The International Journal of Advanced Manufacturing Technology (2018) 96:581–586
https://doi.org/10.1007/s00170-018-1603-z

ORIGINAL ARTICLE

# Cyber acoustic analysis of additively manufactured objects

Thomas Mativo[1] · Colleen Fritz[2] · Ismail Fidan[3]

Received: 30 October 2017 / Accepted: 10 January 2018 / Published online: 24 January 2018

## Abstract

The potential for intellectual property theft has been shown in the additive manufacturing industry using acoustic side-channel attacks lately. This paper aims to discuss the rate of success for recreating the G-Code of an object from the acoustic features and further elaborates on regression model analysis that provides the G-Code. Acoustic and G-Code data was analyzed in a training phase and an attack phase. In the training phase, a supervised machine learning algorithm was trained using Python, which is an interpreted, object-oriented, high-level programming language. During the attack phase, the created algorithm was used to process new acoustic data and to reconstruct the G-Code. The accuracy of the classification models and the regression models were determined. The classification accuracy was determined with k-fold cross validation, and the regression model accuracy was determined by scoring the regression models within the algorithm. Although classification and regression algorithms developed showed promising results, lower model accuracy was observed when the $X$ and $Y$ motors moved together. In the future, the team hopes to further increase the model accuracy so that an unknown shape can be replicated successfully. While security measures for cyber-security have previously been investigated, very little research has considered acoustic side-channel attacks on their ability to reconstruct G-Code and steal intellectual property. The findings of this novel research project showed some promising preliminary results on a sample case study.

**Keywords** Additive manufacturing · Cyber-security · Intellectual property · Supervised machine learning · Python

## 1 Introduction

Considered the start of a new industrial revolution with industry revenue expected to exceed 21 billion dollars by 2020 [1], additive manufacturing (AM) allows production of objects in a one step process, making product iterations readily available and without the overhead cost of molding [2]. An example of a cyber-physical system (CPS), an AM system incorporates physical hardware with a software system that is typically connected to a network [3]. But along with the convenience of AM, potential risks associated with the intellectual property (IP) rights are present. It has become easier than ever to copy a product with AM methods, creating the potential for a loss of IP [2]. IP in AM includes the structure of the device being printed and the parameters of the process as well as the specific machine used. A printed object can be reproduced solely with the G-Code (IP), which gives the machine directions in relation to speed, temperature, and extrusion amount [1]. Current security measures have focused on securing machines against cyber based attacks with cloud-based resources and software programs. However, hackers are able to steal computer-aided design (CAD)-based models which can be used to reproduce the parts with the same qualities as the original component [4]. Previous research has focused on protecting the cyber domain from IP theft, such as by altering design features in CAD files, resulting in inferior prints if a unique set of slicing conditions and other parameters are not met prior to printing. Security features embedded into the files can help prevent the exposed vulnerabilities due to the digital nature of the cyber-domain technology [4]. Nevertheless, attacks in the physical domain have also occurred. The cyber-physical domain consists of physical infrastructures that are utilized to provide cyber services [5]. Past research studies illustrate the importance of security within the cyber-physical domain ([3, 6], and [7]). The physical component of AM machines

✉ Ismail Fidan
ifidan@tntech.edu

[1] University of Dayton, Dayton, OH, USA

[2] University of Alabama, Tuscaloosa, AL, USA

[3] Tennessee Tech University, Cookeville, TN, USA

opens up the system to vulnerabilities due to side-channels. Side-channels are indirect pathways that lead to the access of desired data such as obtaining G-Code from vibrational, acoustic, magnetic, or power emissions. Previous analysis of side-channels has been used to infer information about cyber domain data. Therefore, it is important to analyze these side-channels to better secure the system and prevent leakage of IP [8].

## 2 Background and related work

The typical AM process chain is depicted in Fig. 1. In order to first begin, a CAD model is created which is then converted into an .STL file, that consists of coded instructions for the slicing software (i.e. Cura) to create the G-Code [9]. Triangular facets are created on the surface of the solid model, which correspond to numerical data. It is through these triangular facets that the model can be sliced to get the contours of each layer thereby creating the G-Code [10]. After being sliced, the G-Code is given to the AM system and the part is printed. After printing is completed, any creative supports are removed, and the part is finished. Previous research has successfully shown a proof of concept for acoustic side-channel attacks on AM systems using either a Zoom H6 recording device or a Nexus 5 smartphone in addition to electromagnetic signals in order to extract the G-Code [1, 11]. Al Faruque's team achieved nearly 90% accuracy using the sound copying process to duplicate a key-shaped object in his laboratory. After such a study, Xu's team gathered enough acoustic data to enable his researchers to replicate printing a simple object with a 94% accuracy rate.

The purpose of this work was to try to improve upon what the previous works could accomplish by incorporating machine learning and deep learning to the training and attack algorithms. Inclusion of these techniques will allow for more accurate shape reconstruction in AM.

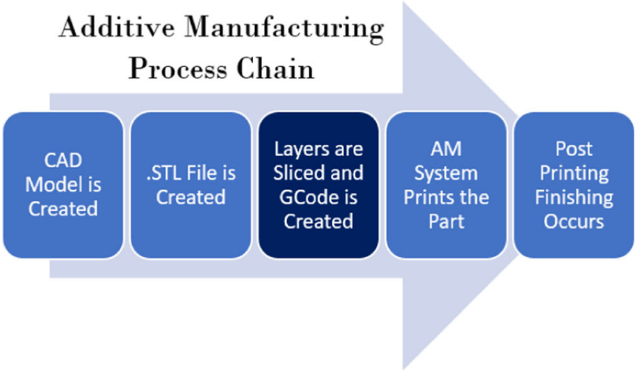
Additive Manufacturing Process Chain
CAD Model is Created
.STL File is Created
Layers are Sliced and GCode is Created
AM System Prints the Part
Post Printing Finishing Occurs

**Fig. 1** Additive manufacturing process chain

## 3 Methodology

The algorithm works in two phases, the training phase and the testing (attack) phase. During the training phase, the machine specific algorithm is developed using supervised machine learning which links the sounds emitted from the printer to the G-Code file. This algorithm is feasible because the stepper motors used in printers, which create audible sound while printing, emit different sounds based on movement. Each of the four motors for $X$, $Y$, and $Z$ movement as well as one for extrusion create different audio signals because the load on each motor is different from motor to motor [1]. Priority is given to $X$ axis motion in start up since both $X$ and $Y$ axes have the same stepper motors. Compared to the $X$, $Y$, and $Z$ effects of the stepper motors, the acoustics of the extrusion nozzle was low, and it was not considered in this study. The accelerating and decelerating effect of the motors was not considered during this study either. The audible sounds of an object are then paired with the known G-Code and used in classification and regression analysis to create an algorithm to be used in the attack phase. During the attack phase, a potential attacker would place a recording device near a printer of the same model that was used during the training phase and record the audio signals. Then, using the classification and regression algorithms created in the training phase, the attacker could determine the G-Code of the object without ever having to steal the CAD file that contained it. Thus, the potential of IP theft is inherently present in modern day AM systems.

## 4 Training phase set-up

In order to complete the training phase, an Ultimaker 2 Extended+ was used to print the objects tested in this paper [12]. As it can be seen in Fig. 2, a Zoom H1 recording device was used at a distance of 20 cm from the corner of the printer at a 45-degree angle from the front of the device, thereby preventing one of the $X$ or $Y$ motor sounds from overpowering the other [13]. The Zoom H1 incorporated a 90° $X|Y$ stereo microphone recorded at 24 bit/96 kHz into a waveform audio file (WAV). In order to prevent interference from another audio

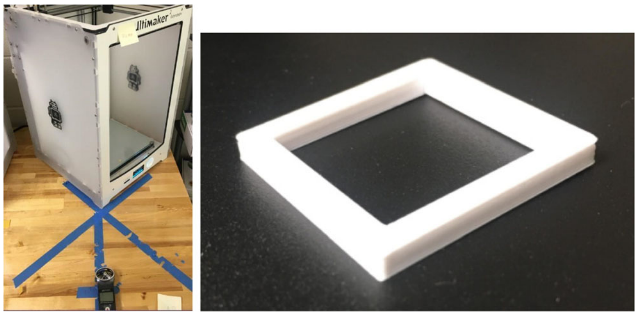

**Fig. 2** 3D printer with audio recorder (left); 3D printed shape (right)

source, testing was done in a silent room. The team chose to keep the recordings at about 15 minutes, therefore preventing the algorithms, especially the classification cross-validation analysis, from taking too long to run. However, an increase in the length of the audio could potentially increase the accuracy for simple shapes as it would provide additional layers which could be averaged out to determine the true shape of the structure, more testing is needed to confirm or deny this hypothesis.

## 5 Preprocessing the data

Before the data could be run through the algorithm, preprocessing was done on the audio data through Audacity (an audio file editor). The Zoom H1 recording device records in stereo which was converted to mono in Audacity. To prevent sound frequencies unrelated to printing from affecting the data, low-pass and high-pass filters were applied to the audio signals in Audacity. The audio file was also cut to eliminate any sound that occurred before and after printing of the object so that the audio and G-Code files would line up correctly. The audio was then converted from 96 to 16 kHz. The low-pass filter was off on the recording device. However, high- and low-pass filter options were used in Audacity. The sampling rate from the Zoom H1 was set to 96 kHz at a 24-bit rate in stereo (wav format). This data was then loaded into Audacity and converted to mono at a 16 kHz, 16-bit sampling rate.

## 6 Algorithm creation

The collected audio data was chopped into frames of 50 ms and put into a matrix that corresponded to the G-Code file length. Then supervised machine learning was used for feature extraction. Supervised machine learning is the branch of machine learning where the data in the training set has known inputs and outputs. Correlations are determined that create a predictive model for new data [14]. The created algorithm contained both classification and regression models for the data. The classification models were used to determine which motor was running at any given audio frame for a new set of untested audio data. After this was determined, the newly classified data was then run through the regression algorithms much like the previous known training data would be. The regression models were used to predict the speed at which the motor was running for both the training data and the testing data. The theory is expressed in Fig. 3. The goal of this research was to take known audio and G-Code data and complete a regression model that can predict the G-Code of unknown audio data after being run through the classification and regression algorithms. However, accuracy could only be determined for two different types of models with the known data files during the training phase. In future work, the goal is to be able to complete the full classification and regression models, and focus on the attack phase with an unknown audio file.

## 7 Classification algorithm

Figure 4 is a representation of the classifiers that were used. There were four classifiers used in the algorithm, presented here. The *Z* classifier determined if there was movement in the *Z* direction. The second classifier determined whether both the *X* and *Y* motors were moving or if only one of them was moving. If only one of the motors was moving, the third classifier determined which axis the movement was in. If both motors were moving, the last classifier determined whether they moved at the same speed or if they moved at different speeds. For classification, features in both the time and frequency domain were used, consisting of short time Fourier Transforms, mel frequency cepstral coefficients, zero crossing rate, and frame energies. To perform the classification, support vector classification (SVC) was completed for each of the four different classifiers.

## 8 Regression algorithm

The regression models were based on the extracted features named above. Three regression models were developed. One for the *X* direction, one for the *Y*, and one for both *X* and *Y* motion. Because the distance moved by the *Z* motor was constant, a regression model for the *Z* direction was not necessary. The predicted feature matrix generated by the regression models was then fed into another algorithm to achieve the predicted speed of the printer head. After the predicted speed data is obtained, it is used to calculate the predicted travel distance of the nozzle. The values of travel distance and speed are then used to reconstruct the G-Code based on regression model predictions. The regression model that was first tested, a linear regression model, had an accuracy between 20–50%. The model was then changed to a logistic regression model which saw the accuracy increase to 70–80%. It is believed that accuracy remains below 90% due to the enormous size of the

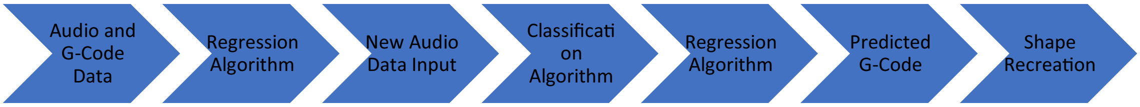

**Fig. 3** Process for G-Code recreation

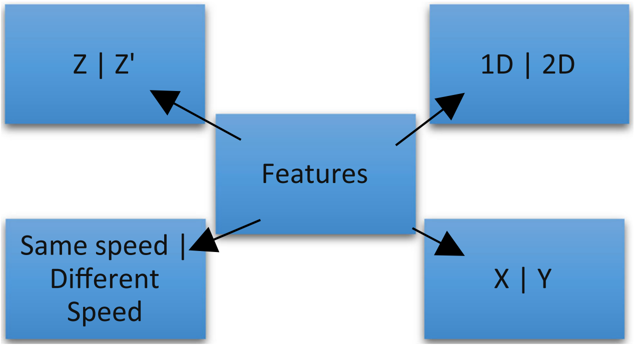

**Fig. 4** Decision tree for motor classification

feature matrix. With future work, unrelated features could be determined and removed to increase the accuracy and efficiency of the algorithm. The input data had four features which generated 442 different divisions of the data, which make it difficult for logistic regression to work effectively.

## 9 G-Code reproduction

To recreate the G-Code, outputs from the classification and regression models need to be combined. The team mainly focused on increasing the accuracy of the classification and regression models with the training data. Therefore, the recreated G-Code was based upon the accuracy of the regression models that were used. The speed was able to be determined for each of the different classifications of *X*, *Y*, *XY* at the same speed, and XY at different speeds. The speed was then multiplied by the frame size in order to determine the distance traveled because the G-Code uses change in distance not speed to direct the printer. Finally, predicted axis, distance, speed, and extrusion values were all concatenated into the same matrix. These values were then saved as a comma separated values file (csv) and exported into an Excel document. A few minor changes were made in Excel to ensure that the values obtained from Python had the correct format [15]. After this, the values were saved as a .gcode file and the file was exported to a printer simulator environment.

## 10 Results

### 10.1 Classification accuracy

The classification accuracy was determined for each branch of the decision tree and recorded as seen in Table 1. A support vector machine classification model was used for each of the four data sets. To determine the accuracy score k-fold cross-validation was used. The data was split into ten different data sets. Then, a support vector machine was trained on nine tenths of the data and tested on the last tenth, ten different times, each time leaving out a different tenth of the data. The accuracy for each of these ten tests was determined and the mean was taken and reported in Table 1. The classification model was then used to determine which motor was moving at a given time for a new set of audio data during an attack phase. The information from the classifiers was important for regression accuracy. Once the active motor was determined, it was paired with the audio features to determine the speeds and distance traveled at each respective motor.

**Table 1** Classification accuracy for case study hollow rectangle

| Classifier | Cross Validation Accuracy Score* (%) |
|---|---|
| *Z* \| *Z'* | 99.98 |
| 1D \| 2D | 93.30 |
| *X* \| *Y* | 93.30 |
| Same speed \| different speeds | 61.18 |

*The k-fold cross-validation accuracy score given was based upon the hollow rectangle (discussed in the case study). Different shapes yielded different cross validation accuracy scores, but generally were in the same range as the score shown above

### 10.2 Regression accuracy

A logistic regression model was used in determining the speed values of the printer. The logistic regression model proved to be more accurate than the linear regression model. Because movement in the *Z* direction was set to standard increments by the printer, a regression model for the *Z* direction was omitted. A regression score for each of the motors was determined and recorded as shown in Table 2.

### 10.3 Shape reproduction

A simulation environment was used to recreate the object obtained from the final G-Code. The predicted data and actual data did not match completely, so the print data was simulated in GCodeSimulator (a Java applet) to avoid physical printing issues and errors [16]. When confident that no harm would come to the actual printer, the predicted shape was printed out.

**Table 2** Regression accuracy scores for case study hollow rectangle

| Regression model | Regression score** (%) |
|---|---|
| *X* | 79.05 |
| *Y* | 82.17 |
| *XY* same speeds | 97.40 |
| *XY* different speeds | 78.81 |

**The regression score was based upon one shape and different shapes gave different regression scores, but generally the scores were in the same range as the scores shown in this table.

The printed shape was post-processed to more resemble the original shape. Any extra filament in the center created by an error in the regression algorithm was removed and gaps in the square were filled in. As it can be seen in Fig. 5, the basic general outline of the original shape was almost identical to the printed/predicted shape after the post processing operations.

## 11 Discussion

Although the team was not able to fully test the capabilities of the algorithm to test an unseen audio file, the team was able to test the accuracy of two algorithms that might eventually be able to do this recreation. The biggest setback in the experiment was being unable to correctly incorporate the XY regression model when both motors were moving into the G-Code, especially when the *X* and *Y* motors were moving at different speeds. In order to solve this issue, the team attempted to average the movements of the *X* and *Y* motors and subtract it from when the motors moved at the same time. However, it was still difficult for the created algorithm to properly distinguish the distance. Because of this, the most accurate model was the creation of the hollow rectangle (square). Due to the shape, most of the printing was done in either the *X* or *Y* domain, separately. For the square, the printing that was done with both motors active was due to the fill structure and therefore did not affect the general shape. The *X* and *Y* regression models could determine the general shape of the object being reproduced. With more accurate regression models, better recreation of the square may be obtained.

## 12 Case study: hollow rectangle

Classification and regression models were completed on the audio and G-Code data and the accuracy of the results may be found in Tables 1 and 2. The purpose of the classification model was to later determine which motor was running from untested audio data. This information was then put into the regression algorithm. The purpose of the regression algorithm was to determine the changing distance of the motor by using pre-classified data from the training phase for the newly classified data based upon the classification algorithm from the testing phase. Like the classifier, there were also four different regression models. For the classifiers, one model determined whether or not the *Z* axis was running. The next determined whether movement was in one dimension or two. The third was for one dimensional data and determined whether or not it moved in the *X* or *Y* direction, and the last was for two-dimensional data and determined whether or not the two *X* and *Y* motors were moving at the same speeds or at different speeds. For analysis, efforts were focused on the regression algorithm in the training phase and the processed data was used to recreate the shapes as shown in Fig. 5. The outer sides of the created shape closely resembled the model that was printed with the original G-Code file, with the exception being the addition of the inner wall matrix which was removed before the photo and placed next to the finished post-processed model. To see if the accuracy could be improved, the data for concurrent *X* and *Y* motor movements at the same speeds as well different speeds was added. When included, the model did not appear to be as accurate as the previous model. The team believes this is due to the inner structure of the original printed shape, which was created with the movement of the *X* and *Y* motor at the same time in a diagonal motion. This data was not important for obtaining the general shape of the model, so the team chose to stick with the analysis done in the *X*, *Y*, and *Z* directions.

In the future, the goal is to eliminate this reduction of accuracy with the inclusion of more regression models to create a more holistic model that is capable of the recreation of shapes that do not only contain right angles and straight lines. An object similar to the real object was able to be obtained through regression analysis in the testing phase as shown in Fig. 5.

## 13 Conclusion

This research study developed a new working algorithm which successfully ran on both simulation software and on an Ultimaker Extended 2+ printer which could determine the general shape of a sample AM object with very high accuracies. The current study shows promising results for the possibility of using this model for complex shapes in the future,

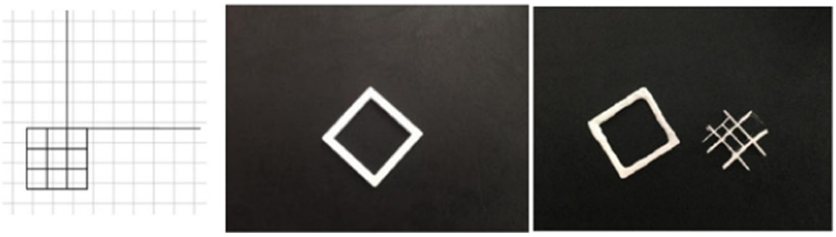

**Fig. 5** Simulated predicted shape (left), original shape (center), printed/predicted shape (right)

which previous research studies have not addressed. The inconsistency present in the shape was due to the created algorithm, which could not accurately predict the speed, resulting in the data being inconsistent with the original G-Code. To omit the inaccurate data, the experiment focused on motor movements that occurred either in the *X*, *Y*, or *Z* direction only. Tests were then limited to shapes created with right angles. For future work, once the accuracy of the classification and regression models is increased or a more accurate algorithm is introduced, any untrained audio file could be run through the entire algorithm and more complex shapes with different angles could be introduced. As model accuracy increases, the possibility of theft of more complex IP data will rise in the AM industry, potentially undetected by traditional security measures.

**Acknowledgements** The authors would like to convey their deepest thanks and appreciation to Nick Russell, Serhat Sahin, Mahmoud Nabil, Justin Medley, Astrit Imeri, Cesar Ortiz, Yolnan Chen, and Kyle Wendt for their help and support during this research study.

**Funding information** This research was supported by the National Science Foundation Grant Awards 1461179 and 1601587.

## References

1. Mohammad Abdullah Al Faruque, Sujit Rokka Chhetri, Arquimedes Canedo, and Jiang Wan, (2016). Acoustic side-channel attacks on additive manufacturing systems. In Proceedings of the 7th International Conference on Cyber-Physical Systems (ICCPS '16). IEEE Press, Piscataway, NJ, USA, Article 19
2. Weller C, Kleer R, Piller F (2015) Economic implications of 3D printing: market structure models in light of additive manufacturing revisited. Int J Prod Econ 164:43–56. https://doi.org/10.1016/j.ijpe.2015.02.020
3. Sturm L, Williams C, Camelio J, White J, Parker R (2017) Cyber-physical vulnerabilities in additive manufacturing systems: a case study attack on the .STL file with human subjects. J Manuf Syst 44(Part 1):154–164
4. Chen F, Mac G, Gupta N (2017) Security features embedded in computer aided design (CAD) solid models for additive manufacturing. Mater Des 128:182–194. https://doi.org/10.1016/j.matdes.2017.04.078
5. Rao N, Poole S, Ma C, He F, Zhuang J, Yau D (2016) Defense of cyber infrastructures against cyber-physical attacks using game-theoretic models. Risk Anal: Int J 36(4):694–710. https://doi.org/10.1111/risa.12362
6. Petnga Leonard & Xu Huan. (2016). Security of unmanned aerial vehicles: dynamic state estimation under cyber-physical attacks. pp. 811–819. International Conference on Unmanned Aircraft Systems (ICUAS), Unmanned Aircraft Systems (ICUAS), https://doi.org/10.1109/ICUAS.2016.7502663
7. Taormina, R, Galelli, S, Tippenhauer, N, Salomons, E, & Ostfeld, A, (2017). 'Characterizing cyber-physical attacks on water distribution systems', J Water Resour Plan Manag, 5
8. Rokka Chhetri Sujit & Al Faruque Mohammad Abdullah. (2017). Side-channels of cyber-physical systems: case study in additive manufacturing. IEEE Design Test 18, PP. 1–1. https://doi.org/10.1109/MDAT.2017.2682225, 4, 25
9. Kurfess T, Cass W (2014) Rethinking additive manufacturing and intellectual property protection. Res Technol Manag 57(5):35–42. https://doi.org/10.5437/08956308X5705256
10. Xu Hongwei, Jing Weihua, Li Minjuan, Li Wei (2016). A slicing model algorithm based on STL model for additive manufacturing processes. 1607–1610. https://doi.org/10.1109/IMCEC.2016.7867489
11. Song, C, Lin, F, Ba, Z, Ren, K, Zhou, C, & Xu, W, (2016). My smartphone knows what you print. Conf Comput Commun Secur, p 895
12. "Dynamism - Ultimaker 2 Extended ". (n.d.). *Dynamism.com, next-Generation technology*, available at: http://www.dynamism.com/3d-printers/ultimaker-2-extended-plus.shtml?APC=P4500&gclid=CjsKDwjw5arMBRDz9cK2uen9ORIkAAqmJewXdeu7lwT8tQ0U22o5n-l95VHsgt8WyC6oiWCD83ohGgLH9vD_BwE (accessed 10 October 2017)
13. "Zoom H1 Handy Recorder". (2017), *Zoom*, 23 June, available at: https://www.zoom-na.com/products/field-video-recording/field-recording/zoom-h1-handy-recorder (accessed 10 October 2017)
14. Fabris F, Magalhaes J, Freitas A (2017) A review of supervised machine learning applied to ageing research. Biogerontology 2:171
15. "Welcome to Python.org". (n.d.). *Python.org*, available at: https://www.python.org/ (accessed 10 October 2017)
16. "GCodeSimulator for PC and Android". (n.d.). *GCodeSimulator*, available at: http://3dprintapps.de/gcodesimulator.html (accessed 10 October 2017)