UNIVERSITY OF CALIFORNIA,
IRVINE

Cyber-Physical Attack Detection and Localization in Additive Manufacturing Systems

THESIS

submitted in partial satisfaction of the requirements
for the degree of

MASTER OF SCIENCE

in Electrical and Computer Engineering

by

Ashley Sayuri Masuda

Thesis Committee:
Professor Mohammad Abdullah Al Faruque, Chair
Professor Fadi J. Kurdahi
Assistant Professor Yasser Shoukry

2023

DEDICATION

To my parents who have been by my side through the best and worst of times. May your love and devotion be recognized as the reason I am here today.

To the friends I have made along this journey, who have supported and encouraged me this past year, I wish you all healthy and happy memories ahead.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my committee chair, Professor Mohammad Abdullah Al Faruque, with his constant support that has allowed me to grow and prosper as a researcher. For allowing me this opportunity to work alongside some of the best and brightest minds in your laboratory, I sincerely thank you.

I would like to thank my committee members, Professor Fadi J. Kurdahi and Associate Professor Yasser Shoukry, for taking the time to provide me with excellent guidance and helpful feedback to create a refined and well-received thesis paper.

I also want to acknowledge my fellow lab mates, Yasamin Moghaddas and Rozhin Yasaei, for their time dedicated to this project, perseverance through setbacks, and insightful knowledge of Neural Networks and Side-Channel Analysis.

ABSTRACT OF THE THESIS

Cyber-Physical Attack Detection and Localization in Additive Manufacturing Systems

By

Ashley Sayuri Masuda

Master of Science in Electrical and Computer Engineering

University of California, Irvine, 2023

Professor Mohammad Abdullah Al Faruque, Chair

Additive Manufacturing (AM) systems continue to revolutionize various industries with their ability to mass-produce complex and custom products. This growing influence, however, offers its own new set of cybersecurity vulnerabilities. Analog side-channel emissions across the cyber-physical domain can provide attackers with enough information to compromise the integrity and performance of the AM Process Chain. Leakage of confidential CAD models, sabotage of the product's structure, and manipulation of the printer's internal instruction sequence are examples of the potential consequences. Our methodology incorporates multi-modal machine-learning techniques via a Graph Neural Network (GNN) to accurately detect and localize the source of discrepancies. By identifying the nature of the anomaly, companies and consumers can benefit from effective/rapid threat detection and proactive vulnerability mitigation while being assured of the product's quality. The logic behind this model is anticipated to be used in numerous other fields beyond 3D printing applications.

Chapter 1

Introduction

Recent studies on Additive Manufacturing security have become an area of growing interest in the cybersecurity community. For instance, advances in medical practices, including the creation of artificial tissues/organs [44] and effective new drug structures using stereolithography [31], highlight the flexibility and effectiveness of the practice. However, as 3D manufacturing becomes more commercialized and computationally dependent, these devices become increasingly prone to malfunction and sabotage [54]. To safeguard the integrity and security of this promising industry, it is essential to perform diverse and thorough analyses to identify possible disruptions that could negatively impact the functionality and performance of the desired product.

Taking advantage of AM side-channel emissions and modern machine-learning algorithms has led to innovative methodologies and adversary models, such as sensor fusion [23] and digital product duplication [36]. Papers [51] and [40] discuss the comparisons of physical and cyber data from network- and host-based IDS's for intrusion detection, while methods in [27] utilize image processing through convolutional neural networks. Despite the promising results of these projects in malware detection, there has been relatively less emphasis on differentiating between particular types of anomalies (e.g. hostile attacks and environmental interruptions).

At first, the distinction between anomalies and malware might seem insignificant in the context of Additive Manufacturing, for the result is equally unusable in either case. However, delving deeper into this classification approach could have significant ramifications for AM devices' long-term security and effectiveness. A quick and detailed incident response to users regarding potential hostile attempts to compromise the system or instances where environmental debris becomes entangled during printing enables proper troubleshooting and diagnosis. Early detection of any product-related issues during this stage can help prevent future complications from arising.

It is within our interest to develop a multi-modal machine-learning algorithm that uses sensor fusion of vibration, acoustic, magnetic, and power side-channel emissions to accurately distinguish between malicious malware and irregular anomalies.

1.1 Research Challenges

While there have been previous studies on AM side-channel emissions [1] [2], collecting data from various external sensors to create a sufficiently large dataset for the learning phase of our GNN model in order to ensure the accuracy of the attack localization classifiers presents a significant challenge. The dataset needs to illustrate consistent patterns during training for accurate forecasting of future values, as well as be extensive enough to differentiate abnormalities under zero-day conditions.

Another challenge involves precisely identifying the behaviors exhibited by various types of anomalies, whether malicious, environmental, or user-based. The classifier for localization should be capable of distinguishing specific analog signals associated with each kind of anomaly, producing reliable and consistent classification results. Multiple experimental scenarios must be implemented to validate the classifier's performance across different datasets, emulating its effectiveness in real-world situations.

The following are recognized as research challenges for our experiment:

1. Acquiring a comprehensive and synchronized dataset, consisting of both sensor and G-code information for effective model training/testing.
2. Designing an algorithm for assigning attack labels during the testing phase based on compromised times defined by modified G-code deviation
3. Generating a thorough multi-modal Graph Neural Network with optimized weight correlations between sensors that produce consistent results
4. Testing the accuracy and reliability of classifiers to identify the nature of the anomaly: G-code or sensor

1.2 Contribution

Past lab research have established a measurable and dependable connection between acoustic side-channel analog emissions and instructions in the cyber domain [15][1]. Recent experiments have further explored this relationship by employing an algorithmic multi-modal sabotage attack system to demonstrate the correlation between side-channel data and abnormal behavior [55]. Their model incorporated various sensor types for increased system understanding and precision. Our research contributions will involve applying neural network machine-learning techniques coupled with post-processing attack localization methods. This comprehensive approach aims to advance the field by leveraging sophisticated computational algorithms and analysis to further enhance the detection and localization of attacks in the context of side-channel emissions.

The novelty of our contribution to this project can be summarized as follows:

1. **Graph Neural Network Model**: Application of multi-modal neural network model in an Additive Manufacturing environment capable of accurate forecasting of side-channel behavior

2. **Adversary Recreation**: Injection, modification, and deletion of discrepancies within G-code printer instructions and external sensor data to emulate attacks along various points in the AM Process Chain

3. **Attack Label Generation**: Supervised labeling of tampered data under the assumption that the user has no prior knowledge of G-code modifications

4. **Attack Localization**: Classification for both G-code and sensor anomalies based on their error scores from a calculated threshold for future predictions

1.3 Motivational Example

Industrial integration of 3D manufacturing into a company's products and technology has become increasingly popular in a variety of mass-scale applications [6]. The potential impact it has on corporations' supply chains is an important topic of discussion to understand the reasoning behind its explosive growth over the past decade. Generative, facilitated, and selective services have flourished in response to the growing demand for Additive Manufacturing customized and applicable results [42][24].

From a cybersecurity standpoint, authenticating 3D-printed products' consistency, functionality, and durability is imperative to ensure the quality of results and the safety of its benefactors. For example, when creating a complex part in a car model, one of many unforeseen problems could occur: an attacker injects malware or additional instructions into the automated system, a blunt external force damages some of the mechanisms, exposure to the elements affects the printer's consistency of product, etc. Recognizing the difference in quality early on in the manufacturing process will help avoid potential risks down the line, such as improper fitting of customized parts within the overall system, unexpected breakage during operation, or the ultimate failure of the entire car to function as intended.

Identifying and localizing these errors involves training a neural network with a large dataset of sensors' side-channel readings. Taking it a step further, our process consists of using a classifier to categorize the nature of the anomaly by backtracking through the data to find the most likely source of the issue.

Challenges in industry and construction have been identified among seven categories: material, printer, software/computational, architecture and design, construction management, regulations, and stakeholders [20]. Our research tackles the issues of printer-related and computational errors using machine-learning techniques to observe and effectively mitigate the possibility of both faults and malware early in the processing pipeline.

Chapter 2

Background and Related Works

This section reviews relevant topics and current research on Additive Manufacturing and its cybersecurity concerns. We compare and analyze previous works' progress to formulate the best approach for this project.

2.1 Additive Manufacturing Vulnerabilities

Cybersecurity attacks within Additive Manufacturing systems can be localized into three sections along the AM process chain: the CAD Files, the Network, and the 3D printer itself [34]. Computer-Aided Design (CAD) files contain details for the product's overall design, which can be converted to stereolithography instructions for the printer to read. Stereolithography involves the creation of 3D models through the deposition of filament materials layer-by-layer. These design files are susceptible to extraction through network connections [49]. Conducive to this experiment, vulnerabilities within the 3D printer and Network space include modified firmware [35][47], side-channel emission analysis [48], printer spoofing [19], and file tampering during data transfers [34]. This involves physical-to-cyber and cyber-to-physical domain security analysis to identify potential breaches in confidentiality [16]. Figure 2.1 outlines the general process:

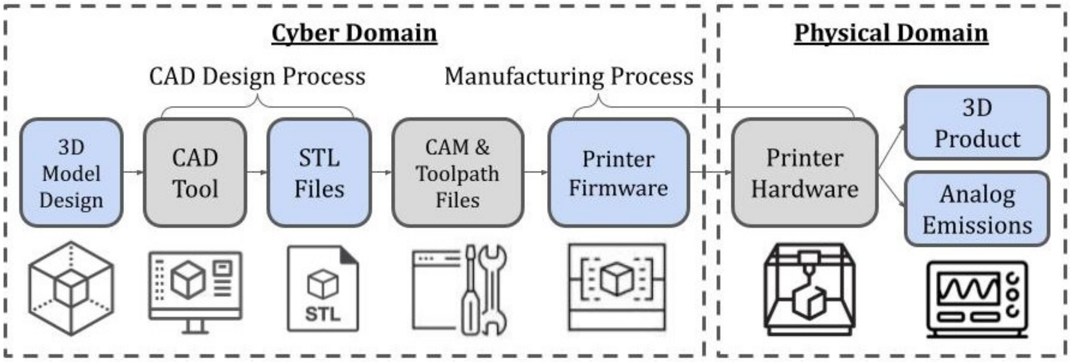

Figure 2.1: AM Process Chain

In previous works, attack vectors were identified and injected during CAD Modeling, STL and Toolpath file processing, and the one-site machine parameter configuration. The experiments were conducted under the assumption that these modifications were made via network connectivity and physical access ports. Access to the software compilation toolchain, where attackers can drastically increase the amount of leaked information while remaining undetected, present a new set of confidentiality concerns [9]. Analog emissions from external sensors collected data from 3D prints that were not previously tampered with and compared them to the sabotaged attack prints [55][4]. Quantifying cross-domain vulnerabilities of physical-to-cyber system attacks is imperative to re-evaluate potential countermeasures in future product iterations [4][13][3]. In addition, detection of zero-day kinetic cyber-attacks through mapping performance between analog emissions and cyber-domain data was conducted with a 77.45% accuracy, providing a solid proof of concept for related sensor behavior [10]. Acoustic side-channel studies for digital twin recreation has also proved possible, collecting data on print extrusion motors to evaluate position and velocity at a fixed sampling rate, ultimately recreating a software duplicate of the intended print [15][1].

Studies have been done to help manage the risk factors involved in the construction industry, where 3D printing has become a potential future resource [33]. Any data leakage or malware modification to the design can have huge implications on the structure's integrity and the company by association. Examples of more specific types of attacks that have become increasingly more viable with the popularity of 3D printers involve: manipulating components of the printer to cause physical disruption to the printing process [43], altering/injecting print commands that appear valid but will fail upon application [35], collecting leaked proprietary data [34], and overloading the printer with tasks to the point of interruption.

2.2 Side-Channel Analysis

The concept of side-channel attacks (SCA) was introduced over 20 years ago under the pretense that the execution of specific processes will result in a physical leakage that can be quantified. SCAs became a primary concern in cryptography, where attackers could determine decryption keys that were thought to be secure. These unintentional emissions include but are not limited to acoustic [17][1], electromagnetic [45], thermal [14][2], power consumption [41], vibration [29], magnetic [22], and photon side-channels [46][26][8]. While there are methods in mitigating the chances of an SCA - such as masking or encasing high-processing components to avoid electromagnetic information leakage or randomizing signal values during power consumption analysis via algorithms [28] - recent breakthroughs in deep-learning have allowed attackers to create multi-input models to create a systematization of knowledge (SoK) that can bypass any one countermeasure [38].

machine-learning techniques are more effective than pre-existing template attacks when developing an exploitable leakage model based on physical data [37][30]. Template attacks are a type of power consumption analysis where the adversary has access to the same machine/system as the victim. These types of attacks should be the most effective when exploiting a system. However, learning algorithms and deep-learning have advanced to the point where SoK and wholistic data compilation has proved equally, if not more successful. Just as this behavior can be analyzed to inject malware into the system, a predictive model can be trained to emulate the future readings of these emissions.

As mentioned previously, research in acoustic analog emissions present a concerning breach in confidentiality of Additive Manufacturing products. Correlation between this data leakage and the G-codes of the internal printer commands has been proven with over 80% accuracy [36][11]. Specifically acoustic and vibration side-channels have demonstrated a high percentage of mutual information with the 3D printer's control parameters, in comparison to magnetic and power modalities [55][8]. In summary, accurately defining the relation between side-channel data and G-code instructions executed by the printer will yield more substantial results needed for adversary recreation and develop the proper methods to counteract them.[12].

2.3 Graph Neural Networks

In the past decade, there has been a significant advancement in the maturity of neural networks. Graph Neural Networks, in particular, are machine-learning models designed to be highly proficient in processing graph-structured data, where edges between nodes are updated with the weights and probabilities of their behavioral connections. The growing popularity stems from their unique capability to capture intricate relationships and patterns within multiple input data. Model performance analysis usually focuses on one of the following techniques: node classification, link prediction, and clustering [56]. Applications of learned networks have expanded into social networking [52], drug side-effect prediction [5], and computer vision deep-learning techniques for image processing [39].

Graph Neural Networks (GNNs) have become prominent since they can process and predict non-Euclidean data [53]. Multivariate time-series data can be formatted as a graph, where the nodes represent featured data types and the edges emulate the relation between them, whose weights are determined by correlation matrix. A graph is denoted as $G = (V,\ E)$, where V is the set of nodes and E is the set of edges. The neighborhood of a node would be $N(i) = \{j \in V \mid e_{i,j} \in E\}$.

GNNs can aggregate the information of a node's neighbors to develop a richer representation of a node and its relationships with other nodes [53]. This process produces a better understanding of the graph's topology. We train a GNN to learn the inter-dependencies between nodes, updating node embeddings after each training iteration. **Sensor embedding** is the process of representing separate data in a flexible and comprehensive way, such as employing a multidimensional **embedding vector** to analyze any underlying behavioral correlations. After training these vectors to imitate the performance of the system, we test the GNN to make forecasts for future timestamps.

Our approach builds on the novelty presented in [18], which proposes a Graph Deviation Network (GDN). As the name suggests, GDN learns the structure of a graphical representation of a sensor network and labels deviations from the learned baseline as anomalous. For further insight on GDN processes, see Section 3.3.

2.4 Attack Localization

Anomaly detection methods have been used to determine the accuracy and validity of systems. Examples include unsupervised graph clustering, neural networks, parametric statistical modeling, and algorithm-based prediction [7]. Although these types of analyses are valid, in some cases, knowing exactly where the operation is compromised provides a much more insightful and practical course of action. This is known as **attack localization**. By understanding the source of the anomaly, the user can backtrace what parts of the system are affected and appropriately classify the nature of the error.

The methodology of pinpointing the source of security breaches is the employment of **classifiers**. These algorithms draw upon network traffic and, in our case, external sensor readings to identify attack types. Standard prediction algorithms whose performance has been tested against one another include Adaptive Boosting (AB), Classification And Regression Tree (CART), K-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), Logistic Regression (LR), Multi-Layer Perception (MLP), Naive Bayes (NB), and Random Forest (RF) [25]. Determining which classifier best suits specific experiments is highly dependent on the application. Future work in this field attempts to combine these methods, where ensemble methods train and compare multiple classifiers to output the best result, while hybrid methods use different classifiers at various stages of training iterations [32].

Not all anomalies should be correlated to malicious attacks. Harmless types include operational events such as power outages and human mistakes, flash crowds involving large amounts of traffic, and measurement anomalies during data collection [21]. For the purposes of this experiment, these lapses in system functionality will be localized to external sensor data. Supposed attacks on the printer will be limited to G-code anomalies extrapolated from sensor data. That being said, it is essential to recognize that anywhere along the AM Process chain that compromises our target areas of interest - sensor and G-code - should be considered a potential cybersecurity threat during application.

Chapter 3

Methodology

In this section, we explain the pipeline of our methodology in Figure 3.1 at each of the following stages: data collection, data pre-processing, GDN model generation (training), adversary model and attack label creation, and attack localization.

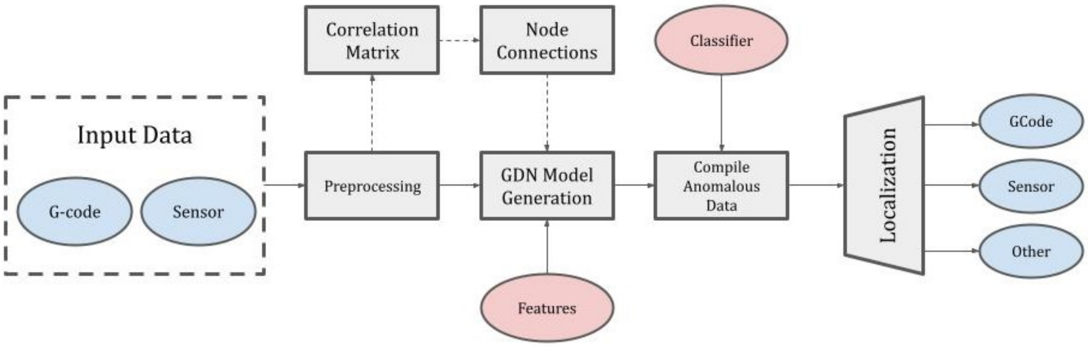

Figure 3.1: Process Chain for Experiment

3.1 Data Pre-processing

The dataset used to train our model includes both sensor and printer data. In order to assure that the data collected internally and externally is being properly processed, the timestamps associated with each are synchronized. Not only do these modified files need to be compared line-by-line to the original, but the pre-processing stage needs to take into account G-code execution. Once these files execute along the same timeline, analysis of injected G-code instructions is used to determine the attack label for testing the model.

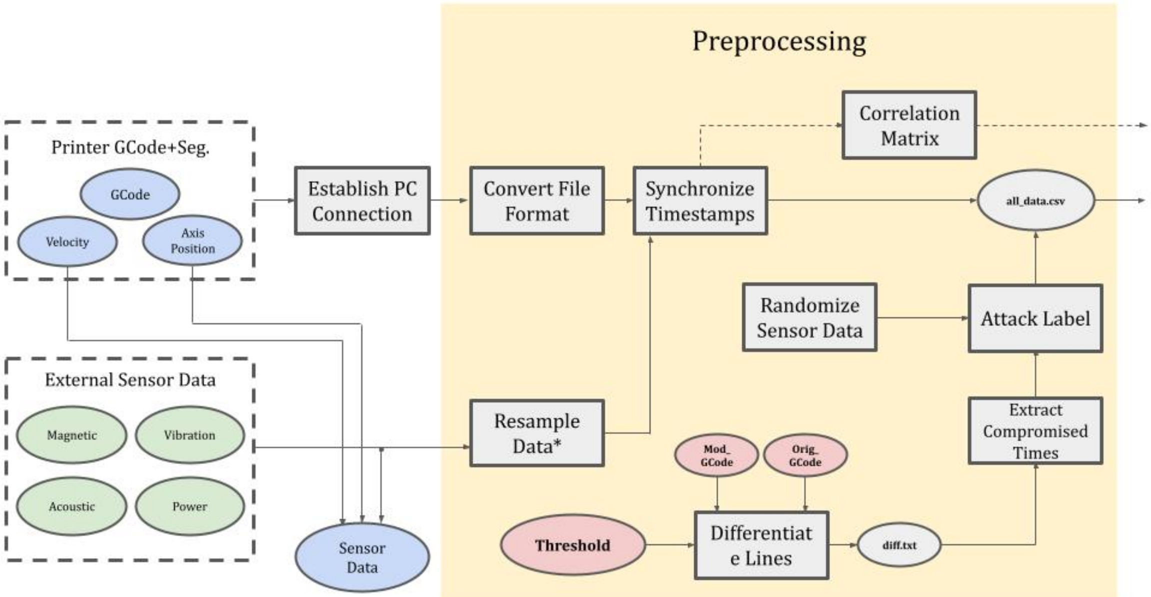

Figure 3.2: Preprocessing Pipeline

3.1.1 G-code Synchronization

The files to be processed involve two types: original and modified data. The untouched data is collected as a control group for training the GDN. Modified data contains injections of randomized G-code instructions, which range in frequency and magnitude. Within each of these categories are three types of data: (1) G-code instructions, (2) position and velocity of the printer head, and (3) external sensor data. Synchronizing this information into a singular test file requires extensive data processing. Our project will be handling the information collected from previous experiments and, thus, will need to address the following issues before applying our model:

1. **Timezone Difference**: The internal clock of the printer is executing at a rate approximately 7 hours ahead of the PCTime.

2. **PC Time vs Printer Time**: Timestamps have been assigned for each G-code instruction for both PCTime and PrinterTime. However, PCTime has been deemed unreliable with repeat values and missing information.

3. **Data Collection Start Time**: The time when the external sensor data is collected does not necessarily match with the time that the G-code instructions are executed.

Starting with the first point, all G-code instruction timestamps are shifted by 7 hours to match the PCTime. Next, the average difference between PrinterTime and PCTime is analyzed with a 100 datapoint sampling rate. PCTime for the G-code instructions is updated to this value. A G-code instruction is selected to evaluate the start time, and its associated timestamp will determine when the print file begins. Understanding that G-code lines need to be synonymous with the modified version as well as other print designs is imperative for the following steps. The process of synchronization is shown in Figure 3.3.

3.2 Adversary Model

The novelty of this paper can be attributed to analyzing and testing different methodologies of anomaly classification. Our experiment will utilize multiple side-channel emissions and training models to create the most sophisticated way to accurately flag a potential discrepancy. For this experiment, there are three ways of classifying the data:

- **Compromised G-code Instructions**: an unknown attacker injects/changes corrupted G-code into the printer which is then executed by the compiler. The results may range from disfigurement of the product to permanent damage of the equipment.

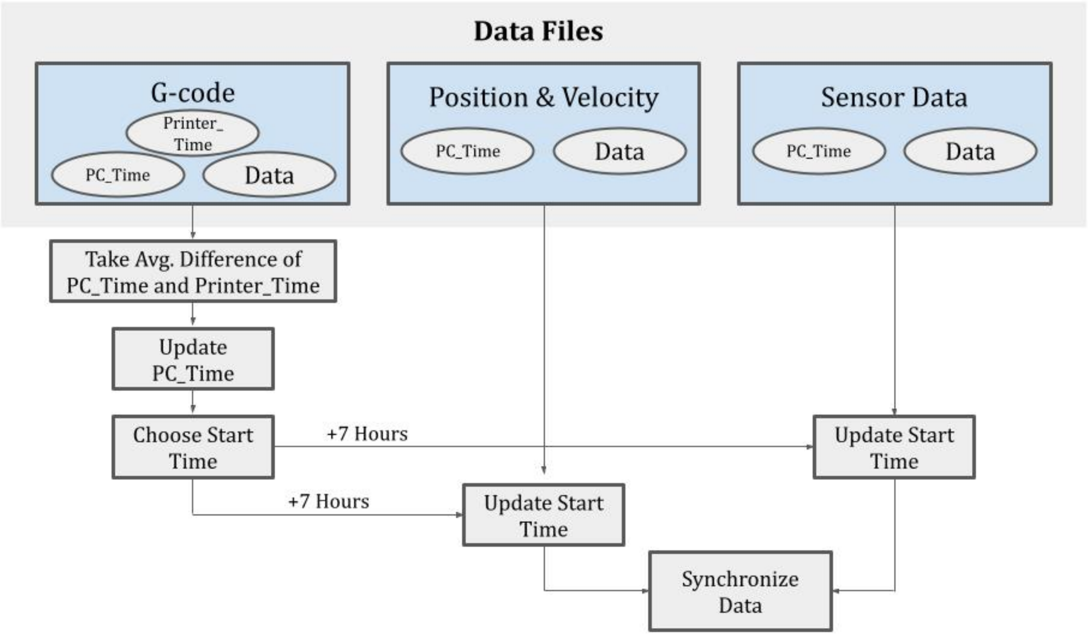

Figure 3.3: Pipeline of data synchronization between G-code instructions, printer-head position/velocity, and external sensor data

- **Faulty Sensor Data**: external sensor data which serves as a check for detecting G-code anomalies may experience unexpected readings that are outside a specific magnitude of standard deviation. Sensor malfunctions and environmental hazards are considered.

- **Normal/Non-Adversary**: non-tampered and acceptable data whose values fall within a set threshold defined by the GDN.

To observe the model's reaction when exposing the original data to the above modifications, three experiments of G-code only anomalies, sensor only anomalies, and a combination of both were conducted.

3.2.1 G-code Attack Label

Labeling where the data has been determined as anomalous is imperative for an accurate model during training. Generating these test files involve supervised differentiation between original and modified G-code files. These files are then compared to a set threshold from the pre-processing stage, where any values above said threshold are deemed as anomalous. After the synchronization of the internal and external data as seen in Figure 3.3, if the combined data timestamp falls within an anomalous time range - determined by the G-code - an attack label of ”1” is assigned. Otherwise, the data is seen as non-tampered with a ”0” label. Algorithm 1 contains pseudo code that illustrates the process of assigning the attack label to our test file.

Algorithm 1: Pseudocode for Generating Attack Label

```
Input  : Modified Timing+G-code file, external sensor data, threshold
Output: Sensor and position data w/ attack label
Isolate G-code information from original and modified Timing files;
Create diff.txt that isolates differences;
Parse through each line in diff file and compare values to the threshold;
if $|original\_value - modified\_value| <= threshold$ then
    Line considered NOT anomalous
else
    Anomalous Data
end
Record timestamps of anomalous data under $anomalous\_timeranges$;
Synchronize $anomalous\_timeranges$ with sensor data;
Anomalous lines have their timestamps recorded, where an $anomalous\_timeranges$ is determined for potential compromised data:;
if Timestamp in modified sensor data in $anomalous\_timeranges$ then
    AttackLabel=1
else
    AttackLabel=0
end
```

3.2.2 Sensor Anomaly Injection

Using a similar logic to the G-code, sensor anomalies will also be determined via a predetermined threshold value. By evaluating the average value of each sensor throughout the print, the standard deviation can be determined:

$$\sigma = \sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \overline{x})^2} \tag{3.1}$$

By conducting tests on different ranges of standard deviations, we can determine the threshold at which modified values are considered anomalous during the testing phase. It enables us to simulate environmental disruptions, sensor malfunctions, and printer readings defects by injecting unknown and abnormal sensor data. It is important to note that these changes are implemented, assuming the original sensor data is intact and trustworthy. Next, modification starting points are randomly selected, separated by a distance of at least 500. After these starting points, a specific sensor's data is changed within a randomized size range. These steps provide a test dataset similar to the expected abnormal behaviors mentioned previously, such as individual sensor malfunctions and external force interference.

3.3 Graph Deviation Network

GDN model generation begins by creating a graph with the sensors as nodes. These nodes are inter-connected by directed edges whose dependency patterns are asymmetrical. It selects the *top k* nodes with the highest cosine similarity between the source node embedding, x_i, and any other node embedding, x_j, to produce e_{ji} for each candidate relation sensor i is dependent on (as seen in Equation 3.2). The adjacency matrix, A_{ji}, captures the information of related sensors based on the computed cosine similarities. It is used to construct the layers of the model:

$$e_{ji} = \frac{x_i^T x_j}{|x_i| \cdot |x_j|} \tag{3.2}$$

$$A_{ji} = \begin{cases} 1, & \text{if} j \in Top - K(\{e_{ki} : k \in S_i\}) \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

where S_i is the set of embeddings that does not include x_i.

In order to capture the relationships between connected nodes, the GDN uses a Graph Attention Network (GAT) [50]. The GAT combines the node embedding's information with that of its neighbors using a sliding window of predetermined size for both training and testing. **Attention coefficients** are generated to define the relative importance of information to one another. The feature extraction process results in aggregated representation, $z_i^{(t)}$:

$$\mathbf{z}_i^{(t)} = ReLU\left(\alpha_{i,i}\mathbf{W}\mathbf{x}_i^{(t)} + \sum_{j \in N_{(i)}} \alpha_{i,j}\mathbf{W}\mathbf{x}_j^{(t)}\right) \tag{3.4}$$

The three equations below are used to calculate the attention coefficients, α:

$$\mathbf{g}_i^{(t)} = \mathbf{v}_i \oplus \mathbf{W}\mathbf{x}_i^{(t)} \tag{3.5}$$

$$\pi\left(i,j\right) = LeakyReLU\left(\mathbf{a}^{T}\left(\mathbf{g}_{i}^{(i)} \oplus \mathbf{g}_{i}^{(j)}\right)\right) \tag{3.6}$$

$$\alpha_{i,i} = \frac{exp\left(\pi\left(i,j\right)\right)}{\sum_{k \in N_{(i)} \cup \{i\}} exp\left(\pi\left(i,k\right)\right)} \tag{3.7}$$

, where W is a trainable weight matrix that performs a linear transformation on a node's feature vector, $\mathbf{x}_{i}^{(t)}$. $\mathbf{g}_{i}^{(t)}$ concatenates the sensor embedding with the linearly transformed result. LeakyReLU is used to introduce non-linearity, where the attention coefficients are then normalized with a softmax function 3.7. Predictions are made by taking each aggregated embedding and element-wise multiplying it with its time-series embedding, v_i. This is done for each node to form the input of a fully-connected layer, whose result is a vector of predictions, $\hat{s}^{(t)}$, for each sensor value at time step, t:

$$\hat{s}^{(t)} = f_{\theta}\left(\left[\boldsymbol{v_i} \circ \boldsymbol{z_i^{(t)}}, \ldots, \boldsymbol{v_N} \circ \boldsymbol{z_N^{(t)}}\right]\right) \tag{3.8}$$

The loss function for minimization applies the Mean Squared Error, which uses the predicted output, $\hat{s}^{(t)}$, and the ground truth data, $s^{(t)}$.

When classifying timestamps as anomalous, we continue with the GDN author's method by calculating error scores at time t for each sensor in the dataset [18]:

$$Err_i(t) = |s(t)_i - \hat{s}(t)_i| \tag{3.9}$$

The error scores are then normalized in order to smooth out any extreme deviations due to the diversity of sensor characteristics:

$$a_i(t) = \frac{Err_i(t - \widetilde{\mu_i})}{\widetilde{\sigma_i}} \tag{3.10}$$

where $\widetilde{\mu_i}$ and $\widetilde{\sigma_i}$ are defined as the median and inter-quartile range of each sensor's array of errors across the time window. The anomaly score at time t is the maximum $a_i(t)$ across all nodes. The reasoning behind finding the maximum value is that in real-world scenarios, it is reasonable to assume that the cause of anomalies localized within the sensor data would only affect a fraction of the dataset as a whole. The highest F1-score is then selected as the threshold. Any error score greater than the threshold is labeled as anomalous, while any score that is within the threshold is considered normal. More information about the technical details of GDN can be found in [18].

Chapter 4

Evaluation

For this section, we first evaluate any thresholds and pre-processing steps necessary for optimal model performance. Next, we describe the layout of our experiment, from tools and equipment to device setup. The discussion portion assesses the precision and F1 score of the GDN model during the testing phase.

4.1 Experimental Setup

The equipment used for verifying our methods will involve the Ultimaker 3 3D-printer, a Data Acquisition (DAQ) device, two Arduino micro-controller boards coupled with MCP4725 boards for digital-to-analog conversion (DAC), and a personal computer for managing communication and data acquisition. The DAQ served as a hub to connect the GPIO of the external sensors with a synchronized timestamp.

Similar to [55], our side-channel analysis will consist of three 3-axis magnetometers, three 3-axis accelerometers, four high-definition microphones, a DC current clamp, and internal sensor data from the 3D printer. These are located on the sides of the printer, as well as on

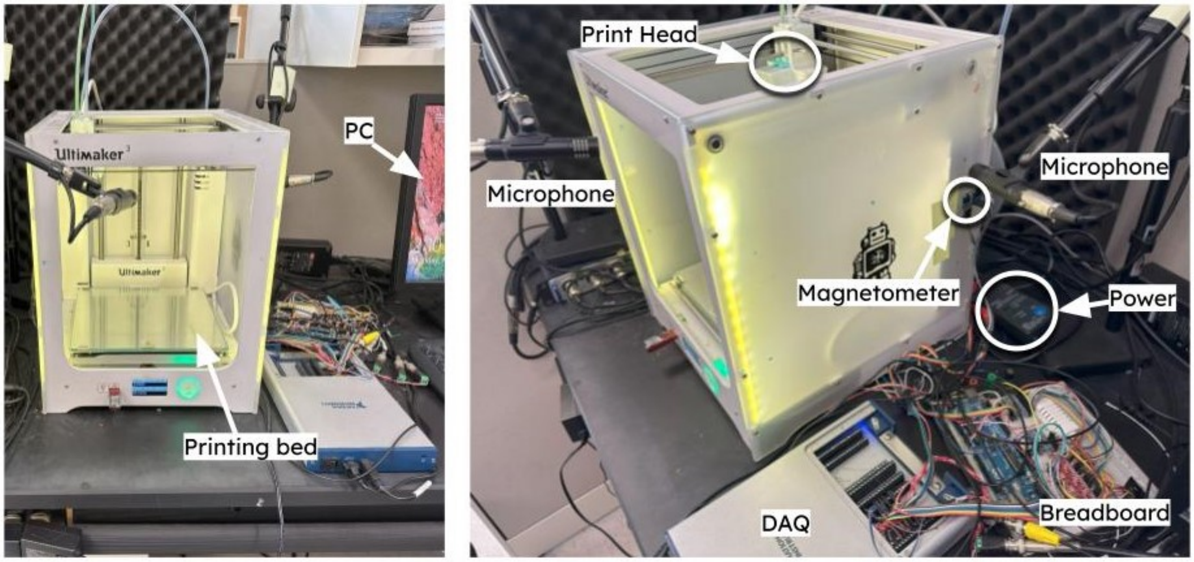

Figure 4.1: Experimental Setup of Printer for Data Collection

the stepper motors that create the most vibration. Internal sensor data includes the position of the printer head through X, Y, and Z coordinates.

Regarding data collection, a connection is established between the printer and the PC to transmit the printer's internal data. At the same time, the DAQ compiles and sends external sensor data at a fixed sampling rate of 0.05 milliseconds. The combined dataset consists of 24GB of data, divided into 48 files of 500k rows each. Each timestamp within these 500k data points encompasses all analog sensor and printer head positioning/velocity values. The synchronization of these timestamps is explained in Section 3.1.

The positioning data was filtered to feature X-, Y-, and Z-axis movements exclusively to capture the actions of the printer extruder in a 3-Dimensional space (A_x, A_y, A_z). Due to the limited Z-axis movement for the print through testing, since the models chosen were relatively flattened, only the X-velocity and Y-velocity were recorded as part of the trained dataset (V_x, V_y).

4.2 Threshold Analysis

The logic in Algorithm 1 was executed for multiple G-code modified datasets of 200k data points with a resampling rate of 5 data points. Due to the 0.05ms sampling rate for sensor data in comparison to the inconsistent rate of G-code instructions, some of the extraneous data was excluded in order to properly assess the general ratio of compromised data points. A default threshold of +/-5.0 was established, maintaining an average of 9.71% of anomalous data (See Table 4.1).

Table 4.1: Attack Label Threshold Analysis

Anomaly Percentage from 200k Resampled Dataset		
Threshold	Anomalies	Percentage
1	25212	12.13%
3	21811	10.49%
5	20190	9.71%
7	19629	9.44%

Various methodologies were employed to create a balanced adversary model necessary for testing. As seen in Figure 4.2, achieving around a 10% anomaly ratio involved carefully refining a supervised detection algorithm, as well as taking into account a threshold based on observed irregular behavior. In other words, by monitoring the behavior of sensor and G-code values during printing, a reasonable threshold could be extrapolated, where it could be tested amongst various practices for attack labeling. Among these include developing an in-depth line-by-line comparison (Method 1), flagging the surrounding data as also being untrustworthy (Method 2), and labeling the rest of the data as compromised once an anomaly threshold is broken until an acceptable value is reached (Method 3). Our study concluded that due to the nature of the G-code sampling rate and the accuracy of the threshold during testing, Method 1 was used with a threshold of +/-5.0.

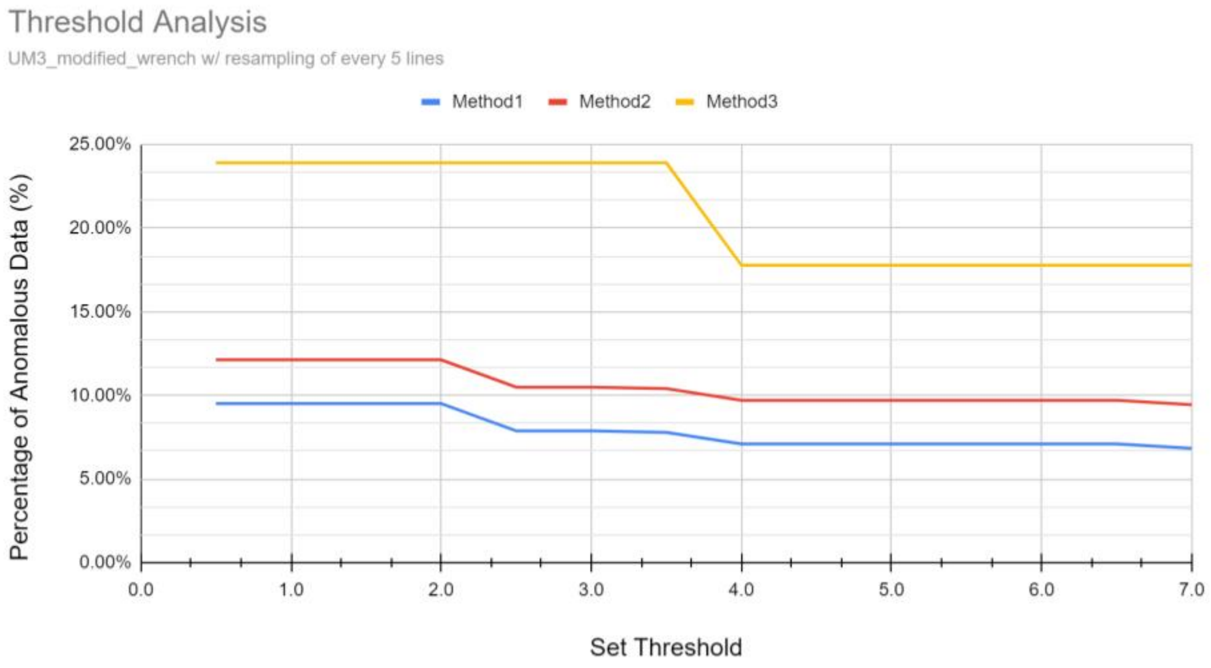

Figure 4.2: Methodology Performance of Adversary Model Threshold Testing

Similarly, in order to emulate anomalies generated by environmental interference or malfunctioning sensors, values from the original sensor data were modified by a calculated threshold of 12 magnitudes of standard deviation above and 15 magnitudes of standard deviation below the predicted value. This extensive range allows for unintentional noise and false positives not to affect the overall accuracy of the results. In the case of both G-code and sensor anomaly interference, data lines that were previously labeled as compromised by an attack label from the G-code injections were not altered further by the sensor adversary program. By adopting this approach, we prioritize the detection of malware and attacks targeting the behavior of printer instructions, placing them above any issues related to faulty data collection.

4.3 Attack Detection and Localization Performance

We evaluate the model based on the f1-score, which is considered a mean of precision and recall scores:

$$Prec = \frac{TP}{TP + FP} \text{ , } Recall = \frac{TP}{TP + FN} \tag{4.1}$$

$$F1 - score = \frac{2 \times Prec \times Recall}{Prec + Recall} \tag{4.2}$$

, where TP is the number of true positives, FP is the number of false positives, and FN is the number of false negatives. Each row in Table 4.2 shows the model results for the datasets with G-code anomalies only, sensor anomalies only, and G-code/sensor injections combined. Each dataset consists of G-code from different time frames of printing the same object.

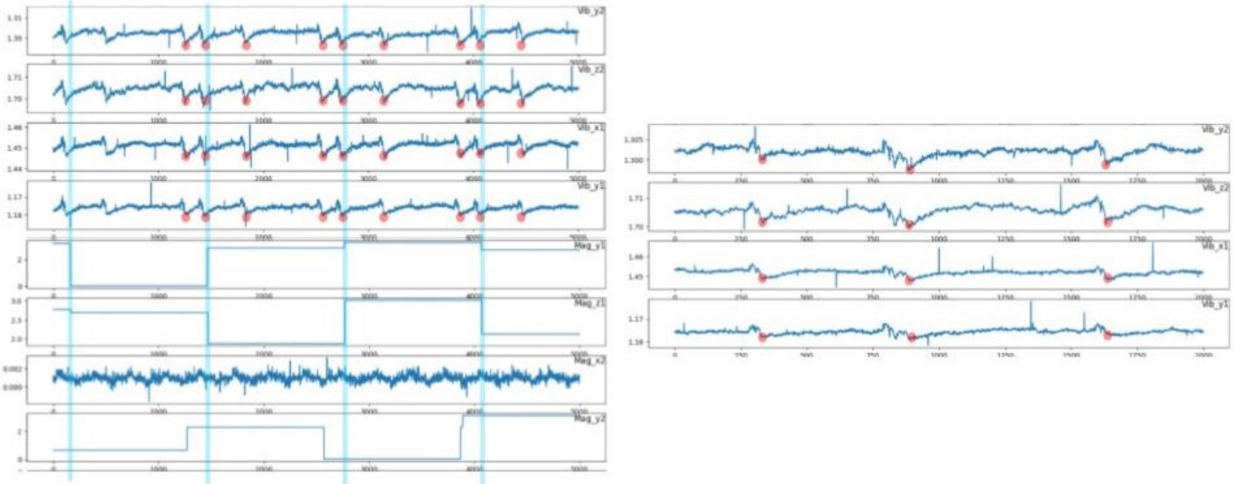

Figure 4.3: Pattern Recognition between Magnetic and Vibration sensors (left) and Threshold Analysis (right)

The model consistently performed better on datasets with G-code and sensor anomalies combined, most likely because the model does not consider uni-variate data individually but rather does multivariate anomaly detection. Therefore, as the number of abnormal values increases within the sensor data and G-code features at a specific timestamp, the attack becomes more conspicuous, allowing the model to detect it more effectively.

Table 4.2: Results of Anomaly Types

Dataset F1-Scores			
Anomaly Type	Dataset 1	Dataset 2	Dataset 3
G-code Only	0.71509	0.89525	0.86738
Sensor Only	0.85531	0.65916	0.66060
G-code and Sensor	0.71774	0.89776	0.86862

Table 4.3 shows the results for G-code anomaly localization, while Table 4.4 shows sensor anomaly localization results for each tested dataset. The method of selecting the threshold for G-code anomaly localization closely resembles our approach to general attack detection, with the exception that we take the sum of the total error scores for the G-code data at each timestamp. This array of errors is then ranked from highest to lowest f1-score, promptly identifying which error value to set as our threshold. By applying this threshold, we can determine the presence of a G-code anomaly at a specific timestamp.

Table 4.3: Results of G-code Anomaly Localization

	Dataset 1	Dataset 2	Dataset 3
F1 Score	0.71509	0.86737	0.89525

Table 4.4: Results of Sensor Anomaly Localization

	Dataset 1: Vibration_y2	Dataset 2: Magnetometer_y0	Dataset 3: Vibration_y2
F1 Score	0.99971	0.99935	0.99961

For sensor localization, we individually consider each sensor's array of error scores for each timestamp. By training a Random Forest classifier using a subset of these errors, predictions for each individual sensor are generated. The exceptional performance of the sensor localization can be attributed to the utilization of the supervised Random Forest algorithm. It is important to note that while this approach achieves higher accuracy when trained on labeled data, it should be acknowledged that our classifier design serves as a proof of concept rather than a direct emulation of real-world results.

Chapter 5

Reflection

5.1 Limitations

Our GDN model applied deep-learning strategies to accurately forecast the behavior of time-series sensor data and localize any detected anomalies. Further correlation analysis and supervised top k selection may improve accuracy and performance.

Another limitation to recognize would be the quantity and quality of data available. The datasets utilized to conduct this experiment were taken from previous data collections of a similar work environment. Testing other CAD models with new firmware provided by the company may improve performance and information handling.

While our methodology could be applied across different systems, the scope of our project is restricted to the Ultimaker 3 model. Locations of side-channel data extraction, such as stepper motors and power flow, will vary from system to system. The position of these sensors may result in deviating behaviors during correlation analysis. On the other hand, because most commercial and industrial 3D printers apply G-code instructions, collecting sensor data with the proper understanding of side-channel emissions will result in a similar methodology.

5.2 Feasibility

Acknowledging the significance of feasibility in cybersecurity is essential, as it directly impacts the effectiveness and sustainability of security measures enforced by industry. Our system primarily consists of affordable sensors and micro-controllers, making it readily usable at the workplace and at home. As mentioned in 5.1, while specific 3D printer models may differ, the underlying techniques remain consistent. Therefore, the sensor types and overall experimental procedure can be replicated to accommodate any additive manufacturing system as long as the quantification and analysis of analog side-channel emissions are feasible.

5.3 Future Work

Subsequent research can be done on future iterations of multi-modal neural network models. One area of improvement involves optimizing the customized edge generation process between data nodes, particularly when handling the top k correlations. Additionally, expanding the range and diversity of side-channel modalities, such as incorporating electromagnetic and optical sensors, has the potential to provide more valuable data for supervised localization training. These modifications would impact the weights assigned to each node in the graph and contribute to overall performance improvements. The results of this paper can be compared to the data obtained from testing attack localization on different classifiers with varying levels of supervision.

Chapter 6

Conclusion

The increasing prevalence of 3D printing in modern society raises concerns about the accompanying security risks. This paper introduces a methodology to amass a multi-modal dataset suitable for training a Graph Deviation Network (GDN) model to identify any divergence from expected values. Alongside essential pre-processing synchronization, anomaly localization using a Random Forest classifier achieved an average accuracy of **82.59%** in detecting G-code abnormalities and an average accuracy exceeding **95%** for external sensor data through supervised learning. While there is still potential for further improvement, this approach demonstrates promising results that extend beyond Additive Manufacturing, suggesting its applicability in other domains.

Bibliography

[1] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan. Acoustic side-channel attacks on additive manufacturing systems. In *2016 ACM/IEEE 7th international conference on Cyber-Physical Systems (ICCPS)*, pages 1–10. IEEE, 2016.

[2] M. A. Al Faruque, S. R. Chhetri, A. Canedo, and J. Wan. Forensics of thermal side-channel in additive manufacturing systems. *University of California, Irvine*, 12(13):176, 2016.

[3] M. A. Al Faruque, J. Wan, and S. R. Chhetri. Defending side channel attacks in additive manufacturing systems, Feb. 19 2019. US Patent 10,212,185.

[4] M. A. Al Faruque, J. Wan, S. R. Chhetri, and S. Faezi. Information leakage-aware computer aided cyber-physical manufacturing, Nov. 16 2021. US Patent 11,178,166.

[5] S. Bang, J. H. Jhee, and H. Shin. Polypharmacy side-effect prediction with enhanced interpretability based on graph feature attention network. *Bioinformatics*, 37(18):2955–2962, 2021.

[6] H. K. Chan, J. Griffin, J. J. Lim, F. Zeng, and A. S. Chiu. The impact of 3d printing technology on the supply chain: Manufacturing and legal perspectives. *International Journal of Production Economics*, 205:156–162, 2018.

[7] V. Chandola, A. Banerjee, and V. Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):1–58, 2009.

[8] S. R. Chhetri and M. A. Al Faruque. Side channels of cyber-physical systems: Case study in additive manufacturing. *IEEE Design & Test*, 34(4):18–25, 2017.

[9] S. R. Chhetri, A. Barua, S. Faezi, F. Regazzoni, A. Canedo, and M. A. Al Faruque. Tool of spies: Leaking your ip by altering the 3d printer compiler. *IEEE Transactions on Dependable and Secure Computing*, 18(2):667–678, 2019.

[10] 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)*, pages 1–8. IEEE, 2016.

[11] S. R. Chhetri, A. Canedo, and M. A. A. Faruque. Confidentiality breach through acoustic side-channel in cyber-physical additive manufacturing systems. *ACM Transactions on Cyber-Physical Systems*, 2(1):1–25, 2017.

[12] S. R. Chhetri, S. Faezi, and M. A. Al Faruque. Fix the leak! an information leakage aware secured cyber-physical manufacturing system. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*, pages 1408–1413. IEEE, 2017.

[13] 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*, 13(9):2333–2344, 2018.

[14] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. Al Faruque. Thermal side-channel forensics in additive manufacturing systems. In *2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS)*, pages 1–1. IEEE, 2016.

[15] S. R. Chhetri, S. Faezi, A. Canedo, and M. A. A. Faruque. Quilt: Quality inference from living digital twins in iot-enabled manufacturing systems. In *Proceedings of the International Conference on Internet of Things Design and Implementation*, pages 237–248, 2019.

[16] S. R. Chhetri, J. Wan, and M. A. Al Faruque. Cross-domain security of cyber-physical systems. In *2017 22nd Asia and South Pacific design automation conference (ASP-DAC)*, pages 200–205. IEEE, 2017.

[17] G. Deepa, G. SriTeja, and S. Venkateswarlu. An overview of acoustic side-channel attack. *International Journal of Computer Science & Communication Networks*, 3(1):15, 2013.

[18] A. Deng and B. Hooi. Graph neural network-based anomaly detection in multivariate time series. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 4027–4035, 2021.

[19] 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*, 11(10):2174–2186, 2016.

[20] S. El-Sayegh, L. Romdhane, and S. Manjikian. A critical review of 3d printing in construction: Benefits, challenges, and risks. *Archives of Civil and Mechanical Engineering*, 20:1–25, 2020.

[21] G. Fernandes, J. J. Rodrigues, L. F. Carvalho, J. F. Al-Muhtadi, and M. L. Proença. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, 70(3):447–489, 2019.

[22] A. Gangwal and M. Conti. Cryptomining cannot change its spots: detecting covert cryptomining using magnetic side-channel. *IEEE Transactions on Information Forensics and Security*, 15:1630–1639, 2019.

[23] 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*, 2(3):1–27, 2018.

[24] G. D. Goh, S. L. Sing, and W. Y. Yeong. A review on machine learning in 3d printing: applications, potential, and challenges. *Artificial Intelligence Review*, 54(1):63–94, 2021.

[25] Á. M. Guerrero-Higueras, N. DeCastro-Garcia, and V. Matellan. Detection of cyber-attacks to indoor real time localization systems for autonomous robots. *Robotics and Autonomous Systems*, 99:75–83, 2018.

[26] F. Hu, H. Wang, and J. Wang. Multi-leak deep-learning side-channel analysis. *IEEE Access*, 10:22610–22621, 2022.

[27] Z. Jin, Z. Zhang, X. Shao, and G. X. Gu. Monitoring anomalies in 3d bioprinting with deep neural networks. *ACS Biomaterials Science & Engineering*, 2021.

[28] G. Joy Persial, M. Prabhu, and R. Shanmugalakshmi. Side channel attack-survey. *Int. J. Adv. Sci. Res. Rev*, 1(4):54–57, 2011.

[29] Y. Kim, W. S. Lee, V. Raghunathan, N. K. Jha, and A. Raghunathan. Vibration-based secure side channel for medical devices. In *Proceedings of the 52Nd Annual Design Automation Conference*, pages 1–6, 2015.

[30] T.-H. Le, C. Canovas, and J. Clédiere. An overview of side channel analysis attacks. In *Proceedings of the 2008 ACM symposium on Information, computer and communications security*, pages 33–43, 2008.

[31] C. Li, D. Pisignano, Y. Zhao, and J. Xue. Advances in medical applications of additive manufacturing. *Engineering*, 6(11):1222–1231, 2020.

[32] H. Liu and B. Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *applied sciences*, 9(20):4396, 2019.

[33] B. R. Mantha and B. G. de Soto. Cyber security challenges and vulnerability assessment in the construction industry. In *Creative Construction Conference 2019*, pages 29–37. Budapest University of Technology and Economics, 2019.

[34] M. McCormack, S. Chandrasekaran, G. Liu, T. Yu, S. D. Wolf, and V. Sekar. security analysis of networked 3d printers. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 118–125. IEEE, 2020.

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

[36] T. Mortlock, D. Muthirayan, S.-Y. Yu, P. P. Khargonekar, and M. A. Al Faruque. Graph learning for cognitive digital twins in manufacturing systems. *IEEE Transactions on Emerging Topics in Computing*, 10(1):34–45, 2021.

[37] S. Picek, A. Heuser, A. Jovic, S. A. Ludwig, S. Guilley, D. Jakobovic, and N. Mentens. Side-channel analysis and machine learning: A practical perspective. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 4095–4102. IEEE, 2017.

[38] S. Picek, G. Perin, L. Mariot, L. Wu, and L. Batina. Sok: Deep learning-based physical side-channel analysis. *ACM Computing Surveys*, 55(11):1–35, 2023.

[39] P. Pradhyumna, G. Shreya, et al. Graph neural network (gnn) in image and video understanding using deep learning for computer vision applications. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1183–1189. IEEE, 2021.

[40] R. Prasad and Y. Moon. Adaptive intrusion detection system for cyber-manufacturing system. In *ASME International Mechanical Engineering Congress and Exposition*, volume 85567, page V02BT02A010. American Society of Mechanical Engineers, 2021.

[41] M. Randolph and W. Diehl. Power side-channel attack analysis: A review of 20 years of study for the layman. *Cryptography*, 4(2):15, 2020.

[42] H. Rogers, N. Baricz, and K. S. Pawar. 3d printing services: classification, supply chain implications and research agenda. *International Journal of Physical Distribution & Logistics Management*, 2016.

[43] T. Ryan and D. Hubbard. 3-d printing hazards: Literature review & preliminary hazard assessment. *Professional Safety*, 61(06):56–62, 2016.

[44] M. Salmi. Additive manufacturing processes in medical applications. *Materials*, 14(1):191, 2021.

[45] A. Sayakkara, N.-A. Le-Khac, and M. Scanlon. A survey of electromagnetic side-channel attacks and discussion on their case-progressing potential for digital forensics. *Digital Investigation*, 29:43–54, 2019.

[46] A. Schlösser, D. Nedospasov, J. Krämer, S. Orlic, and J.-P. Seifert. Simple photonic emission analysis of aes: photonic side channel analysis for the rest of us. In *Cryptographic Hardware and Embedded Systems–CHES 2012: 14th International Workshop, Leuven, Belgium, September 9-12, 2012. Proceedings 14*, pages 41–57. Springer, 2012.

[47] 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*, pages 1–10, 2017.

[48] 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*, pages 895–907, 2016.

[49] L. D. Sturm, C. B. Williams, J. A. Camelio, J. White, and R. Parker. Cyber-physical vulnerabilities in additive manufacturing systems: A case study attack on the. stl file with human subjects. *Journal of Manufacturing Systems*, 44:154–164, 2017.

[50] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio. Graph attention networks. *arXiv preprint arXiv:1710.10903*, 2017.

[51] M. Wu and Y. B. Moon. Intrusion detection system for cyber-manufacturing system. *Journal of Manufacturing Science and Engineering*, 141(3), 2019.

[52] Y. Wu, D. Lian, Y. Xu, L. Wu, and E. Chen. Graph convolutional networks with markov random field reasoning for social spammer detection. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1054–1061, 2020.

[53] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

[54] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici. Security of additive manufacturing: Attack taxonomy and survey. *Additive Manufacturing*, 21:431–457, 2018.

[55] S.-Y. Yu, A. V. Malawade, S. R. Chhetri, and M. A. Al Faruque. Sabotage attack detection for additive manufacturing systems. *IEEE Access*, 8:27218–27231, 2020.

[56] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun. Graph neural networks: A review of methods and applications. *AI open*, 1:57–81, 2020.