WattShield: A Power Side-Channel Framework for Detecting Malicious Firmware in Fused Filament Fabrication

Muhammad Ahsan
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia 23284
Email: ahsanm5@vcu.edu

Irfan Ahmed
Department of Computer Science
Virginia Commonwealth University
Richmond, Virginia 23284
Email: iahmed3@vcu.edu

***Abstract*—With the growing popularity of Additive Manufacturing (AM) and its use in security-sensitive applications, these devices have become lucrative targets for adversaries. Given the globalized nature of the supply chain, a robust security solution to detect malicious firmware and ensure a secure printing process and environment is necessary. While researchers have explored solutions for detecting attacks at the designing and slicing stages of the printing process, the firmware aspect remains underexplored. Moreover, the nature of attacks also allows adversaries to sabotage the printer hardware, which is not detectable by current in-situ monitoring solutions. This paper proposes WattShield, a security framework capable of detecting sabotage attacks on the print object and printing environment. WattShield utilizes the power side channel to capture thermal and kinetic process variables and detect malicious manipulations. It is designed to be minimally intrusive and adaptable to monitor different print geometries without requiring per-object training and profiling. To evaluate the WattShield framework, we employed nine firmware attacks across multiple categories, including intellectual property theft, compromising the print object, and targeting the print environment. Our results demonstrate that WattShield can successfully detect all implemented attacks, raising alerts and generating logs in each instance.**

***Index Terms*—Additive Manufacturing, Side Channel, Cybersecurity**

I. Introduction

With the evolving industrial landscape, additive manufacturing (AM) is gaining increasing attention. As a crucial component of Industry 4.0, additive manufacturing enables the creation of customized products. Unlike subtractive manufacturing, AM allows for the creation of more complex geometries with less material wastage and production time. Due to this capability, additive manufacturing has been increasingly adopted across various industries, including aerospace [1], medical implants [2], and automotive. By 2033, the industry size is projected to grow from $18 billion in 2023 to $110 billion, with an annual growth rate of 19.85% [3].

Cyber-physical systems because of their widespread availability as commodity devices and their use in many security-critical applications, are increasingly attracting the attention of adversaries [4], [5], [6]. While the true potential of these applications has yet to be fully realized, the potential vulnerability of devices to adversarial manipulations could deter users by making them skeptical of the benefits offered [7], [8], [9]. Moreover, the current industrial trend of fully connected IT and industrial networks [10], has potentially extended the adversarial reach to the manufacturing environment.

AM attacks can be categorized as stealing intellectual property (IP) information or sabotaging the printing process, wherein sabotage attacks can target the printed part or the print environment. For instance, an adversary might create a hidden cavity that compromises the mechanical strength of a printed part [11] or aim to critically damage the printer nozzle, rendering it dysfunctional. While researchers have proposed solutions to detect attacks targeting printed objects, monitoring the print environment is under-explored. Moreover, firmware attacks with implementation constraints [12], [13], require different monitoring strategies, compared to other stages of the printing process (design, slicing, etc.).

This paper proposes WattShield, a framework capable of detecting firmware attacks aimed at stealing IP information, sabotaging the print, or damaging the print environment. WattShield uses the power side channel to collect information on the printing process. The data collection approach is designed to be minimally intrusive and generic for a particular printer model, making the WattShield framework adaptable across multiple printers. The framework analyzes the captured sensor data to validate the thermal and kinetic process variables.

WattShield uses an automated trigger to monitor the process during the printing state. To validate the print geometry framework, use G-code as ground truth and employ per-command analysis, wherein features extracted from the signal are used to train the model and predict kinetic activities. For thermal activities, the framework uses empirical analysis to train the estimation function for predicting temperature values. During inactivity states, the framework monitors and logs different printer activities, which later can be used to identify anomalous/unauthorized activities.

Multiple experiments were conducted to calculate the detection threshold of the proposed framework in measuring kinetic and thermal process parameters. To evaluate the framework for detecting firmware attacks, nine attacks from different categories, including surveillance, sabotage to the print object, and sabotage to the print environment, are implemented from the literature. WattShield, given the detection threshold, can detect the attacks accurately.

The major contributions are summarized as follows;

- We propose a minimally intrusive monitoring technique capable of estimating thermal and kinetic process parameters. Through in-situ and offline process monitoring, the framework detects attacks targeting both the print object and the printer environment.
- We designed the framework to be adaptable to changing printing needs, i.e., without requiring per-object training data, thereby enabling the framework to start monitoring from the first print job.
- We evaluate the proposed framework against nine different firmware attacks to demonstrate its effectiveness across various firmware attack categories.

II. Background and Threat Model

A. Fused Filament Fabrication (FFF)

Fused Filament Fabrication, a material extrusion-based AM process, uses polymer as a material and infuses it layer by layer to create a final print geometry. The FFF process chain contains multiple components, starting with generating a 3D design file (CAD, obj). The design file is then transferred to a slicer software (e.g., Cura), where the user controls different parameters optimized for different geometries and material types. The slicer software converts the .stl file into corresponding machine instructions interpretable by the printer. This set of instructions also called the G-code file, contains all the design parameters set in the previous stages. The printer is connected to a control PC that sends the generated G-code file via the communication channels (e.g., Ethernet, USB/SD card, Serial, etc.). The printer firmware on the main board then interprets these instructions and correspondingly controls the printer actuators to produce a print object.

B. Threat Model and Assumptions

There are multiple components involved in the AM process chain that are susceptible to different attacks. Researchers have shown vulnerabilities in slicer software [14], communication protocols [15], [16], [17], firmware [12], [18], and control PC [11]. By exploiting such weaknesses an adversary can change the entire G-code file or manipulate commands and printing parameters being communicated to the printer.

We used firmware attacks in this study and based our approach on established adversarial techniques documented in the literature [12], [19]. For example; an adversary can introduce malware into the firmware by infiltrating the supply chain [20], or by hijacking the firmware update mechanism [21]. An adversary can also add malware to an open-source firmware repository, tricking an unsuspecting user into downloading and installing the malicious firmware onto their printer. Regardless of the technique employed, once the firmware is compromised, it can be used to damage the printer or print object, and can even be leveraged for surveillance purposes [13].

III. Related Work

This section details the AM attacks studied in the literature, followed by different defense techniques proposed in the literature for monitoring the printing process.

A. AM Attacks

Attacks on the AM process could majorly be categorized into surveillance and sabotage attacks. In surveillance attacks, the goal is to collect potentially valuable information e.g. intellectual property (IP) [22]. IP in terms of AM is the 3D geometry which could be a trade secret for a manufacturer and leaking such information to their competitors could jeopardize their market standings.

Sabotage attacks on the other hand are more direct and can be defined as any manipulations intended to deter the part being printed [23] or damage the print environment [24]. Multiple object-oriented sabotage attacks have been explored in the literature, where the goal is to either add obfuscated changes that go unnoticed from the production and quality assurance checks but fail during operations [11] or make obvious changes to render the part useless [18]. The print environment can be defined as the printing equipment and the print facility where the printer is placed. Yampolskiy et al. [25] discuss different implications of weaponizing the 3D printer to target the print environment and object.

An adversary can target different components including thermal or kinetic parameters to manipulate the printing process [26]. For example; an adversary could change filament density to create voids at critical positions in print geometry to degrade mechanical strength [27], [28]. Similarly, an adversary could target the nozzle temperature to cause clogged, partially clogged, or over/under-extrusion of the filament [19].

B. AM Process Monitoring

Side channel, in terms of cyber-physical systems, can be defined as any unintentional information being leaked from the physical process [24]. Analysis of this side channel information can provide critical information about the process and help identify malicious behavior [29]. Table I provides a comparative overview of the existing AM monitoring solutions using different side channels.

Current efforts in AM security could be categorized into offline verification of the printed product using non-destructive testing techniques [30], [31], or in-situ monitoring of the printing process using collected sensor data [32], [33], [34]. Based on sensor deployment they could be highly intrusive, have high calibration complexity, or noise sensitivity. For instance, acoustic monitoring [35], [36], though less intrusive, is more susceptible to noise sensitivity. Similarly, optical camera monitoring requires an unhindered view of the print geometry [37], [38], and is more complex to calibrate because of factors such as lighting conditions and deployment angles.

Based on the physical nature of the side channel some techniques are explored to monitor either kinetic or thermal process parameters. To handle such limitations some researchers have proposed using multiple sensors for each physical process. For example, Rais et. al. [39] accurately estimated multiple parameters using sensors deployed at each physical process, however, the employed sensing is intrusive and requires retrofitting/augmenting the printing setup.

Methodology	Ref.	Side Channel	Master Profile Req.	AM Process Stage	Process Type		Montr. Target		
					Kin.	Thrm.	Obj.	IP	Env.
Product Verif.	[30]	X-ray CT	✓	Slicer	✓		✓		
	[35]	Acoustic	✓	Design	✓		✓		
	[31]	X-ray CT	✓	Firmware	✓	✓	✓		
In-situ Process Montr.	[36]	Acoustic	✓	Slicer	✓		✓		
	[33]	Acoustic	✓	Firmware	✓		✓		
	[40], [41]	Acoustic	-	Printing	✓			✓	
	[32]	Elect. Current	✓	Design	✓		✓		
	[42]	Opt. Camera	-	Printing	✓			✓	
	[37], [43]	Opt. Camera	✓	Design	✓		✓		
	[38], [44]	Opt. Camera	✓	Design	✓		✓		
	[45]	Multiple Sens.	✗	Firmware	✓		✓		
	[39]	Multiple Sens.	✗	Slicer	✓	✓	✓		
	[46]	IR Camera	✓	Design		✓	✓		
	[47]	Elect. Current	✗	Slicer	✓		✓		
	[34]	Elect. Current	✗	Slicer	✓		✓		
	WattShield	Elect. Current	✗	Firmware	✓	✓	✓	✓	✓

TABLE I: Comparison of monitoring techniques in additive manufacturing; last row is the proposed framework.

The majority of the proposed techniques require a master profile of the design file to analyze the data. For example, Gatlin et al. [32] used the electric current side channel to generate a master signature for the normal print and used it as a baseline to identify any malicious G-code command manipulations. The use of such techniques assumes that the generated baseline signature is of unaltered print and is also limited to a specific geometry of print at a time.

The proposed framework doesn't necessitate a master print signature and can detect thermal and kinetic process variations resulting from malicious firmware. The framework uses the minimally intrusive approach to capture electric current side-channel data, wherein instead of adding sensors to each actuator the current is measured at the socket level. The framework can also identify any malicious changes intended to physically damage the print environment, we have found no current solution addressing such attacks.

IV. Side Channel Monitoring

A. Electric Current Side Channel

The power side channel is defined as the analysis of electric current consumption patterns, with the voltage remaining constant, to identify and differentiate between various activities. At any given time, the electric current side channel (SC_e) representing the printer state contains two types of variables, i.e., kinetic and thermal. The kinetic variable includes the nozzle and the print bed position, whereas, the thermal consists of the nozzle and the print bed temperature. If the instantaneous values of the current side channel for kinetic and thermal parameters are represented by $K(t)$ and $TH(t)$, then;

$$K(t) = [N_{(x,y)}, B_z]_i$$

$$TH(t) = [N_T, B_T]_i \qquad i = 1, 2, ...N-1$$

Where N represents the total number of samples taken at sampling rate f_s. Complete side-channel information can therefore be modeled as $SC_e(t) = [K(t), TH(t)]_i$.

1) Kinetic parameters modelling: For process monitoring, the SC_e captured over an interval provides more insights than instantaneous data. Let $K_a(T,d)$ be the signal captured over time 'T' for a G-code command moving actuator 'a' for distance 'd'. $f_a(S_f, t)$ is a function representing the instantaneous value, at a time 't,' of the actuator 'a' signal moving at speed S_f. Consider there are two states of the signal in which the actuator 'a' is either moving or idle, then the captured signal is represented as;

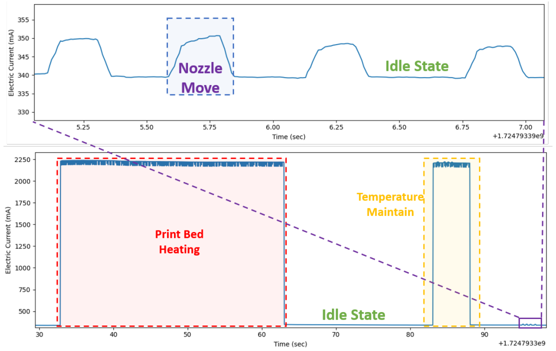


Fig. 1: Electric Current Signal for Kinetic and Thermal Actuation

$$K_a(T,d) = \begin{cases} f_a(S_f, t) & t_s \leq t \leq t_s + t_d \\ idle\ state & otherwise \end{cases}$$

2) Thermal Parameter Modelling: Unlike actuators, where the current state could be represented by either an idle or moving state, the thermal parameters take 3 states, i.e., idle, heating, and temperature-maintaining states. The current consumption patterns differ while heating to achieve the target temperature rather than maintaining it. If time to attain target temperature 'T_r' is represented by 't_r' then the signal captured over the total duration 't_n' can be represented as;

$$TH_h(T_r, t) = \begin{cases} f_h(T_r, t) & for \quad t_s \leq t \leq t_s + t_r \\ f'_h(T_r, t) & for \quad t_s + t_r < t \leq t_n \\ idle & otherwise \end{cases}$$

Where, $f_h(T_r)$ represents the signal for the duration where the printer is attaining the target temperature, and $f'_h(T_r)$ is the captured signal during the temperature-maintaining phase. The two thermal parameters, nozzle (N_T) and print bed (B_T) temperature are modeled using this equation.

B. Challenges in Monitoring

Researchers have used electric current side-channel for labeling activities e.g. identifying activities in IoT devices to detect malware [48]. However, limited research is available on identifying activities for 3D printers due to challenges and limitations associated with the physical process. The rest of the section will detail these challenges.

1) Limited Relationship: The relationship between certain process parameters and corresponding side-channel information could be weak. For example; for heating processes; i.e. nozzle and print-bed temperature the current consumption pattern provides stronger mutual information in the captured data as compared to stepper motor actuation. Due to this, it becomes practically impossible to detect certain kinetic-process information from the capture signal. We addressed this limited information of kinetic information availability by defining the activity detection to measurable activities i.e. instead of individually mapping stepper motor activities (x, y,

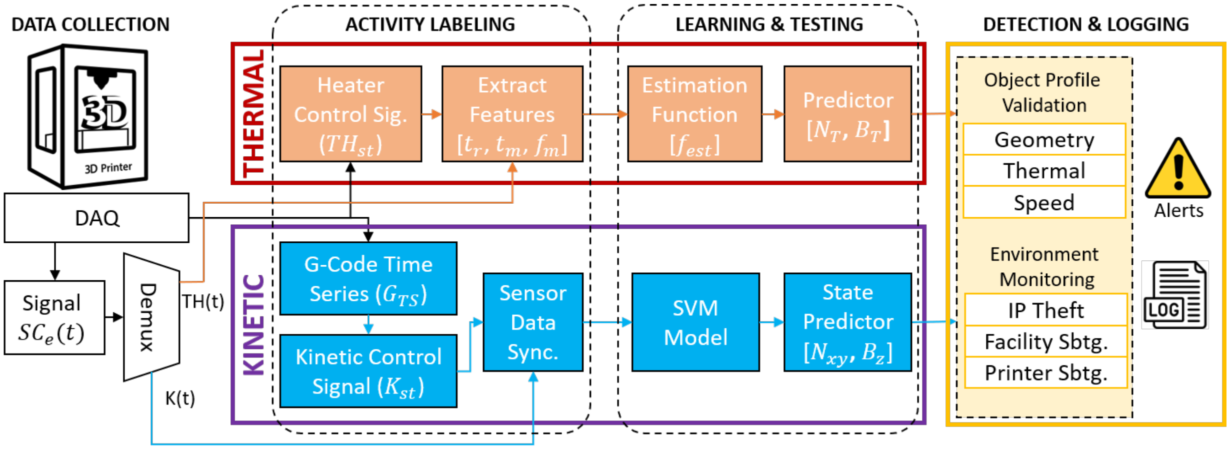


Fig. 2: Framework for detecting Sabotage attacks

and z-axis) we defined the scope to the nozzle (x and y-axis) and printer bed movement (z-axis).

2) Physical Process Properties: For heating processes, the captured data does not directly translate into interpretable thermal information. For instance, while side-channel data can provide insights into the activity of the nozzle-heating element as it reaches the desired temperature, it doesn't directly indicate the current temperature of the nozzle itself. To overcome this challenge, we leveraged process knowledge to develop an estimation function capable of predicting the nozzle temperature from its initial state.

In contrast to heating elements, the current consumption patterns of stepper motors used for kinetic actuation are influenced by factors such as motor model, rating, and the load they bear. Typically, 3D printers utilize identical motors for the x, y, and z-axis movements, and the loads on the x and y-axis motors are usually balanced. As a result, the current consumption patterns for movements along the x and y-axes become indistinguishable. However, the print bed's motor, due to its different load, exhibits a more distinct current consumption pattern compared to the other actuator motors. Therefore, we train our model on the measurable activities i.e. nozzle and print-bed moves.

3) Sensor Sensitivity: The current consumption for heating elements and actuators ranges from amperes to milliamperes. For instance, in the Ultimaker 2+, the bed heater can draw up to 2.2 A, while the motor actuators generate current changes as small as 10 mA, as demonstrated in Figure 1. Consequently, the sensor must be capable of detecting these small milliampere variations while also measuring currents in the ampere range. This wide measurement range poses challenges for sensor sensitivity, as it can increase noise levels in the milliampere range and reduce the sensor's ability to respond quickly and accurately to small current changes. As a result, any variations in the sub-millimeter range will be lost in the signal noise. Additionally, process variations faster than the sensor's sampling rate might not be captured in the data.

4) Data Complexity: During the normal printing process, multiple activities are performed simultaneously. For instance, maintaining the nozzle and bed temperatures requires the firmware to monitor the current temperature and toggle the heating elements on or off based on the difference from the target temperature. As a result, at any given moment, components such as the x-axis motor, y-axis motor, print-bed motor, nozzle heater, and print-bed heater may be activated concurrently or individually. We addressed this challenge by implementing a demultiplexer that separates the heating element signals from the motor actuation signals. Once these signals are separated, classifying different activities becomes significantly less challenging.

V. Proposed Framework

The proposed framework, illustrated in Figure 2, is structured into four distinct stages: data collection, activity labeling, learning and testing, and attack detection. Each stage operates in conjunction where the data from one stage is transferred to the other for further processing and analysis. Since the kinetic and thermal parameters are modeled differently the frameworks assume two separate paths, where the signal in each path is processed differently at each stage and is detailed in the following subsections.

A. Data Collection

This module collects the time series sensor and printer firmware data to be used in the later stages. The module also collects the G-code file, which is initially used for training and then later for attack detection. The acquired sensor signal captured at a sampling rate of 'f_s' is first passed through a filter to demultiplex the kinetic and thermal signal data. The signals are then further used as input to the corresponding later stages to detect kinetic and thermal process anomalies.

B. Kinetic Module

1) G-code Transformation: The first step in the activity labeling is to transform the G-code into the time series representation. The G-code has two types of instructions: move commands (G0/G1) and control commands (M). The temperature is set using M commands and only contains target temperature information. The G-code move instructions contain the initial and final coordinates of the nozzle print

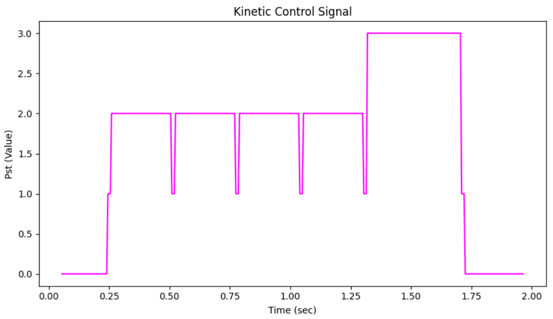


Fig. 3: G-code control signal for nozzle and print-bed move commands

head, along with the extrusion information, and therefore, can be transformed into a time series representation. 'G1' and 'G0' instructions correspond to move commands with and without extrusion, respectively. The transformation module first filters these commands from the G-code file, along with other parameters such as coordinates and speed information.

To convert the G-code to a time-series representation, second-order motion equations are used. These equations, taking into account the maximum speed and acceleration, calculate the nozzle position at the given point in time. Algorithm 1 (Appendix B) is employed to convert the end-point information into time series representation (P_{st}). The firmware, controlling the actuators, accelerates and decelerates the motors using a trapezoidal or triangular speed profile. Wherein, given maximum speed, distance, and acceleration, if the calculated speed is less than the maximum speed, then a triangular profile is used to generate time-series data. Otherwise, a trapezoidal profile containing the acceleration, constant speed, and deceleration phase is used. The algorithm given the initial (x_i, y_i) and final (x_f, y_f) endpoints along with maximum speed (v_{max}) and acceleration (a_{max}), calculates the nozzle position at every time step Δt. The algorithm 1 details steps for the nozzle triangular speed profile, similar steps will be followed for trapezoidal and print bed position.

2) Kinetic control signal: The transformed G-code data (G_{TS}) is used to ascertain the state of the nozzle and bed actuators at a given time t. If the position of the actuator at time instance t differs from the previous position at $t-1$, then the actuator is considered in the active state at time t. Conversely, if the actuator's previous and current position is unchanged then it is considered inactive. Therefore, the printer's state can be represented by a time series signal of combined actuator states, where the value at any given time reflects the printer's current state.

Four kinds of activities will be labeled using this time series data including; idle state, G-code command change, nozzle moving, and print bed moving state. Labeling all these activities with an integer value of 0-3, respectively, gives a control signal as shown in Figure 3. The control signal P_{st} obtained from the G-code time series data will be used to label the current signal captured during printer activity to train the model. In the detection stage, P_{st} is then used to compare against the predicted values to detect malicious activities.

3) Synchronization and Signal Labelling: To detect the start (t_s) and end $(t_s + t_d)$ of a particular move instruction in the captured current signal, the signal is labeled with the kinetic control signal (P_{st}). However, the signal first needed to be synchronized with the captured sensor data. The synchronization can be achieved by using an external trigger to start and stop capturing the sensor data. Later, the two time-series signals are merged based on time stamps.

4) Model Training: Once the activities are accurately labeled, the next step is to train a classification model using the captured data. The model takes the electric current signal as the input feature and the synchronized labels as the output. To achieve precise predictions while accounting for the influence of neighboring values, we employed a non-linear Support Vector Machine (SVM). The SVM works by finding an optimal hyperplane in a high-dimensional space that best separates the different classes, allowing for complex decision boundaries that can accurately classify the activities. If the input features are represented as $K_i \in \mathbb{R}^d$ and the output labels as $P_{st_i} \in \{0, 1, 2, 3\}$ than the training function can be abstracted as;

$$\text{TrainSVM}\left(\{(\mathbf{K}_i, P_{st_i})\}_{i=1}^{N}\right) \rightarrow \text{Model parameters}\{\mathbf{w}_k, b_k\}_{k=0}^{3}$$

Where N represents the total number of training data points. Using the above model parameters the predicted classes $(\hat{P_{st}})$ on the new input feature K_x can be represented as;

$$\hat{P_{st}} = argmax_{k\in\{0,1,2,3\}}\left(\mathbf{w}_k \cdot \phi(\mathbf{K_x}) + b_k\right)$$

5) Kinetic Sabotage Detection: This module detects attacks targeting the print geometry. The detector utilizes the trained model to make predictions based on real-time current signals. These predictions provide the time signatures of each executed G-code command. The predicted signatures are then compared with the expected time signatures generated from the G-code file (P_{st}). Detection occurs after a predefined set of executed commands, and if the predicted time signature deviates from the G-code-generated time signature beyond a defined threshold (see Section VI-D), the detector triggers an alert.

C. Thermal Module

The current signal is first processed to decouple the nozzle and print bed temperature signals. The signal is then synchronized and labeled with the heater control signal to extract multiple features. The features are then used for empirical analysis to generate the dataset used by the estimator function to train and predict the target temperature. The detector in the last stage uses the information to trigger alerts. The steps are further detailed in the following subsections.

1) Thermal Activity Labelling: To label the current signal for thermal processes, we used the temperature readings reported by the printer. The reported data contains temperature and heater information for both thermal processes. The information will help profile the current behavior to identify idle, heating, and temperature-maintaining activities. The firmware achieves the target temperature in a controlled manner where

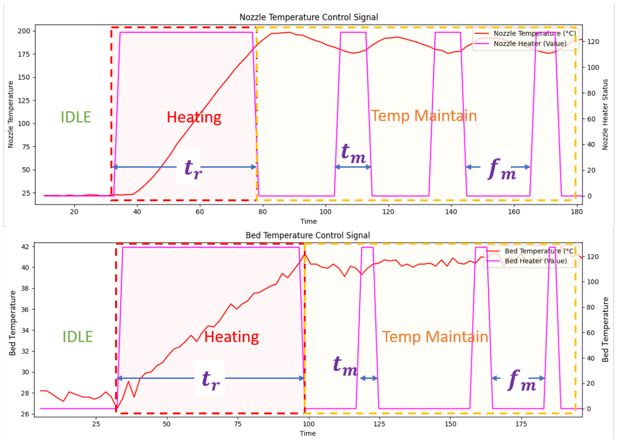


Fig. 4: Thermal control signal for temperature control commands

the temperature difference is used as input to control the heater value. The heater value is fully turned on till the temperature goes beyond the target temperature and then turned off for it to cool down and attain the target value. Once the target temperature is achieved the firmware maintains the temperature by using a closed-loop temperature control and regularly switching the heater for a short duration.

The heater control signals for both nozzle and bed temperature, along with the temperature values, are shown in Figure 4. Where t_r represents the time taken to achieve the target temperature, t_m represents the pulse width used to maintain the temperature, and f_m represents the frequency of the maintaining pulse. The thermal state of the printer using control signal at time instance 't' is represented by $TH_{state} = [N_H, B_H]$, where N_H and B_H represent the nozzle heater and bed heater value, respectively. The control signal is used to label data during each mode of heating operation.

2) Thermal Empirical Analysis: The captured current signal doesn't directly contain the temperature information. However, the current consumption pattern could be used to help identify the rise in temperature. The time taken to achieve the target temperature (t_r) is the function of the temperature difference between the initial (T_i) and target temperature (T_r). i.e. $T_r - T_i = f_{est}(t_r)$, where f_{est} is the non linear estimation function. The larger t_r value indicates a larger temperature difference and vice versa. Using this observation, one can estimate the target temperature value. Using empirical analysis to generate data points an estimation function can be derived and could later be used to predict the target temperature provided t_r value and the initial temperature.

3) Thermal Sabotage Detection: To detect sabotage to the print object thresholds are set for multiple parameters including target temperature (T_r), the maintaining signal frequency (f_m), and the maintaining signal pulse width (t_m). If the difference between the target and the predicted temperature falls outside the set threshold range, an alert is raised. Similarly, if the expected and measured maintaining signal pulse width or frequency deviates beyond the threshold, it indicates possible malicious thermal activity, triggering an alert.

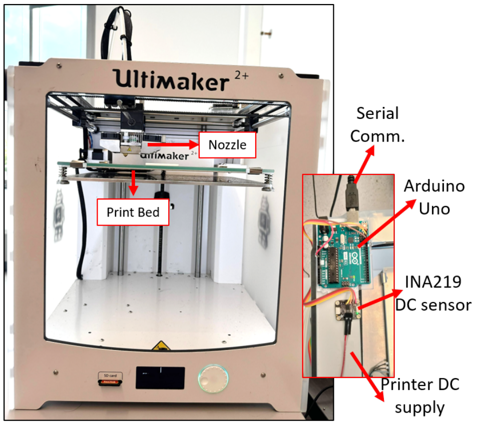


Fig. 5: Experimental setup used for evaluation of proposed framework

D. Activity Monitor

To identify attacks aimed at sabotaging the print environment, the module logs the printer activities in a file. Activities include SD card insertion, printer active and idle state, nozzle and print bed moving, and the nozzle and bed thermal activity. All these activities are logged alongside the time of the activity to facilitate monitoring of any unauthorized action performed targeted at damaging the printing environment.

VI. Implementation Details

The experimental setup (Figure 5) includes an Ultimaker2+ printer using PLA material, controlled by slicer software (Cura) running on a Windows PC. An air-gapped monitoring system, also running on Windows PC, operates the framework. The setup features a DC sensor (INA219) for collecting power signals, with an Arduino Uno communicating with the monitoring system and serially transmitting the sensor data. The following sections delve into the implementation-specific details of each module.

A. Data Collection

The INA219 sensor with a resolution of 0.8 mA, can measure up to 3.2A of current. The sensor communicates with the Arduino Uno via the I2C interface and is configured in continuous mode to average 16 samples with a conversion rate of 8.51 ms. To accommodate this conversion time the sampling frequency (f_s) is set to 10 ms. The Arduino Uno polls the sensor data, monitors the external trigger pin, and serially sends the data, along with a timestamp, to the monitoring PC. The communication module on the monitoring system along with the sensor data, also collects temperature data from the printer. Before the time-series sensor data is transmitted to later stages, it is pre-processed to filter noise using a simple moving average (SMA) filter. The filtered signal then passes through a demultiplexer algorithm to separate the thermal and kinetic actuation signals. The algorithm uses knowledge of thermal signal patterns to define a variable that is subtracted from the signal to isolate the kinetic actuation data.

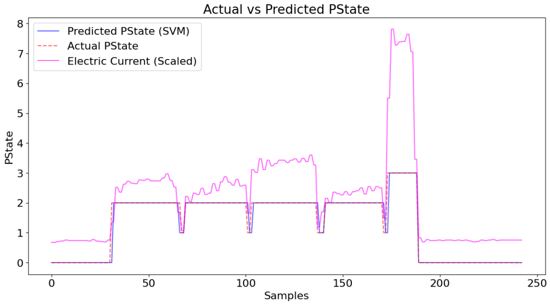


Fig. 6: Prediction results for the nozzle and print-bed moves

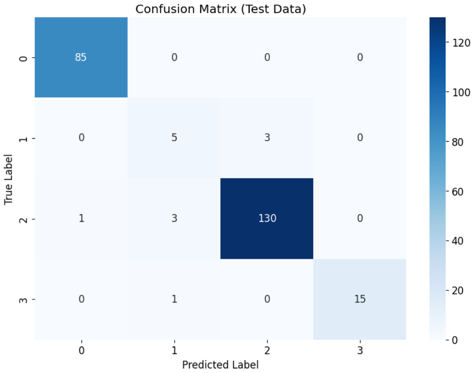


Fig. 7: Confusion Matrix for the SVM classifier

B. Activity Labelling

The collected sensor and temperature data along with the G-code file are used for labeling thermal and kinetic actuation activities in the corresponding signals. The 'M105' command is used to get the temperature report from the printer. The response is parsed to get the nozzle and print bed temperature and heater values. The heater value ranges from 0-127, where 127 means the heater is operating at maximum capacity to achieve/maintain the target temperature. The captured data is then used to label the nozzle and bed thermal activities. The labeled thermal data is then used to identify t_r, t_m, and f_m values that are later used to train the estimation function.

For kinetic actuators, the kinetic control signal (P_{st}) data is used to label the current signal. The control signal is synced using the external trigger. The trigger is set in the G-code file at the start of printing. The command M42 P13 S255 is used for that purpose which sets the pin 13 with an analogue value of 255. The trigger tells the framework that the printing process is being started which in response starts capturing sensor data, thereby syncing with the start-time of G-code generated P_{st}.

C. SVM Model training

The labeled sensor data is used to train the SVM model, which is a robust method for classification tasks. The training dataset, consisting of 22665 samples, is generated by executing multiple G-code commands. These G-code commands include variable-length movements along different dimensions, which helps diversify the dataset and enhance the model's ability to generalize and make accurate predictions on unseen data.

Command	**Distance (mm)**	**Expected Time (s)**	**Predicted Time (s)**	**Δt (s)**
Nozzle Move	15	0.26	0.245	0.015
	10	0.18	0.155	**0.025**
	5	0.1	0.085	0.015
	3	0.065	0.05	0.015
	1	0.04	0.035	0.005
Print-bed Move	15	0.27	0.265	0.005
	10	0.185	0.19	0.005
	5	0.1	0.115	**0.015**
	3	0.07	0.08	0.01
	1	0.035	0.05	0.015

TABLE II: Prediction results for Nozzle and Print-bed moves

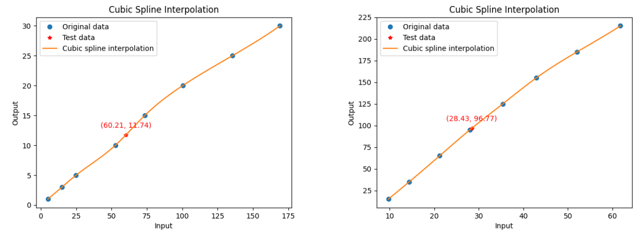
To ensure compatibility with the SVM model, the input features are first normalized using the MinMaxScaler, a preprocessing step that scales the data between 0 and 1. Following normalization, the data is segmented into fixed-length sequences of five data points each. This step ensures that the temporal dependencies inherent in time-series data are captured, allowing the model to learn patterns over time. The sequences are then fed into the SVM model, which is configured with an RBF (Radial Basis Function) kernel. The RBF kernel is particularly effective for classification tasks as it can handle non-linear relationships between features. The hyperparameters were carefully optimized to improve the model's predictive accuracy, with the regularization parameter 'C' set to 10 and gamma set to 'scale'. The prediction results are shown in Figure 6 with the reported overall accuracy of 97.12%, and the confusion matrix is shown in Figure 7.

D. Kinetic Sabotage Detection

The trained model is used to make predictions based on real-time sensor data, with predictions occurring after every 5 G-code command executions. These predictions identify which activity (nozzle or print-bed) is being performed and the duration of each activity as indicated by the command changes. The predicted time, calculated by multiplying the number of samples by the sampling frequency, is then compared to the actual time derived from P_{st}. The results are tabulated across multiple commands (Table II), with the minimum detectable distance recorded as 1 mm for both nozzle and print-bed movements. Additionally, the maximum difference between the actual and predicted results for nozzle and print-bed movements is 0.025 and 0.015 seconds, respectively, and are used as thresholds by the framework for generating alerts.

E. Thermal Parameters Estimation

After labeling the current signal for heating and temperature maintenance states using the thermal control signal (TH_{state}), the next step is to determine the t_r, t_m, and f_m values. We developed a simple algorithm that uses static thresholds to extract these values from the signal. The t_r values will be used to train the estimation function f_{est}, while the t_m and f_m values will help profile the current during the temperature maintenance phase. For each thermal parameter, we trained a separate estimation function with details provided in the following subsections.



(a) B_T Est. Function

(b) N_T Est. Function

Fig. 8: Estimation function curve for thermal parameters (red indicates the test data).

1) Print Bed Estimation function: To train the estimation function we collected multiple data points using different target temperature ranges. The collected parameters are tabulated in Table VI (Appendix A), where for each experiment, the starting bed temperature was set to 32 $°C$. Each experiment was conducted three times, with the average value taken for analysis. From the results, it was observed that the bed temperature is raised and maintained in the multiple of 5 sec activity window. Similarly, for f_m the temperature is maintained at a frequency of multiple of 5 and depends on the target temperature, i.e. at higher temperatures the heater needs to turn on more frequently to maintain the temperature.

The measured t_r values from the experiments were used to train the estimation function. We employed cubic spline interpolation that fits a cubic polynomial between each pair of data points to output a smooth curve that passes through them. The plot for the estimation function is shown in Figure 8a. The trained estimation function was then evaluated on untested data points. Given the t_r value observed from the current signal and the initial temperature provided by the user, the function predicts the final target temperature. For example, we tested the estimation function wherein the bed temperature was raised from 21 to 33 °C. The observed t_r from the current signal was 60.21 sec, which, when inputted into the estimator function yielded a value of 11.74. Given the initial temperature of 21°C, the predicted target bed temperature was calculated to be 32.74 °C which is within 1 °C of the original set target of 33 °C.

2) Nozzle Estimation function: Similar to the print bed, multiple experiments were conducted for the nozzle with the observations tabulated in Table VII (Appendix A). For each experiment, we recorded t_r, along with the maximum observed t_m and f_m values, averaged over 3 readings. The nozzle's initial temperature was set to 35 °C. We used the same cubic spline interpolation function as for the print bed, and the plot of the fitted estimation function is shown in Figure 8b.

To evaluate the fitted estimation function, we tested it with an unknown data point. The nozzle temperature was increased to 140 °C from the initial temperature of 45.6 °C. The t_r value (28.43 sec) obtained from the current signal, when inputted to the estimator function $f_{est}(28.43)$ yielded a value of 96.77. Given the initial temperature, the predicted target nozzle temperature was calculated to be 142.37 °C which is close to the original set target of 140 °C.

Parameter	**Target Temp**	**Initial Temp**	**Predicted Temp**	ΔT	**Max t_m**	**Max f_m**
Bed Temperature	60	22.2	61.2	1.2	9.97	20
		40	62.9	2.9	5.11	14.95
		44.6	63.8	3.8	9.97	**20.06**
		44	64.3	**4.3**	5.12	15.08
		50.3	62.03	2.03	**9.97**	15.07
Nozzle Temperature	210	53.8	219.67	**9.67**	9.04	19.3
		39	212.15	2.15	9.18	19.58
		49	217.51	7.51	**9.83**	**20.17**
		51.5	218.24	8.24	9.04	19.15
		51.2	218.37	8.37	9.57	19.4

TABLE III: Bed and Nozzle parameters thresholding for alert generation

F. Thermal Attack Detector

The framework employs multiple thresholds including target temperature, maintaining signal frequency, and maintaining signal width to differentiate between anomalous and baseline behavior. We established these thresholds through extensive experiments to avoid false positives/negatives. For both the nozzle and bed temperature, a test object was printed five times, and the t_r, t_m, and f_m values were recorded. The t_r value is used as input by the estimator function to generate target temperature prediction. The maximum recorded deviation (ΔT) between predicted and actual values is used as a threshold for generating the alert. The results from these experiments are shown in Table III. For bed temperature, the maximum $\Delta T, t_m$, and f_m values recorded were $4.3, 9.97$, and 20.06, respectively; thus thresholds of $5, 15$, and 25 were set for generating alerts.

Similarly for nozzle temperature, the maximum recorded $\Delta T, t_m$, and f_m values from the experiments were $9.67, 9.83$, and 20.17, respectively. Therefore, thresholds of $15, 15$, and 25 were set for generating alerts. For thermal sabotage detection, a deviation in the t_r value will indicate an initial target temperature modification, an increase in t_m value will suggest that the temperature is being increased during operation, and an increase in the f_m value will indicate a decrease in temperature during the printing process.

G. Activity Monitoring

The module passively monitors the printer and generates alerts. The alert algorithm takes in sensor data every 5 minutes, and based on predefined thresholds learned from previous experiments, logs different activities present in the signal. A total of 7 activities including; idle state, SD card insert/remove, nozzle heating, print-bed heating, printer active, nozzle moving, and print-bed moving conditions are logged along with the timestamps by the algorithm. The logs can help identify any unauthorized access to the printer or any anomalous printer behavior during the printer's inactivity state.

VII. Framework Evaluation

We evaluated the proposed framework against nine different attacks across three categories: surveillance (IP theft), sabotage of the print environment, and sabotage of the printed object. These attacks, sourced from the literature [13], were executed by manipulating the open-source Marlin firmware.

A. Surveillance

1) IP theft: The attack, proposed by Rais et al. [13], involves retrieving the print geometry by inserting a malicious SD card into the printer. The compromised firmware captures the sketch of the printed object using various approximations and stores it in the controller's EEPROM region. An authentication token is embedded to identify the attacker's SD card, and once authenticated, the firmware downloads the geometry information onto the card. The insertion and removal of the SD card generate a distinguishable signature in the current signal. The proposed framework leverages this signature to log such activities, allowing for the easy identification of unauthorized SD card activities from the logs. The entire process takes 3 seconds for the firmware to register the card and copy the file. The captured current signal and a snippet of the log file are shown in Figure 9 (Appendix C).

B. Sabotage to Print Environment

In this attack category, the adversary aims to sabotage the printer environment, which includes the printer's components (such as the nozzle, print bed, etc.) and the facility where the printer is located. The framework utilizes its offline monitoring capability to capture, label, and log printer activities along with corresponding timestamps. These logs can later be analyzed to identify any sabotage attempt performed by potentially malicious firmware. To evaluate the framework's effectiveness in detecting such attacks, three firmware-based attacks from the literature were employed.

1) Nozzle Burning: In this attack, the adversary disables the software-defined safety limits in the firmware that regulate the nozzle temperature along with the nozzle cooling fan. These firmware manipulations allow nozzle temperature to exceed safe limits, potentially causing physical damage to the printer nozzle. The monitoring framework successfully generated logs of the nozzle's thermal activity during the attack. To prevent actual damage to the printer, the attack was limited to 5 minutes, during which the temperature reached 300°C. The logs generated as a result of the attack, along with the captured signal, are shown in Figure 10 (Appendix C).

2) Print Your Own Grave: The attack proposed by Rais et al. [13] uses malicious firmware to break the print bed glass, which is securely held in place by end clamps. To achieve this, the firmware first prints a destructive tool to aid in removing the clamps. Once the clamps are removed, the firmware then pushes the print bed glass out of the printer. This attack triggered multiple activities, including nozzle and print-bed thermal activity, the printer entering in active state, print-bed movement, and nozzle movement. The attack took 30 minutes to complete. All the sequences of activities generated during the attack were logged in a file for easy detection by the user. The printed destruction tool along with the captured logs are shown in Figure 11 (Appendix C).

3) Print Facility Air Quality: In this attack, the adversary compromises the print facility's air quality by increasing volatile organic compounds (VOCs) and microparticle emissions [13]. The compromised firmware initiates a high-burst cold extrusion to chip filament particles, followed by an uncontrolled rise in nozzle temperature by disabling the temperature feedback loop. This sequence burns the filament in the nozzle chamber, producing harmful fumes and microparticles. The attack involves two key activities: the printer's active state and nozzle thermal activity. The 1 min captured signal of the attack along with the resulting logs are shown in Figure 12 (Appendix C).

Attack	Attack Temp. (°C)	Attack Magnitude (°C)	Predicted Temp. (°C)	Accuracy (%)
Warping	50	10	47.68	95.15
	40	20	38.45	96.13
First-layer adhesion	35	25	32.96	94.2

TABLE IV: Detection performance of print-bed temperature attacks

C. Sabotage to Print Object

We evaluated the framework's effectiveness in detecting print object sabotage by employing five different attacks. We implemented warping, first-layer adhesion, and filament underflow to demonstrate thermal attacks. To showcase attacks on print geometry, we used object feature scaling and cavity through filament density. We conducted these attacks by printing a bar object ($50 \times 20 \times 3\ mm^3$) using PLA material, with the standard bed temperature for PLA set at 60°C and the nozzle temperature at 200 °C.

1) Warping Defects: In this attack, the malicious firmware alters the bed temperature, leading to warping defects in the printed object due to thermal stresses. To make the attack stealthier, the firmware falsely reports the normal temperature to the user. For a standard print (Print A), the bed temperature is maintained at 60 °C. However, to induce warping, the malicious firmware reduces the bed temperature at the start of the print. Two attack instances are demonstrated alongside the normal print object in Figure 13 (Appendix C). In these attacks, the firmware reduces the print bed temperature by 10 °C (Print B) and 20 °C (Print C). As observed, at 50 °C, warping is minor, appearing only at the bottom right corner of 'Print B.' However, with a more significant reduction to 40 °C, warping is visible around all the edges of 'Print C.' The framework detected both attack instances with up to 96% accuracy as shown in Table IV.

2) First-Layer Adhesion: In this attack the firmware changes the bed temperature such that the printed object doesn't adhere to the build plate, resulting in a garbage print. While the footprint of the attack is obvious the user cannot determine the root cause because of the normal temperature being reported by the firmware. To demonstrate the attack the malicious firmware reduces the bed temperature to 35°C. The result of the attack is shown in Figure 14 (Appendix C), wherein the print object is detached from the print bed after the first layer is completed. The print was stopped once separated from the build plate. With initial temperature at 21 °C the bed t_r value was found to be 65.14 giving the predicted temperature 32.96 °C. Therefore, the framework detected the attacked bed temperature with 94.2% accuracy.

Category	Firmware Attacks	Detection Type	Attack Magnitude	Detection Performance
Surveillance	IP Theft	Logs	3 sec	Manual
Sabotage to Print Environment	Nozzle Burning	Logs	5 mins	Manual
	Print Your Own Grave	Logs	30 mins	Manual
	Facility Air Quality	Logs	1 min	Manual
Sabotage to Print Object	Warping Defects	Alert	10, 20 °C	96.1%
	First Layer Adhesion	Alert	25 °C	94.2%
	Filament Underflow	Alert	30, 50 °C	94.74%
	Object Feature Scaling	Alert	0.04mm	On attack command
	Filament Density	Alert	200%	On first command

TABLE V: Performance evaluation of WattShield on performed attacks

3) Filament Underflow: In this attack, the firmware is altered to lower the nozzle temperature during printing, leading to a filament underflow condition where the nozzle becomes partially or fully clogged. As a result, the affected portion of the print exhibits reduced material density. We conducted two instances of this attack where at the start of layer-5 the nozzle temperature was reduced to 170°C (Attack-A) and 150°C (Attack-B). The results of these attacks are shown in Figure 15 (Appendix C). In Attack-A, the nozzle was partially clogged, resulting in material underflow, while in Attack-B, the nozzle was completely clogged. The framework successfully detected both attacks, with the maximum measured f_m value for Attack-A and Attack-B recorded at 43.91 and 62.54 sec, respectively, both exceeding the set threshold of 25 sec.

4) Object Feature Scaling: In this attack, the adversary alters the object's dimensions by adding an extra layer to the outer wall structure. The malicious firmware first identifies a closed shape that constitutes the outer shell and prints an additional outer shell by shifting the coordinates according to the wall thickness [13]. Figure 16 (Appendix C) illustrates the attacked and the normal print objects. The normal object (Print-A) has three wall structure. However, the attacked print (Print-B) has an additional wall with a thickness of 0.8 mm around it, resulting in an overall dimensional increase — the framework using the kinetic detection module able to detect the attack. After the infill and outer wall print commands, the next expected command was anticipated to take 0.82 sec, however, the prediction on captured results came out to be 0.33 sec due to extra added commands for the outer wall, exceeding the threshold and triggering detection.

5) Filament density: The attack manipulates the extruder-to-filament speed ratio to under-overflow material during printing. The malicious firmware during printing looks for the speed parameter in the move commands and increases it while keeping the extruded filament amount the same. This results in the material being less deposited (under-extrusion). Figure 17 (Appendix C) illustrates the first layer of the attacked print, wherein, the speed is increased by a factor of 2, i.e. from 60 mm/s to 120 mm/s. The framework successfully detected the attack on the first prediction instance (after 5 commands). The expected execution times for these 5 commands were 0.465, 0.135, 0.32, 0.135, and 0.15 seconds, respectively. However, during the attack, the predicted times were 0.26, 0.07, 0.17, 0.08, and 0.09 seconds. The calculated differences for each command (0.205, 0.065, 0.15, 0.055, and 0.06 seconds) exceeded the threshold, thereby triggering an alert.

VIII. Discussion and Limitations

Table V summarizes the detection performance of the framework against the performed firmware attacks. The framework was able to raise alerts or help identify unauthorized actions through manual log analysis. The rest of the section discusses the security, scalability, and limitations of the framework.

WattShield framework Scalability. The framework is designed to monitor print geometries without requiring prior learning. Additionally, it employs a minimally intrusive sensing technique that doesn't necessitate printer retrofitting and can be adapted to any printer using a fused filament fabrication technique. Since power consumption patterns depend on the printer's make and model, printers of the same make and model can be monitored using the same framework.

Injection Attacks. An adversary could attempt to modify sensor readings to evade detection. For example, researchers have shown using electromagnetic interference to modify sensor readings [49]. However, such interference can be nullified using proper shielding and introducing low/high pass filters.

Adaptable Attacks. An adversary could evade detection by manipulating parameters to remain below the defined thresholds. For instance, thresholds of 5 °C and 10 °C for nozzle and bed temperatures, respectively, could allow an adversary to operate undetected. However, the effects of such manipulations should also be considered. For example, PLA has an operating range of nozzle temperatures between 190-220 °C and bed temperatures between 50-70 °C. Consequently, an attack with a low footprint is unlikely to affect the print object. However, for kinetic attacks small dimensional variations below defined thresholds have been proven to affect the part's mechanical performance [27]. Increased sensor sensitivity and sampling can help lower the detection thresholds to identify such attacks.

Extrusion Attacks. While the framework effectively monitors and detects most printing parameter manipulations, filament extrusion was not detectable through the power side channel. As a result, attacks targeting the extrusion process would go undetected. Integrating an additional sensor, such as a microphone to monitor the extrusion state, could further enhance the framework and is proposed as a direction for future research.

IX. Conclusion

As the global supply chain expands, firmware malware threats are growing. Additive Manufacturing, with its applications in security-critical industries, is particularly vulnerable to such attacks. This study introduces WattShield, a monitoring framework designed to detect malicious firmware targeting the printed object and the print environment. WattShield analyzing the printer power consumption could estimate kinetic and thermal process parameters with up to 97% and 96% accuracy, respectively. Using G-code as ground truth, WattShield could detect malicious process variations. WattShield was evaluated on nine different firmware attacks from the literature, including

stealing print geometry and sabotaging the printer and print object. WattShield could successfully detect attacks and generate alerts and logs based on defined thresholds.

References

[1] GE Aviation. (2018) New manufacturing milestone: 30,000 additive fuel nozzles. [Online]. Available: https://www.ge.com/additive/stories/new-manufacturing-milestone-30000-additive-fuel-nozzles

[2] N. Gupta, C. Weber, and S. Newsome, "Additive manufacturing: status and opportunities," *Science and Technology Policy Institute, Washington*, 2012.

[3] Precedence Research, "Additive Manufacturing Market," 2024. [Online]. Available: https://www.precedenceresearch.com/additive-manufacturing-market

[4] B. Imran, M. Ahsan, A. H. Akbar, and G. A. Shah, "D4GW: DTLS for gateway multiplexed application to secure MQTT(SN)-based pub/sub architecture," *Internet of Things*, vol. 26, p. 101172, 2024.

[5] A. Muhammad, B. Afzal, B. Imran, A. Tanwir, A. H. Akbar, and G. Shah, "oneM2M Architecture Based Secure MQTT Binding in Mbed OS," in *2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, 2019, pp. 48–56.

[6] M. Ahsan and M. Ali, "Lsstk: Lightweight solution to preventing stack from buffer overflow vulnerability," in *2023 17th International Conference on Open Source Systems and Technologies (ICOSST)*, 2023, pp. 1–7.

[7] H. Alam, M. S. Yaqub, and I. Nadir, "Detecting iot attacks using multi-layer data through machine learning," in *2022 Second International Conference on Distributed Computing and High Performance Computing (DCHPC)*, 2022, pp. 52–59.

[8] B. Imran, B. Afzal, A. H. Akbar, M. Ahsan, and G. A. Shah, "MISA: Minimalist Implementation of oneM2M Security Architecture for Constrained IoT Devices," in *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, pp. 1–6.

[9] M. S. Yaqub, H. Mahmood, I. Nadir, and G. A. Shah, "An ensemble approach for iot firmware strength analysis using stride threat modeling and reverse engineering," in *2022 24th International Multitopic Conference (INMIC)*, 2022, pp. 1–6.

[10] S. R. Chhetri, S. Faezi, N. Rashid, and M. A. Al Faruque, "Manufacturing supply chain and product lifecycle security in the era of industry 4.0," *Journal of Hardware and Systems Security*, vol. 2, no. 1, pp. 51–68, 2018.

[11] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, "dr0wned–{Cyber-Physical} attack with additive manufacturing," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, 2017.

[12] H. Pearce, K. Yanamandra, N. Gupta, and R. Karri, "Flaw3d: A trojan-based cyber attack on the physical outcomes of additive manufacturing," *IEEE/ASME Transactions on Mechatronics*, 2022.

[13] M. H. Rais, M. Ahsan, and I. Ahmed, "Sok: 3d printer firmware attacks on fused filament fabrication," in *18th USENIX WOOT Conference on Offensive Technologies*, 2024.

[14] E. Kurkowski, A. Van Stockum, J. Dawson, C. Taylor, T. Schulz, and S. Shenoi, "Manipulation of g-code toolpath files in 3d printers: Attacks and mitigations," in *International Conference on Critical Infrastructure Protection*. Springer, 2022, pp. 155–174.

[15] M. McCormack, S. Chandrasekaran, T. Yu, S. Wolf, and V. Sekar, "C3po: A security analysis tool for networked 3d printers (cmu-cylab-19-002)."

[16] L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, and R. Parker, "Cyber-physical vulnerabilities in additive manufacturing systems," in *2014 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2014.

[17] Y. Forihat, C. Taylor, R. Asmar Awad, and I. Ahmad, "Security assessment of lbp16 protocol-based cnc machine," in *18th IFIP International Conference on Critical Infrastructure Protection (ICCIP)*, 2024.

[18] S. B. Moore, W. B. Glisson, and M. Yampolskiy, "Implications of malicious 3d printer firmware," in *Proceedings of the 50th Hawaii International Conference on System Sciences*, 2017.

[19] C. Xiao, "Security attack to 3d printing," *xFocus Information Security Conference*, 2013.

[20] F. E. McFadden and R. D. Arnold, "Supply chain risk mitigation for it electronics," in *2010 IEEE International Conference on Technologies for Homeland Security (HST)*. IEEE, 2010, pp. 49–55.

[21] Y. Wu, J. Wang, Y. Wang, S. Zhai, Z. Li, Y. He, K. Sun, Q. Li, and N. Zhang, "Your firmware has arrived: A study of firmware update vulnerabilities," in *USENIX Security Symposium*, 2023.

[22] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu, "My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 895–907.

[23] M. Ahsan, E. Pak, K. Jackson, M. H. Rais, B. Najarro-Blancas, N. Lewinski, and I. Ahmed, "Biosafe: Bioprinting security framework for detecting sabotage attacks on printability and cell viability," in *40th Annual Computer Security Applications Conference (ACSAC'24)*. IEEE, 2024.

[24] M. Ahsan, M. H. Rais, and I. Ahmed, "Sok: Side channel monitoring for additive manufacturing-bridging cybersecurity and quality assurance communities," in *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)*. IEEE, 2023, pp. 1160–1178.

[25] M. Yampolskiy, A. Skjellum, M. Kretzschmar, R. A. Overfelt, K. R. Sloan, and A. Yasinsac, "Using 3d printers as weapons," *International Journal of Critical Infrastructure Protection*, vol. 14, pp. 58–71, 2016.

[26] M. H. Rais, Y. Li, and I. Ahmed, "Dynamic-thermal and localized filament-kinetic attacks on fused filament fabrication based 3d printing process," *Additive Manufacturing*, vol. 46, p. 102200, 2021.

[27] M. H. Rais, M. Ahsan, V. Sharma, R. Barua, R. Prins, and I. Ahmed, "Low-magnitude infill structure manipulation attacks on fused filament fabrication 3d printers," in *Critical Infrastructure Protection XVI: 16th IFIP WG 11.10 International Conference, ICCIP 2022*. Springer, 2022, pp. 205–232.

[28] M. H. Rais, M. Ahsan, and I. Ahmed, "Sabotaging material extrusion-based 3d printed parts through low-magnitude kinetic manipulation attacks," *ACM Trans. Cyber-Phys. Syst.*, Nov. 2024.

[29] ——, "FRoMEPP: Digital forensic readiness framework for material extrusion based 3D printing process," *Forensic Science International: Digital Investigation*, vol. 44, p. 301510, 2023, selected papers of the Tenth Annual DFRWS EU Conference.

[30] Z. Yu, Y. Chang, S. Zhai, N. Deily, T. Ju, X. Wang, U. Jammalamadaka, and N. Zhang, "{XCheck}: Verifying integrity of 3d printed {Patient-Specific} devices via computing tomography," in *32nd USENIX Security Symposium (USENIX Security 23)*, 2023, pp. 2815–2832.

[31] L. Graves, W. King, P. Carrion, S. Shao, N. Shamsaei, and M. Yampolskiy, "Sabotaging metal additive manufacturing: Powder delivery system manipulation and material-dependent effects," *Additive Manufacturing*, vol. 46, p. 102029, 2021.

[32] J. Gatlin, S. Belikovetsky, S. B. Moore, Y. Solewicz, Y. Elovici, and M. Yampolskiy, "Detecting sabotage attacks in additive manufacturing using actuator power signatures," *IEEE Access*, vol. 7, pp. 133 421–133 432, 2019.

[33] S. R. Chhetri, A. Canedo, and M. A. Al Faruque, "Kcad: kinetic cyber-attack detection method for cyber-physical additive manufacturing systems," in *2016 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*. IEEE, 2016, pp. 1–8.

[34] M. Rott and S. A. S. Monroy, "Power-based intrusion detection for additive manufacturing: A deep learning approach," in *International Conference on Industrial IoT Technologies and Applications*. Springer, 2020, pp. 171–189.

[35] C. Bayens, T. Le, L. Garcia, R. Beyah, M. Javanmard, and S. Zonouz, "See no evil, hear no evil, feel no evil, print no evil? malicious fill patterns detection in additive manufacturing," in *26th USENIX Security Symposium (USENIX Security 17)*, 2017, pp. 1181–1198.

[36] 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, 2018.

[37] M. Wu, H. Zhou, L. L. Lin, B. Silva, Z. Song, J. Cheung, and Y. Moon, "Detecting attacks in cybermanufacturing systems: Additive manufacturing example," in *MATEC Web of Conferences*, vol. 108. EDP Sciences, 2017, p. 06005.

[38] A. Al Mamun, C. Liu, C. Kan, and W. Tian, "Securing cyber-physical additive manufacturing systems by in-situ process authentication using streamline video analysis," *Journal of Manufacturing Systems*, vol. 62, pp. 429–440, 2022.

[39] M. H. Rais, Y. Li, and I. Ahmed, "Spatiotemporal G-code modeling for secure FDM-based 3D printing," in *Proceedings of the ACM/IEEE 12th International Conference on Cyber-Physical Systems*, 2021, pp. 177–186.

[40] S. R. Chhetri, S. Faezi, and M. A. Al Faruque, “Information leakage-aware computer-aided cyber-physical manufacturing,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2333–2344, 2018.

[41] ——, “Fix the leak! an information leakage aware secured cyber-physical manufacturing system,” in *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, 2017, pp. 1408–1413.

[42] S. Liang, S. A. Zonouz, and R. Beyah, “Hiding my real self! protecting intellectual property in additive manufacturing systems against optical side-channel attacks.” in *NDSS*, 2022.

[43] M. Wu, Z. Song, and Y. B. Moon, “Detecting cyber-physical attacks in cybermanufacturing systems with machine learning methods,” *Journal of intelligent manufacturing*, vol. 30, no. 3, pp. 1111–1123, 2019.

[44] A. Al Mamun, C. Liu, C. Kan, and W. Tian, “Real-time process authentication for additive manufacturing processes based on in-situ video analysis,” *Procedia Manufacturing*, vol. 53, pp. 697–704, 2021.

[45] Y. Gao, B. Li, W. Wang, W. Xu, C. Zhou, and Z. Jin, “Watching and safeguarding your 3d printer: Online process monitoring against cyber-physical attacks,” *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, no. 3, pp. 1–27, 2018.

[46] A. Slaughter, M. Yampolskiy, M. Matthews, W. E. King, G. Guss, and Y. Elovici, “How to Ensure Bad Quality in Metal Additive Manufacturing: In-Situ Infrared Thermography from the Security Perspective,” in *Proceedings of the 12th International Conference on Availability, Reliability and Security*, ser. ARES ’17. Association for Computing Machinery, 2017.

[47] S. Li, Y. Chen, X. Chen, Z. Li, D. Fang, K. Liu, S. Lv, and L. Sun, “Powerguard: Using power side-channel signals to secure motion controllers in ics,” *IEEE Transactions on Information Forensics and Security*, 2024.

[48] F. Ding, H. Li, F. Luo, H. Hu, L. Cheng, H. Xiao, and R. Ge, “Deeppower: Non-intrusive and deep learning-based detection of iot malware using power side channels,” in *Proceedings of the 15th ACM Asia conference on computer and communications security*, 2020, pp. 33–46.

[49] Y. Tu, S. Rampazzi, B. Hao, A. Rodriguez, K. Fu, and X. Hei, “Trick or heat? manipulating critical temperature-based control systems using rectification attacks,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2301–2315.

APPENDIX A
ESTIMATION FUNCTION TRAINING DATA POINTS

Avg t_r (sec)	Max t_m (sec)	Max f_m (sec)	Target B_T ($°C$)	ΔB_T ($°C$)
14.95	5	45.01	35	3
24.99	5	50.2	37	5
52.71	5.10	34.98	42	10
73.55	5	30.1	47	15
100.37	5.01	20.05	52	20
135.44	5.02	15	57	25
169.07	10.1	9.93	62	30

TABLE VI: Bed temperature rise time (t_r) starting at $32°C$ and averaged over 3 readings

Avg t_r (sec)	Max t_m (sec)	Max f_m (sec)	Target N_T ($°C$)	ΔN_T ($°C$)
9.69	1.82	41.5	50	15
14.33	3.8	35.3	70	35
21.22	5.4	37.1	100	65
28.01	6.65	28.6	130	95
35.39	7.46	23.9	160	125
42.93	9.12	24	190	155
52.06	9.83	17.2	220	185
61.79	11.21	15.8	250	215

TABLE VII: Nozzle temperature rise time (t_r) starting at $35°C$ and averaged over 3 readings

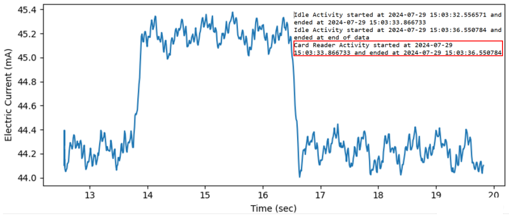
APPENDIX B
G-CODE TIME SERIES CONVERSION

Algorithm 1 Time Domain Transformation Function

```
Require: $x_i, y_i, x_f, y_f, v_{max}, a_{max}$
Ensure: $G_{TS}$
  $x_1 \leftarrow x_i,\ x_2 \leftarrow x_f,\ y_1 \leftarrow y_f,\ y_2 \leftarrow y_f$
  $D \leftarrow \sqrt{(x_2 - x_1)^2 - (y_2 - y_1)^2}$
  $\theta \leftarrow tan^{-1}((y_2 - y_1)/(x_2 - x_1))$
  $S \leftarrow \sqrt{D * a_{max}}$
  if $S < v_{max}$ then                                ▷ Triangular motion
      $T \leftarrow 2\sqrt{D/a_{max}}$
      $t \leftarrow T/2,\ d \leftarrow a_{max}t^2/2$
      while $temp < T$ do
          $t_i \leftarrow temp * \Delta t$
          if $temp < t_i$ then                         ▷ Acceleration phase
              $\Delta d \leftarrow 0.5 * a_{max} * t_i^2$
          else                                         ▷ Deceleration phase
              $\Delta d \leftarrow d + S * t_i - 0.5 * a_{max} * t_i^2$
          end if
          $intp\_x \leftarrow \Delta d cos\theta$
          $intp\_y \leftarrow \Delta d|sin\theta|$
          $G_{TS} \leftarrow insert(t_i, intp\_x, intp\_y)$
      end while
  end if
```

APPENDIX C
FIRMWARE ATTACKS RESULTS

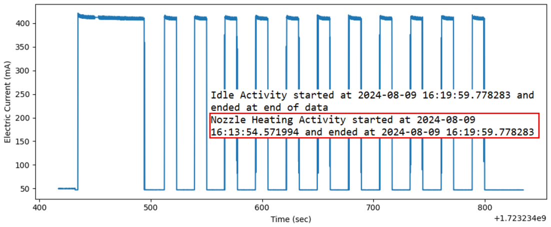
A. IP theft



```
Idle Activity started at 2024-07-29 15:03:32.556571 and
ended at 2024-07-29 15:03:33.866733
Idle Activity started at 2024-07-29 15:03:36.550784 and
ended at end of data
Card Reader Activity started at 2024-07-29
15:03:33.866733 and ended at 2024-07-29 15:03:36.550784
```

Fig. 9: Signal along with logs showing SD card activity.

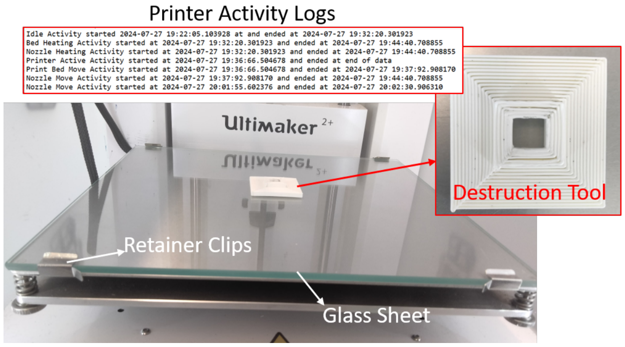
B. Nozzle Burning



```
Idle Activity started at 2024-08-09 16:19:59.778283 and
ended at end of data
Nozzle Heating Activity started at 2024-08-09
16:13:54.571994 and ended at 2024-08-09 16:19:59.778283
```

Fig. 10: Nozzle Burning attack signal with 2 recent activity logs.

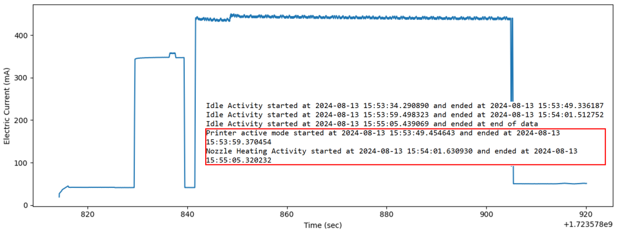
C. Print Your Own Grave



```
Idle Activity started 2024-07-27 19:22:05.103928 at and ended at 2024-07-27 19:32:20.301923
Bed Heating Activity started at 2024-07-27 19:32:20.301923 and ended at 2024-07-27 19:44:40.708855
Nozzle Heating Activity started at 2024-07-27 19:32:20.301923 and ended at 2024-07-27 19:44:40.708855
Printer Active Activity started at 2024-07-27 19:36:66.504678 and ended at end of data
Print Bed Move Activity started at 2024-07-27 19:36:66.504678 and ended at 2024-07-27 19:37:92.908170
Nozzle Move Activity started at 2024-07-27 19:37:92.908170 and ended at 2024-07-27 19:44:40.708855
Nozzle Move Activity started at 2024-07-27 20:01:55.602376 and ended at 2024-07-27 20:02:30.906310
```

Fig. 11: Print Your Own Grave (PYOG) attack along with captured activity logs.

D. Print Facility Air Quality



```
Idle Activity started at 2024-08-13 15:53:34.290890 and ended at 2024-08-13 15:53:49.336187
Idle Activity started at 2024-08-13 15:53:59.498323 and ended at 2024-08-13 15:54:01.512752
Idle Activity started at 2024-08-13 15:55:05.439069 and ended at end of data
Printer active mode started at 2024-08-13 15:53:49.454643 and ended at 2024-08-13
15:53:59.370454
Nozzle Heating Activity started at 2024-08-13 15:54:01.630930 and ended at 2024-08-13
15:55:05.320232
```

Fig. 12: Electric current capture along with logs showing print facility attack.

E. Warping Defects

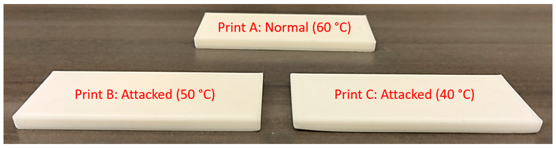


Fig. 13: Warping attacks demonstrated at 50°C and 40°C bed temperature; Print B attack is more subtle with little deformation at the bottom right corner, whereas Print C has deformation around all the corners.

F. First Layer Adhesion

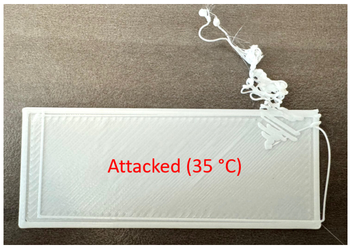


Fig. 14: First Layer Adhesion attack demonstrated at 35°C bed temperature.

G. Filament Underflow

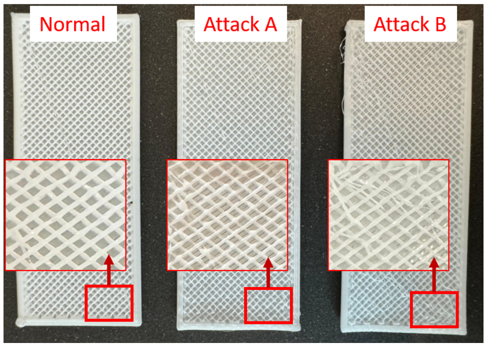


Fig. 15: Filament underflow attack demonstrated at 170°C (Attack A) and 150°C (Attack B) nozzle temperature.

H. Object Feature Scaling

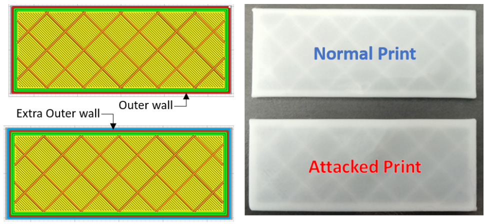


Fig. 16: Scaling attack by adding an extra outer layer.

I. Filament Density

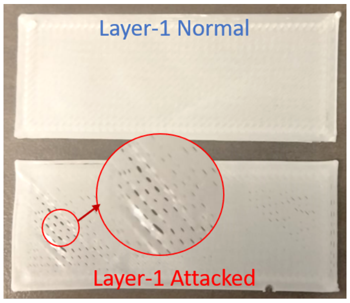


Fig. 17: Filament density attack by increasing the print speed.