mathur, "Noise Matters: Using Sensor and Process Noise Fingerprint to Detect Stealthy Cyber Attacks and Authenticate sensors in CPS," in *Proceedings of the 34th Annual Computer Security Applications Conference*, San Juan PR USA, Dec. 2018, pp. 566–581. doi: 10.1145/3274694.3274748.

[18] A. Ypma, D. M. J. Tax, and R. P. W. Duin, "Robust Machine Fault Detection With Independent Component Analysis And Support Vector Data Description," in *Proceedings of the 1999 Ieee Workshop on Neural Networks for Signal Processing*, 1999, pp. 67–76.

[19] P. Koprov, A. Ramachandran, Y.-S. Lee, P. Cohen, and B. Starly, "Streaming Machine Generated Data via the MQTT Sparkplug B Protocol for Smart Factory Operations," *Manufacturing Letters*, vol. 33, pp. 66–73, Sep. 2022, doi: 10.1016/j.mfglet.2022.07.016.

[20] "Universal Robots e-Series User Manual." Accessed: Oct. 30, 2022. [Online]. Available: https://s3-eu-west-1.amazonaws.com/ur-support-site/115737/99404_UR5e_User_Manual_en_Global.pdf

[21] M. Tuupola, "MicroPython MPU-9250 (MPU-6500 + AK8963) I2C driver." Oct. 19, 2022. Accessed: Oct. 31, 2022. [Online]. Available: https://github.com/tuupola/micropython-mpu9250

[22] R. K. Ing, N. Quieffin, S. Catheline, and M. Fink, "In solid localization of finger impacts using acoustic time-reversal process," *Appl. Phys. Lett.*, vol. 87, no. 20, p. 204104, Nov. 2005, doi: 10.1063/1.2130720.