In cyber-physical systems, due to the tight integration of the computational, communication, and physical components, most of the information in the cyber-domain manifests in terms of physical actions (such as motion, temperature change, etc.). This leads to system being prone to physical-to-cyber domain attacks that affect the confidentiality. Physical actions are governed by energy flows which may be observed. Some of these observable energy flows unintentionally leak information about the cyber-domain, and hence are known as the side-channels. Side-channels such as acoustic, thermal, and power allow attackers to acquire the information without actually leveraging the vulnerability of the algorithms implemented in the system. As a case study, we have taken cyber-physical additive manufacturing systems (fused deposition modeling based 3D printer) to demonstrate how the acoustic side-channel can be used to breach the confidentiality of the system. In 3D printers, geometry, process, and machine information are the intellectual properties, which are stored in the cyber domain (G-code). We have designed an attack model that consists of digital signal processing, machine learning algorithms, and context-based post processing to steal the intellectual property in the form of geometry details by reconstructing the G-code and thus the test objects. We have successfully reconstructed various test objects with an average axis prediction accuracy of 86% and an average length prediction error of 11.11%.

CCS Concepts: •Computer systems organization → Embedded and cyber-physical systems; •Security and privacy → Systems security;

General Terms: Side-Channel Attack, Security, 3D printer, Confidentiality

Additional Key Words and Phrases: Cyber-Physical Systems, Additive Manufacturing, Side-Channels

ACM Reference Format:
DOI: 0000001.0000001

1. INTRODUCTION

Cyber-Physical Systems (CPS) consist of the integration of computation, physical, and networking components. The synergy of these components results in a new form of vulnerabilities, which cannot be addressed by traditional security solutions designed for the individual components. Additive manufacturing is an example of CPS, where 3D objects are created layer by layer. Fused Deposition Modeling (FDM) is one of the technologies used in additive manufacturing, where plastic or metal filaments, heated slightly above their melting point, are deposited to construct a 3D object. Several sectors, such as medical and aerospace, are increasingly adopting the use of these additive manufacturing systems [Leukers et al. 2005; Platform and Recycling 2014]. In addition, agencies like the U.S. Air Force, Navy, and NASA are also incorporating additive manufacturing into their manufacturing processes [Platform and Recycling 2014].
The revenue of the additive manufacturing industry is expected to exceed $21B by 2020 [Wohlers 2014]. This trend shows that new vulnerabilities, such as cross-domain attacks, can have large economic impact on manufacturing industries utilizing additive manufacturing. Attackers who target additive manufacturing systems are often motivated by either industrial espionage of Intellectual Property (IP), alteration of data, or denial of process control [NDIA 2014; Reznick 2015]. The world economy relies heavily on IP-based industries, which produce and protect their designs through IP rights. In the U.S. alone, the IP-intensive industries have been known to account for 34.8% of the gross domestic product [Economics and Administration 2012], and they are bound to face security issues. In fact, it has been estimated that, by 2018, 3D printing of pirated designs will result in annual IP losses of $100 billion [Hvistendahl 2016]. IP in additive manufacturing consists of the internal and external structure of the object, the process parameters, and the machine specific tuning parameters [Yampolskiy et al. 2014]. To produce a 3D object, design information (which contains IP) is supplied to the manufacturing system in the form of G-code. G-code, a programming language, is primarily used in FDM to control the system components and parameters such as speed, extrusion amount, etc [Gibson et al. 2010]. If these designs are stolen, they can be manipulated to harm the image of the company, or even worse, can cause the company to lose its IP (as it is stolen before production) [Ashford 2014]. Currently, IP theft mainly occurs through the cyber domain (e.g., Operation Aurora, GhostNet) [Branigan 2010; Wall and Yar 2010], but IP information can also be leaked through the physical domain (side-channels). A common example of this is to use side-channel information (e.g., timing data, acoustics, power dissipation, and electromagnetic emission) from devices performing cryptographic computation to determine their secret keys [Standaert et al. 2009]. This paper highlights the possibility of physical-to-cyber attacks on CPS, and motivate a general research interest in novel ways to minimize side-channel leakage during design time and manufacturing.

Research Challenges and Our Novel Contributions: It is probably not possible to make a system completely secure. This is because many vulnerabilities are not known during design time. Hence, it is necessary to continue to investigate this field to identify novel threats. We contribute to the security research in cyber-physical additive manufacturing systems with the following:

— **Source of Acoustic Emission (Section 3)** where we, in detail, analyze the source of acoustic emission in a FDM technique based 3D printer, and provide various equations to better understand the working principles of the proposed attack model.

— **Acoustic Leakage Analysis (Section 4)** where we provide the side-channel leakage model and leakage quantification to understand the relation between the cyber-data and acoustic emission.

— **An Acoustic side-channel attack model to breach confidentiality (Section 5)** which describes our novel attack methodology. It consists of exploration of time and frequency domain features, learning algorithms trained to acquire specific information (axis of movement, speed of the nozzle) about the G-code, context-based post processing, and algorithms used to reconstruct the G-code by reverse engineering.
2. BACKGROUND AND RELATED WORK

A typical digital process chain in additive manufacturing systems is presented in Figure 1. Designers start their design of 3D objects with 3D Computer-Aided Design (CAD) modeling tools such as Sketchup [SketchUp Make 2015] and the extended version of Photoshop Adobe Photoshop CC. Next, the CAD tool generates a standard STereoLithography (STL) file for the manufacturing purpose. Computer Aided Manufacturing (CAM) software is then required to slice the STL file into layer-by-layer description file (e.g., G-code, cube, etc.). Then, the layer description file is sent to the manufacturing system (e.g., 3D printer) for production [Gibson et al. 2010].

In the physical domain of the additive manufacturing, components such as a stepper motor, fan, extruder, base plate etc., carry out operations on the basis of information provided by the cyber domain (G-code). In carrying out the operation, these physical components leak cyber domain information (G-code) from the side-channels, such as acoustic and power, which may be used to steal IP by performing physical-to-cyber attack. The issues regarding the theft of IP and the framework for preventing IP theft have been studied in [Holbrook and Osborn 2014; Yampolskiy et al. 2014]. The study of attack in the process chain, starting from the 3D object design to its creation, along with a case study of cyber attacks in STL file, is presented in [Sturm et al. 2014]. However, physical domain attacks are not well studied by the existing works.

There are several publications utilizing the side-channel information to gather data related to the cyber domain in other systems. [Backes et al. 2010] has used the acoustics emanated from the dot matrix printer while printing to recover the text it was sent to print. Authors in [Toreini et al. 2015] have been able to decode the keys pressed in the Enigma machine by analyzing the sound made by the device while pressing the keys. However, these methodologies are not applicable to 3D printers since, unlike printed words on paper, a 3D printer’s movement has infinite possibilities. Recently, researchers from MIT have found that even the minor movement of physical devices can leak information about the cyber domain. In [Davis et al. 2014], they have successfully retrieved digital audio being played by capturing the vibration of objects near a sound source by a high speed camera. However, in a 3D printer, there are multiple sources of sound and vibration. Therefore, the task of analyzing sound for G-code reconstruction requires a completely new approach. Authors in [Vincent et al. 2015] have considered using side-channel for providing security, but they have not demonstrated any methodology for using it to steal the IP. In summary, the related work is focused on retrieving the text being printed (either in keyboard or dot matrix printer), analyzing acoustic emissions for observing mechanical degradation of the physical components in a manufacturing plant, etc. However, the possibility of using the acoustic emissions for reconstruction of a 3D object has not been considered. Hence, in this paper, we have designed an acoustic side-channel attack model to breach the confidentiality of cyber-physical additive manufacturing systems.

3. SOURCES OF ACOUSTIC EMISSION

A 3D printer has various sources of acoustic emissions. A strong attacker maybe able to describe the behavior of acoustic emission in the side-channel based on the various
G-codes using a single physical model (using a single equation). However, this task is not trivial, as it has to consider all the sources of vibration in the system. Moreover, they may require the system design knowledge to model the mechanical and electrical model of the system, which may not readily be available. Instead, a data-driven modeling of the system can easily be created by treating the system as a black box. However, to better understand why these data-driven models work, we will provide a preliminary analysis of individual sources of sound in a typical FDM technique based 3D printer. Based on this, in Section 4, we will perform the leakage analysis to understand why various information about G-code can be inferred from the acoustic emission.

3.1. System Description

State-of-the-art FDM based additive manufacturing systems consist of four to five stepper motors depending upon their structural design and number of filaments available for extrusion. Due to high torque/size ratio, hybrid two-phase stepper motors have been widely used in these 3D printers. However, these stepper motors are the major source of sound that enable the leakage of cyber-domain information from the side-channel. Hence, in the rest of the sections we will focus on describing various mathematical models of the stepper motor, and analyze how they aid in sound production.

The energy conversion steps for a stepper motor is shown in Figure 2. Here, the electrical energy (current $i$) is first converted to electric and magnetic field, which in turn guides the rotors. The electromagnetic field acting upon the various components produces force $F_{em}$, which causes them to vibrate and produce sound with power $P$.

However, the electrical energy is controlled by the G-codes, which gives the printer instruction to control the dynamics of the system (such as nozzle speed, axis movement, etc.). Hence, the acoustics emitted by the printer, eventually depends on the supplied G-code.

![Fig. 2: Energy Conversion.](Image)

### 3.2. Equation of Motion

To understand the production of mechanical energy, we can analyze the equation of motion for hybrid stepper motor, presented as follows [Hughes and Lawrenson 1975]:

$$J \frac{d^2 \theta}{dt^2} + D \frac{d \theta}{dt} + p \Psi_m i_A \sin(p \theta) + p \Psi_m i_B \sin(p(\theta - \lambda)) = 0$$

(1)

Where $J$ is the moment of inertia of the rotor and the load combined, $J = J_M + J_L$, and $D$ is the damping coefficient based on eddy current, air friction, hysteresis effects, etc. $i_A$ and $i_B$ are the current flowing through the two phases. $\Psi_m$ is the maximum stator flux linkage, $p$ is the number of rotor pole pairs, $\lambda$ is the angle between the two stator winding, and $\theta$ is the mechanical rotational angle. Equation 1 states that the inertia in the motor depends on the current being supplied to the two phases of the stator. This inertia determines the amount of vibration produced by the motor when it is rotating. Moreover, based on this equation we can also derive the natural oscillation frequency of the rotor, which plays a major role in generating unique acoustics for the given stepper motor.

### 3.3. Natural Rotor Oscillation Frequency

The radiated sound power is higher when the stepper motor vibrates with the rotor’s natural oscillation frequency. Using Equation 1, we may calculate the natural frequency of rotor oscillation as follows [Kenjö and Sugawara 1994]:

$$\omega_n^2 = \frac{2p^2 \Psi_m I_s \cos(\frac{p \lambda}{2})}{J}$$

(2)
Where $I_o$ is the stationary current flowing in the two phases $A$ and $B$. When the stepper motor is rotating with the harmonic frequency of the natural frequency such as $\frac{\omega_{np}}{4}, \frac{\omega_{np}}{5}, \frac{\omega_{np}}{6}, 2\omega_{np}, 3\omega_{np}, 4\omega_{np}, ...$ the vibration is more prominent due to resonance. Given the fact, that $J$ is the inertia of load and the rotor combined, varying load in the stepper motor will change its natural rotor oscillation frequency as well.

3.4. Stator Natural Frequency

Natural rotor oscillation frequency corresponds to the dynamic response of the stepper motor. However, the stator itself has a natural frequency which depends on various parameters. One of the major parameter is the vibration modes. Due to the prominence of the radial force acting on the stator, we will only consider the circumferential radial vibration modes and the corresponding stator natural frequencies. The structure of the stator is complex and many attempts have been made to calculate the natural frequencies of the stator with various considerations, an example being single-ring type stator [Heller and Hamata 1977; Yang 1981]. Since the external structure connected to the stator also influences its mass and stiffness, the natural frequency of the stator with circumferential vibration mode $m$ and axial vibration mode $n$ of the frame may be calculated as follows [Gieras et al. 2005]:

$$\omega_{stator\; np}^2 \approx \frac{K_m^{(c)} + K_{mn}^{(f)}}{M_c + M_f}$$  \hspace{1cm} (3)

Where $K_m^{(c)}$ is the lumped stiffness of the stator core, $K_{mn}^{(f)}$ is the lumped stiffness of the frame, and $M_c$ and $M_f$ are the mass of the stator and the frame, respectively. Equation 3 has been derived by assuming that the lumped stiffness of the core and the frame are in parallel.

3.5. Source of Vibration

The main sources of vibration in stepper motors are electromagnetic, mechanical, and aerodynamic [Timár-P and Timár 1989]. These vibrations help in radiating sound from the stepper motor stator surface and the frame to which the motor is connected. In this paper, we will consider the electromagnetic and mechanical sources as they are the major sources of leakage.

**Electromagnetic Source:** The fundamental source of vibration in hybrid stepper motors is due to the fluctuation of electromagnetic force produced by the winding of the stator. The two types of vibration produced by the electromagnetic force are:

i) **Radial Stator Vibration:** In a hybrid stepper motor, both stator and the rotor are responsible for exciting the magnetic flux density in the air gap between the rotor and the stator. These magnetic flux contribute in generating the radial force. If $\sigma_{l,k}$ be the radial force at pole $l$ for $k^{th}$ harmonic then the total radial force acting on the stepper motor may be calculated as follows [So et al. 1993]:

$$\sigma_{total} = \sum_{k=1}^{\infty} \sigma_{l,k} \cos(k\omega t + \phi_{l,k})$$  \hspace{1cm} (4)

Where $\phi_{l,k}$ is the phase angle of the radial force at pole $l$ for $k$ harmonic, $\omega = 2\pi f$, and $f$ is the frequency determined by the stepping rate of the motor. This radial force acts on the stator and rotor surface and deforms its structure. This produces vibration and eventually sound in the stepper motor. When the radial force excites the harmonics of the natural frequencies of the stator/frame structure and the rotor oscillation, vibration is more prominent due to resonance.

Let us assume that the structure of the stator of the hybrid-stepper motor is cylindrical (as shown in Figure 3). With this, we may express the total sound power radiated...
According to [Timár-P and Timár 1989; Yang 1981], the radiated sound power \( P \) can be calculated using the following equation:

\[
P = 2 \rho c \pi f^2 A_{rd}^2 (4 \pi rl/2) I_{rel} = 4 \rho c \pi f^2 A_{rd}^2 rl I_{rel}
\]

where \( P \) is the radiated sound power (W), \( \rho \) is the density of the medium (kg/m\(^3\)), \( c \) is the speed of the sound in the medium (m/s), \( f \) is the excitation frequency of the vibration with multiple harmonics (Hz), \( A_{rd} \) is the surface vibratory displacement (m), \( r \) is the radius of the cylindrical stator (m), \( l \) is the length of the stepper motor (m), and \( I_{rel} \) is the relative sound intensity. \( I_{rel} \) depends on the mode of stator vibration \( R \), the radius, and the length-diameter ratio. Hence, stepper motors with different geometry and design in the 3D printer will emit different sound power.

**ii) Torque Ripple:** Even though torque ripple is substantially reduced by using the microstepping for driving the stator windings, microstepping position ripple is still produced due to non-conformity to the ideal sine/cosine waves required for absolute removal of the torque ripple. However, the vibration produced by the torque ripple is less compared to the radial stator vibration.

**Mechanical Source:** The rotor and load connected to the stepper motor may also produce vibration and sound at various frequencies due to friction, rotor unbalance, shaft misalignment, loose stator lamination, etc. These vibrations produce a loud noise due to resonance.

In summary, the major source of acoustic emission are the stepper motors. However, any actuator (for example DC motors used in cooling fans) that corresponds to the G-code are also capable of leaking information from the acoustic side-channel. A strong attacker will consider all these facts to make their attack model accurate. In the subsequent sections, we will present detail analysis on the information that may be leaked through the acoustic emission.

### 4. ACOUSTIC LEAKAGE ANALYSIS

#### 4.1. Side-Channel Leakage Model

Using an acoustic data acquisition device, an attacker may physically observe analog emissions \([o_1, o_2, o_3, ..., o_i]\), where \( o_i \) denotes the \( i \)th sample observed in the time domain. This is in fact the measurement of the acoustic power radiated by the stepper motor, as given in Equation 5. Let \( O \) be a random variable denoting observable analog emissions, then we can model the side-channel leakage as follows:

\[
O = f_d(\cdot) + N
\]

Where \( f_d(\cdot) \) represents a deterministic leakage function, that may be modeled by an attacker. \( N \) represents a random variable denoting noise independent from \( f_d(\cdot) \) added to the side-channel. Leakage function \( f_d(\cdot) \) depends on the G-code instruction \([g_1, g_2, g_3, ..., g_j]\), where \( g_j \) is the \( j \)th instruction supplied to the 3D printer. The G-code may be denoted by a random variable \( G \), then the leakage model may be re-written as follows:

\[
O = f_d(G) + N
\]
Since some of the G-code take longer time to execute than others, for each G-code instruction \( g_j \), there will be \( k \) samples of analog emissions corresponding to it. The length \( k \) depends on the sampling frequency of the audio device used. The fundamental information contained in \( g_j \) are speeds in each axis \( \{v_{x_j}, v_{y_j}, v_{z_j}\} \in \mathbb{R}_+ \), presence or absence of axis movements \( \{a_{x_j}, a_{y_j}, a_{z_j}\} \in \{0, 1\} \) with 0 representing absence of movement, positive or negative distance moved in each axis \( \{d_{x_j}, d_{y_j}, d_{z_j}\} \in \mathbb{R}_+ \), and extrusion amount \( d_{e_j} \in \mathbb{R} \). Hence, the Equation 7 may be rewritten as follows:

\[
o_i = f_d(v_{x_i}, v_{y_i}, v_{z_i}, a_{x_i}, a_{y_i}, a_{z_i}, d_{x_i}, d_{y_i}, d_{z_i}, d_{e_i}) + n_i \tag{8}
\]

Where the smaller alphabets represents the value of the random variables \( O, G \) and \( N \) during \( i^{th} \) time interval, with random random variable \( G \) further partitioned into random variables \( \{V_{x}, V_{y}, V_{z}, A_{x}, A_{y}, A_{z}, D_{x}, D_{y}, D_{z}\} \) representing speed, axis movement, and distance, respectively, with their corresponding values passed to the function \( f_d(.) \) in Equation 8. Using data-driven approach, various machine learning algorithms such as regression, classifications, etc., may be used to estimate a function, such that for the \( i^{th} \) sample of observable emissions, we can estimate the corresponding G-code \( \hat{g}_i = \hat{f}_d(o_i, \alpha) + n_i \). Where \( \hat{f}_d \) represents the estimated function, and \( \alpha \) represents the tuning parameter learned for the function. Due to the presence of multiple parameters, classification and regression machine learning algorithms may be used to estimate multiple functions to estimate individual parameters separately.

**ASSUMPTION 1.** Given the acoustic leakage \( O \), the frequency of the radiated sound varies according to the speed of the nozzle in \( X \) and \( Y \) axis, respectively.

The radial force generated in Equation 4 in each pole depends on the magnetic flux density, stator tooth width, and rotor cap thickness. The magnetic flux density depends on the current passing through the each winding. To increase the angular speed of the stepper motor, the stepping rate is increased. From Equation 4, we can see that this increases the frequency of the radial force acting on the stepper motor. From Equation 5, we also can see that the radiated power increases with the excitation frequency of the vibration.

**ASSUMPTION 2.** Given the acoustic leakage \( O \), the power frequency spectrum of the radiated sound from the stepper motors \( X, Y, Z \), and the one for the extruder are different.

The natural rotor oscillation frequency in Equation 2 is inversely proportional to the moment of inertia of the load and the motor \( (J_L + J_M) \). The load moved by each stepper motors \( X, Y, Z \), and \( E \) are different in state-of-the-art stepper motors. The natural frequency in Equation 3 also depends on the mechanical structure of the frame to which the stepper motor is connected. Due to the mechanical structure of the 3D printers, stepper motors are placed in various locations and are connected to different frame structures. Therefore, the natural frequencies of the stepper motors vary according to the load and the frame to which they are attached. This means that the resonance can occur at different frequencies of the vibration for different stepper motors and the frame structure. This causes the power spectrum of the radiated sound to vary according to the source of sound, i.e., the stator motor and the frame structure.

**ASSUMPTION 3.** Given the acoustic leakage \( O \), the intensity of the radiated power will vary according to the direction of the nozzle movement in different directions from the audio device.

According to the inverse square law, the intensity of the sound decreases drastically with the square of the distance from the sound source. If \( P \) is the power of the sound

source and $r$ be the distance from the sound source, then we have:

$$I = \frac{P}{4\pi r^2}$$

(9)

Hence, for analyzing the direction of movement, the intensity of the sound radiated by each motor and frame structure may be measured.

**Assumption 4.** The direction of movement in Z-axis during printing is always in either positive or negative direction.

In additive manufacturing systems, materials are extruded layer wise. Hence, direction in Z-axis should always be in one direction. This allows us to exclude the estimation of direction motor in Z-axis.

### 4.2. Leakage Quantification

In our attack model, reconstruction of the G-code will depend on accurate estimation of the individual parameters of the G-code. Hence, in this paper we will focus on capability of the attack model to accurately estimate the individual parameters. The estimation is done using regression models to predict the continuous speed value, whereas, classification models are used to predict the axis information. Hence, leakage may simply be quantified by measuring the separability of the nozzle movement and prediction accuracy of the nozzle speeds in each axis. The separability is specifically measured using the Area Under the Curve (AUC) of the Receiver Operating Characteristic (ROC) curve, whereas the speed prediction accuracy can be measured using the Mean Square Error (MSE). We also calculate the accuracy of the classifiers as follows:

$$\text{Accuracy} = \frac{TP + TN}{\text{Total Sample}}$$

(10)

Where True Negative (TN) and True Positive (TP) are the total number of right classifications made by the classifier.

### 4.3. Leakage Exploitation

The accuracy with which an adversary is able to exploit the acoustic leakage depends on their ability to estimate the functions $f_d(.)$. Breaking the process for estimating $g_i$ into multiple estimation functions improves the adversarial attack model by focusing on only those parameters in $g_i$ that are required for breaching the confidentiality of the system. However, the accuracy of the attack model now becomes a function of successful estimation of individual functions. Moreover, to completely steal the intellectual property inherent in the geometry of a 3D object, each line segment in each layer must be reconstructed with error $e \geq e_P$. Where $e_P$ is the error introduced due to process variation. A strong attacker will be able to acquire accuracy with $e = e_P$. However, in a competitive market for products, losing even minute details about the geometry of their product can be disastrous for a company. Hence, in such scenarios, an attacker may just have to infer about the geometry of the 3D objects without 100% accuracy. Based on the domain knowledge of an attacker, they may be able to reconstruct the 3D object with further processing. However, we will not consider this capability (what kind of domain knowledge an attacker might have) of an attacker, as it is out of scope of this paper.

### 5. ATTACK MODEL DESCRIPTION

#### 5.1. Attack Model

In our attack model (shown in Figure 4), intention of an attacker is to steal the geometry details of an object which is one of the intellectual properties for a company.
using a 3D printer. Even though these details maybe stolen by acquiring the 3D object itself, our attack model is most suitable for stealing the IP during the prototyping stage, where 3D printers like the one based on FDM technique are used to visualize the geometry of the company products. An attacker maybe be a person who has a low level access (can be in a vicinity of the 3D printer) to the 3D printer but not to the digital process chain files (for example STL, G-codes, etc.) itself. Moreover, they do not have any access to the digital process chain tools and software either. They will have an access to replica model of the target machine, that has to have the same physical structure as the target 3D printer. On this replica model, they can perform any sort of prior experiments to train their learning algorithms to estimate the function $\hat{f}_d(.)$. However, the learning should be performed in an environment that is as close to the target machine as possible. The learning algorithms consists of multiple estimated functions $\hat{f}_d(.)$ to predict individual parameters of a G-code. These predicted values are then combined to reconstruct the G-code, and eventually the geometry of an object.

5.2. Components of the Attack Model

5.2.1. Data Acquisition. The first step of acquiring the observable analog emissions, $[o_1, o_2, \ldots, o_i]$, is to place an audio recording device such as a mobile phone near the 3D printer. The sampling frequency of the recording device must be higher than 40 kHz to capture the sound in the audible range to avoid aliasing effect [Oppenheim et al. 1989]. The distance of the audio device from the 3D printer and the angle to the different sources of sound (stepper motor X and stepper motor Y) will also determine the accuracy of leakage exploitation. Moreover, in case of devices that are enclosed, a contact microphone maybe utilized to estimate $\hat{f}_d(.)$. This will also reduce the influence of environmental noise on the acquired acoustic emission. For better accuracy, even multiple microphones maybe placed to localize and remove environmental noise during the attack.
5.2.2. Noise Filtering. We use a digital finite impulse response band pass filter to eliminate the noise from low frequency alternating current from the power source, and the high frequency noise generated by the hybrid stepper motor winding when it is in charged and in idle state. The passband frequency for the noise removal is between 100 Hz and 20 kHz.

5.2.3. Maximal Overlap Discrete Wavelet Transform and Multiresolution Analysis. In [Al Faruque et al. 2016], a fixed length window was used to extract various time and frequency domain features. With this configuration, feature extraction is more challenging. Capturing smaller movements requires smaller frame sizes to improve the temporal resolution, while improving the speed and frequency resolution requires larger frame sizes. This trade-off prescribes the use of discrete wavelets transforms, to preserve both time and frequency domain features. Maximal Overlap Discrete Wavelet Transform (MODWT) is a redundant form of discrete wavelet transform, where a time series signal is transformed into coefficients related to the variation of set of scales of a mother wavelet. Due to the added redundancy, MODWT makes it easier for the alignment of the decomposed wavelet and scaling coefficients at each level with the original time series. The MODWT allows us to decompose the time series data into multiple levels of scales and different frequency regions without having to worry about the length of the window. These signals are then passed through the multiresolution analysis block to decompose it into multiple signals ($S_1, S_2, \ldots, S_n$) for feature extraction. In [Al Faruque et al. 2016], uniform frequency scales are used to extract features using Short-Term Fourier Transform (STFT). Even though, Mel-Frequency Cepstral Coefficients (MFCC) are used in [Al Faruque et al. 2016], which uses non uniform scaling focusing on higher number of frequency features in the lower frequency range, we have found that this feature extraction is not feasible for the 3D printer acoustic analysis due to variation of frequency distribution based on the travel feed-rate. Hence, we analyze the signals decomposed by MODWT to define the non-uniform frequency scales for feature extraction. Moreover, we have computed five levels of MODWT signals to define the non-uniform frequency scales.

5.2.4. Feature Extraction. We use features commonly used in speech pattern recognition [Theodoridis et al. 2010] in the time and frequency domains to train our learning algorithms. In the time domain, the features extracted are frame energy, Zero Crossing Rate (ZCR), and energy entropy [Theodoridis et al. 2010]. The features extracted from the frequency domain are spectral entropy and spectral flux. Multiple signals ($S_1, S_2, \ldots, S_n$) are obtained from the multiresolution analysis block. For each signal, varying number of frequency scales are allocated to extract spectral energy values. We use two frame sizes, 50 ms and 20 ms, for testing the performance of the attack models with MODWT and STFT based features. From each frame, we extract features and create a feature vector to supply the training algorithm. For a given frame of length $F_L$ with audio signals $x(i) = 1, 2, \ldots, F_L$, different features are extracted as follows:

$$Frame\ Energy (E) = \sum_{i=1}^{F_L} |x(i)|^2$$ (11)

Frame energy is enough to predict direction when the printer is only printing in one axis, however spectral energy is required while predicting the direction in multiple axes movement. ZCR is calculated as follows:

$$ZCR = \frac{1}{2F_L} \sum_{i=1}^{F_L} |\text{sign}[x[i]] - \text{sign}[x(i-1)]|^2$$ (12)

ZCR is high when the printer is not making any sound, due to the noise, and low when it is printing. For energy entropy, we divide the frame into short frames of length $K$. If $E_j$ is the energy of the $j^{th}$ short frame, then we have:

$$\text{Energy Entropy} = - \sum_{j=1}^{K} e_j \log_2(e_j)$$  \hfill (13)

Where $e_j = \frac{E_j}{\sum_{i=1}^{K} E_i}$  \hfill (14)

Energy entropy measures the abrupt change in the energy of the signal, and may be used to detect the change of motion. For frequency domain data, let $X_i(k)$, $k = 1, ..., F_L$ be the magnitude of the Fast Fourier Transform (FFT) coefficient of the given frame. For spectral entropy, we divide the spectrum into $L$ sub bands. Let $E_f$ be the energy of the $f^{th}$ sub band then we have:

$$\text{Spectral Entropy} = - \sum_{f=1}^{L-1} n_f \log_2(n_f)$$  \hfill (15)

Where $n_f = \frac{E_f}{\sum_{j=1}^{L-1} E_j}$  \hfill (16)

Spectral flux measures spectral change between two successive frames, and may be used to detect the change of speed of the nozzle while printing within each layer:

$$\text{Spectral Flux}_{i,i-1} = \sum_{k=1}^{F_i} (EN_i(k) - EN_{i-1}(k))^2$$  \hfill (17)

Where $EN_i(k) = \frac{X_i(k)}{\sum_{j=1}^{F_{L-1}} X_i(j)}$  \hfill (18)

**Fig. 6: Regression Model for Motor Speed Prediction.**

5.2.5. Regression Model. The regression model consists of a collection of models, each using a supervised learning algorithm for regression as shown in Figure 6. These models are used for estimating the functions $v_{x,i} = f(o_i, \alpha)$ and $v_{y,i} = f(o_i, \alpha)$. These functions are used to extract information about speed in $X$ direction given only one axis movement, and speed in $X$ direction given the motion in two axis. Similarly, this is done for speed in $Y$ direction as well.

**Assumption 5.** The speed in the $Z$ direction while printing the given model with the given printer is fixed and the speed of extrusion can be calculated as a function of layer height and nozzle diameter.
For a given 3D printer, the layer height is assumed to be fixed. This relaxes the complexity for leakage exploitation by reducing the need for estimating speeds $v_x$ and $v_y$. The speed of the printing, also known as the travel feedrate is determined by training these regression algorithms [Pedregosa et al. 2011]. After gaining the information about the travel feedrate, we may calculate the distance moved by the nozzle as follows:

$$\text{Distance} = \text{Framesize (ms)} \times \text{Speed (mm/ms)} \quad (19)$$

### ALGORITHM 1: Feature Processing and Speed Calculation with Motion in XY Axes.

**Input:** Feature Vectors $xy_\beta; x_\beta; y_\beta$  
**Output:** Speed $\vartheta_x, \vartheta_y$  

1. $\vartheta_{x\text{mean}_i} = \frac{1}{N_i} \sum_{n=1}^{N_i} x_{\beta_n,j}$  
2. $\vartheta_{y\text{mean}_i} = \frac{1}{N_i} \sum_{n=1}^{N_i} y_{\beta_n,j}$  
3. For each $xy$ do  
4. For $\vartheta_i$ in range($v^1, v^n$)  
5. \[ xy_{\beta}(xy-y) = xy_{\beta} - \vartheta_{y\text{mean}_i} \]  
6. $\vartheta_{x} \leftarrow \text{RegressionModel1}(xy_{\beta}(xy-y))$  
7. $xy_{\beta}(xy-x) = xy_{\beta} - \vartheta_{x\text{mean}_i}$  
8. $\vartheta_{y} \leftarrow \text{RegressionModel2}(xy_{\beta}(xy-x))$  
9. $\text{diff}_{fi} = |\vartheta_{y} - \vartheta_{y}|$  
10. $\vartheta_{y} = \vartheta_{y}^i$ with minimum $\text{diff}_{fi}$  
11. $\vartheta_{x} = \vartheta_{x}^i$ with minimum $\text{diff}_{fi}$  
12. Return $\vartheta_x, \vartheta_y$

When the nozzle is moving in only one axis, the regression model may just take the features directly without further processing, however, when the nozzle is moving in two or more axes, the audio signal from one motor is combined with the others. Hence, it becomes imperative to separate these signals before the regression model can be used to predict the speed. Algorithm 1 provides the pseudo code for performing the spectral subtraction necessary when motion is involved in both the X and Y axes. It takes features, extracted from the audio when both the X and Y motors are running, and the features from the training phase for individual motor X and Y as the input. Spectral subtraction is not performed for Z motor because it only moves one layer at a time and the distance it moves is normally fixed for a given object. While training, $n$ number of speeds, in incremental number is taken to train the regression models. For each of these speeds, lines 1 and 2 calculate the average magnitude of spectral features. Then, for each of the speeds, line 6 assumes the speed of the Y motor and the spectral components are subtracted from the combined spectral features of X and Y. By subtraction, we remove the spectral components present in Y from the combination of these features. Line 7 gives the predicted speed for the given value of speed in Y direction. We use this speed to subtract the spectral features of X in the particular speed and again use this value to predict the speed for motion in Y-axis. In lines 11 and 12, the speed of X and Y that gives the minimum difference in the predicted speed and output speed in Y-axis is chosen as an output.

#### 5.2.6. Classification Model.

As shown in Figure 7, to determine the axis in which the nozzle is moving, the classification model consists of collection of classifiers to convert the classification problem into two-class separation model. This in fact will estimate the functions $a_{x_i} = f(o_i, \alpha), a_{y_i} = f(o_i, \alpha)$, and $a_{z_i} = f(o_i, \alpha)$. We have found that this model gives us better prediction results than multi-class classifier models. Each of these classifiers consists of supervised learning algorithms for classification. Algorithm
2 gives the pseudo code which takes the output from the classifiers to determine the axis of movement. It also gives information such as whether the layer has changed or not, and whether the nozzle is moving in X and Y axis with the same or different speed.

**Algorithm 2: Estimate the Axis of Movement.**

**Input:** Classifier Outputs $\phi_1, \phi_2, \phi_3, \phi_4$

**Output:** Axis Parameters $A_x, A_y, A_z, \Theta_{1D}, \Theta_{2D}, Layer_{flag}, XY_{speedflag}$

1. $\Theta_{1D} = 0, \Theta_{2D} = 0$ // $A \rightarrow \text{axis}, \Theta \rightarrow \text{dimension}$
2. $Layer_{flag} = 0, XY_{speedflag} = 0$ // Initialize to zero
3. if $\phi_1 = 1$
   4. $Layer_{flag} = 1, A_z = 1$
5. else
6. if $\phi_2 = 1$
   7. $\Theta_{1D} = 1$
   8. if $\phi_3 = 1$ // Movement in X-axis
   9. $A_x = 1$
10. else
11. $A_y = 1$ // Movement in Y-axis
12. else
13. $\Theta_{2D} = 1, A_x = 1, A_y = 1$
14. if $\phi_4 = 1$ // X and Y move with same speed
15. $XY_{speedflag} = 1$
16. else
17. $XY_{speedflag} = 0$ // Different speed
18. return $A_x, A_y, A_z, \Theta_{1D}, \Theta_{2D}, Layer_{flag}, XY_{speedflag}$

5.2.7. Direction Prediction Model: Most of the 3D printers have motors in a fixed location. However, the base plate, the nozzle or combination of both are always in motion while printing. Therefore, vibration is conducted from the motor to the nozzle and the base plate of the printer. This means that the audio source physically gets closer or away from the recording device while printing. We can use the frame energy of the audio signal to check the direction of motion. For multiple motor movements, we utilize the difference of feature in frequency domain to calculate the energy of only those spectral components that represent the specific motor. In order to suppress the high fluctuation, median filtering is applied to the sequence of frame energies to smooth the curve of frame energies. The prediction model will output 1 if the frame energy is increasing and 0 if the frame energy is decreasing. In order to aid the direction prediction model
and the post-processing, a feature comparison block measures the distance (Euclidean
distance) between consecutive frame features. If the motion of direction changes, then
there is a large difference in the features between the consecutive frames. We use this
spike to detect the change in direction of motion of the nozzle.

5.2.8. Model Reconstruction: For reconstructing the G-code, we need to determine
whether the 3D printer nozzle is actually extruding the filament or not. From our
analysis, we have found that the printer nozzle moves at a higher speed when it is
not extruding the filament. Hence, determining whether it is printing or not printing
becomes a task of finding out the speed at which the nozzle is moving. This informa-
tion is acquired from the regression model. The extrusion amount for a given segment
is machine-specific, and can be calculated as a function of the layer height, and the
nozzle diameter. After acquiring the output from the regression model, classification
model, and direction prediction model, Algorithm 3 calculates the positive or negative
distance movement. Finally, Algorithm 4 reconstructs the G-code for the printed object.

5.2.9. Post-Processing for Model Reconstruction. We have found a high mutual infor-
mation between the G-code and the sound retrieved from the physical medium. For G-
codes, let $G$ be a discrete random variable with $f(g)$ as its probability distribution
function at $g$. Let $O$ be a discrete random variable representing the feature extracted
from the acoustics with $f(o)$ as its probability distribution function. Then the entropy
of each of these random variables may be given as:

$$H(G) = - \sum_{g \in G} f(g) \log_2 f(g)$$

ALGORITHM 3: Calculate Distance Moved in Each Axis, and Check Extrusion.

**Input:** Output from Classifier and Regression Models $\theta_x, \theta_y, \theta_z, w, A_x, A_y, A_z, \delta_x, \delta_y, \delta_z$

**Output:** Distance Values $d_x, d_y, d_z, d_E$

1. $d_x = 0, d_y = 0, d_z = 0$

   // $d \rightarrow$ Distance, $x, y \rightarrow$ Speed in X-axis

2. for each $i$ in $x, y, z$ do

3.   if $A_i = 1$ then

4.     if $\delta_i = 1$ then

5.       $d_i = \theta_i \times w$

6.       // Positive distance

7.     else

8.       $d_i = -\theta_i \times w$

9.       // Negative distance

10. if $\theta_x \geq Speed_{High} || \theta_y \geq Speed_{High}$ then

11.   $d_E = 0$

12.   // No extrusion in high speed

13. else

14.   $d_E = e_d$

15.   // $e_d \rightarrow$ Machine specific extrusion

16. return $d_x, d_y, d_z, d_E$
ALGORITHM 4: Generate G-code of the Object.

**Input:** Distance and Frame Length \( d_x, d_y, d_z, d_E, w \)

**Output:** G-code

1. \( dr_x = 0, dr_y = 0, dr_z = 0, dr_E = 0 \) // Initialize to zero

2. \( \vartheta = \frac{\sqrt{d_x^2 + d_y^2 + d_z^2}}{w} \) // Travel feedrate

3. \( dr_x = dr_x + d_x \) // Distance moved in X-axis

4. \( dr_y = dr_y + d_y \) // Distance moved in Y-axis

5. \( dr_z = dr_z + d_z \) // Distance moved in Z-axis

6. \( dr_E = dr_E + d_E \) // Extrusion amount

7. \( \text{G-code} \leftarrow G1 \, F(\vartheta) \, X(dr_x) \, Y(dr_y) \, Z(dr_z) \, E(dr_E) \)

8. return G-code

\[
H(O) = - \sum_{o \in O} f(o) \log_2 f(o) \tag{21}
\]

If \( f(g,o) \) and \( f(g|o) \) are the joint and conditional probabilities of the random variables, respectively, then the conditional entropy \( H(G|O) \) is calculated as:

\[
H(G|O) = - \sum_{o \in O} \sum_{g \in G} f(g,o) \log_2 f(g|o) \tag{22}
\]

The conditional entropy measures the amount of information required to describe outcome of a random variable \( G \), given the information about a random variable \( O \). In this context, in addition to the information gathered from \( O \), the amount of additional additive manufacturing context-based information required to reconstruct the G-code is directly related to the mutual information. This is calculated as:

\[
I(G;O) = H(G) - H(G|O) \tag{23}
\]

We have found that the uncertainty of reconstruction of G-code or the entropy \( H(G|O) \) increases when the distance of the microphone is further away from the printer or when there is added noise in the environment. It also increases when the speed of the printer is high and there are more short and rapid movements. During these scenarios, we can use the properties of additive manufacturing to post-process the data achieved from the learning algorithms. Specifically, we have used two post-processing stages which utilizes specific additive manufacturing context-based information.

**Post-Processing Stage I:** In this stage, we reduce \( H(G|O) \) by utilizing the fact that until the change of motion occurs, the nozzle moving in one particular dimension with a particular speed has a similar feature vector. By taking the output from the feature comparison model, we segment the acquired acoustic data into sections with similar movement. In this post-processing stage, we then choose the output of the classifiers to be the highest occurring value in the given segment, and for regression, we average the speed obtained within the same section. This is similar to averaging used in digital signal processing to increase the Signal-to-Noise Ratio (SNR).

\[
SNR_{dB} = 10 \log_{10} \left( \frac{Power_{signal}}{Power_{noise}} \right) \tag{24}
\]

When we increase the SNR, the entropy of the signal is reduced. As there is high correlation among the features extracted from successive frames of the audio collected from the 3D printer, averaging the output of the classification and the regression model increases the SNR and thus reduces \( H(G|O) \).

**Post-Processing Stage II:** After applying post-processing stage I, the second stage measures the similarity between the two layers. The similarity of two layers is measured in
terms of number of segments, the sequence of motions in each layer, and the length of each segment. This post-processing stage helps the attack model in reducing the error due to miscalculated direction and fluctuating lengths by taking the average of segment lengths and direction among the similar layers of the 3D object.

5.3. Attack Model Training and Evaluation

Testbed for Training and Testing: Our testbed, shown in Figure 9, consists of a Printrbot 3D printer [Printers 2015] with open source marlin firmware. It has four stepper motors. Motion in the X-axis is achieved by moving the base plate, whereas the nozzle itself can be moved in the Y and Z directions. The audio is recorded using a cardioid condenser microphone (Zoom H6) [Handy Recorder 2015], which has a sampling frequency of 96 kHz and stores the data at 24 bit per sample. We have placed the audio recorder within 20 cm of the 3D printer. From our experiments, we have analyzed that for the direction prediction model to work efficiently, the audio device has to be placed at 45° angle to both the X and the Y-axis as shown in Figure 9. This allows the audio device to capture the variation of sound in both X and Y directions.

The digital signal processing, feature extraction, and post-processing are performed in MATLAB [MATLAB 2015], whereas the training of learning algorithms, their evaluation and testing is done using Python [Python 2.7.10 2015]. The attack model consists of supervised learning algorithms described in Section 5.2. For training these algorithms, initial training data has to be determined. The training data consists of G-code to move the printer nozzle at different speeds (500 mm/min to 4500 mm/min) and different axes. The speed range chosen is specific to the 3D printer. The G-code for training phase consists of movement in just one axis (X, Y, and Z), two axes (XY, XZ, and YZ), and all three axes (XYZ). The audio signal corresponding to each of these G-codes is pre-processed and labeled for training the learning algorithms. The total length of audio recorded for training is 1 hour 48 minutes. The total numbers of features extracted is 328 for the window size of 50 ms and 318 for the window size of 20 ms. For regression model, we have used Decision Trees, boosted using Gradient Boosting algorithm, whereas for the classification model we have used Decision Tree Classifier, boosted using AdaBoost algorithm [Pedregosa et al. 2011]. We have trained the learning algorithms and have performed K-fold cross validation, with k = 4, to test the efficiency of the learners as well as to avoid over or under fitting of the learning algorithms. In our experiments, the regression model is trained only for the nozzle movements in the X and Y directions.

Classification Models: Table I shows the accuracy of the various classifiers. The MODWT and STFT based features extraction are compared with the features ex-
Table I: Accuracy of the Classification Models.

<table>
<thead>
<tr>
<th>Classification Model</th>
<th>Classifying</th>
<th>Accuracy(%) [Al Faruque et al. 2016] win 50 ms</th>
<th>Accuracy(%) (MODWT and STFT) win 50 ms</th>
<th>Accuracy(%) (MODWT and STFT) win 20 ms</th>
</tr>
</thead>
<tbody>
<tr>
<td>φ₁</td>
<td>Z or ~ Z Axis</td>
<td>99.86</td>
<td>99.9559</td>
<td>99.9400</td>
</tr>
<tr>
<td>φ₂</td>
<td>1D or 2D Axis</td>
<td>99.88</td>
<td>99.9700</td>
<td>99.9606</td>
</tr>
<tr>
<td>φ₃</td>
<td>X or Y Axis</td>
<td>99.93</td>
<td>99.9910</td>
<td>99.9778</td>
</tr>
<tr>
<td>φ₄</td>
<td>XYsame or XYdifferent</td>
<td>98.89</td>
<td>99.4644</td>
<td>99.2349</td>
</tr>
</tbody>
</table>

For measuring the accuracy of the classifiers, Receiver Operating Characteristics (ROC) curves are also analyzed. The classifiers capability to separate the two classes is high if the graph lies closer to the upper right corner. This region corresponds to 100% sensitivity (zero false negatives) and 100% specificity (zero false positives). As the collection of classifiers are arranged in a hierarchy, the bottleneck in terms of accuracy is the longest path followed while making the decision. In this case, the longest path involves all the classifiers. From Figures 10 (a)-(d), it can be observed that the classifiers have a high sensitivity and specificity with a high Area Under the Curve (AUC). It means the different classes can be accurately classified based on the observed leakage from the side-channel. The AUC for the classifier classifying whether the movement X and Y axis have same speed or different speed is comparatively less than other AUCs. This is intuitive as, in multiple axis movement, separation of individual movement is difficult.

Regression Models: The accuracy of the regression model is measured in terms of Mean Square Error (MSE) with the data normalized with zero mean and unit variance. We have also presented the mean absolute error to understand how the speed prediction varies from the real speed. From Table II, we can see that the MSE with new features is less than compared to the MSE of the regression models used in our previous work.
With the new features, we are able to achieve lower MSE when the window size is reduced to 20 ms. Lower window size allows us to have higher 3D object reconstruction accuracy when the dimension of the 3D object being printed is lower. We can see that the MSE is relatively higher for the value predicted by the regression model for the motion in Y-axis when the motion is occurring at two axes. However, this error can be removed during the post-processing stage as the travel feed rate is generally similar between consecutive frames in each layer of printing.

**Table II: Accuracy of the Regression Models.**

<table>
<thead>
<tr>
<th>Regression Model</th>
<th>Movement Axis</th>
<th>MSE [Al Faruque et al. 2016] Normalized (win 50 ms)</th>
<th>MSE Normalized (MODWT and STFT)(win 50 ms)</th>
<th>Mean Absolute Error (win 50 ms) (MODWT and STFT) mm/minute</th>
<th>Mean Absolute Error (win 20 ms) (MODWT and STFT) mm/minute</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>Only X</td>
<td>0.00616</td>
<td>0.00292</td>
<td>5.876</td>
<td>8.8786</td>
</tr>
<tr>
<td>Y</td>
<td>Only Y</td>
<td>0.01874</td>
<td>0.01201</td>
<td>18.3721</td>
<td>23.5821</td>
</tr>
<tr>
<td>X</td>
<td>X and Y</td>
<td>0.1658</td>
<td>0.04641</td>
<td>91.4723</td>
<td>140.045</td>
</tr>
<tr>
<td>Y</td>
<td>X and Y</td>
<td>0.4290</td>
<td>0.25120</td>
<td>268.07</td>
<td>294.0423</td>
</tr>
</tbody>
</table>

Figure 11 shows that there is a linear relationship between the real speed and the predicted speed computed by the regression model.

**Fig. 11: Prediction Results for Regression Models.**

**Fig. 12: Features Segmented with Varying Motion.**

Figure 12 shows the feature comparison conducted for the audio recorded while the 3D printer is printing an object. We can observe when the nozzle changes its direction by analyzing the distance of features between successive frames. Peaks are extracted by applying the threshold obtained during the training phase. A value higher than the threshold is 1, otherwise 0.
6. RESULTS FOR TEST OBJECTS

In order to test our attack model, we have defined various benchmark parameters which affect the accuracy of the attack model. While printing, multiple similar 2D layers are printed to achieve a 3D object. Hence, our test objects also consists of 3D objects with simple 2D geometry (such as square, triangle, etc.) repeated over multiple layers. The various benchmark parameters in designing these test objects are as follows:

1. Speed of Printing: The fixed frame rate affects the temporal and spectral features extracted from the audio. With the increase in the speed, faster rate of change of spectral features will not be captured and this can degrade the performance of the attack model. Hence, the speed of printing is varied to test the accuracy of the attack model.

2. Dimension of the Object: With smaller objects, shorter nozzle movements are present. To represent these shorter movements, temporal resolution of the features is increased by making the frame size smaller. To test our attack model with smaller objects, we vary the size of the object being printed.

3. Complexity of the Object: Printing a complex object incorporates movement in more than one axis. Hence, to increase the complexity of the object being created, we have tested the acoustic model with shapes consisting of simultaneous multiple axis movement, such as a triangle.

For each of the test objects, to test the reconstruction capability of the attack model for 3D objects, we have printed it in five layers. Since the attack model uses post-processing to remove any layer that is not overlapping with the previous layers, having larger number of layers with same base 2D outline throughout would increase the accuracy of the attack model. Hence, we have chosen smaller number of layers as a proof of concept, and for demonstration purpose. In order to provide the result in a meaningful manner, instead of calculating the mean square error, we use the Mean Absolute Percentage Error (MAPE) for the distance prediction.

\[
MAPE = \frac{1}{n} \sum_{t=1}^{n} \left| \frac{A_t - P_t}{A_t} \right|
\]

Where \(A_t\) is the actual speed and \(P_t\) is the speed predicted by the attack model. Since the frame size (20 ms) is same for all the features, the distance calculation error is also be given by Equation 25. Table III provides results for the different parameters used to test the accuracy of the attack model. The average classification accuracy and regression MAPE before the post-processing stage are 74.43% and 13.17%, respectively. This is an improvement over the results of our previous work [Al Faruque et al. 2016], where the classification accuracy is 66.29% and regression MAPE is 20.91%. After the post-processing stages, the classification accuracy is 86%, and the regression MAPE is 11.11%. This result is also an improvement over the classification accuracy of 78.35% and regression MAPE of 17.82% after post processing stages presented in our previous work [Al Faruque et al. 2016] due to better feature extraction achieved by combination of MODWT and STFT.

6.1. Reconstruction of a Square

A square incorporates movements of stepper motors in all axes, however, one at a time. From Table III, we can see that the accuracy of the classifier for reconstructing the G-code is as high as 92.56% with MAPE of just 4.83%. After post-processing stages the same accuracy has been increased to 99.51% for the classifier with MAPE of just 3.39%.
### Table III: Test Results for Square and Triangle.

<table>
<thead>
<tr>
<th>Dimension (mm)</th>
<th>Speed (mm/min)</th>
<th>Regression MAPE (%)</th>
<th>Classification Accuracy (%)</th>
<th>Classification Accuracy App I (%)</th>
<th>Classification Accuracy App II (%)</th>
<th>Regression MAPE App II (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square (side)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>900</td>
<td>4.83</td>
<td>92.56</td>
<td>99.51</td>
<td>99.51</td>
<td>3.39</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>6.80</td>
<td>91.82</td>
<td>97.68</td>
<td>97.68</td>
<td>5.14</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>9.37</td>
<td>85.91</td>
<td>89.34</td>
<td>89.34</td>
<td>7.65</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>15.47</td>
<td>82.43</td>
<td>88.27</td>
<td>88.27</td>
<td>11.21</td>
</tr>
<tr>
<td>10</td>
<td>900</td>
<td>8.72</td>
<td>76.72</td>
<td>84.63</td>
<td>86.62</td>
<td>5.61</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>12.64</td>
<td>69.44</td>
<td>74.07</td>
<td>74.07</td>
<td>8.91</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>17.44</td>
<td>62.05</td>
<td>62.64</td>
<td>62.64</td>
<td>12.51</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>22.91</td>
<td>57.29</td>
<td>58.71</td>
<td>58.71</td>
<td>18.99</td>
</tr>
<tr>
<td>5</td>
<td>900</td>
<td>4.48</td>
<td>90.85</td>
<td>95.06</td>
<td>98.71</td>
<td>4.46</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>5.19</td>
<td>88.70</td>
<td>90.18</td>
<td>98.71</td>
<td>5.11</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>6.85</td>
<td>85.15</td>
<td>86.18</td>
<td>91.08</td>
<td>6.73</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>10.33</td>
<td>78.77</td>
<td>84.30</td>
<td>84.30</td>
<td>9.82</td>
</tr>
<tr>
<td>Triangle (base, height)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>30,20</td>
<td>900</td>
<td>5.45</td>
<td>87.50</td>
<td>94.77</td>
<td>97.35</td>
<td>4.80</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>7.41</td>
<td>84.01</td>
<td>90.14</td>
<td>90.14</td>
<td>6.37</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>9.22</td>
<td>79.99</td>
<td>85.61</td>
<td>85.61</td>
<td>8.88</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>16.02</td>
<td>77.41</td>
<td>79.39</td>
<td>83.13</td>
<td>14.25</td>
</tr>
<tr>
<td>20,20</td>
<td>900</td>
<td>16.57</td>
<td>70.84</td>
<td>78.62</td>
<td>82.37</td>
<td>14.77</td>
</tr>
<tr>
<td></td>
<td>1200</td>
<td>21.43</td>
<td>68.23</td>
<td>74.90</td>
<td>74.90</td>
<td>19.69</td>
</tr>
<tr>
<td></td>
<td>1500</td>
<td>32.78</td>
<td>63.93</td>
<td>72.28</td>
<td>74.44</td>
<td>28.55</td>
</tr>
<tr>
<td></td>
<td>1700</td>
<td>38.41</td>
<td>58.51</td>
<td>65.71</td>
<td>65.71</td>
<td>35.21</td>
</tr>
<tr>
<td>10,5</td>
<td>900</td>
<td>13.17</td>
<td>74.43</td>
<td>84.89</td>
<td>86.00</td>
<td>11.11</td>
</tr>
<tr>
<td></td>
<td>Average</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>20.91</td>
<td>69.29</td>
<td>76.04</td>
<td>78.35</td>
<td>17.82</td>
<td></td>
</tr>
</tbody>
</table>

We can also observe that as the travel feedrate increases to 1700 mm/min, the accuracy of the classifier and the regression model decreases. Also for short movements such as 5mm, the accuracy of the attack model decreases. Figure 13 (a) shows the square reconstructed by the attack model for a side length of 20 mm.

6.2. Reconstruction of a Triangle

While constructing a triangle, both the X and Y stepper motors move, and this affects the reconstruction accuracy of the attack model. From Table III, we can see that the accuracy of the classifier of the attack model is as high as 90.85% with MAPE of just 4.48% before post-processing and after post-processing they are 98.71% and 4.46%, respectively. As expected, the accuracy of the learning algorithms decreases with an increasing speed and decreasing length of the movement. Also, classification and regression accuracy of reconstructing the triangle are less when compared to the square. Figure 13 (b) shows the reconstructed shape of the triangle. For higher travel feedrates, the accuracy of the reconstruction is lower.

6.3. Case Study: Outline of a Key

As a case study, to test the attack model against a combination of shapes, we have printed an object representing the outline of a key at 900 mm/min travel feedrate. The classification accuracy obtained for this object before post-processing is 88.83% and the regression MAPE is 5.62%. This is better than the results presented in our earlier work [Al Faruque et al. 2016], where the classification accuracy obtained before the post-
processing is 83.21% and the regression MAPE is 9.15%. After the post-processing, the classification accuracy obtained is 96.32% and the regression MAPE obtained is 3.92%. This is also an improvement over the result presented in our previous work [Al Faruque et al. 2016], where the classification accuracy obtained is 92.54% and the regression MAPE obtained is 6.35%. The object reconstructed by the attack model is shown in Figure 14. As we can see, before post-processing stage II, there are some non-uniform lengths in each of the layers of the object. However, after post-processing stage II, these errors are corrected. In terms of dimension, we can observe that the reconstructed key varies in length and width compared to the original object. Nevertheless, the general outline of the key is reconstructed accurately. Moreover, the accuracy in terms of the length obtained after the post-processing stage is 92.48%, which is calculated by dividing the difference between the original length and the predicted length of each segment in each of the layers by the total length of all the segments in all the layers. This is better than the results presented in [Al Faruque et al. 2016], where the accuracy is 89.72%. On increasing the travel feedrate for the given test case, we can ex-
pect higher distortion as demonstrated by the results for reconstruction of test objects such as square and triangle. Where increasing the speed caused drastic misalignment of layers, error in direction prediction, and imprecise dimension prediction.

7. DISCUSSION

Technology Variation: We have focused our experiment in the FDM technology based 3D printers. The key assumption for the attack model is that there is a correlation between the G-code and the radiated sound, and an attacker is capable of acquiring these radiated signal. With this, any 3D printer that is capable of suppressing the acoustic emission from the printer can effectively avoid this kind of attack. For example, other 3D printing technologies such as StereoLithography (SLA), Selective Laser Melting (SLM), Electron Beam Melting (EBM), etc., that uses UV-laser, high beam laser, electron beam etc., to harden either the liquid resin or metal powder, have minimum number of components that can generate acoustic emissions. In such scenarios, different analog emissions (magnetic, power, thermal, etc.) may be used for breaching the confidentiality of the system.

Sensor Position: In our experiment, we have placed the sensor at a fixed position (20 cm) from the 3D printer. However, the sensor position plays an important role in the accuracy of the system. For instance, if the sensor is placed on top parallel to Z-axis of the 3D printer, the variation in X and Y axis direction will be difficult to capture. We acknowledge that a better sensor position exploration can be done to understand how the position affects the accuracy of the attack model. We will incorporate this analysis in our future work.

Sensor Number: In this paper, we have only used a single acoustic sensor to acquire the acoustic emissions. However, a motivated attacker might have multiple sensor placed around the 3D printer to acquire the signals. This may change the accuracy of the attack model. It will help in better localization of the source of acoustic, and remove other environmental noise (for example using blind source separation method). However, angle and distance between these sensors have to be explored before it can be used for data acquisition.

Dynamic Window: Due to the fixed frame size incorporated for feature extraction in our experiments, the accuracy of the attack model is reduced for higher speeds and smaller dimensions. In order to capture smaller movements, the temporal resolution of the features extracted has to be increased by making the frame size smaller. However, for faster speeds, we need larger frame size to increase the frequency resolution for better spectral features. This trade-off dictates that we should incorporate adaptive frame size to increase the accuracy of the attack model. We will incorporate this in our future work. We have tried to address this issue with MODWT based features, however, it does not incorporate adaptive framing.

Feature Separation during Multiple Axis Movement and Noise: The separation of sound source from combination of sound is a well known problem in speech processing. In our attack model, we have incorporated spectral subtraction to acquire features that are unique to each of the stepper motors. However, there are other sound separation methods [Barry et al. 2004; Pedersen et al. 2007]. Moreover, we have used only one audio sensor for separating two features. Incorporating two or more audio sensors may improve the results further.
**Target Machine Degradation:** Over longer period of time, due to mechanical degradation, the vibration produced by the 3D printer will vary compared to the new models. Since an attacker cannot access the target model for estimating the leakage function, they might have less accuracy in stealing the information. However, one possible solution to tackle this issue would be to continuously update the model function using a 3D printer model that closely handles the same work load as is done in the industry through accelerated aging test methods, and capture the degradation trend to isolate any noise caused by the mechanical wear and tear that does not aid in information leakage. We will incorporate this in our future work.

In our recent work [Rokka Chhetri et al. 2016], we have incorporated the result of the finding of this work to provide a methodology for detection of attacks on firmware of the 3D printers. Moreover, we have also presented security measures for preventing the information leakage from the manufacturing system [Rokka Chhetri et al. 2017] by providing leakage aware computer aided manufacturing tools.

8. CONCLUSION

This paper introduces and demonstrates a novel acoustic side-channel attack model on additive manufacturing systems. We present an analysis on sources of acoustic emission in a fused deposition modeling technique based 3D printer, and perform the leakage analysis to highlight the parameters of G-code that can be inferred from the acoustic side-channel. We have incorporated the maximum overlap discrete wavelet transform in our feature extraction to acquire better G-code and 3D object reconstruction results than short term fourier transform and mel-frequency cepstral coefficient features based attack model. The validation of our novel attack model with a state-of-the-art 3D printer shows that objects with different benchmark parameters such as speed, dimension, and complexity can be reconstructed using acoustic side-channel attacks. Our experiments show an average axis prediction accuracy of 74.43%, and average length prediction error of 13.17%. Furthermore, with post-processing, we have achieved a high average axis prediction accuracy of 86% and average length prediction error of 11.11%. As it is explained in the discussion section, there are several research challenges that remain open. However, our work serves as a proof of concept for the possibility of physical-to-cyber attacks on cyber-physical additive manufacturing systems.

REFERENCES


Takashi Kenjō and Akira Sugawara. 1994. *Stepping motors and their microprocessor controls*. Oxford University Press, USA.


François-Xavier Standaert and others. 2009. A unified framework for the analysis of...


**Sujit Rokka Chhetri** is currently pursuing his PhD in Computer Engineering at the University of California Irvine (UCI). He has been working at the Advanced Integrated Cyber-Physical Systems Lab under the supervision of Professor Mohammad Al Faruque since Summer of 2015. His current research is focused on embedded and cyber-physical systems modeling, design automation, and security. He received Distinguished Best Poster Award at NDSS 2016.

**Arquimedes Canedo** is currently a Principal Key Expert Scientist at Siemens Corporate Technology. In this role, he provides technical leadership to a team of 150 scientists and engineers in the area of Automation and Control. His research interests include cyber-physical (production) systems, design automation, programming languages and compilers. He holds more than 30 patents and has published more than 40 papers in top journals and conferences.

**Mohammad Abdullah Al Faruque** is currently with the University of California Irvine (UCI), where he is a tenure track assistant professor and directing the Cyber-Physical Systems Lab. Before, he was with Siemens Corporate Research and Technology in Princeton, NJ. Prof. Al Faruque is the recipient of the IEEE CEDA Ernest S. Kuh Early Career Award 2016. He received the B.Sc. degree in computer science and engineering from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, in 2002; the M.Sc. degree in computer science from RWTH Aachen Technical University, Aachen, Germany, in 2004; and the Ph.D. degree in computer science from Karlsruhe Institute of Technology, Karlsruhe, Germany, in 2009. Prof. Al Faruque’s current research is focused on system-level design of embedded systems and Cyber-Physical-Systems (CPS) with special interest on model-based design of embedded software and CPS security. Prof. Al Faruque received several best paper awards, e.g., the 2015 DAC Best Paper Award, the 2016 DATE Best Paper Award, 2009 ICCAD Best Paper Award, the 2008 HiPEAC Paper Award, and the 2016 NDSS Distinguished Poster Award. Besides 60+ IEEE/ACM publications in the premier journals and conferences, he holds 5 US patents.