Ensemble Machine Learning for Securing 3D Printers: A Heterogeneous Model of TON_IoT and SideChannel-3D to Safeguard the CIA Triad

by Adam R. Imran

B.S. Statistics, December 2018, University of Wisconsin-Madison
M.S. Analytics concentration Data Science, December 2020, Georgetown University

A Praxis submitted to

The Faculty of
The School of Engineering and Applied Science
of The George Washington University
in partial fulfillment of the requirements
for the degree of Doctor of Engineering

August 15, 2025

Praxis directed by

Thembani Togwe
Professorial Lecturer in Engineering and Applied Science

The School of Engineering and Applied Science of The George Washington University certifies that Adam R. Imran has passed the Final Examination for the degree of Doctor of Engineering as of July 14th, 2025. This is the final and approved form of the praxis.

Ensemble Machine Learning for Securing 3D Printers: A Heterogeneous Model of TON_IoT and SideChannel-3D to Safeguard the CIA Triad

Adam R. Imran

Praxis Research Committee:

Ken Clements, Professorial Lecturer in Engineering and Applied Science, Praxis Chair

Thembani Togwe, Professorial Lecturer in Engineering and Applied Science, Praxis Director

Andrea Molina, Professorial Lecturer in Engineering and Applied Science, Committee Member

© Copyright 2025 by Adam R. Imran
All rights reserved

Dedication

The author dedicates this work to his family.

To my mother Naomi, the wind beneath my sails, who spent countless hours researching key gaps in my paper and who supported me completely.

To my wife Safiya, my rock, who spent many weekends as a newly married couple editing my paper and staying by my side.

To my father Mudassar, my role-model, who's always instilled in me the importance of research and rigorous scientific thought.

To my entire family for their continued support.

Acknowledgments

The author chooses to acknowledge Dr. Nour Moustafa and Ahmed Alsaedi for their continued advancements in publicly available datasets for IoT attacks traffic as well as Nathan Costa and Shih-Yuan Yu for generating the SideChannel-3D dataset. The author chooses to also thank Dr. Thembani Togwe, the advisor for this work for his expertise.

Abstract of Praxis

Ensemble Machine Learning for Securing 3D Printers: A Heterogeneous Model of TON_IoT and SideChannel-3D to Safeguard the CIA Triad

The Fourth Industrial Revolution (Industry 4.0) shifts the paradigms of manufacturing from automation to an interconnected system of manufacturing. Industry 4.0 encompasses Internet of Things (IoT), cloud computing, Artificial Intelligence (AI), Additive Manufacturing, and other emerging technologies. 3D printers can now integrate with other IoT devices wirelessly on networks, elevating them from simple stand-alone tools. 3D printers possess sensors and detectors that collect valuable data for analysis both in real-time and offline. Although these IoT features provide efficiency benefits and quality enhancements, they also invite malicious attackers to target all members of the confidentiality, integrity, and availability (CIA) triad.

An Ensemble Machine Learning detection model based on IoT Attacks is a valuable capability to safeguard 3D printers against a segment of targeted attacks, ultimately preserving the CIA triad. One fundamental question addresses this: What components of an ensemble machine learning (ML) model created on supervised ML models and trained on the TON_IoT and SideChannel-3D data can detect a subset of attacks that can affect a 3D printer's CIA with a high truly malicious detection rate?

A model experimentation methodology approach simulated individual and ensemble classifiers to accurately detect benign and malicious instances for relevant attack vectors with high rates of true malicious detection (i.e., recall). Two data sources were used to train these models, TON_IoT and SideChannel-3D. TON_IoT datasets are central to cybersecurity research for intrusion and detection. However, the datasets are

not specific to 3D printers, or any other single IoT device, but rather includes IoT network sensors to generalize IoT devices. To leverage TON_IoT for specifically 3D printers, the praxis maps the properties of IoT devices with the characteristics of network 3D printers together with representing network traffic for a subset of known 3D printer attack vectors. Ultimately this dataset was used to train the confidentiality and availability components of the CIA triad. The SideChannel-3D which is specific to 3D printers trains the G-code segment data for tampered 3D printed objects affecting integrity.

The results of the analysis displayed a complete success in constructing an ensemble stacking detector with a series of base classifiers and logistic regression meta model achieving a recall greater than the 98.2% benchmark (the average of 3 relevant sources). The high recall rate concludes that a high number of truly malicious instances and events are detected across all members of the CIA Triad. The mean recall observed in this praxis outcomes is 98.97%, performing better than the average of the 3 sources, (Booij et al., 2022; Alotaibi & Ilyas, 2023; Yu et al., 2020).

This praxis successfully filled gaps in prior research by (1) combining two established detection methods into a single ensemble stacking model (2) generating additional representative features for SideChannel-3D data (3) mapping characteristics of IoT devices to networked 3D printers proving they are IoT devices (4) constructing an attack vector schematic to map them to CIA risk and state (5) advancing the IoT detection capabilities using TON_IoT.

Table of Contents

Dedication iv

Acknowledgments v

Abstract of Praxis vi

Table of Contents viii

Tables xi

Glossary of Terms xii

Chapter 1. Introduction 1

1.2 Research Motivation 2

1.3 Problem Statement 4

1.4 Thesis Statement 4

1.5 Research Objective 4

1.6 Research Question & Hypothesis 5

1.7 Scope of Research 5

1.8 Research Limitations 5

Chapter 2. Background & Relevant Research 7

2.1.1 General Trends in Additive Manufacturing 9

2.1.2 Cyber Security & 3D Printing 11

2.1.3 3D Printing Tampering 18

2.2.1 Safeguarding CIA 19

2.2.2 Network Intrusion and Detection Datasets 24

2.2.3: Network Intrusion and Detection Modeling 28

2.2.4 Network Attack Execution 33

2.3 Hypothesis Testing in Attack Detection 36
Chapter 3. Methodology 37
3.1 Dataset Overview 38
3.2 Data Usage & Labeling 44
3.3 Feature Engineering 51
3.4 Individual Model Evaluation 53
3.5 Ensemble Aggregation of Models 55
3.6 Fine-tuning 58
Chapter 4. Results 62
4.1 Introduction 62
4.2 Individual Model Results 62
4.3 Ensemble Model Results 65
4.4 Research Question Hypothesis Testing 70
Chapter 5. Conclusion 74
5.1 Research Question Discussion 74
5.2 Conclusion 76
5.3 Contributions to the Body of Knowledge 77
5.4 Future Works 80
References 82
Appendix 90

Figures

Figure 1: Chapter 2 Roadmap 8

Figure 2: Attack Vector Schematic 11

Figure 3 & 4: Material Removal & Relocation (Pearce et al., 2022) 22

Figure 5: Example Confusion matrix (Encord, 2025) 29

Figure 6: Roadmap of Methodology 37

Figure 7: IoT-based 3D printer design (Prianto et al. 2021) 40

Figure 8: Mapping of preprocessing TON_IoT 45

Figure 9: Highly correlated values from the sets of confidentiality & availability 47

Figure 10: Normal and tampered wrench (Yu et al., 2020) 49

Figure 11: Puncture is a small angle between segments 59

Figure 12: Chapter 4 Outline 62

Figure 13: Individual Model Evaluation on the Confidentiality dataset (TON_IoT) 64

Figure 14: Individual Model Evaluation on the Availability dataset (TON_IoT) 65

Figure 15: Individual Model Evaluation on the Integrity dataset (SideChannel-3D) 65

Figure 16: Confidentiality Confusion Matrix Avg. for Stacking LR with 5-fold CV 68

Figure 17: Integrity Confusion Matrix Avg. for Stacking LR with 5-fold CV 68

Figure 18: Availability Confusion Matrix Avg. for Stacking LR with 5-fold CV 69

Figure 19: Confidentiality ROC Curve 69

Figure 20: Availability ROC Curve 70

Figure 21: Integrity ROC Curve 70

Figure 22: Boxplots for Accuracy, Precision, and Recall for 5-fold CV 73

Tables

Table 1: C3PO Vulnerabilities Category, Observations, and Attack Vectors 17

Table 2: Network Intrusion and Detection Modeling 30

Table 3: Simplified Behavioral Signatures based on TON_IoT 35

Table 4: Mapping TON_IoT Attack Vectors to 3D printing Attack Vectors 42

Table 5: Conn.log Features of TON_IoT 43

Table 6: Features of SideChannel-3D Segments 44

Table 7: Total counts of Attack Categories 45

Table 8: Subsetted Attack vectors and Labels 46

Table 9: Description of final variable transformation 48

Table 10: Variable removal from modeling 50

Table 11: Example of segment data for the benign wrench 51

Table 12: Relevant resources for Recall Benchmark 60

Table 13: Individual Model Performance Metrics 63

Table 14.1: Ensemble Stacking Model Performance Metrics for Confidentiality 66

Table 14.2: Ensemble Stacking Model Performance Metrics for Availability 66

Table 14.3: Ensemble Stacking Model Performance Metrics for Integrity 67

Glossary of Terms

ARP	Address Resolution Protocol
AUC	Area Under the Curve
ASaRP	As Secure as Reasonably Practicable
AAN	Authenticity, Authorization, Non-repudiation
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
CSV	Comma-Separated Variables
CAD	Computer-Aided Design
CIA	Confidentiality, Integrity, and Availability
CNN	Convolutional Neural Network
DT	Decision Trees
DNN	Deep Neural Network
DoS	Denial-of-Service
DNS	Domain Name Service
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FFF	Fused-Filament Fabrication
G-code	Geometric Code

GBC	Gradient Boosting Classifier
HGBC	Histogram Gradient Boosting Classifier
HTAcc	.htaccess file
IIoT	Industrial Internet of Things
IP	Intellectual Property
IoT	Internet of Things
IDS	Intrusion Detection System
KNN	K-Nearest Neighbors
LightGBM or LGBM	Light Gradient Boosting Machine
LAN	Local Area Network
LR	Logistic Regression
MitM / MITM	Man-in-the-Middle
MLP	Multi-Layer Perceptron
NB	Naïve Bayes
NN	Neural Network
OS	Operating System
PCAP	Packet Capture
PBF	Powder Bed Fusion
PLA	Polylactic Acid
RF	Random Forest
ROC	Receiver Operating Characteristic

RNN	Recurrent Neural Network
SSL	Secure Sockets Layer
SAE	Sparce Auto Encoder
SLA	Stereolithography
SCADA	Supervisory Control and Data Acquisition
SVM	Support Vector Machines
TTPs	Tactics Techniques and Procedures
TON_IoT	Telemetry, Operating System, Network Internet of Things
TCP	Transport Control Protocol
TN	True Negative
TNR	True Negative Rate
TPR	True Positive Rate
TP	True Positive (or Recall)
WAAM	Wire Arc Additive Manufacturing
XGB	Extreme Gradient Boosting or XGBoost
XSS	Cross-Site Scripting
ZTA	Zero-Trust Architecture

Chapter 1. Introduction

1.1 Background

Over the past decade, Additive Manufacturing has emerged as a viable alternative to traditional manufacturing methods. 3D printing, the method by which users additively manufacture objects, adds value to both the public sector (U.S. Department of Defense & Office of Inspector General, 2021) and the private sector. Additive Manufacturing enhances operational efficiency by facilitating the prototyping of bespoke components and the creating of customizable products.

Commonly utilized methods of 3D printing include Fused-Filament Fabrication (FFF), Stereolithography (SLA), and Powder Bed Fusion (PBF). Less common methods of 3D printing include Jetting, Sheet Head Lamination, and Wire Arc manufacturing. FFF, the most widely implemented method of Additive Manufacturing, builds an object from the bottom up, by layering extruded plastic (Polylactic Acid (PLA) or Acrylonitrile Butadiene Styrene (ABS)). SLA, typically used for lightweight architectural molds and medical prosthetics, leverages resin-based materials to build a 3D-printed object; this method requires considerable pre- and post-processing. PBF utilizes granular metal powder and a laser to precisely melt the metal material resulting in the construction of an object. This method is of importance to the US government. The current practiced forms of Additive Manufacturing display versatility across different use cases.

Metal 3D printing has been a significant area of research and development across various industries. The aerospace and automotive industries benefit from metal printing to enhance rapid prototyping, material readiness, productivity, and development of end-state products. According to the Additive Manufacturing Industry report by Wohlers

Associates (2023), the sales for metal Additive Manufacturing parts increased by 27.2% between 2021 and 2022 likely due to an increase in available versatile materials and evolving use case requirements**.** Large industries are modifying their system practices to incorporate metal material 3D printing, which can withstand high impact and temperatures.

1.2 Research Motivation

Additive Manufacturing can accommodate a range of use cases, each having specific requirements. 3D desktop printers with USB connections are sufficient for hobbyist prints. While in industrial and operational settings, 3D printers may be hosted on a network allowing for wireless 3D printing and the integration of other emerging technologies, such as Artificial Intelligence and Automation. Since 3D printers are hosted on networks, they are vulnerable to network attacks, typically in the form of Internet of Things (IoT) attacks with specific attack vectors like man-in-the-middle attacks, Denial-of-Service Attacks, and malicious modification from unauthorized parties. IoT devices on a network require malicious attack detection capabilities while encouraging good security practices.

To safeguard against a selection of the attack vectors that can threaten 3D printers on a network, a machine learning detection can identify IoT attacks in conjunction with physical 3D printing tampering attacks, preserving all elements of the CIA triad. NIST SP 800-12r1 states: confidentiality protects privacy by restricting unauthorized entities; integrity guards against modification and destruction of a system or information; and availability ensures the ability to access data and systems. The absence of safeguards against any element may place the 3D printer, other devices, and the network at risk (Nieles et al., 2017). Authentication, Authenticity, and Non-Repudiation (AAN) augment

the CIA triad in assessing 3D printer security. Authentication for 3D printers ensures that the user initiating a 3D print process is genuine. Authenticity verifies that a 3D print file is genuine and comes from an authorized source. Non-repudiation ensures that a receiver of a file cannot deny receipt of a 3D print file. Certain aspects of the AAN components of the CIA triad are encompassed in the integrity and confidentiality elements, but they are not a focus of this praxis' model detection capability. Beyond the model, AAN is essential to evaluate and assess IT security controls, such as identity access management and digital signatures, which extend beyond a Machine Learning model.

Of the two-hundred-year history of traditional manufacturing, Additive Manufacturing has only been part of the last decade (Riikonen et al., 2020). As 3D printers advance and further integrate with IoT devices, they increase their risk of cyberattacks. Cybersecurity Manufacturing Innovation Institute (CyManII) was formed as a partnership between Government agencies, Universities, and private businesses to innovate and collaborate on issues involving Additive Manufacturing cybersecurity (CyManII, 2025).

The level of security applied to Networked 3D printers should follow the 'As Secure as Reasonably Practicable (ASaRP)' standard providing security against identified risks without hindering technological advancements in 3D printing. In the public and private sectors, unauthorized users may pose a threat to Intellectual Property (IP) by attempting to access its contents. The capability to transfer data wirelessly on a network increases the 3D printer's risks by specifically allowing file access, system availability, and product G-code file integrity to be open to malicious actors. As the public and private sectors continue to advance their usage of Additive Manufacturing and 3D printers, there

is a greater need for security to detect attacks on 3D printers and protect these systems.

Unauthorized access to, and exfiltration of proprietary Intellectual Property (system, stl, and/or G-code files), pose a risk to the competitive nature and reputation of an organization or agency. In addition to accessing valuable and sometimes classified data, attackers can tamper with and alter the original files to produce a different product or distort the original files, affecting the integrity of the Intellectual Property. The availability of 3D printers is at risk of denial-of-service attacks, whereby an attacker can flood the IP address running the 3D printer with pings or requests, significantly slowing it down or rendering it unusable.

1.3 Problem Statement

The increasing use of 3D printers connected to a Local Area Network (LAN) leaves file access, system availability, and product G-code file integrity vulnerable to malicious actors.

1.4 Thesis Statement

An Ensemble Machine Learning detection model based on IoT attacks is a valuable capability to safeguard 3D printers against a segment of targeted attacks, ultimately preserving confidentiality, integrity, and availability (CIA).

1.5 Research Objective

The objective of this praxis is to build a scalable ensemble machine learning model to detect attacks to a 3D printer's confidentiality, integrity, and availability (CIA) with a high true malicious detection rate.

1.6 Research Question & Hypothesis

Research Question (RQ): What components of an ensemble ML model created on supervised machine learning models and trained on the TON_IoT and SideChannel-3D data can accurately detect attacks to a 3D printer's confidentiality, integrity, and availability (CIA)?

- **Hypothesis Question (HQ)**: An Ensemble Stacking Model with a logistic regression meta-model will achieve a true malicious detection rate (recall) of 98.2% for a segment of attack vectors that can compromise a 3D printer's confidentiality, integrity, and availability.

1.7 Scope of Research

Through this praxis, a detector will be developed to identify attacks on 3D printers connected to a Local Area Network (LAN). Methodology in order:

(1) conduct a literature review to understand the attack vectors that are known to affect 3D printers' confidentiality, integrity, and availability

(2) identify and evaluate data on benign and malicious IoT traffic and G-code

(3) develop and train a detection capability on the chosen benign and malicious datasets for a segment of attack vectors that affect 3D printers

1.8 Research Limitations

This research limits the attack vectors to those provided in the chosen datasets corresponding to each element of the CIA triad. Confidentiality correlates to unauthorized access to data at rest and in transit. Based on the limitations of the datasets for this research, confidentiality attacks are limited to Scanning and Man-in-the-Middle (on-path) Attacks. Integrity attacks are limited to tampering with G-code in the form of

puncture attacks as adversarial inclusions. Availability attacks are limited to Denial of Service (DoS), that can render a device unusable.

No Industrial Internet of Things (IIoT) publically available dataset exists that is specific to 3D printers and on the scale of TON_IoT. An experimental physical testbed can create data that advances research into 3D printers and other Industry 4.0 devices. Although this is feasible, as methodologies exist to create, test, and validate the model, it is beyond the scope of this praxis.

The entire corpus of rows and columns between TON_IoT and SideChannel-3D exceeds the size limitations of what the desktop computer running Python and scikit-learn can process without cloud integration.

Chapter 2. Background & Relevant Research

This praxis applies ensemble stacking and voting models trained on labeled IoT network traffic data and G-code to detect malicious instances against 3D printers. The detection capability guards against a subset of cybersecurity attack vectors that compromise confidentiality, integrity, and availability. This praxis specifically examines physical attacks that compromise the structural integrity, coupled with malicious cyberattacks that impact the confidentiality and availability of 3D printing. A body of literature is dedicated to network attack detection towards IoT devices while another body of literature describes and detects tampering attacks on 3D printers.

The first body of research focuses on (1) identifying network attacks datasets that exhibit behavior of benign and malicious behavior onto IoT devices with unique IP addresses and (2) detecting malicious attacks leveraging machine-learning-based solutions. These datasets stem from packet capture (PCAP) data before being processed into to ZEEK logs. They usually display normal and malicious behavior, spanning network attacks, including but not limited to: Scanning, various types of Denial-of-Service (DoS) attacks, and Web-based attacks (like Backdoor, XSS, and Injection). Researchers have leveraged these datasets to detect both single cyberattacks and multiple cyberattacks. One example, leveraging the CICIDS-2017 dataset, identifies multiple cyber-attacks (Darweesh et al., 2024) while another source uses the CICDDoS2019 dataset to investigate Distributed Denial-of-Service (DDoS) attacks exclusively (Xie, 2023).

The second body of research focuses on detecting tampered 3D printed objects. Adversaries can compromise the structural integrity of 3D printed objects by tampering

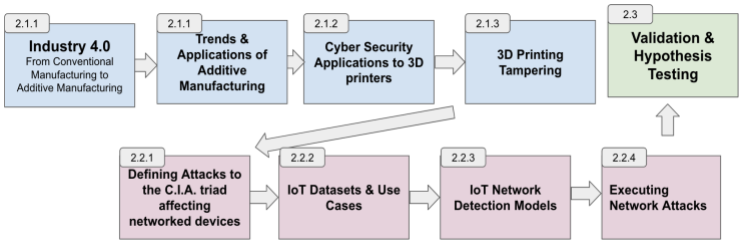
with the G-code instructions used by 3D printers. These attacks can happen when files are at rest or when data is being transferred. Tampering with G-code files can cause disruptions in the supply chain, this action can also lead to performance problems within systems reliant on Additive Manufacturing (Yu et al., 2020). G-code files contain information that instructs 3D printing, such as the minimum temperatures of the build plate and extrusion nozzle, coordinates by which the extruder is to travel (x, y, z coordinates), and the level of extrusion.

The security of networked 3D printers and its integrated components (data, and devices) is placed at risk by both network intrusions and malicious physical tampering. Despite an abundance of research published in each of the two mutually exclusive research silos, there is a dearth of research at the intersection of them safeguarding all of confidentiality, integrity, and availability of networked 3D printers within one detection capability.

This praxis contributes to the aforementioned research communities by developing an ensemble model that safeguards the three elements of the CIA triad simultaneously. By creating an ensemble model, each component of the CIA triad is safeguarded without sacrificing the safety of the other elements.

Figure 1

Chapter 2 Roadmap



2.1.1 General Trends in Additive Manufacturing

Additive Manufacturing is a key technological advancement within Industry 4.0, also referred to as the Fourth Industrial Revolution, or "Smart Industrial Technologies" Revolution. Additional technological advancement of Industry 4.0 include Artificial Intelligence (AI), Internet of Things IoT), Automation, and big data (Ishfaq et al., 2023). Prior to Industry 4.0, the predominant manufacturing approach was subtractive manufacturing, relying on the removal of material through machining techniques such as milling, cutting, grinding, and turning. Starting in the late 1700s with steam-powered manufacturing, the process evolved to the assembly line in the early 1900s, and in the 21st century, advancements in computer-aided design (CAD) have occurred. Each industrial revolution offers more versatility and efficiency compared to the previous (Riikonen et al., 2020). Additive Manufacturing constructs objects layer by layer, allowing for customization, material efficiency, and reproducibility providing a viable alternative to a diver application of manufacturing use cases, especially those reliant on constructing geometrically intricate objects at scale

Additive Manufacturing is continuing to evolve, leading to even greater integration with Industry 4.0 advancements in the future, most notably the Internet of Things (IoT), Automation, and Artificial Intelligence/Machine Learning (AI/ML). Advancements in the interconnectivity of 3D printers with IoT, and Cyber-Physical Systems (Tamir et al., 2023) can increase machine performance, output, and monitoring. This interconnectivity can also lead to risks in a connected environment.

Additive Manufacturing has applications across a broad spectrum of use cases and industries, including but not limited to aerospace, automotive, healthcare, and consumer

goods. The aerospace industry uses Additive Manufacturing to engineer complex, high-performance components that are able to withstand extreme temperatures and create intricate geometries previously unattainable by conventional manufacturing processes. Similarly, the automotive industry uses Additive Manufacturing to produce bespoke spare parts on an as-needed, ad hoc basis (Wohlers Associates, 2023). The healthcare sector leverages on Additive Manufacturing to create customized implants and prosthetics. The consumer goods market leverages Additive Manufacturing for personalized and customizable items. Despite the broad range of use cases, Additive Manufacturing augments rather than replaces, as it is not a fully automated process and still requires pre- and post-processing (Tamir et al., 2023).

The construction industry has started to implement Additive Manufacturing, primarily through the Wire Arc Additive Manufacturing (WAAM) process. This approach uses a metal wire and a welding torch to melt and deposit the wire, layer by layer, to construct a 3D printed object. Two examples of WAAM are the dragon bench and the metal pedestrian footbridge (Shah, Aliyev et al., 2023). The use of Additive Manufacturing in the construction industry is expected to expand by 101.9% between 2023 and 2030 (Horizon Grand View Research, 2023).

Material usage is dependent on the use case and the desired properties of the 3D-printed object. The materials used for 3D printing include resins, thermoplastics, metals, and ceramics. Shape memory materials enable 3D printing to change over time (Srivastava et al., 2022).

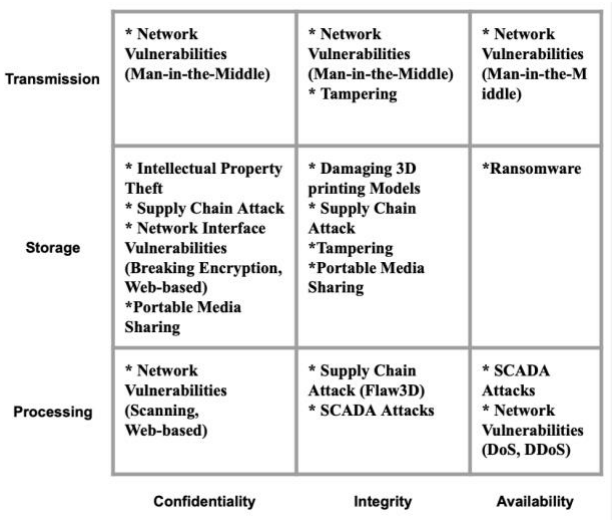
2.1.2 Cyber Security & 3D Printing

Networked 3D printers offer the same benefits as other IoT devices – but also pose additional risks. IoT devices enhance the functionality and interoperability of devices within a network by enabling them to interact with one another natively or via enhancements like web-based monitoring. Two examples of IoT devices include wireless printing and smart laundry machines.

The 3D printing process comprises multiple phases, starting with the CAD design phase and concluding with the post-processing stage. Each stage of the 3D printing process consists of unique risks and avenues where an adversary can exploit. The general pipeline for 3D printing is described herewith, in conjunction with a broad selection of attack vectors that could detrimentally affect a 3D printer at each phase (Šlajs et al., 2023; McCormack et al., 2020; McCormack et al., 2021).:

Figure 2

Attack Vector Schematic



(1) Designing a 3D Printing Model: Computer-Aided Design (CAD) files design the

structure of the object to be 3D printed. These files support types .stl or .obj. CAD tools allow users to design bespoke parts while customizing measurements and structural choices (Šlajs et al., 2023). Standard repositories, such as printables, contain existing CAD files for immediate use or as an initial starting point for bespoke modifications. Unique mission requirements and files associated with a 3D printed object are intellectual property (IP) and critical information. A malicious attacker could exploit these and consequently hurt the business process.

- **Intellectual Property Theft**: Intellectual Property Theft is a risk to company operations, where an adversary may steal the stored assets (databases, data files, and model design prototypes) and either release them to the market or expose them to a competitor (Šlajs et al., 2023). Disclosure of this information could result in financial loss and breach confidentiality, as unauthorized users now have access to proprietary and potentially sensitive data.
- **Damaging 3D printing Models**: Stored CAD design models can become corrupted, resulting in the incomplete or incorrect creation of the set of instructions that guides the 3D print (G-code) (Šlajs et al.,2023). Damaging 3D printing models can compromise the structural integrity of the object, making it time-consuming and expensive to repair.
- **Malware**: Malicious code can spread by sharing files and downloading software containing potentially malicious add-ons and plugins. Malicious code can be hidden from the user executing the 3D printing (Šlajs et al., 2023).

- **Network Interface Vulnerabilities:**
    - **Breaking Encryption**: Adversaries put the confidentiality of a system at risk by breaking encryption and viewing sensitive files stored on a device.
    - **Web-based:** Attacks like backdoors can generate an unauthorized tunnel for adversaries to exfiltrate files from a system.
- **Ransomware:** Files stored on a system can be maliciously encrypted by an adversary, hindering availability as they become inaccessible.

(2) Slicing and 3D print Preparation: after the 3D printable design file is created or gathered from an outside source, the files are then sliced into layers and transformed into Geometric-Code (G-code) that instructs a 3D printer's nozzle extruder and print bed to build a structure from the bottom up.

- **Tampering**: Every device or individual that interacts with files during the pre-printing stage (Design and Slicing) has the potential to compromise the system. Malicious attackers can damage STL files prior to slicing. It is challenging to distinguish between genuinely malicious behavior, anomalies, and normal expected behavior (Šlajs et al., 2023). With G-code tampering, however, it is a more common approach to attack the sliced instruction file, thereby directly affecting the 3D-printed object. Insider threats are an example of potential attack vectors related to tampering; however, infiltrators who are already within the network may also pose risks of tampering. Tampering affects a 3D printer's integrity while

targeting G-code stored on a device hosting the slicing software or in transit from the slicing software to the printer.

- **3D Printing Supply Chain Attacks**: Like malware infecting CAD files, individuals can download unsafe slicing software or tampered files from online sources, which can automatically embed itself in the 3D printing process (Šlajs et al., 2023). Supply chain attacks at this stage compromise the integrity and confidentiality of the 3D print. Downloading a faulty slicing software or a tampered file may result in the lack of necessary security safeguards, leaving it vulnerable to confidentiality attacks or unauthorized breaches; tampered G-code can compromise the integrity of the print.

(3) G-code data is transferred to the 3D printer: Data is transferred from the device to the 3D printer via the following media: a USB cable, a Removable storage device like a USB drive or SD card, and Wirelessly through an internet connection (Šlajs et al., 2023).

- **Network Vulnerabilities (Man-in-the-Middle Attacks):** After a device establishes a wireless connection to a 3D printer, an adversary can plant themselves between the sender and the receiver of a communication channel. This intrusion enables unauthorized individuals to intercept the transmitted data (Šlajs et al., 2023), thereby compromising the confidentiality of data in transit. Files transferred to a 3D printer may be stolen/altered, and replaced with maliciously encrypted data by an adversary. This action hinders integrity and availability, making the 3D printer inaccessible.

- **Portable Media Sharing:** Portable media sharing devices, such as USB drives or SD cards, can connect to multiple different media sources, increasing the risk of malware spreading to 3D printers (Šlajs et al., 2023). The confidentiality and integrity of the data are at risk when the transmitted data is stored and downloaded.

(4) 3D Printing: Attacks that could hinder performance of a 3D print during the print are:

- **Supply Chain Attacks** - Firmware attacks, such as FLAW3D, alongside Material Malfunctions, can reduce the structural integrity of the 3D printed object. The Flaw3D Trojan horse bootloader affects 3D printers during the print.
- **SCADA Attacks** - Manufacturing interruptions due to power disruptions pose a risk to the integrity of active 3D print jobs by causing incorrect extrusion of filament. These attacks compromise availability, rendering the printer unusable.
- **Network Vulnerabilities (further expanded)**:
    - **Scanning:** Placing confidentiality at risk by seeking active printers, open ports, and device metadata. Scanning attacks can lead to further attacks on 3D printers.
    - **Web-based:** Backdoor vulnerabilities and injections can affect the confidentiality of the system by creating a potential pathway for adversaries to exfiltrate data or manipulate 3D prints. These attacks can lead to further C2 (Command and Control) attacks.

    - **DoS/DDoS:** Denial-of-Service attacks inundate an active 3D print job by flooding it with requests, slowing down or stopping performance.
- **Weak Credentials** - Users should avoid using default credentials and easy-to-brute-force passwords whenever possible (Grassi et al.,2017).

(5) Post-Processing: Following the completion of the 3D printing process, the object is removed from the build plate and sanded, cut, and cleaned to model the ideal properties of the desired object (Šlajs et al, 2023). Since this is a manual and customized process for the manufacturers, it is difficult to share the attack vectors for this component of 3D printing.

In addition to the attack vectors presented in "Cyber-Attack Vectors Focused on Additive Manufacturing" (Šlajs et al., 2023), the Connected 3D Printer Observer (C3PO) research effort, initiated by Carnegie Mellon University, studies and executes possible attacks on networked 3D printers. The presence of other IoT devices on the same network makes 3D printers increasingly vulnerable to cyber risks. C3PO analyzed 8 commercially available 3D printers whose vulnerabilities spanned four categories (McCormack et al., 2020; McCormack et al., 2021). The specific observations and attack vectors are in Table 1.

Table 1

C3PO Vulnerabilities Category, Observations, and Attack Vectors (McCormack et al., 2020; McCormack et al., 2021)

Vulnerabilities	Possible Attack Vector Description
Data Transfer **"8** 3D printers did not use authentication or encryption during data transfer"	**MitM**: "Attackers [may] spoof a networked 3D printer and create a man in the middle situation between a PC and the 3D printer."
Network Services "**6** 3D printers had [exposed] network services such as unused open ports and public IP addresses" **"3** 3D printers surveyed had network services vulnerable to known exploits"	**Scanning**: "specifically, open network ports and any services they were running were identified. This was performed by scanning all possible TCP ports and the 100 most common UDP ports on the 3D printer." **Backdoor**: "These unused open ports could be indicators of insecure design or a compromised machine (e.g., attacker installed backdoor)"
3D printer Application "**3** 3D printers had insecure 3D printer applications"	**Malicious Firmware**: "a network capture of this operation was collected and analyzed …to determine the potential for an attacker to send malicious firmware"
Availability "**8** 3D printers were vulnerable to [DoS] attacks"	**DoS**: "Our tool performs the following basic DoS attacks: SYN flood … and TCP connection exhaustion."

2.1.3 3D Printing Tampering

3D Printing Tampering

Tampering, within the context of networked 3D printers, implies that an adversary can remotely or directly access files or data for modification. Three types of tampering are explored in this section: FLAW3D, extrusion modifications, and sabotage attacks.

FLAW3D is a Trojan-horse bootloader which can alter the firmware at the system level, allowing malicious actors to relocate and reduce material extrusion at specific segments of the build process (Pearce et al., 2022). This can affect the integrity of the 3D printed object.

Additional research administers malicious integrity tampering attacks (red team) and uses statistical methods to detect them (blue team). The vulnerabilities introduced were the following: (1) Converting every 4th G1 command to G0 (2) Convert every 4th G1 command to G0 and add blob extrusion (3) Reduce global extrusion by 50% (4) Convert every fourth G1 command extrusion value set to previous extrusion (5) Every 4th G1 Command has extrusion value set to previous + 0.0001 (6) Delete every 4th G1 line with no replacement (Beckwith et al., 2021).

Previous research has used acoustic, vibrational, and various other forms of side-channels to detect sabotage attacks to 3D printer's integrity (Al Faruque et al., 2016; Yu et al., 2020**)**. The SideChannel-3D dataset includes an in-depth approach to tampering with a normal collection of G-code (Costa et al.,2021), this dataset is further expounded in Chapter 3 of this praxis.

2.2.1 Safeguarding CIA

Confidentiality, integrity and availability (CIA) are three foundational pillars of cybersecurity (Nieles et al., 2017). Attacks and vulnerabilities can be mapped to one or more of these elements of the triad. As adversaries utilize more sophisticated attacks and complex infiltration and persistence methods, it becomes difficult to map attacks to one specific element of the CIA triad. This highlights the need to build a detector to safeguard all elements of the triad, regardless of the ambiguity.

Defining Confidentiality and Scope:

Breaches to confidentiality include unauthorized access or an 'interception' of data in transit (Singh et al., 2011). Attacks to confidentiality include the deployment of malicious software (malware), social engineering tactics (phishing), and large-scale data breaches. Confidentiality attacks target sensitive assets such as intellectual property, financial records, and proprietary customer data (Mitchell & Osazuwa, 2023). Attacks to confidentiality are identified as an unauthorized person or persons viewing the status of a 3D printer or data passed through it.

Blockchain aids in Additive Manufacturing security by safeguarding confidentiality and integrity while ensuring transparency. The decentralized nature of blockchain systems facilitates secure device communication within a "shared manufacturing" environment, where multiple entities work together to exchange digital resources with no centralized ownership. This can help ensure resiliency in the event of unauthorized interference or manipulation. Additionally, blockchain employs smart contracts that enable advanced encryption and authentication when customers interact with providers (Lupi et al., 2023).

Zero-trust architecture (ZTA) operates on the core principle that there are no implicit trust or default security measures placed on the users, applications, environments, or systems attempting to access services. Traditional perimeter-based security architecture implies that all elements within a corporate network are protected by a singular defensive perimeter (Paul & Rao, 2022). Technological advancements from Industry 4.0, like IoT, benefit from the safeguards in Zero-Trust-Architecture Frameworks specifically, attribute-based access controls and granular microsegmentation. This ensures that access is strictly confined to only those users and devices that are authorized and required - in accordance with the principle of least privilege.

This praxis does not focus on Blockchain or Zero-trust architecture (ZTA) but includes them as examples of technologies used to safeguard against risks to confidentiality and unauthorized access. The scope of this praxis is limited to confidentiality attacks that can be investigated from network flow data. Attack vectors proposed as threats to confidentiality include Scanning, Man-in-the Middle, and Backdoor attacks.

- Scanning: Malicious actors scan a victim's network during the reconnaissance phase of an attack to reveal devices names/IP addresses, operating systems, active services, and additional device information that could be compromised. The unauthorized scanning of these devices can lead to exfiltration, exploitation, command & control jeopardizing the confidentiality of the device and network (Booij et al., 2022).

- Man-in-the-Middle: Malicious actors intercept communication between endpoints stealing, rerouting, and accessing potentially sensitive information. When Man-in-the-Middle (MitM) attacks occur, the confidentiality of the communication is at risk and if data is modified in transit, the integrity is also at risk. Man-in-the-Middle attacks contain different methods of execution (Nayak & Samaddar, 2010), two of the most common include:
    - ARP Spoofing: one common method of executing a man-in-the-middle attack whereby a message intended for the target device gets redirected because the attacker uses the MAC address of the victim computer.
    - DNS Spoofing: an attacker may spoof a DNS and reroute a victim to a malicious site.
- Backdoor Attacks: Malicious Actors can embed malicious code with firmware, establishing a remote connection from a victim's Machine to an Adversary's system. Backdoor attacks may go undetected, particularly within environments comprising IoT devices (Hashemi & Zarei, 2021). The confidentiality of the system is compromised as backdoors create an unauthorized channel of communication, which allow for the exfiltration of data and remote control of devices. One example of a backdoor attack is the SolarWinds incident, where a compromised software update to the SolarWinds Orion platform affected over 30,000 organizations, creating an unauthorized channel for adversaries to infiltrate these networks (Oladimeji & Kerner, 2023).

***Defining Integrity and Scope**:*

By "modifying" or tampering with data files at rest and in transit the integrity of the 3D printer and its data (Singh et al., 2011) are put at risk. Viruses, Trojan horses, and ransomware can compromise confidentiality, integrity, and in the case of ransomware availability. Whereas FLAW3D is a threat to integrity, G-code watermarking strengthens integrity. The FLAW3D Trojan horse bootloader is an example of an attack on integrity, where materials can be removed or replaced during the build process. G-code watermarking safeguards against malicious tampering or malicious modification, that ensures sections of G-code are free from alterations.

Flaw3D Bootloader: Flaw3D is a Trojan Horse bootloader that initiate when the Marlin firmware-based 3D printer is powered on. The Trojan horse bootloader hooks onto system processes executing tampering attacks. Two examples of its malicious effects include material relocation and reduction. Although this praxis research does not detect FLAW3D, this use case highlights one example of how 3D-printed objects can be tampered. Figure 3 reduces the infill of a 3D print by 50%; Figure 4 takes three consecutive sets of instructions and iteratively increases by 1mm. If material extrusion is skipped, the following extrusion command should be repeated (Pearce et al., 2022).

Figure 3

*Material Removal (*Pearce et al., 2022*)*

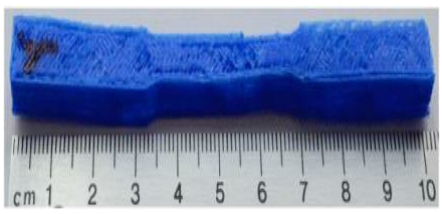
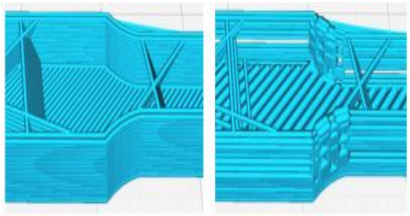


Figure 4

*Material Relocation (*Pearce et al., 2022*)*



Watermarking: G-code files, the instruction files leveraged in an FFF 3D printer, can be chunked and assigned a random mapping. Each mapping contains a unique segment of G-code, including a determination of whether and where a malicious modification has occurred (Li et al., 2021). While confidentiality safeguards against the unauthorized interception of data both at rest and in transit, integrity examines malicious modifications that could impact the structure and usability of a 3D printed object.

Aside from Watermarking and the Flaw3D Bootloader, malicious tampering of any kind can compromise the efficacy, durability, and structural integrity of a 3D printed object. The praxis research assumes that an adversary has obtained access to the network prior to executing the tampering attack, which occurs at rest. The remainder of this praxis will leverage the SideChannel-3D dataset, which detects attack to segmented G-code.

Defining Availability and Scope*:*

Attacks on the availability of a device hinder regular access to data or an environment, in the form of an "interruption" (Singh et al., 2011). Denial of Service (DoS) and Distributed Denial of Service (DDoS) attacks threaten availability, which target corporations and government systems. This praxis research explores DoS attacks in 3D printers. DoS attacks flood a device or system with illegitimate requests, shutting down or slowing down traffic and communication. A DDoS attack amplifies DoS attacks by utilizing numerous machines to execute attacks against the same victim machine.

There are multiple datasets available to the public that show network traffic to and from IP addresses that contain availability attacks executed (i.e. DoS and DDoS). The CICDDoS2019 dataset provides a large set of DDoS attack events (University of New Brunswick, 2019b; Sharafaldin et al., 2019). SYN Flood, UDP Flood, and TCP Flood are

three types of DDoS attacks mentioned in the CICDDoS2019 dataset; however, Malicious actors can execute DoS attacks in a myriad of other ways. Naive Bayes, Random Forest, Support Vector Machine, and XGBoost models attempt to detect DDoS attacks, with the Random Forest model performing the best in terms of Accuracy, Precision, and Recall.

Unlike malicious tampering, which affects integrity, attacks on confidentiality and availability comprise vectors often found in corpora of network traffic data. This praxis seeks to train and evaluate a model based on both.

2.2.2 Network Intrusion and Detection Datasets

Internet of Things (IoT) devices are found on commercial, government, and home networks and are increasingly vulnerable to the evolving landscape of cyber threats. Consequently, networked 3D printers are subjected to the same vulnerabilities, with the potential for destruction due to data sensitivity and physical tampering. To mitigate and detect vulnerabilities in IoT devices and, in turn, 3D printers, security measures such as regularly patching systems, adhering to industry best practices, and deploying advanced detection methods can be converted into Intrusion Detection System (IDS) rules.

Supervised, unsupervised, and deep learning is widely researched to detect vulnerabilities in IoT devices using a diverse range of approaches. In one example, a Deep Learning Neural Network (DLNN) detection mechanism for DoS and Brute Force attacks was trained on the NWSB-15 dataset (Kandhro et al., 2023). In addition, supervised machine learning detection solutions have been previously explored.

KDD-99: Network intrusion dataset included as the 1998 DARPA Benchmark as part of the *Third International Knowledge Discovery and Data Mining Tools Competition*

(Devarakonda et al., 2022) contains 41 features, primarily: duration, source bytes, destination bytes, number of failed login events, error rates, protocol used, and service used. This dataset originates from Zeek log data, which is then subsequently formatted into CSV files. Each row was assigned a label indicating whether the network traffic was normal or malicious. The dataset comprises a training set and a testing set, each containing 5.2 million instances of network traffic. KDD-99 includes attack vectors such as Denial-of-Service, Probing, User-to-Root, and Remote-to-Local. Denial-of-Service (DoS) attacks are the most prevalent attack vector in the dataset, with 7 distinct types of DoS attacks present in the data. Probing attacks are the second most frequent category in the dataset, involving scanning attacks conducted with various tools.

NSL-KDD: This dataset is the successor to the KDD-99 dataset. Researchers Tavallee et al. (2009) sought to improve the KDD-99 dataset by de-duplicating and rebalancing the malicious events within each attack (Devarakonda et al., 2022). This dataset contains approximately 148,000 rows, which is significantly smaller than the KDD-99 dataset. The features and the attacks included in this source are the same as in the KDD-99 dataset.

UNSW-NB15: In 2015, researchers at the University of New South Wales generated the UNSW-NB15 dataset, which includes both normal and malicious data traffic (Moustafa & Slay, 2015). Researchers utilized the PerfectStorm tool to generate a network traffic dataset comprising benign events and malicious attacks for nine attack families. PerfectStorm is a tool created by IXIA that simulates attacks from a CVE site. This tool generated network traffic over a network for a total of 16 hours. The output files contain Argus and Bro-IDS features, leveraging 47 of them in the context of network

traffic. Overarching feature categories include: Flow Features (source and destination IP addresses, protocols, and port numbers), Basic Features (time and duration of bytes sent and received, packets lost, duration of flow), Content Features (flow packet sizes, and source and destination sequence number for TCP connections), Time Features, custom generated Features, and Labeled features (attack types, benign/malicious). This dataset also contains 9 attack types rather than the 4 presented in KDD-99. Additional attack vectors include backdoors, where attackers exploit vulnerabilities to access a device; fuzzers, which allow an adversary to disrupt a network by feeding it random data; generic attacks; and worms (Moustafa & Slay, 2015).

CIC - A repository of datasets by the Canadian Institute of Cybersecurity (CIC), each dataset includes attacks with realistic background traffic. CIC provides IoT, Intrusion and Detection System (IDS), Malware, and Ground truth datasets, among others. Data is collected through packet capture for the network traffic, formatted in a CSV file with timestamps, IP addresses, ports, protocols, and attack labels. CICIDS-2017, one of the most popular datasets in the CICIDS repository, continuously captures 5 days' worth of network traffic labeled by attack type and containing more than 80 features (University of New Brunswick, 2019a). Attack vector categories included in this dataset are: Brute Force, (Distributed-)Denial-of-Service Attacks, Heartbleed, Web Attacks, Infiltrations, and Botnets. The 80 features include: number of packets sent forward/backward, flow bytes, flag counts in both directions, and time-based features. There are shortfalls in the dataset, such as misclassification of labels and packet misorder (Dube, 2022; Lanvin et al., 2022). Notwithstanding, it is still widely used due to its features and various attack types.

IoT network intrusion dataset - An IoT dataset created by the Hacking and Countermeasure Research Lab in 2019. Researchers placed IoT devices, specifically SKT NUGU and EVIZ Wi-Fi camera, as targets of cyberattacks, along with other network devices on the same wireless network (Kang et al., 2019). This dataset comprises 42 datasets, each corresponding to a specific point in time and a particular attack category and sub-category. Among the most prevalent attacks executed on the network were Normal data, Man-in-the-Middle attacks, Denial-of-Service attacks, Scanning attacks, Mirai Botnet Flooding, and Mirai Botnet Brute Force attacks. Although the dataset is raw packet capture data and does not run the risk of incorrectly generating features, attack files are separated based on the type of attack conducted. However, the dataset does not label normal or attack instances, nor does it contain the ground truth necessary for classification.

Aposemat IoT-23 - As a widely used dataset created in 2018 and 2019, it comprises 20 attack scenarios related to the precise execution of malware, alongside 3 benign scenarios (Garcia et al., 2020). The Aposemat Project of the Avast-AIC Laboratory captured network traffic from a variety of different devices. Each attack scenario ran for almost 24 hours, generating PCAP data subsequently converted into ZEEK data. IoT devices, such as Amazon Echo, Somfy Smart Door Lock, and Philips Hue Device, were present on the network. Due to the size of the dataset, it is too large to leverage for detection and model development while still accounting for all its attributes.

TON_IoT - The TON_IoT network intrusion dataset contains a collection of raw PCAP data files from 10 different attacks. Comparatively, the Aposemat IoT-23 also contains features for 10 attack types; however, TON_IoT offers a richer feature set with

22 million log entries. Researchers tested the Gradient Boosting Classifier (GBC) model, the Neural Network (NN) deep learning model, and the Random Forest (RF) supervised machine learning model. IoT-23 revealed the better model evaluation metrics, but TON_IoT has a richer dataset with strengths in its heterogeneous nature. The TON_IoT dataset contains Operating System (OS), Network, raw PCAP, and labeled and processed ZEEK CSV data for 10 different types of attacks (Booij et al., 2022).

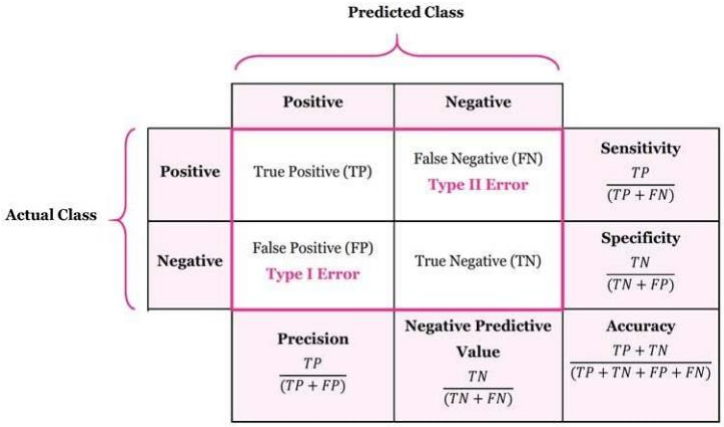
2.2.3: Network Intrusion and Detection Modeling

Once an appropriate dataset has been selected, a machine learning detector may be created to protect devices against one or multiple attack vectors. Table 2 identifies key research detecting malicious instances through network traffic datasets.

Table 2 also synthesizes models by including the following: attack vectors detected, dataset(s) used, preprocessing techniques employed, and metrics associated with the evaluation of the best-performing model. Confusion Matrices identify True Positives, True Negatives, False Positives, and False Negatives. Figure 5 is an example of a confusion matrix model evaluation.

Figure 5

Example Confusion Matrix (Encord, 2025)



Within the context of evaluating a detection model the following terms can be elaborated:

- *"Positive" & "Negative"*: Malicious Instances & Normal Instances Respectively
- *"Predicted Class"*: Model Identified Label
- *"False Positive"*: Incorrectly identifying a Normal Instance as Malicious
- *"False Negative"*: Incorrectly identifying a Malicious Instance as Normal
- *"Sensitivity" = "Recall" = "True Positive Rate"*

Table 2

Network Intrusion and Detection Modeling

Attack Vectors	Dataset	Preprocessing	Best Model %	
DDoS	CICDDoS2019	*Random selection of each sub-category *ExtraTreeClassifier, was used to select the best 20 features	RF (99.32%)	(Xie, 2023)
Smurf Attacks, Port scans, ARP Spoofing, DoS Attacks, Backdoor, Ransomware, Injection, Cross-Site Scripting, MiTM	ECU-IoHT, TON_IoT, Edge_IIoTset	*Data Concatenation, Bert Tokenization *Categorical Variable *Encoding	LightGBM (AUC 99%)	(Ghourabi, 2022)
DoS Analysis Exploits Generic	NSL-KDD, KDDCup99 UNSW-NB15	*Data normalization *Zscore scaling	* Most performed well for certain attacks	(Kandhro et al., 2023)
Backdoor, DDoS, DoS, MitM, Password, Ransomware, Scanning,XSS Injection	TON_IoT Aposemat IoT-23	*Down Sampling Attack Groups *Normalizing via Z Scores *Kolmogorov Smirnoff revealing what models to test to limit noise and false alarm rate	RF AUC 99.63 TON 99.9 Aposemat	(Booij et al., 2022)
sqlattack, smurf, snmpgetattack Spy, httptunnel	KDD-99 NSL-KDD	*labeling *Zscore normalization *onehotencoding	RF (Recall 99.7%)	(Devarakonda et al., 2022)
DDoS	CSE-CIC-2017	*Removal of not useful, irrelevant, and highly correlated features	RF (96.65%)	(Bindra & Sood, 2020)

DDoS, DoS, Brute Force, Botnet, Infiltration, Web Attack	CIC-IDS-2018	*Feature selection for top 20	GBC (99%)	(Leevy et al., 2020)
Backdoor, DDoS, DoS, MitM, Password, Ransomware, Scanning,XSS Injection	TON_IoT	* Label encoding * Min-Max normalization * Feature Selection	Stacking (Base: DT, RF, KNN, LR, Meta: LR) (98.63%)	(Alotaibi & Ilyas, 2023)

** **Note**: Area Under the Curve (AUC) for ROC curves

Non-Ensemble Supervised Machine Learning Models

Logistic Regression: Multiple Linear Regression calculates a y-predicted value from the summation of all Beta values and X inputs. Beta values explain the strength of the coefficient as a trend. A function in k dimensions, where k is the number of features in the data, generates a line of best fit by minimizing the sum of squares. Logistic Regression relies on the sigmoid function to perform binary classification, utilizing the principles of Linear Regression. This model can be used to identify benign and malicious network event logs (Leevy et al., 2020; Alotaibi & Ilyas, 2023). Logistic Regression is easily interpretable and has a smaller model size (Bold et al., 2022).

Linear Regression Model *Logistic Regression Model*

$$Y = B_0 + B_1X_1 + B_2X_2 \ldots . + B_kX_k \qquad p = \frac{e^{B_0+B_1X_1+B_2X_2+\ldots+B_kX_k}}{1+e^{B_0+B_1X_1+B_2X_2+\ldots+B_kX_k}}$$

Decision Trees: A Supervised Machine Learning model that uses tree-based logic, which applies a condition to evaluate each child node, maximizing information gain to determine whether an input is benign or malicious. It is then repeated, ultimately providing a tree determining labels based on conditions that maximize information gain (Bold et al.,2022). One of the key advantages of this approach is its interpretability,

offering clear insights into the decision-making process.

Random Forest: An ensemble model that leverages a voting scheme on the output of K number of decision trees and chooses the most common class of an input vector (Breiman, 2001). Random Forest models are not as susceptible to over-fitting compared to Decision Trees.

Random Forest Model

$RF(x) = Mode(Tree1(x)\ \ Tree2(x)\ \ \ldots\ \ TreeK(x))$

x = feature vector (x1, x2, … xK)

Gradient Boosting Classifier: An ensemble model where each tree is iteratively trained to minimize the errors of the previous iteration, minimizing a loss function, the difference between predicted and actual values (He et al., 2019). This method enables the model to predict edge cases, often resulting in higher predictive accuracy compared to other ensemble methods.

Support Vector Machine: In n-dimensional space, the support vector machine creates a boundary hyperplane that separates data into classes. The model will perform as well as the boundary separates benign and malicious correctly. Support vectors are data points closest to each boundary, where the goal is to widen the margin between the boundary and support vectors (Bold et al., 2022).

Neural Network: Widely studied models that enhance Regression by leveraging input layers, hidden layers, and output layers, with each output acting as the input for the next layer. Activation functions are also applied (Bold et al., 2022).

Voting: A series of base classifiers individually model the data to create an ensemble classifier. Each model assigns a label to each input vector. The label with the majority

vote is assigned to the input (Bold et al., 2022).

Stacking: A series of chosen base classifiers makes individual predictions that are used as inputs for a meta model. The results of the meta classifier will be the final prediction model (Bold et al., 2022).

2.2.4 Network Attack Execution

Execution of Network Attacks in TON_IoT

The TON_IoT dataset was ultimately selected as the network intrusion data source for this praxis. TON_IoT contains useful attacks to the confidentiality and availability of IoT devices. TON_IoT provides raw PCAP data from normal and malicious network traffic and processed ZEEK data in the form of CSV labeled attack instances. The dataset contains 22 million log entries pertaining to both benign and malicious events. Housed in the dataset are events from 9 different attack types. Descriptions have been provided for each attack in a way that is replicable (Alsaedi et al., 2020).

- *Scanning Attack*—Nessus, a vulnerability scanning tool and the Kali Linux Nmap tools were executed against the target and all other public vulnerable systems. This could be useful to the construction of attacks to execute against the 3D printer's confidentiality.
- *Denial-of-Service (DoS) Attack*—Python scripts using Scapy were utilized alongside IoT Devices to administer TCP Flood DoS attacks.
- *DDoS Attack*—Distributed Denial of Service attacks leverage multiple devices to overwhelm the network with traffic. The author of the paper only mentions that TON_IoT leveraged Python scripts with ufonet.

- *Ransomware Attack*—Ransomware attacks were executed against targets running Windows OS that are monitoring IoT services, by means of Metasploit exploiting server message block vulnerabilities.
- *Backdoor Attack*—The researchers execute backdoor attacks by Metasploit executing a bash script with the command "runpersistence-h."
- *Injection Attack*—Adversarial devices inject data and launch attacks like (SQL injection, client-side injection, broken authentication and data management, and unintended data leakage) to Web applications of DVWA and Security Shepherd VMs, and to webpages of other IoT services.
- *Cross-Site Scripting (XSS) Attack*—XSS code was injected to the same machines and services as targeted with injection attacks, XSSer toolkit in python leveraged malicious bash scripts to execute.
- *Password Attack*—Kali Linux Hydra and Cewl toolkits were used. They were configured by means of automated bash script to concurrently launch password hacking attacks against any vulnerable nodes in the testbed.
- *Man-in-the-Middle (MITM) Attack*— Creators of the dataset leveraged ARP spoofing, ICMP redirection, port stealing, and DHCP spoofing attacks. The primary tool to execute these was Kali Linux Ettercap.

Among these types of attacks — Scanning attacks, Backdoor attacks, XSS, Password and Man in the Middle attacks are most representative of attacks to confidentiality of 3D printers. Denial-of-Service Attacks, Distributed Denial of Service attacks, and Ransomware are attacks to availability. Both attack families are combined with the vast number of benign events.

Behavioral signatures, once identified, can validate successful execution of attacks and be employed as checkpoints to assess both the performance and interpretability of detection models. Table 3 describes the various attack types in conjunction with the corresponding behavioral patterns that are expected to be performed.

Table 3

Simplified Behavioral Signatures based on TON_IoT

Attack Vector	Behavioral Patterns
Man-in-the-Middle	Bytes sent to an unknown IP address or Bytes sent to a spoofed IP address in an attempt to trick the network
Scanning	A small number of packets being sent but many ports being accessed in a short amount of time
Brute Force	A very high number of failed attempts in a very short amount of time
Denial-of-Service (DoS)	A very high number of bytes sent/received
Web Attacks (XSS)	Bytes/Packets targeting web ports like 80, 8080, 43, 8000
Ransomware	Large number of Bytes sent to an unknown IP address

Simplified behavior signatures identified in Table 3 invert the attack profiles and execution procedures into recognizable behavioral signatures (Booij et al., 2022). These signatures, in turn, embed in advanced detection mechanisms and heuristic-driven pattern matching. MitM and ransomware attacks, for instance, are typically identified through masking or spoofing, sometimes accompanied by anomalous or unknown IP addresses. A high volume of bytes or packet flow characterizes DoS attacks. Scanning attacks exhibit an abnormally high number of port attempts, whereas a rapid succession of failed authentication attempts within a short timeframe characterizes brute force attacks. Web-based attacks, although diverse in their implementations, generally involve the

exploitation of specific web ports, including ports 80, 8080, 43, and 8000.

2.3 Hypothesis Testing in Attack Detection

Reduction of Type 2 Errors

False Positives (Type 1 errors) occur when an event is falsely identified as malicious when it is benign. False Negatives (Type 2 errors) occur when an event is falsely identified as benign when it is malicious. Researchers evaluate machine learning algorithms to reduce false negatives in ransomware detection because of the financial and technological consequences of misclassifying a ransomware attack as benign (Bold et al., 2022). Bold et al., (2022) leveraged Artificial Neural Network (ANN) and Random Forest models to classify benign and malicious events across 1465 events. Ultimately, researchers determined that the Random Forest classifier had a false-negative rate of 0%, despite having a precision of only 92.73%, which was the lowest among the two models.

Power Analysis for Sampling Size

To obtain the minimum sample size conduct a power analysis with chosen alpha, effect size, and power value (Uttley, 2019). Alpha is the Type 1 error, or false positive rate, from Figure 5. Effect size, also known as Cohen's d, measures the relationship between two groups - in conventional approaches, the closer the value is to 1, the more obvious the relation is, the smaller it is, the less obvious it is. Finally, Power represents 1-False Negative Rate. The result of a power analysis is a Sample Size by which a statistical test detects true effects to a specific power value.

Chapter 3. Methodology

This chapter defines the methodology used to answer the RQ "What components of an ensemble Machine Learning model created on supervised machine learning models and trained on TON_IoT and SideChannel-3D data can accurately detect attacks to 3D printer's confidentiality, integrity, and availability (CIA)?" The steps taken include:

(1) **Labeling**: label, preprocess, and subset Network and SideChannel-3D data

(2) **Feature Engineering**: create custom features to best augment the existing variable set, specifically to transform the data into a model-ready form

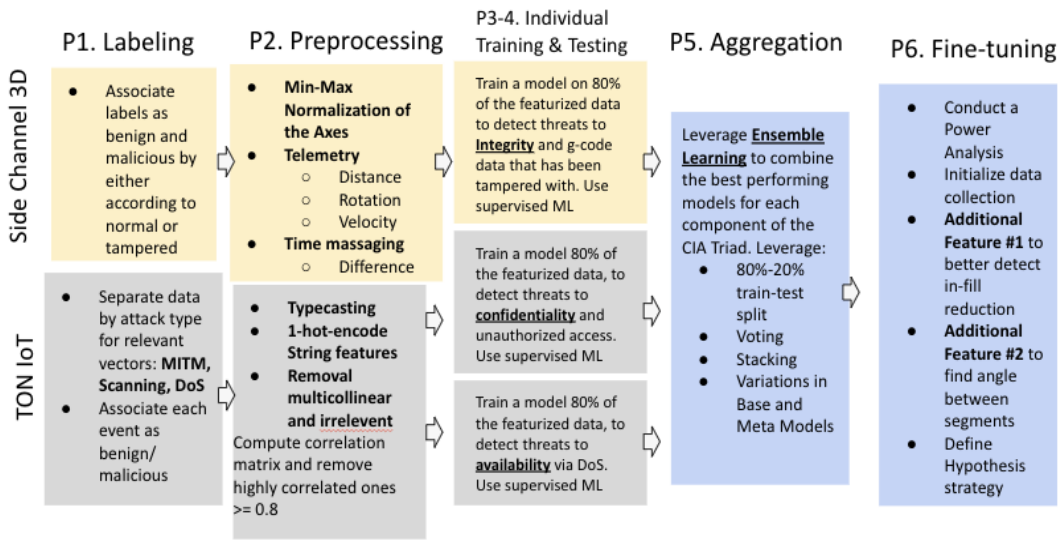
(3) - (4) **Individual Model Selection**: train and test various models for optimal base model and meta model selection, repeat for each element of the CIA Triad

(5) **Ensemble Aggregation**: combine representative models to account for CIA

(6) **Fine-tuning**: introduce additional preprocessing and define statistical tests

Figure 6

Roadmap of Methodology



3.1 Dataset Overview

TON_IoT is the chosen network traffic dataset (Alsaedi et al., 2020; Ashraf et al., 2021; Moustafa, 2019; Moustafa, 2021; Moustafa et al., 2020a; Moustafa et al., 2020b; Booij et al., 2022). SideChannel-3D (Costa et al., 2021) is the chosen tampered G-code collection. TON_IoT and corresponding literature contain relevant attack vectors that could affect 3D printers confidentiality and availability, detailed attack definitions, and a broader range of attacks, compared to KDD 99, UNSW NB-15, and Aposemat IoT-23. The SideChannel-3D dataset is the only available dataset that features printed tampering, focusing on the movement of the 3D printer, and timestamped G-codes for a collection of four unaltered and altered versions of the same objects. SideChannel-3D augments the network intrusion events in TON_IoT with integrity-targeting attacks specifically for 3D printers. TON_IoT data, downloaded from the UNSW Research site, was preprocessed and analyzed for modeling purposes. The SideChannel-3D dataset, acquired from IEEE Dataport, is referenced in the article "Sabotage Attack Detection for Additive Manufacturing System" (Yu et al., 2020), which describes the use case of the dataset.

TON_IoT Overview

TON_IoT is a heterogeneous dataset comprising raw PCAP, processed ZEEK logs, training, testing, and ground truth data collected from IoT and IIoT endpoint sensors. Network events are in log, CSV, and PCAP formats. TON_IoT is the primary source for training and testing a Confidentiality and an Availability model for this praxis, with 47 key features spread across nearly 22.3 million rows. Machine learning is applied to a labeled dataset with the goal of network security (Hashemi & Zarei, 2021).

The dataset was gathered at the UNSW Canberra Cyber IoT Lab, within the School of Engineering and Information Technology (SEIT), part of the Australian Defense Force Academy (ADFA) (Alsaedi et al., 2020; Ashraf et al., 2021; Moustafa, 2019; Moustafa, 2021; Moustafa et al., 2020a; Moustafa et al., 2020b; Booij et al., 2022). Various cyber-attacks are launched on an Industry 4.0 testbed, which comprises virtual machines, physical systems, cloud platforms, hacking platforms, and IoT/IIoT sensors (Alsaedi et al., 2020). Wireshark captures the network traffic while ZEEK simplifies and aggregates it. The RedNode Service (OpenJS Foundation & Contributors, n.d.) is the primary attack simulation environment. The dataset features 47 distinct attributes, including connection, statistical, DNS, SSL, HTAcc, and violation activities. It contains labels such as normal, backdoor, DDoS, DoS, injection, MITM, password, ransomware, scanning, and XSS, with value counts provided for each category.

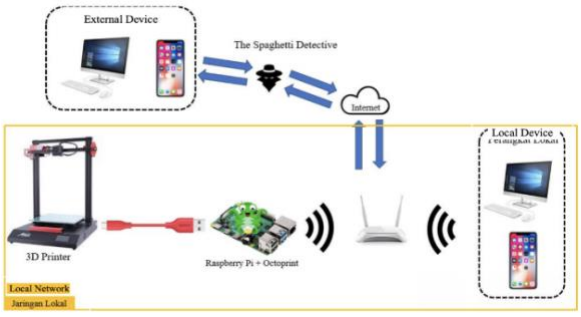
3D Printers as IoT Devices

The TON_IoT dataset includes a plethora of network devices, ranging from weather radars to garage door openers, that have specific features categorizing them as IoT devices. Features of IoT devices include: (1) contain a unique IP address, (2) send and receive network traffic, and (3) can be remotely monitored or controlled (Li et al., 2017). The IoT devices in TON_IoT have a unique IP address. These devices also send and receive network traffic, which gets collected, processed, and labeled. Lastly, these devices have the potential to be remotely monitored and controlled via the HTTP broker and are subject to Command-and-Control (C2) attacks and web-based attacks. 3D printers connected to a LAN can likewise follow the same criteria, making them fall within the IoT umbrella.

- Unique IP Address: By placing a 3D printer on a network, it contains a unique IP address, or in other words, a unique identifier by which the network addresses it. If a 3D printer does not natively connect to a network, it may use OctoPrint ("Download & Setup OctoPrint," 2025) to connect to a Raspberry Pi, allowing it to obtain an IP address.
- Send and receive network traffic: 3D printers can integrate and communicate with the internet as well as other devices. This connection enables the device to send and receive network traffic from any of the unique devices (Prianto et al., 2021; Wu et al., 2016). Additionally, if a 3D printer does not natively connect to a network, OctoPrint ("Download & Setup OctoPrint," 2025), in conjunction with a Raspberry Pi, allows it to send and receive communication via its IP address.

Figure 7

IoT-based 3D printer design (Prianto et al. 2021)



- Remotely Monitored or Controlled: A realization use case depicts "multiple 3D printers working under the control of the manager through the 3D printing monitoring platform based on the Internet of things" (Wu et al., 2016); highlighting another principal idea of IoT that these devices can be controlled or monitored wirelessly. OctoPrint ("Download & Setup OctoPrint," 2025), a web

interface, enables the monitoring of 3D print processes and the control of temperature, position, and status. At the process level, there are security risks to confidentiality, which make them susceptible to web-based and man-in-the-middle attacks that intercept and exfiltrate data.

Since 3D printers connected to a network follow the principles of IoT devices, they may be considered IoT devices. Therefore, they may appear as another device in the TON_IoT dataset. These devices are of special importance to government systems as they are "incorrectly categorized as standalone systems" (Department of Defense Office of Inspector General, 2021).

While training the detector on SideChannel-3D safeguards a 3D printer's integrity against tampering, TON_IoT is the only dataset that presents network traffic for a general classification of IoT devices and attack vectors known to affect 3D printers (Šlajs et al., 2023; McCormack et al., 2020; McCormack, 2021).

Figure 2 illustrates the attack vectors affecting 3D printers in transit, at rest, and in use. The table below lists the attack vectors affecting 3D printers and those present in TON_IoT. TON_IoT presents and leverages methods of executing these attacks. Further insight about the attack vectors that affect 3D printers are visible in section 2.1.2 and Table 1. Section 2.2.4 highlights the execution of TON_IoT attacks and the method of generating PCAP and ZEEK logs.

Table 4

Mapping TON_IoT Attack Vectors to 3D printing attack vectors

TON_IoT Attack Vectors	**3D Printer Attack Vector** (Šlajs, Klaban, & František, 2023)	**C3PO Attacks** (M. McCormack et al., 2020; Mccormack, 2021)	**CIA Element**
Man-in-the-Middle	Man-in-the-Middle	MitM/ Spoofing	Confidentiality
Ransomware	Network Vulnerabilities	*Not Mentioned*	Availability
Scanning	Network Vulnerabilities	Scanning	Confidentiality
DoS/DDoS	Network Vulnerabilities	TCP & SYN Flood	Availability
XSS/Backdoor/ Injection	Network Vulnerabilities	Backdoor	Confidentiality
Password	Weak Credentials	*Out of Scope*	Confidentiality

After researching publicly available databases, no dataset is currently available that explicitly targets cyberattacks on 3D printers connected to a network. The use of TON_IoT is the most appropriate benchmark, as 3D printers on a network are considered IoT devices. The dataset contains attack vectors known to affect 3D printers, namely network vulnerabilities. As such, the TON_IoT data will be the predominant data source describing attacks to a 3D printer's confidentiality and availability. Man-in-the-Middle attacks, specifically ARP Spoofing, can put the confidentiality of data in transit as it traverses from the device to the 3D printer. The data can further be viewed, altered, or maliciously encrypted. DoS attacks can overwhelm a 3D printer in use, compromising its availability. Additionally, file tampering can occur to the data at rest.

Table 5 shows the key features (Alsaedi et al., 2020; Ashraf et al., 2021; Moustafa, 2019;

Moustafa, 2021; Moustafa et al., 2020a; Moustafa et al., 2020b; Booij et al., 2022).

Table 5

Conn.log Features of TON_IoT

Variable	Description
ts <int>	Timestamp of the start of the event
src_ip <str>	IP Address which sent the network event from
dst_ip <str>	IP Address which received the network event from
src_port <str>	IP Address port which sent the network event from
dst_port <str>	IP Address port which received the network event from
proto <str>	Protocol which made connection (UDP,TCP,ICMP,ARP)
service <str>	Communication Service Used (ssl, dns, http)
duration <int>	Length in seconds of the network event
src_bytes <int>	Number of bytes the Source IP Address sent
dst_bytes <int>	Number of bytes the Destination IP Address received
conn_state <str>	State of the connection (SYN, ACK, RST, REJ)
missed_bytes <str>	Number of unsuccessful bytes in the traffic
src_pkts <int>	Number of packets the Source IP Address sent
dst_pkts <int>	Number of packets the Destination IP Address received
label <int>	0 if Normal; 1 if Malicious
type <string>	Type of attack

SideChannel-3D

SideChannel-3D: Researchers generated a collection of datasets for tampered and untampered 3D printable objects. "*Sabotage Attack Detection for Additive Manufacturing Systems*" was the basis for this dataset. Four test cases were included (square, rotating gear, wrench, and gear) to describe normal and modified structures (Yu et al., 2020). A 3D printer prints each use case, with various telemetry probes extracting information that has been passed through a machine learning model after labeling (Costa et al., 2021):

- **data files**: data for magnetometer, microphone, accelerometer, and power.
- **segments**: all straight-line segments printed in the test case
- **G-code files**: sliced instructions on how to print

- **stl images**: original CAD image for each test case

The primary value of this dataset for this praxis research lies in the segments.csv files.

Table 6

Features of SideChannel-3D Segments

Features	Description
timeStart & timeStop	Start and Stop time of the Extrusion Segments
X1 & X2	Start and Stop X-axis value of the Extrusion Segments
Y1 & Y2	Start and Stop Y-axis value of the Extrusion Segments
Z1 & Z2	Start and Stop Z-axis value of the Extrusion Segments
Vx & Vy & Vz	Velocity of the segment on the axes.

3.2 Data Usage & Labeling

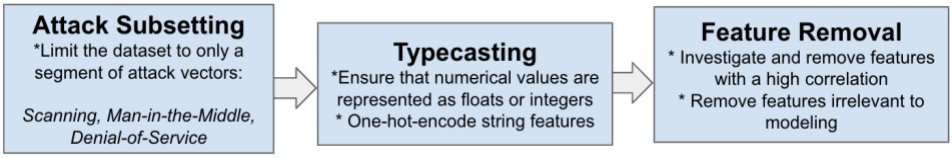
TON_IoT

Prior to machine learning modeling, TON_IoT is preprocessed in a manner that satisfies the formatting requirements for scikit-learn (Pedregosa et al., 2011), while also accounting for the essential attributes of this research. The Processed Network Dataset contains labeled event-level data rooted in ZEEK log files and, in turn, packet capture (PCAP) files. Each row in the dataset describes a single event at a specific time. The general framework for preprocessing the TON_IoT dataset was the following:

1. **Attack Subsetting**: Subsetting the dataset for only the attack labels that can affect the confidentiality and availability of 3D printers in operational environments
2. **Typecasting**: Ensuring that feature values have a numerical representation
3. **Removing Features**: Removing high multicollinearity and irrelevant features

Figure 8

Mapping of preprocessing TON_IoT



(1) **Attack Subsetting:** Man-in-the-Middle Attacks, Scanning Attacks, and Denial-of-Service Attacks were chosen.

- **Man-in-the-Middle attacks**, also known as on-path attacks pose a threat to confidentiality by intercepting packets sent and received by two authorized communicators.
- **Scanning attacks** describe another threat to confidentiality where an adversary passively leverages usage monitoring and metadata to find out information about active devices on a network. This often leads to further complex attacks.
- **DoS and Distributed Denial-of-Service DDoS attacks** describe threats to a 3D printer's Availability where an adversary hinders the normal usability of the 3D printer.

Table 7

Total counts of Attack Categories

Category	Attacks	Count	Processed Network File Number
Normal	Normal	16428261	All
Confidentiality	Scanning	7140161	1,2,3,4,5,6,7,**8**
	MitM	1052	**23**
Availability	DoS	3375328	**8**,9,10,11

According to the ground truth dataset, files 8 and 23 contain malicious events that

execute Scanning, MITM, and DoS, as indicated in **Table 7** above. Processed_Network_Events_**8**.csv comprises Normal, Scanning, and DoS attacks, and Processed_Network_Events_**23**.csv comprises Normal and Man-in-the-Middle attacks (MitM).

P1 in Figure 6 starts by aggregating the Malicious and Normal labeled data for each attack collection. Then, it separates into confidentiality and availability for individual modeling (section 3.4), predicting threats to each of these elements. An event labeled as normal was labeled as "Normal" in the training data for the confidentiality and availability event collections. On the contrary, if an event had "type" Scanning or MitM, it was labeled as "Malicious" in the training data for the confidentiality event collections; if an event had "type" DoS, it was labeled as "Malicious" for the availability event collections.

Table 8

Subsetted Attack Vectors and Labels

Category	Attacks	Labels	Processed Network File Number
Confidentiality	Normal	Normal	8 & 23
	Scanning	Malicious	8
	MitM	Malicious	23
Availability	Normal	Normal	8 & 23
	DoS	Malicious	8

(2) **Typecasting:** Typecasting occurs when assigning numeric types to attributes or variables appropriately. **Table 5** displays the variables and their current types, associated with their values, specifically in the form of numerical and categorical data detailing a specific attribute of the ZEEK event. Two primary forms of type transformations were leveraged: (1) casting numerical values appropriately as integers or floats; (2) applying

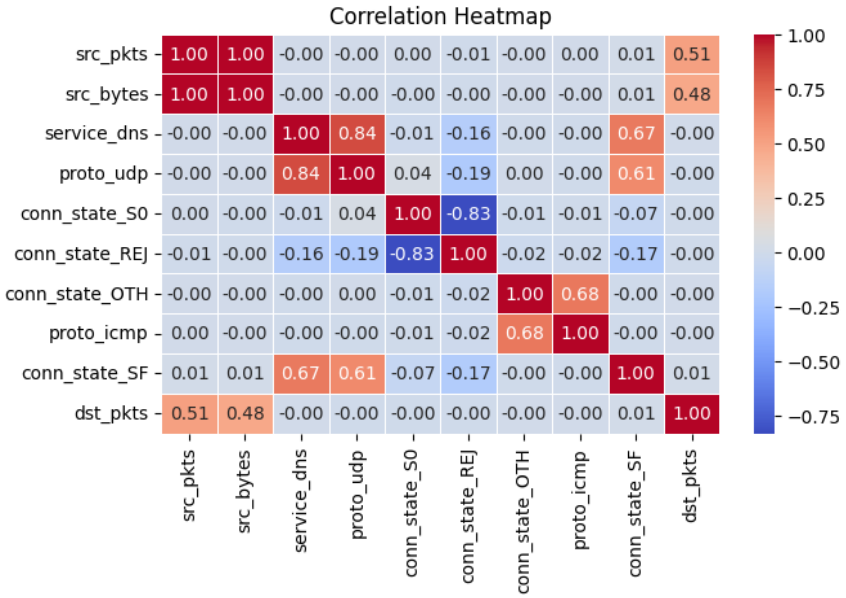
"one-hot encoding", the method of transforming a string-based variable column into a series of 1's and 0's, with new column values being the unique categories.

(3) **Variable Removal:** A correlation matrix identifies multicollinear features that subtly bias the model. This praxis removed variables based on:

- **Irrelevance**: IP addresses, attack types, and timestamps, because they cannot be modeled as features. Doing so would bias the model.
- **Multicollinearity**: Figure 9 displays highly correlated values in deep blue and red, resulting in the removal of packet features.

Figure 9

Highly correlated values from the set of confidentiality & availability



By referencing the information in Figure 9, the necessary transformations described in Table 9 ensure the variables are in an appropriate state for modeling.

Table 9

Description of final variable transformation

Variable	Type	Transformation (s)
src_ip	<string>	Removed for modeling
dst_ip	<string>	Removed for modeling
src_port	<string>	Cast type to integer
dst_port	<string>	Cast type to integer
proto	<string>	One-hot Encoded Values
service	<string>	One-hot Encoded Values
duration	<integer>	No Change
src_bytes	<integer>	No Change
dst_bytes	<integer>	No Change
conn_state	<string>	One-hot Encoded Values
missed_bytes	<integer>	Removed for modeling
src_pkts	<integer>	Removed to Multicollinearity with src_bytes
dst_pkts	<integer>	Removed to Multicollinearity with dst_bytes
ts	<integer>	Removed for modeling
type	<string>	Removed for modeling

SideChannel-3D

SideChannel-3D provides a corpus of benign and tampered G-code, stl files, and side-channel findings that highlight attacks to a 3D printer's integrity. The steps used to preprocess the data and prepare it for modeling include:

1. Acquiring the Normal and Tampered timestamped G-code segments **for one use case presented**, the wrench. This praxis labels G-code segments according to

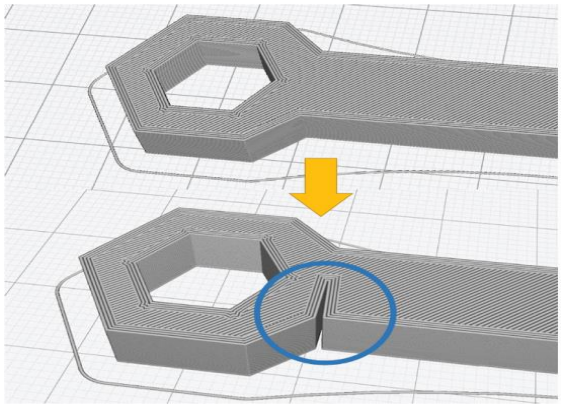
whether they were tampered. **The concatenated benign and tampered wrench segments define the dataset.**

2. Removing irrelevant variables that do not represent the behavior of the data needed to answer the primary research objective of this praxis.

(1) Segments.csv: The wrench use case presented in Figure 10 was the training data for SideChannel-3D and integrity model (Costa et al., 2021). A "segment" is defined as a linear movement in one direction in the build region. Both the normal and tampered versions of the use case were leveraged as the dataset. Each segment is then subsequently column-appended by whether it came from a "normal" collection or a "tampered" collection. A master file containing the row-wise appended segments of each use case for both normal and tampered segments, creating an extra column that holds the label for each timestamped segment.

Figure 10:

Normal and tampered wrench (Yu et al., 2020)



(2) Removal of variables: Table 10 presents an overview of features removed from segments.csv. Only the following variables remained: timeStart, timeStop, X1, X2, Y1, Y2, Z1, Z2, Vx, Vy, Vz, and Label; descriptions of which are in Table 10.

Table 10

Variable removal for modeling

Variable	Description	Reason for Removal
Axy	A binary value denoting if the segment showed movement in the X & Y coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values
Axz	A binary value denoting if the segment showed movement in the X & Z coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values
Ayz	A binary value denoting if the segment showed movement in the Y & Z coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values
layer	Layer of the current segment	All values are 0
OorI	O or I* Details unclear	Unclear what the goal of this variable is
Ax	A binary value denoting if the segment showed movement about only the X coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values
Ay	A binary value denoting if the segment showed movement about only the Y coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values
Az	A binary value denoting if the segment showed movement about only the Z coordinates	Information already accounted for in the X1, Y1, Z1, X2, Y2, Z2 values

Each G-code file consists of linear G-code segments of extruded material over time and space. After labeling each G-code segment, additional features are generated, followed by machine learning. A snapshot of the segment data appears in Table 11:

Table 11

Example of the segment data for the benign wrench

timeStart	timeStop	X1	Y1	Z1	X2	Y2	Z2	Vx	Vy	Label
2019-09-10 17:13:39.916	2019-09-10 17:13:39.92	67.3	93.9	0.3	67.8	93.71	0.3	63.02	-38.6	Normal
2019-09-10 17:13:39.923	2019-09-10 17:13:39.94	67.8	93.7	0.3	68.2	93.51	0.3	38.76	-15.9	Normal
2019-09-10 17:13:39.935	2019-09-10 17:13:39.95	68.2	93.5	0.3	68.7	93.4	0.3	35.76	-8.47	Normal

3.3 Feature Engineering

After the TON_IoT and SideChannel-3D data have been labeled and preprocessed, additional features are generated that detail and describe the characteristics of the dataset. Specifically, this praxis seeks to incorporate additional pieces of information, aggregation steps, and functional representations of the data that can aid the model, while improving performance characteristics, variable importance values, and model interpretability.

TON_IoT

TON_IoT only retains the subset of attack vectors pertinent to this praxis, appropriately typecasts specific attributes in the form of numerical values, and removes irrelevant and highly correlated values. Since the current form of the data has the necessary information and format to feed the model, the praxis generated no additional features for TON_IoT.

SideChannel-3D

The three broad categories of Feature Generation for the SideChannel-3D dataset are: Normalizing Axes, Physics-based features, and Time Differencing. Normalizing Axes

creates a representation of relative x, y, z position about the minimum and maximum build coordinates. Physics-based representations of speed and trajectory of movement about the build plate over time further augment the feature set. The Time Differencing feature displays the duration of the segment. Each of the feature generation functions is detailed below:

- **Normalizing Axes:** To enhance the reusability of the model and allow G-code data with different dimensions, Normalizing Axes creates relative representations of position based on their extreme values, i.e., the minimum and maximum values. Normalizing standardizes the values to the build size limitation of each 3D printer. This praxis explored the Z-score scaling approach, which standardizes data relative to a given distribution; however, it would not be ideal for identifying specific attack vectors related to integrity.
    - Normalized Start & Stop X: $\frac{Xi-min\{X\}}{max\{X\}-min\{X\}}$; scaling every Xi value between 0 and 1 per the min and max values of the feature.
    - Normalized Start & Stop Y: $\frac{Yi-min\{Y\}}{max\{Y\}-min\{Y\}}$; scaling every Yi value between 0 and 1 per the min and max values of the feature.
    - Normalized Start & Stop Z: $\frac{Zi-min\{Z\}}{max\{Z\}-min\{Z\}}$; scaling every Zi value between 0 and 1 per the min and max values of the feature.
- **Physics-based**: Additional physics-based features describe the distance, angle of rotation, and the magnitude of velocity (speed) of each G-code segment. X1, Y1, and Z1 are the starting points of each segment, and X2, Y2, and Z2 are the end points of each segment. Vx, Vy, and Vz are the velocities in each of the 3 axes.

- Distance: Magnitude in all directions.

  $\sqrt{(X2i - X1i)^2 + (Y2i - Y1i)^2 + (Z2i - Z1i)^2}$

- Velocity Magnitude: Combined Speed in X, Y, Z.

  $\sqrt{(Vx\ i)^2 + (Vy\ i)^2 + (Vz\ i)^2}$

- Rotation (Phi): Angle between the Velocity vectors. $arctan(\frac{Vy}{Vx})$

- **Time** By first converting the timestamped data to datetime, subtracting the time the segment finished (timeStop) from the time the segment started (timeStart), represents the duration of the segment.

3.4 Individual Model Evaluation

Labeled, preprocessed, and feature-generated confidentiality, integrity, and availability data are subsequently modeled using supervised machine learning. Scikit-learn in python: split the dataset into training and testing data, trained various supervised machine learning models, and returned confusion matrix metrics to assess performance (Pedregosa et al., 2011). The pandas package was extensively leveraged to subset and process the data (Pandas, 2025).

The individual modeling workflow includes rebalancing the data, conducting a train-test split, modeling the data with cross-validation, and finally exporting the mean recall scores. The best-performing model for each element of the CIA triad was chosen and stored for the next step, Ensemble Learning. The process is:

(1) Rebalancing the data: The pandas package, alongside a custom function, created a random sample of 2500 benign instances and 2500 malicious instances.

(2) Model Set-up: Rows from the rebalanced data with empty values were removed using the pandas dropna() function in python. This praxis utilized the train_test_split() function in scikit-learn to split the data into an 80/20 train-test split, where training occurred on 80% of the labeled rows and testing occurred on the remaining 20%.

(3) Model Methods Ran:

- Logistic Regression (LR) generates a best-fit line in k dimensions, where k is the number of features used in the model. Logistic regression utilizes a sigmoid function to perform binary classification while adhering to the principles of multiple linear regression. Logistic regression provides an easily interpretable solution; however, the inflexible nature of these models makes them prone to underfitting new test data. (Alotaibi & Ilyas, 2023; Bold et al., 2022)
- Decision Trees (DT) test conditions iteratively to maximize information gain and classify the data into normal and malicious. Although it is an inflexible model, it is easily interpretable (Leevy et al., 2020; Alotaibi & Ilyas, 2023; Bold et al., 2022).
- Support Vector Machines (SVMs) create a boundary hyperplane that separates data into two classes. It is a tested model for detecting DoS attacks, as demonstrated in University of New Brunswick. (2019b).
- Random Forest (RF) ensemble decision-tree models choose the most common class of an input vector via voting (Breiman, 2001). Unlike logistic regression, decision tree, and support vector machine models,

random forests are flexible yet are not susceptible to over-fitting. SideChannel-3D and 6 of the IoT Network Traffic data sources presented in Table 2 utilize Random Forest models for classification. 4 sources determined it the best-performing model.

- Gradient Boosting Classifier (GBC), a tree-based ensemble approach that focuses on minimizing the loss function (He et al., 2019). This model performs well on edge cases.

(4) Cross-Validation: The rebalanced dataset leverages Leave-One-Out cross-validation by creating 5, 20% segments, with each segment acting as the testing set and the remaining 80% the training. The approach averages the scores of the five runs.

3.5 Ensemble Aggregation of Models

The Histogram Gradient Boosting Classifier is an alternative to the Gradient Boosting Classifier, offering improved accuracy and model efficiency through its binning approach to categorizing data (Nhat-Duc & Van-Duc, 2023). The Histogram Gradient Boosting Classifier performed 1% to 5% better than the Gradient Boosting Classifier algorithm for each comparison metric. As such, it replaced the Gradient Boosting Classifier as a base classifier for Ensemble Modeling.

Following the selection of supervised individual models for confidentiality, integrity, and availability, Ensemble Voting and Stacking models better detected a subset of network attacks and physical tampering attacks together. The scikit-learn package and the ensemble function: split the dataset into a train-test split, create various models for testing, and return evaluation metrics (accuracy, precision, recall) to assess performance

(Pedregosa et al., 2011). Beyond scikit-learn, the pandas package was leveraged to perform data subsetting and processing (Pandas, 2025).

(1) Rebalancing the data: As in section 3.4, the pandas package, alongside a custom function, created a random sample of 2500 benign instances and 2500 malicious instances. Samples of 3000 confidentiality instances for each class were scaled down to 2500 for uniformity in analyzing the research question.

(2) Model Set-up: As in section 3.4, rows from the rebalanced data with empty values were removed using the pandas dropna() function in python. This praxis utilized the train_test_split() function in scikit-learn to split the data into an 80/20 train-test split, where training occurred on 80% of the labeled rows and testing occurred on the remaining 20%.

(3) Ensemble Model Methods Run:

- Stacking: Stacking models can harness the power of all the chosen base classifiers while leveraging a chosen meta model. A meta model takes the predictions of all the base classifiers as inputs and models the resulting data. The results of the meta classifier will be the final prediction model. Base classifiers include the Random Forest Classifier, Histogram Gradient Boosting Classifier, and Logistic Regression, representing a sample of models available based on model performance, interpretability, and method. The formula below describes the usage of the base models in the matrix and meta model as the label-determining model. Specifically, “x” is a chosen input vector and “y_predicted” is its resulting label. The following process gets repeated for all input vectors or instances in the

ingested data. A variety of different Meta Models were tested with the chosen Base Model Classifiers (Alotaibi & Ilyas, 2023) and (Ravi et al., 2005).

$$y_{predicted} = MetaModel([RF(x)\ HGBM(x)\ LR(x)])$$

  - Logistic Regression (LR): Provides an inflexible yet interpretable approach.
  - Random Forest (RF): Applies ensemble methodology twice; once in the base classification and once as the meta model.
  - Histogram Gradient Boosting Classifier (HGBC): Best handles the edge cases associated with the results of the base classifiers.

- **Voting**: The ensemble voting models consist of hard and soft voting. "Hard voting" includes a series of chosen base classifiers that feed an ensemble classifier. Based on each input vector, each base model determines a class label. The input vector assigns the label as the majority vote. In the case of "soft voting", the label assigned to each input vector is the one with the highest average predicted probability for each class across all base models. The class with the best average probability is determined (Alotaibi & Ilyas, 2023; Ravi et al., 2005).

(4) Cross-Validation: The rebalanced dataset leverages Leave-One-Out cross-validation by creating 5, 20% segments, with each segment acting as the testing set and the remaining 80% the training. The approach averages the scores of the five runs.

3.6 Fine-tuning

After capturing initial results, this praxis introduced minor changes to augment the ensemble model. To improve model performance, interpretability, and explainability, a power analysis determines the optimal train-test split sizes that minimize false negatives. Translated into an additional feature, the angle between segments better represents the primary forms of integrity attacks in this praxis, punctures. SideChannel-3D was relabeled at the segment level, better highlighting the exact points of tampering.

Optimal Train-Test-Split Sizes

A power analysis determines the minimum sample size required for the training and testing sets to minimize false negative rates. Prior to running the power analysis, the desired alpha (or Type 1 Error), effect size (or distribution distance), and power value (1-Type 2 Error) needed to be determined. For the following chosen values:

- **Alpha** (Type 1 Error) = **0.05**: Implying a False Positive Rate of 0.05
- **Beta** (Type 2 Error) = **0.00**: Implying a False Negative Rate of close to 0.00
- **Cohen's distance** = **0.2:** Implying a small distance between the distribution of groups.

The power analysis with the specifications above revealed that a minimum sample size of **637.61** or **638** is needed to minimize the false negative rate as low as possible. Rebalancing the sample of 5000 rows for both labels for each model with an 80%-20% train-test split at 4000 (training) and 1000 (testing), satisfying that a training size of 5000 and 1000 is greater than 637.61.

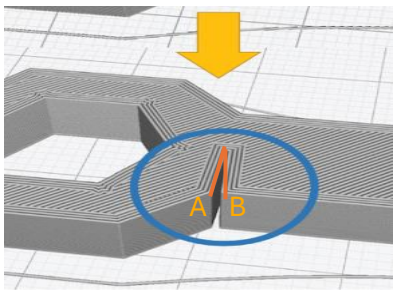
Creation of an additional SideChannel-3D Segment

To effectively capture the underlying characteristics of puncture integrity, this

praxis created a new feature. Punctures are small, quick, recurring "cuts" to G-code, typically taking the form of a slight acute angle between G-code segments near the exterior of the 3D printed object. Figure 11 below shows the puncture presented in the wrench object.

Figure 11

Puncture is a small angle between segments



Additional Feature Generation: Between-segments Angle

- In addition to including Rotation (Phi), it is worth adding a feature to best identify puncturing attacks by investigating the angle between segments in **Figure 11** modified from Yu et al. (2020).
- The angle between segments is $\theta = arccos(\frac{A \cdot B}{|A| * |B|})$
- Python is leveraged to implement it as such.

Update the Labeling of SideChannel-3D Data

The initial run of results showed an accuracy, precision, and recall of nearly 90% for ensemble stacking models. Following, the methodology assigned labels based on each segment depiction as normal or tampered condition. Since there are only 675 of 4752 modified segments, while the remaining are normal segments (Yu et al., 2020), the 675 malicious segments were oversampled to create 2500 more tampered segments using

SMOTE (Cao et al., 2025). The integrity data then followed the preprocessing, feature generation, and modeling procedure.

Defining True Positive Rate (Recall) Threshold

Table 12

Relevant resources for Recall Benchmark

Data	Source	Model Chosen	Recall
TON_IoT	ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets **(Booij et al., 2022)**	Random Forest	0.97264
	Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security **(Alotaibi & Ilyas, 2023)**	Stacking (Base: DT, RF, KNN, LR, Meta: LR)	0.9860
SideChannel-3D	Sabotage Attack Detection for Additive Manufacturing Systems **(Yu et al., 2020)**	Axis Attack Detection Performance GBC & RF	0.9899

<u>*Mean Recall:*</u> 0.982 is the average recall from data sources in **Table 12.** To answer research question, a one-sample z-test for proportions was proposed to assess the number of correctly identified malicious instances among the pool of actual malicious instances. Prior to conducting the hypothesis test, the praxis checked the following assumptions to ensure the prerequisites were met.

One-Sample Z-Test for Proportions Assumptions:

- Random sample of n drawn from population N

    - Satisfied: sample drawn from random.sample() followed by random train/test split
- Binary Outcomes:
    - Satisfied: there are only two outcomes for labels (benign and malicious)
- Independence (10 * Sample (n) <= Population (N))
    - Satisfied: 10 * 572 <= ~27 million
- Sample Size (np >= 10 & n(1-p) >= 10)
    - Satisfied: (572 * 0.982) = 560.56 >= 10
    - Satisfied: (572 * (1 - 0.982)) = 10.296 >= 10

This praxis will use 0.982 as the benchmark for hypothesis testing, as it is the mean recall value from **Table 12**, similarly leveraging detection capabilities on the data sets used in this praxis. A recall value less than 0.982 represents a lower rate of correctly identified malicious instances from the pool of truly malicious events compared to the average recall of representative sources. Conversely, a recall value greater than 0.982 denotes a higher rate of correctly identified malicious instances.

Chapter 4. Results

4.1 Introduction

Chapter 4 presents the findings of TON_IoT and SideChannel-3D dataset trained individual and ensemble machine learning models, which were used to measure a selection of network and tampering attacks affecting the confidentiality, integrity, and availability of 3D printers. Presented at the end of this chapter is an answer to the research question, validated with a hypothesis test.

Results from the analysis conducted through the methodology and the stepwise process directed in Chapter 3, will be presented in the following order: 1) individual supervised machine learning models which established a subset of models for potential ensemble classification, 2) the determination of an ensemble classifier trained on TON_IoT and SideChannel-3D data, 3) hypothesis testing of research question.

Figure 12

Chapter 4 Outline

4.2 Individual Model Results

This praxis assessed a selection of non-neural-network supervised machine learning models mentioned in Chapter 2 and evaluated them in terms of accuracy, precision, and recall. This was an initial step in identifying representative models for base classifiers or meta classifiers in the ensemble modeling in section 4.3. These models were then trained and tested on TON_IoT (Alsaedi et al., 2020; Ashraf et al., 2021; Moustafa, 2019; Moustafa, 2021; Moustafa et al., 2020a; Moustafa et al., 2020b; Booij et al., 2022)

and SideChannel-3D datasets (Costa et al., 2021).

Table 13

Individual Model Performance Metrics

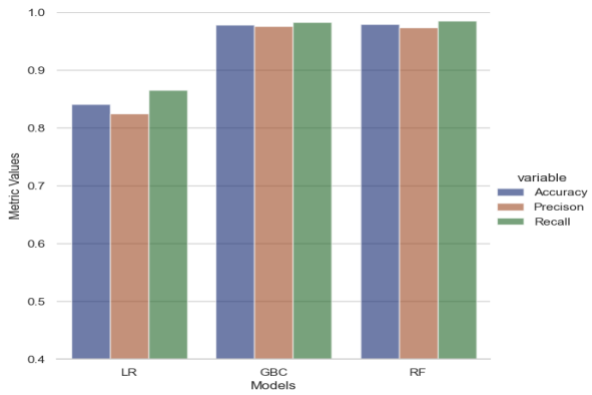
Data	Model	Accuracy	Precision	Recall	Runtime
confidentially	LR	0.84039	0.824915	0.865609	4.222702
confidentially	DT	0.967715	0.96505	0.9711	0.127595
confidentially	GBC	0.978375	0.975445	0.98209	2.659085
confidentially	RF	0.978679	0.9737	0.984478	2.811169
availability	LR	0.973437	0.991034	0.956075	4.243333
availability	DT	0.979063	0.978919	0.979546	0.10564
availability	GBC	0.986875	0.994976	0.978972	2.495723
availability	RF	0.988125	0.996829	0.979588	2.456953
integrity	LR	0.648438	0.651279	0.649163	2.829041
integrity	DT	0.996875	0.996251	0.997509	0.193047
integrity	GBC	0.999063	0.998134	1	2.429313
integrity	RF	0.997188	0.994433	1	4.679132

Table 13 above displays the accuracy, precision, recall, and runtime for each of the individual models with scikit-learn (Breiman, 2001) specifications: **(1)** Random state of 1234 for all models **(2)** 5-fold cross-validation with 80%-20% Leave-One-Out approach.

Bar plots presented in Figures 13, 14, and 15 were leveraged to compare each supervised machine learning model presented above for each element of the CIA triad. The Decision Tree and Gradient Boosting Classifier were removed from the selection of the stacking ensemble. This praxis initially investigated support vector machines, but they were excluded as they required comparatively more training time. Random forests were included because they frequently appear in current research focused on TON_IoT attack detection and SideChannel-3D, and yet they remain interpretable (Yu et al., 2020; Alotaibi & Ilyas, 2023).

Figure 13

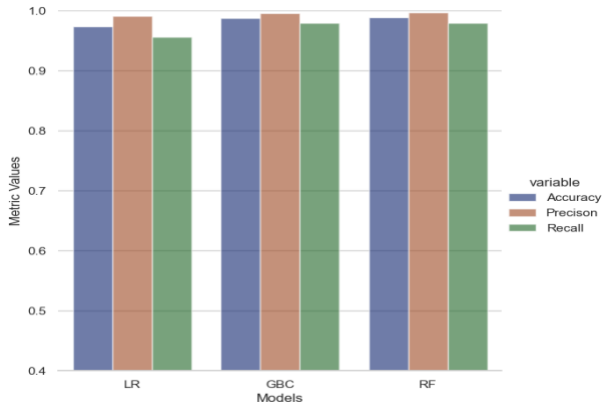
Individual Model Evaluation on the Confidentiality dataset (TON_IoT)



Leveraging 5-fold cross-validation, the overall results in Figure 13 show that the confidentiality dataset yields accuracy, precision, and TPR values above 95% for both the Gradient Boosting Classifier and the Random Forest. Logistic Regression shows accuracy, precision, and TPR values near 80%.

Figure 14

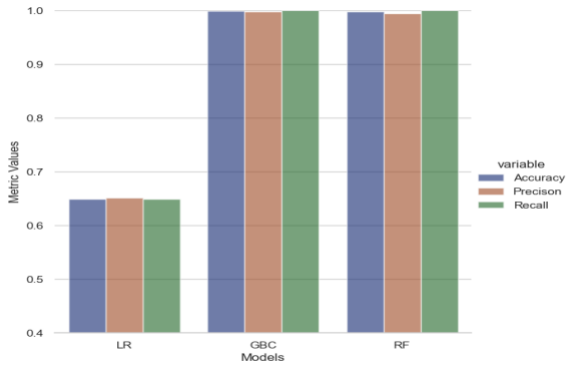
Individual Model Evaluation on the Availability dataset (TON_IoT)



When applying the same specifications to the availability dataset, a near-1.0 accuracy for the GBC and RF performance values, and above 95% for the LR models, is observed from Figure 14.

Figure 15

Individual Model Evaluation on the Integrity dataset (SideChannel-3D)



The model metrics for the integrity dataset, displayed in Figure 15, revealed that Gradient Boosting Machines and Random Forest Models displayed near-perfect classification. The Logistic Regression Model depicted performance metrics of around 65%.

4.3 Ensemble Model Results

Ensemble classifiers like stacking and voting combine and aggregate the effects

of multiple base model classifiers. **Tables 14.1-14.3** present the performance of different forms of ensemble classification; trained and tested on TON_IoT (Alsaedi et al., 2020; Ashraf et al., 2021; Moustafa, 2019; Moustafa, 2021; Moustafa et al., 2020a; Moustafa et al., 2020b; Booij et al., 2022) and SideChannel-3D (Costa et al., 2021).

Table 14.1

Ensemble Stacking Model (with meta LR) Performance Metrics for Confidentiality

Data	Model	Accuracy	Precision	Recall	Runtime
confidentiality	GBC	0.978375	0.975445	0.98209	2.659085
confidentiality	RF	0.978679	0.9737	0.984478	2.811169
confidentiality	Stacking LR	0.981419	0.980321	0.983364	40.04535
confidentiality	Stacking GBC	0.974417	0.973411	0.97834	41.73537
confidentiality	Stacking RF	0.97868	0.980758	0.977238	42.14538
confidentiality	Voting Hard	0.975937	0.971432	0.98272	7.686923
confidentiality	Voting Soft	0.975328	0.968723	0.98327	7.72163

Table 14.2

Ensemble Stacking Model (with meta LR) Performance Metrics for Availability

Data	Model	Accuracy	Precision	Recall	Runtime
availability	GBC	0.986875	0.994976	0.978972	2.495723
availability	RF	0.988125	0.996829	0.979588	2.456953
availability	Stacking LR	0.988438	0.998734	0.978363	38.90948
availability	Stacking GB	0.985625	0.994351	0.977743	41.09697
availability	Stacking RF	0.987813	0.998111	0.977113	41.67283
availability	Voting Hard	0.987813	0.998732	0.977126	7.631066
availability	Voting Soft	0.988125	0.999365	0.977126	7.550876

Table 14.3

Ensemble Stacking Model (with meta LR) Performance Metrics for Integrity

Data	Model	Accuracy	Precision	Recall	Runtime
integrity	GBC	0.999063	0.998134	1	2.429313
integrity	RF	0.997188	0.994433	1	4.679132
integrity	Stacking LR	0.999063	0.998134	1	32.58285
integrity	Stacking GB	0.99875	0.99813	0.999365	34.18321
integrity	Stacking RF	0.999063	0.998761	0.999365	34.25089
integrity	Voting Hard	0.996562	0.993183	1	6.125548
integrity	Voting Soft	0.999063	0.998134	1	6.114238

Gradient Boosting Classifier (GBC) and Random Forests (RF), mentioned in Section 4.2, are ensemble classifiers used to investigate the performance of individual models. Stacking LR represents an ensemble stacking model with Logistic Regression as the meta-classifier. Stacking GBC represents an ensemble stacking model with a Gradient Boosting Classifier as the meta-classifier. Similarly, the Stacking RF comprises an ensemble stacking model with the Random Forest as the meta model. Each stacking model comprises base classifiers: Logistic Regression, Random Forest, and Histogram Gradient Boosted Classifier. The stacking models had significantly higher training times than the voting models and the ensemble models used in individual modeling (Gradient Boosting Classifier and Random Forest).

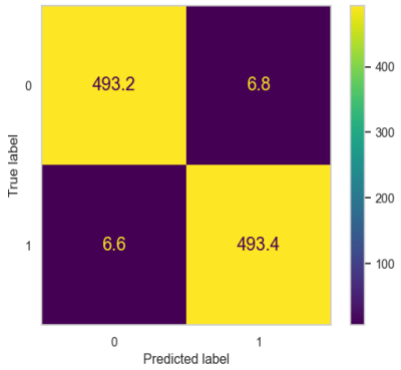
Tables 14.1-14.3 reveal that the training time of stacking models is 6 times longer than Voting, Random Forest, and Gradient Boosting Classifier. Although it only supersedes other models in 4 of the 12 criteria (4 evaluation criteria for 3 CIA elements), Stacking Logistic Regression Model provides a comprehensive, interpretable, and

unbiased model for malicious attack detection.

Confusion Matrices (Ensemble Stacking Model & Logistic Regression Meta Model)

Figure 16

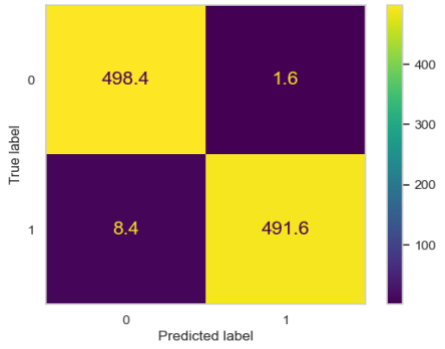
Confidentiality Confusion Matrix Avg. for Stacking LR with 5-fold CV



For the confidentiality data shown in Figure 16, the average number of false positives (FP) is 6.8 (FPR = 1.4%), and FN is 6.6 (FNR = 1.3%).

Figure 17

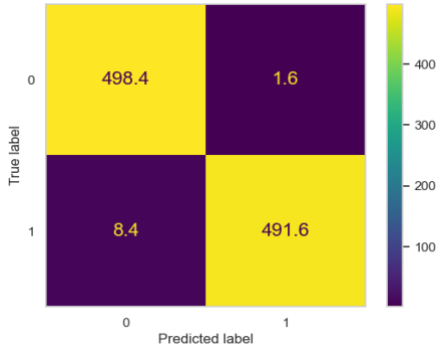
Integrity Confusion Matrix Avg. for Stacking LR with 5-fold CV



For the integrity data shown in Figure 17, the average number of false positives (FP) is 0.6 (FPR = 0.1%), and false negative (FN) is 0.2 (FNR = 0.04%).

Figure 18

Availability Confusion Matrix Avg. for Stacking LR with 5-fold CV



For the availability data shown in Figure 18, the average number of false positives (FP) is 1.6 (FPR = 0.3%), and FN is 8.4 (FNR = 1.7%)

Receiver Operating Characteristic (ROC) Curves

ROC curves provide a visual determination of classification performance, as well as an alternative performance metric (AUC). The chosen model was the Stacking Model: Base Classifiers (LR, RF, HGBC) and LR Meta Classifier (Bradley, 1997).

Figure 19

Confidentiality ROC Curve

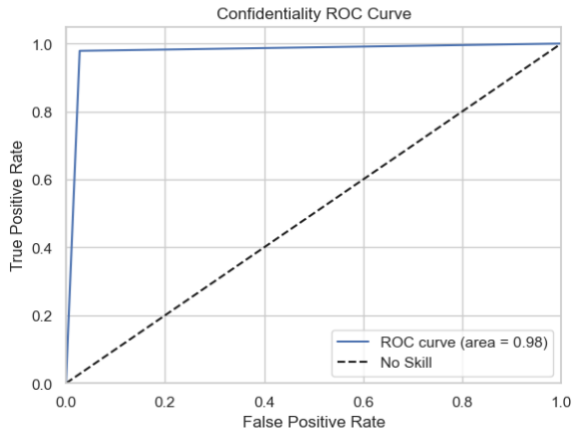


Figure 20

Availability ROC Curve

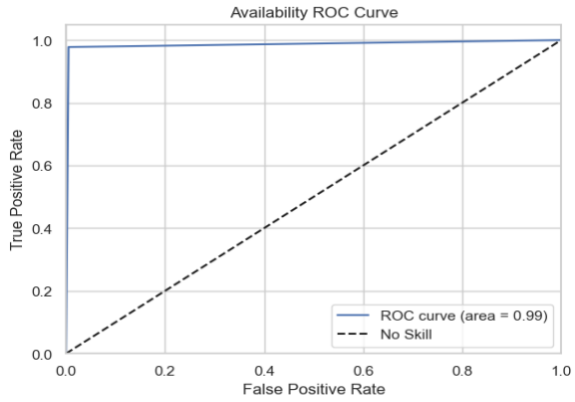


Figure 21

Integrity ROC Curve

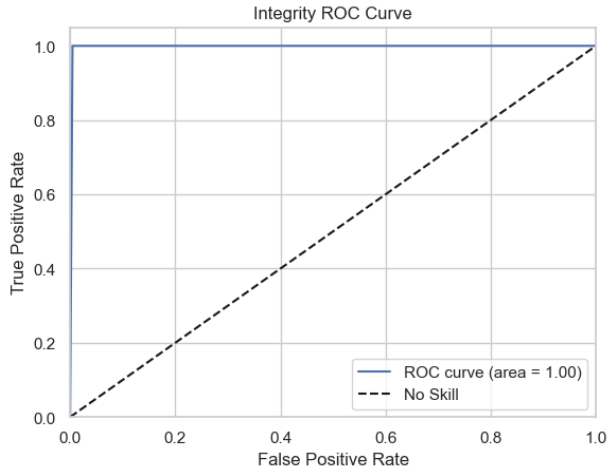


Figure 19, 20, and 21, for all datasets, reveal an Area Under the Curve of close to 1.0 and a distance from the no skill line.

4.4 Research Question Hypothesis Testing

RQ: What components of an ensemble ML model created on supervised machine

learning models and trained on the TON_IoT and SideChannel-3D data can detect a subset of attack vectors that can affect a 3D printer's confidentiality, integrity and availability (CIA) with a high truly malicious detection rate (recall)?

- **Hypothesis**: An Ensemble Stacking Model with a logistic regression meta-model will achieve a true malicious detection rate (recall) of 98.2% for a segment of attack vectors that can compromise a 3D printer's confidentiality, integrity, and availability.
    - **Null Hypothesis:** A Logistic Regression Stacking Model trained on TON_IoT and SideChannel-3D will have a TPR rate equal to 98.2%.
    - **Alternative Hypothesis:** A Logistic Regression Stacking Model trained on TON_IoT and SideChannel-3D will have a TPR rate not equal to 98.2%.
    - **Statistical Tests:** The following measure a statistically significant difference between the rate of True Positives and True Positives + False Negatives (TPR).
        - One-Sample Z-Test for Proportions (Parametric) (Statsmodels.stats.proportion.proportions_ztest, 2025)
- **Hypothesis Testing Inputs:**
    - Count (Average True Positives across all models and cv runs): **495.0**
    - NObs (Average True Positives+False Negatives for all models and cv runs): **500.0**
    - P0 (Comparison Probability): **0.982**
    - Alternative (direction of statistical test): **Two-Tailed**

- **One-Sample Z-Test for Proportions Results**
    - **Test Statistic:** 1.766
    - **p-Value:** 0.077
    - **Conclusion:** Since the p-value is 0.077, we fail to reject the null hypothesis that a Logistic Regression Stacking Model trained on TON_IoT and SideChannel-3D will have a TPR rate equal to 98.2%.

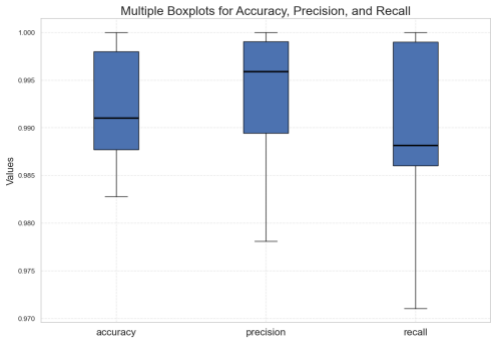
The mean true malicious detection rate (recall) as determined by this praxis, is 98.97%. This is greater than 98.2% but it is not statistically significantly different at the 0.05 level (p-value 0.077). Since the p-value is close to the 0.05 level, the praxis tested the one-tailed hypothesis test to assess if the recall was greater than 98.2%.

- **One-Sample Z-Test for Proportions Results (One-Tailed)**
    - **Test Statistic:** 1.766
    - **p-Value:** 0.0387
    - **Conclusion:** Since the p-value is 0.0387, we reject the null hypothesis that a Logistic Regression Stacking Model trained on TON_IoT and SideChannel-3D will have a TPR rate equal to 98.2%, concluding that the true malicious detection rate is greater than 98.2%.

Although RQ focuses on maximizing recall, it should not sacrifice accuracy or precision; this will be tested. An ANOVA analysis was conducted to assess if there is a statistically significant difference in the mean accuracy, precision, or recall.

Figure 22

Boxplots for Accuracy, Precision, and Recall for 5-fold CV



*Note the y-axis range 0.970 to 1.000

- **One-Way ANOVA Results**
    - Null Hypothesis: There is no statistically significant difference in the means for accuracy, precision, or recall
    - Alternative Hypothesis: There is a statistically significant difference in at least one of the means for accuracy, precision, or recall
    - Results
        - **F_statistic**: 1.309
        - **p-Value:** 0.280
    - Conclusion: The p-Value of 0.280 fails to reject the null hypothesis that there is no statistically significant difference in the means for accuracy precision and recall. Thus, this test along with the results of the one-sample z-test for proportions reveal that all metrics performed above 98.2%.

Chapter 5. Conclusion

5.1 Research Question Discussion

The Research question of this praxis addresses whether an ensemble model, comprising a selection of base and meta classifiers, can predict malicious instances with 98.2% recall or true malicious detection rate when trained and tested on the TON_IoT and SideChannel-3D datasets. An ensemble stacking model with base classifiers including Logistic Regression, Random Forest, and Histogram Gradient Boosting Classifier, in conjunction with a Logistic Regression meta model, was able to identify 98.97% of malicious instances correctly as malicious on average over 5 cross-validation runs. This mean recall rate of 98.97% and median 98.8% exceeds 98.2%, the average Recall or True Positive Rate (TPR) of comparable analyses presented in SideChannel-3D as well as TON_IoT sources (Yu et al., 2020; Booij et al., 2022; Alotaibi & Ilyas, 2023). The application of a One-Sample Z-test for proportions corroborates this assertion with a p-Value of 0.0387, stating that there is only a 3.87% chance that the mean true malicious detection rate is less than or equal to 98.2% given that it is true.

Of the actual malicious instances, 98.97% are identified as malicious (recall or true positive rate). Additionally, of the predicted malicious instances, more than 99.4% of them were actually malicious (precision) on average. Not only does the high model true positive rate (TPR)/recall allow for better detection capabilities, it also enables the government, private sector companies, and public sector companies alike to safeguard critical networked devices by identifying malicious attacks based on IoT network traffic and G-code segments compactly in one ensemble model – with performance recall comparable to similar research works. This praxis aims to minimize the rate at which

malicious instances are incorrectly flagged as normal instances, which can be detrimental to a company's infrastructure by leaving cyberattacks potentially undetected.

ROC Curves and Area Under the Curve (AUC) calculations in Figure 19-21 corroborate the claim of near-perfect classification of benign and malicious instances. Specific findings are:

(1) The curves climb the TPR (y-axis) to ~1.0 at a very low FPR (x-axis) before traversing the x-axis to ~1.0, showcasing a high Recall (True Positive Rate) and low False Positive Rate (FPR).

(2) AUC values of 0.98, 0.99, and 1.0 indicate that the model almost always classifies the label correctly.

(3) The curve is far from the "No skill" lines, displaying that the model has learned relevant information to predict beyond a random assignment of labels.

As mentioned, this research focuses on maximizing the True Positive Rate and, in turn, minimizing the False Negative Rate. Each false negative presents in one of two instances: 1) true malicious network events relating to cyber-attacks, predicted as normal events, and 2) tampered G-code segments, predicted as a normal G-code segment. False Negatives occurs 6.6 times on average in the confidentiality dataset across 5 runs, 8.4 times for availability, and 0.2 times for integrity. The performance is balanced, as the false negative count is under 10 for each element of the CIA triad.

Despite the training time of stacking models taking 6 times longer than voting or traditionally leveraged ensemble models (Random Forest and Histogram Gradient Boosting Classifier), they perform slightly better with values for accuracy, precision, and recall. Although the training time is comparatively longer and the performance is

marginally better for ensemble stacking models, Voting, Random Forest, and Gradient Boosting Classifier are viable alternatives within environments that generate and process large amounts of network flow and G-code data, especially in the context of model retraining.

Although this research question maximizes the true malicious detection rate (recall), a one-way ANOVA revealed that there is no statistically significant difference between the mean accuracy, precision, and recall of the analysis. As such, maximizing the true malicious detection rate (recall) does not sacrifice any of the other two performance metrics.

5.2 Conclusion

An ensemble machine learning model with base classifiers logistic regression, histogram gradient boosting machine, and random forest with a logistic regression meta model achieved a true malicious detection rate (TPR or recall) greater than the benchmark of 98.2% the average of three sources, (p-Value of 0.0387). Among these sources: a recall of 97.26% was observed when training a detector to detect attacks within TON_IoT using a random forest model (Booij et. al., 2022); when applying an ensemble stacking model to TON_IoT attack detection a recall of 98.6% was observed (Alotaibi & Ilyas, 2023); when applying gradient boosting and random forest to detecting adversarial inclusions using SideChannel-3D G-code segments a recall of 98.99% is observed (Yu et al., 2020). The recall observed in this praxis is 98.97%, better than the two of the three sources in the benchmark individually; and better than 98.2%, the average of the three.

Since this hypothesis was validated, the research question was answered revealing that an ensemble ML model created on supervised machine learning models and trained

on the TON_IoT and SideChannel-3D data can detect a subset of attack vectors that can affect a 3D printer's confidentiality, integrity and availability (CIA) with a high truly malicious detection rate (recall). Finally, this reveals that, an ensemble machine learning detection model based on IoT Attacks is a valuable capability to safeguard 3D printers against a segment of targeted attacks, ultimately preserving confidentiality, integrity, and availability (CIA).

5.3 Contributions to the Body of Knowledge

In addition to constructing a usable ensemble machine learning model to identify malicious attacks to a 3D printer's confidentiality, integrity, and availability achieving a recall or a true malicious detection rate of 98.97% -- this praxis addressed the following gaps in the research community:

- ***Filled a methodology gap in 3D printer attack detection by combining two established detection methods into a single ensemble stacking model*:** Extensive research into publicly available repositories i.e. IEEE, ResearchGate, Solid Freeform Fabrication Symposium, ASTM, Ma2JIC, etc. revealed a paucity of datasets and detection models despite a growing need for them (U.S. Department of Defense & Office of Inspector General, 2021; Shaik et al., 2023). This praxis introduces a heterogeneous pipeline and model to detect malicious attacks on both network traffic (ZEEK log) data and G-code segment data, specifically securing all elements of the CIA triad and targeting 3D printers housed on corporate and government networks. Alotaibi & Ilyas (2023) leverages an ensemble stacking model to detect malicious network attacks within the TON_IoT data while Yu et al., (2020) leverages machine learning detection to identify tampered g-code

segments. The ensemble model developed in this praxis combines both methodologies to detect malicious behavior from network traffic logs from industry endpoints, preserving the confidentiality and availability of the system, and/or tampered G-Code segments preserving the integrity.

- ***Filled a theory gap in G-Code segment features by generating additional representative kinematic physics-based features*:** This praxis enhanced the segment.csv features identified by Yu et al., (2020) by introducing physics-based features beyond axes position and velocity to detail the movement of a filament extruder over time. Although Yu et al., (2020) leverages physics-based approaches for side-channel use cases, this praxis allows researchers to produce additional kinematic physics-based features such as speed, distance, rotation, and angle between segments expounded in section 3.3 and 3.6 to further study: the SideChannel-3D segments.csv dataset, self-generated G-Code segments, or additional G-Code files. An axes normalization process standardizes the data to be appropriately used for scalable model usage.
- ***Filled a knowledge and evidence gap in networked 3D printers as IoT devices by mapping characteristics to IoT properties***: 3D printers are referred to as IoT devices in prior literature (Prianto et al., 2021; Wu et al., 2016) without detailing the inherent characteristics of networked 3D printers that align with IoT devices. This praxis establishes an evidence-based mapping between the properties of IoT devices and their application to 3D printers connected to a network. The mapping, presented in section 3.1, showed that 3D printers are IoT devices as they (1) possess a Unique IP Address either natively or via OctoPrint ("Download & Setup

OctoPrint," 2025) (2) send and receive network traffic to and from other devices on a network (3) can be remotely monitored or controlled by web-based interfaces like OctoPrint ("Download & Setup OctoPrint," 2025).

- ***Filled a theory gap in the mapping of 3D printer attack vectors to (1) CIA elements and (2) 3D printer/component states by constructing the attack vector schematic (in Figure 2)***: This praxis synthesizes attack vectors presented in prior literature and has constructed a matrix that outlines the attack vector's impact on confidentiality, integrity, and/or availability and showcasing where in the 3D printing process (storage, transmission, and or processing) these attack vectors pose risks. Figure 2 also provides guidance for future researchers to build timeless detection and heuristic capabilities to safeguard networked 3D printers.
- ***Filled a methodology gap in IoT attack detection using TON_IoT by applying 3D printing attack-vector-specific preprocessing***: This praxis contributes to the growing body of research on IoT attack detection, specifically utilizing the TON_IoT dataset while safeguarding physical integrity against attacks based on G-code segments. TON_IoT was subsetted to include Scanning, MitM, and DoS attacks along with Normal behavior to represent the attack vectors that can compromise 3D printers on a network. A sample of Normal, Scanning, and MitM attacks constituted the confidentiality training data. A sample of Normal and DoS attacks constituted the availability training data. The average confidentiality and availability model accuracy of 98.8% determined in this praxis emerged as a contender with Alotaibi & Ilyas (2023), which shows an accuracy of 98.63%

while also leveraging an ensemble stacking model with base models DT, RF, KNN, LR and meta model LR for detection.

5.4 Future Works

The following are efforts that would further benefit the space of Additive Manufacturing Security. Limitations of this praxis work include a lack of access to larger, high-complexity networks, a lack of a cloud environment for faster processing, and no testbed validation. Provided below are avenues for future research that will augment this praxis and further advance the field.

Investigate web-based cyber intrusions and firmware attacks

Encompassed in the TON_IoT data sets are a series of web attacks, including Backdoor Vulnerabilities, Injections, Cross-Site Scripting, and Ransomware attacks. Further study and detection of web-based cyber intrusions should be explored, particularly as services like OctoPrint ("Download & Setup OctoPrint," 2025) are used to monitor prints across all disparate sectors. Cyber-attacks also impact the firmware of 3D printers. The Flaw3D Bootloader is a Trojan horse affecting Marlin firmware 3D printers; these attacks allow adversaries to remove and relocate extrusion on boot. An investigation and reproduction of this Trojan Horse would secure additional attack vectors.

Apply a mitigation procedure for these attacks using the MITRE ATT&CK framework

A cyber analyst can translate detection procedures presented in this praxis into deterrence and prevention measures observed in private sector cybersecurity frameworks, such as MITRE ATT&CK and the Lockheed Martin Cyber Kill Chain. Attack behaviors pair with a particular procedure for identification, analysis, and response. This will help

safeguard the DoD, contracting agencies, and third-party networks that hold and execute government work.

Explore additional feature generation

Feature generation is a widely implemented component of the data science life cycle, representing and highlighting specific aspects of data to optimize modeling and enhance interpretability. An example of feature generation is edge generation, represented as a binary variable, i.e., identifying whether a segment touches an edge of the 3D printed object or not. An additional feature to explore for better identifying scanning and denial-of-service attacks is converting the number of ports per second or bytes per second into its own feature for each pair of source and destination IP addresses. These features, and those similar to them, will better illuminate the specific heuristics expected for certain reconnaissance and flooding attacks.

Validate model with Testbed

Following this praxis, data should be generated using a physical testbed that contains networked 3D printers, validating the model's performance with a live data feed. Testbeds can advance research into 3D printers and other Industry 4.0 devices.

References

Al Faruque, M. A., Chhetri, S. R., Canedo, A., & Wan, J. (2016). Acoustic side-channel attacks on Additive Manufacturing systems. 2016 ACM/IEEE 7th International Conference on Cyber-Physical Systems (ICCPS). doi:10.1109/ICCPS.2016.7479068

Alotaibi, Y., & Ilyas, M. (2023). Ensemble-Learning Framework for Intrusion Detection to Enhance Internet of Things' Devices Security. Sensors, 23(12), 5568. doi:10.3390/s23125568

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A & Anwar, A. (2020). TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems. IEEE Access 8, 10. doi: 1109/ACCESS.2020.3022862

Ashraf, J., Keshk, M., Moustafa, N., Abdel-Basset, M., Khurshid, H., Bakhshi, A. D., & Mostafa, R. R. (2021). IoTBoT-IDS: A novel statistical learning-enabled botnet detection framework for protecting networks of smart cities. Sustainable Cities and Society, 72. doi:10.1016/j.scs.2021.103041

Beckwith, C., Naicker, H. S., Mehta, S., Udupa, V. R., Nim, N. T., Gadre, V., Pearce, H. Mac, G., Gupta, Nikhil. (2022). Needle in a Haystack: Detecting Subtle Malicious Edits to Additive Manufacturing G-code Files. IEEE Embedded Systems Letters, doi:10.1109/LES.2021.3129108

Bindra, N., & Sood, M. (2020). Evaluating the Impact of Feature Selection Methods on the Performance of the Machine Learning Models in Detecting DDoS Attacks. Romanian Journal of Information Science and Technology.

Bold, R., Al-Khateeb, H., & Ersotelos, N. (2022). Reducing False Negatives in Ransomware Detection: A Critical Evaluation of Machine Learning Algorithms. Applied Sciences, 12(24). doi:10.3390/app122412941

Booij, T. M., Chiscop, I., Meeuwissen, E., Moustafa, N., & Hartog, F. T. H. d. (2022). ToN_IoT: The Role of Heterogeneity and the Need for Standardization of Features and Attack Types in IoT Network Intrusion Data Sets. IEEE Internet of Things Journal, 9(1), 485–496. doi:10.1109/JIOT.2021.3085194

Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition, 30(7), 1145–1159. doi:10.1016/S0031-3203(96)00142-2

Breiman, L. (2001). Random Forests. Machine Learning 45, 5–32. https://doi.org/10.1023/A:1010933404324

Cao, Z., Zhao, Z., Shang, W. et al. Using the TON_IoT dataset to develop a new intrusion detection system for industrial IoT devices. Multimed Tools Appl 84, 16425–16453 (2025). https://doi.org/10.1007/s11042-024-19695-7

Costa, N., Yu, S., Y., Malawade, A., Chhetri S., & Mohammad, A. F. (2021). SideChannel-3D: Acoustic, vibration, magnetic, and power side-channel 3D printer dataset. IEEE Data Port. doi:10.21227/j6cw-y314

CyManII. (2025). CyManII. https://cymanii.org/

Darweesh, M., Abdelaziz, M., Radwan, A., Mousa, H., Saad, A., Abuzaid, A., Abdulrahman, A., Zakzouk S., & Moussa, K., (2024). Random Forest-Based NIDS: Advancing Network Threat Detection. doi:10.21203/rs.3.rs-4737281/v1

Devarakonda, A., Sharma, N., Saha, P., & Lokesh, R. (2022). Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. Journal of Physics: Conference Series, 2161, 012043. doi:10.1088/1742-6596/2161/1/012043

Dube, R. (2022). Faulty use of the CIC-IDS 2017 Dataset in Information Security Research. doi:10.13140/RG.2.2.25435.64809

Encord. (2025). What is a confusion matrix? Machine learning glossary: Encord. https://encord.com/glossary/confusion-matrix/

Garcia, S., Parmisano, A., & Erquiaga, J. E. (2020). IoT-23: A labeled dataset with malicious and benign IoT network traffic (version 1.0.0) [data set]. Zenodo. doi:10.5281/zenodo.4743746

Ghourabi, A. (2022). A Security Model Based on LightGBM and Transformer to Protect Healthcare Systems From Cyberattacks. IEEE Access 10, 48890–48903.

doi:10.1109/ACCESS.2022.3172432

Grassi, P. A., Garcia, M., & Fenton, J. L., (2017). NIST Special Publication 800-63-3. doi: 10.6028/NIST.SP.800-63-3

Hashemi, S., & Zarei, M. (2021). Internet of things backdoors: Resource management issues, security challenges, and detection methods. Transactions on Emerging Telecommunications Technologies, 25. doi:10.1002/ett.4142

He, Z., Lin, D., Lau, T., & Wu, M. (2019). Gradient boosting machine: A survey. doi:10.48550/arXiv.1908.06951

Horizon Grand View Research. (2023). Global 3D Printing Construction Market Size & Outlook. Grand View Research. https://www.grandviewresearch.com/horizon/outlook/3d-printing-construction-market-size/global

Ishfaq, K., Khan, M., Khan, M., Mahmood, M. A., & Maqsood, M. A. (2023). A correlation among industry 4.0, Additive Manufacturing, and topology optimization: A state-of-the-art review. The International Journal of Advanced Manufacturing Technology, 129. doi:10.1007/s00170-023-12515-6

Kandhro, I. A., Alanazi S. M., Ali F., Kehar, A., Fatima K., Uddin M., & Karuppayah S. (2023). Detection of Real-Time Malicious Intrusions and Attacks in IoT Empowered Cybersecurity Infrastructures. IEEE Access. 10.1109/ACCESS.2023.3238664

Kang, H., Ahn, D. H., Lee, G. M., Yoo, J. D., Park K. H., & Huy, K. K. (2019). IoT network intrusion dataset. IEEE Dataport. https://dx.doi.org/10.21227/q70p-q449

Lanvin, M., Gimenez, P., Han, Y., Majorczyk, F., Mé, L., & Totel, É. (2022). Errors in the CICIDS2017 Dataset and the Significant Differences in Detection Performances It Makes. In Risks and Security of Internet and Systems: 17th International Conference, CRiSIS 2022. https://doi.org/10.1007/978-3-031-31108-6_2

Leevy, J., Hancock, J., Richard, Z., & Khoshgoftaar, T. (2020). Detecting Cybersecurity Attacks

Using Different Network Features with LightGBM and XGBoost Learners. 2020 IEEE Second International Conference on Cognitive Machine Intelligence (CogMI), 190-197. doi: 10.1109/CogMI50398.2020.00032

Li, S., Freije, E., & Pe, P. Y. (2017). Monitoring 3D Printer Performance using Internet of Things (IoT) Application. 2017 ASEE Annual Conference & Exposition. https://peer.asee.org/monitoring-3d-printer-performance-using-internet-of-things-iot-application

Li, Z., Gong, D., Tan, L., Luo, X. Liu, F. (2021) Self-embedding watermarking method for G-code used in 3D printing. IEEE International Workshop on Information Forensics and Security (WIFS). https://doi.org/10.1109/WIFS53200.2021.9648386

Lupi, F., Cimino, M. G. C. A., Berlec, T., Galatolo, F. A., Corn, M., Rožman, N., Rossi, A., Lanzetta, M. (2023). Blockchain-based Shared Additive Manufacturing. Computers & Industrial Engineering, 183, 109497. doi:10.1016/j.cie.2023.109497

McCormack, M., Chandrasekaran, S., Liu, G., Yu, T., Wolf, S. D. & Sekar, V. (2020). Security Analysis of Networked 3D Printers. 2020 IEEE Security and Privacy Workshops (SPW), pp. 118–125. doi:10.1109/SPW50608.2020.00035

Mccormack, M., Chandrasekaran, S., Yu, T., Wolf, S., & Sekar, V. (2021). C3PO: A Security Analysis Tool for Networked 3D Printers (CMU-CyLab-19-002) (Version 1). Carnegie Mellon University. https://doi.org/10.1184/R1/16694896.v1

Mitchell, O., Osazuwa, M., Msc, Cpss, Css, & Fty. (2023). Confidentiality, Integrity, and Availability in Network Systems: A Review of Related Literature. doi:10.5281/zenodo.10464076

Moustafa, N. (2019). A Systemic IoT-Fog-Cloud Architecture for Big-Data Analytics and Cyber Security Systems: A Review of Fog Computing. arXiv Preprint https://arxiv.org/abs/1906.01055

Moustafa, N. (2021). A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustainable Cities and Society, 72, 102994. doi:10.1016/j.scs.2021.102994

Moustafa, N. New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets. Proceedings of the eResearch Australasia Conference.

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). 2015 Military Communications and Information Systems Conference (MilCIS), pp. 1–6. doi:10.1109/MilCIS.2015.7348942

Moustafa, N., Ahmed, M., & Ahmed, S., (2020a). Data analytics-enabled intrusion detection: Evaluations of ToN_IoT linux datasets. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 727–735. doi:10.1109/TrustCom50675.2020.00100

Moustafa, N., Keshky, M., Debiez, E., & Janicke, H.,. (2020b). Federated TON_IoT Windows Datasets for Evaluating AI-Based Security Applications. 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), pp. 848–855. doi:10.1109/TrustCom50675.2020.00114

Nayak G. N. & Samaddar S. G. (2010). Different flavours of man-in-the-middle attack, consequences and feasible solutions. doi:10.1109/ICCSIT.2010.5563900

Nhat-Duc, H., & Van-Duc, T. (2023). Comparison of histogram-based gradient boosting classification machine, random forest, and deep convolutional neural network for pavement raveling severity classification. Automation in Construction, 148, 104767. doi:10.1016/j.autcon.2023.104767

Nieles, M., Dempsey, K., & Yan Pillitteri, V. (2017). NIST special publication 800-12. An Introduction to Information Security. https://doi.org/10.6028/NIST.SP.800-12r1

Oladimeji, S., & Kerner, S. M. (2023). SolarWinds hack explained: Everything you need to know. Tech Targets. https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

OpenJS Foundation & Contributors. Node-RED.https://nodered.org

Pandas (2025). Pandas. https://pandas.pydata.org

Paul, B., & Rao, M. (2022). Zero-trust model for smart manufacturing industry. Applied Sciences, 13, 221. doi:10.3390/app13010221

Pearce, H., Yanamandra, K., Gupta, N., & Karri, R. (2022). FLAW3D: A Trojan-Based Cyber Attack on the Physical Outcomes of Additive Manufacturing. IEEE/ASME Transactions on Mechatronics, 27(6), 5361–5370. doi:10.1109/TMECH.2022.3179713

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dunbourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Bruncher, M., Perrot, M., Duchesnay, E. (2011). Scikit-learn: Machine learning in python. Journal of Machine Learning Research, 12, 2825–2830.

Prianto, E., Pramono, H. S., & Yuchofif. (2021, November). IoT-Based 3D Printer Development for Student Competence Improvement. *Journal of Physics: Conference Series*, 2111(1), Article 012002. https://doi.org/10.1088/1742-6596/2111/1/012002

Ravi, N., Dandekar, N., Mysore, P., & Littman, M. (2005). Activity recognition from accelerometer data. AAAI, 3, 1541-1546.

Riikonen, N., Piili, H., Salminen, A., & Nyrhilä, O. (2020). Additive Manufacturing—Past, Present, and the Future. doi:10.1007/978-3-030-46103-4_2

Shah, A., Aliyev, R., Zeidler, H., & Krinke, S. (2023). A Review of the Recent Developments and Challenges in Wire Arc Additive Manufacturing (WAAM) Process. Journal of Manufacturing and Materials Processing, 7(3), 97. https://doi.org/10.3390/jmmp7030097

Shaik, S., Tunc, C., & Morozov, K. (2023). Intrusion detection for additive manufacturing systems and networks. In 2023 20th ACS/IEEE International Conference on Computer Systems and Applications (AICCSA) (pp. 1–3). IEEE. https://doi.org/10.1109/AICCSA59173.2023.10479327

Sharafaldin, I., Lashkari, A. H., Hakak, S., & Ghorbani, A. (2019). Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy. IEEE 53rd International Carnahan Conference on Security Technology. doi: 10.1109/CCST.2019.8888419.

Singh, S., Singh, M., & Singh, D. (2011). A Survey on Network Security and Attack Defense Mechanism For Wireless Sensor Networks

Šlajs, T., Klaban, I., & František, P. (2023). Cyber-Attack Vectors Focused on Additive Manufacturing. doi:10.1109/KIT59097.2023.10297115

Srivastava, M., Rathee, S., Patel, V., Kumar, A., & Koppad, P. G. (2022). A review of various materials for Additive Manufacturing: Recent trends and processing issues. Journal of Materials Research and Technology, 21, 2612–2641. doi:10.1016/j.jmrt.2022.10.015

Statsmodels.stats.proportion.proportions_ztest, (2025). Stats Models. https://www.statsmodels.org/dev/generated/statsmodels.stats.proportion.proportions_ztest.html

Tamir, T. S., Xiong, G., Shen, Z., Leng, J., Fang, Q., Yang, Y., Jiang, J., Lodhi, E., & Wang, F. Y. (2023). 3D printing in materials manufacturing industry: A realm of industry 4.0. Heliyon, 9(9), e19689. doi:10.1016/j.heliyon.2023.e19689

U.S. Department of Defense & Office of Inspector General. (2021). Audit of the cybersecurity of department of defense Additive Manufacturing systems. Defense. https://media.defense.gov/2021/Jul/07/2002757308/-1/-1/1/DODIG-2021-098.PDF

University of New Brunswick. (2019a). Canadian institute for cybersecurity. https://www.unb.ca/cic/

University of New Brunswick. (2019b). DDoS evaluation dataset (CIC DDoS2019). https://www.unb.ca/cic/datasets/ddos-2019.html

Uttley, J. (2019). Power Analysis, Sample Size, and Assessment of Statistical Assumptions—Improving the Evidential Value of Lighting Research. LEUKOS, 15(2-3), 143–162. doi:10.1080/15502724.2018.1533851

Wohlers, T. T., Campbell, I., Diegel, O., Huff, R., Kowen, J., & Wohlers Associates (Firm). (2023). Wohlers report 2023: 3D printing and Additive Manufacturing global state of the industry. https://wohlersassociates.com/product/wr2023/

Wu, J. J., Tan, Y. G., & Ma, G. F. (2016). 3D printing monitoring platform based on the Internet of Things. In Proceedings of the Fifth Asia International Symposium on Mechatronics (AISM 2015). Institution of Engineering and Technology. https://doi.org/10.1049/cp.2015.1569

Xie, Y. (2023). Machine learning-based DDoS detection for IoT networks. Applied and Computational Engineering, 29, 99–107. doi:10.54254/2755-2721/29/20230972

Yu, S., Malawade, A. V., Chhetri, S. R., & Al Faruque, M. A. (2020). Sabotage Attack Detection for Additive Manufacturing Systems. IEEE Access, 8, 27218–27231. doi:10.1109/ACCESS.2020.2971947

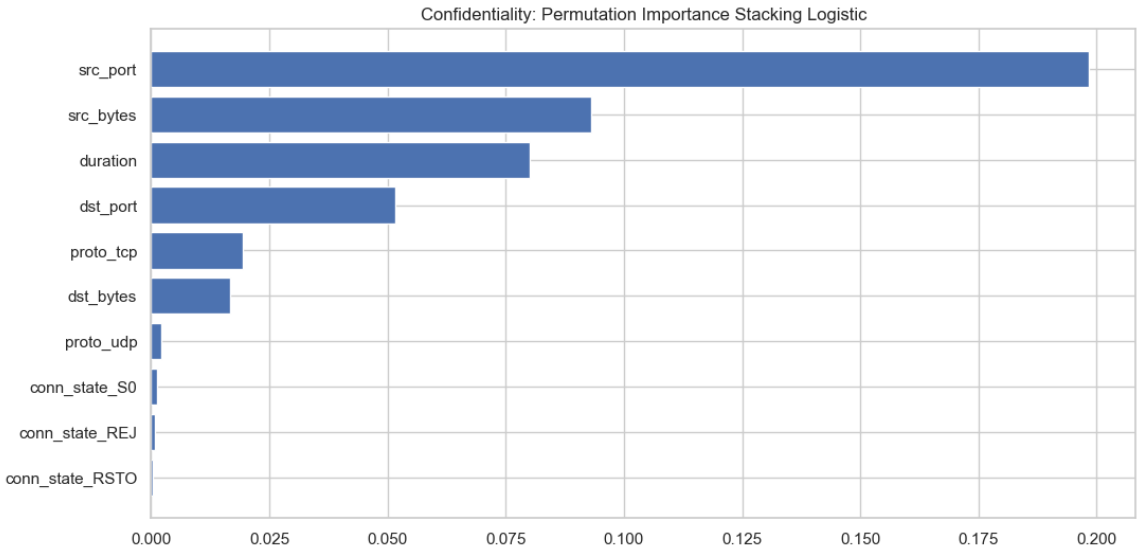
Appendix

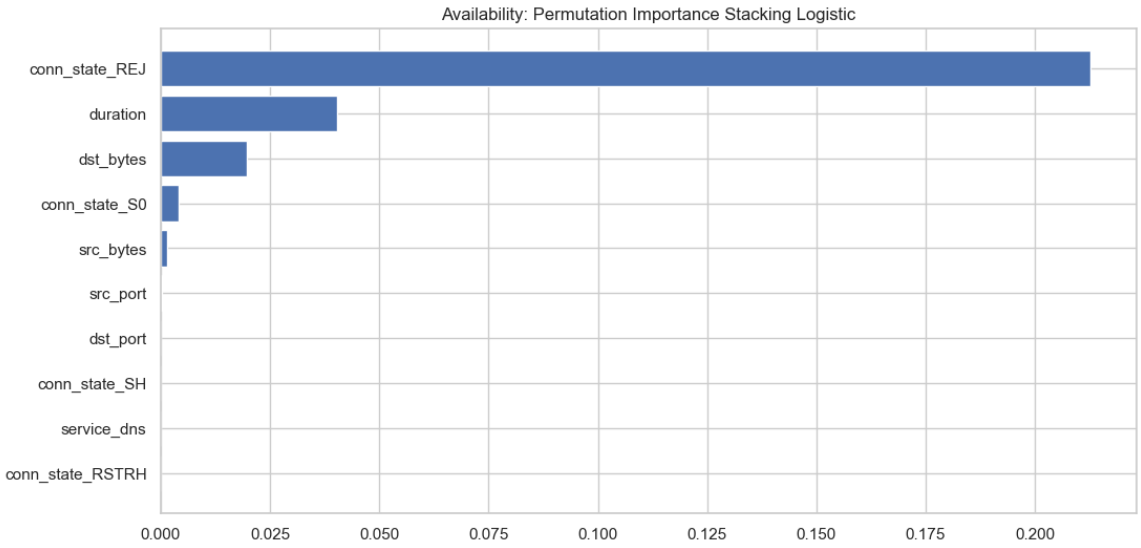
Code for Ensemble Modeling

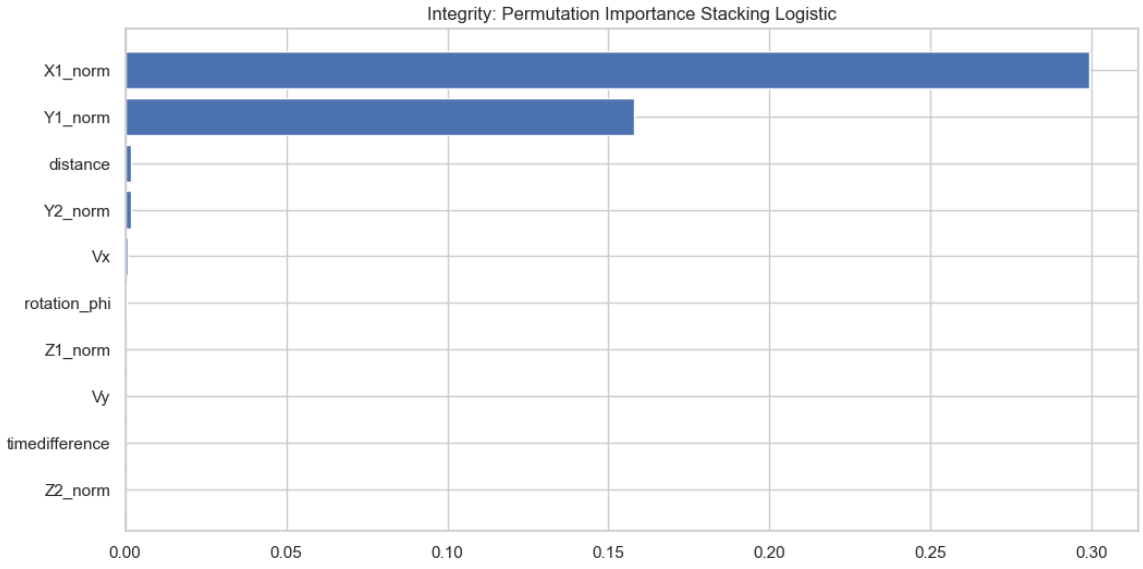
```
from sklearn.linear_model import LogisticRegression
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import HistGradientBoostingClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.metrics import confusion_matrix
from sklearn.metrics import accuracy_score, precision_score, recall_score

import pandas as pd
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
import numpy as np


import warnings
warnings.filterwarnings("ignore")

############ P6: Aggregation
def establish_estimators(seed):
    estimators = [('rf', RandomForestClassifier(n_estimators=10, random_state=seed)),
                  ('hgb',HistGradientBoostingClassifier(random_state=seed)),
                  ('lr',LogisticRegression(solver='lbfgs', max_iter=1000, random_state=seed))]
    return estimators

def evaluator(y_true,y_pred):
    # Calculate accuracy
    accuracy = accuracy_score(y_true, y_pred)
    print("Accuracy:", accuracy)
    # Calculate precision
    precision = precision_score(y_true, y_pred)
    print("Precision:", precision)
    # Calculate recall
    recall = recall_score(y_true, y_pred)
    print("Recall:", recall)
    return [accuracy, precision, recall]

def rebalancer_ensemble(df, sd):
    if set(df['label']) == {'Malicious','Normal'}:
        df['label'] = df['label'].replace({'Malicious': 1.0, 'Normal': 0.0})
    df_0 = df[df['label'] == 0.0]
    df_1 = df[df['label'] == 1.0]
    ratio = (df_1.shape[0]-df_0.shape[0])/(df_0.shape[0])
    print(f"ratio: {ratio}")
    if abs(ratio) >= 0.2:
```

```
        df_0_samp = df_0.sample(n=2500, random_state=sd)
        df_1_samp = df_1.sample(n=2500, random_state=sd)
        df_new = pd.concat([df_0_samp,df_1_samp], axis = 0)
    else:
        df_new = df
    return df_new

def model_setup_ensemble(df):
    df_model = df.dropna()
    df_y = df_model['label']
    df_X = df_model.drop(['label'], axis=1)
    df_X_train, df_X_test, df_y_train, df_y_test = train_test_split(df_X, df_y)
    return df_X_train, df_y_train, df_X_test, df_y_test

def stacking_logistic(X_train, y_train, X_test, y_test, estimators):
    print("### stacking logistics")
    ens = StackingClassifier(estimators=estimators,
final_estimator=LogisticRegression(solver='lbfgs', max_iter=1000))
    ens.fit(X_train, y_train)
    ens_y_pred = ens.predict(X_test)
    tn, fp, fn, tp = confusion_matrix(y_test, ens_y_pred).ravel()
    print(confusion_matrix(y_test, ens_y_pred).ravel())
    print(f"{tp + fn} = {tp}")

if __name__ == "__main__":
    df_N = rebalancer(dataframe, seed)
    df_X_train, df_y_train, df_X_test, df_y_test = model_setup(df_N)
    logreg = LogisticRegression(random_state=seed, solver='lbfgs', max_iter=1000)
    logreg.fit(df_X_train, df_y_train)
    logreg_y_pred = logreg.predict(df_X_test)
    print(confusion_matrix(df_y_test, logreg_y_pred).ravel())
    print(logreg.classes_)
    conf_matrix = confusion_matrix(df_y_test, logreg_y_pred, labels = logreg.classes_)
    disp = ConfusionMatrixDisplay(confusion_matrix=conf_matrix,
display_labels=logreg.classes_)
```

Permutation Importance







Model 5-fold CV Information

Data	TN	FP	FN	TP	runtime	Train/test sizes	TP_FN	recall	precision	accuracy
confidentiality	619	13	8	581	18.39	(4883, 1221)	589	0.986	0.9781	0.983
confidentiality	585	6	7	623	18.39	(4883, 1221)	630	0.989	0.990	0.989
confidentiality	589	8	8	616	18.39	(4883, 1221)	624	0.987	0.987	0.987
confidentiality	607	7	10	597	18.39	(4883, 1221)	607	0.984	0.988	0.986
confidentiality	611	7	7	595	18.39	(4884, 1220)	602	0.988	0.988	0.989
availability	503	2	8	487	18.23	(4000, 1000)	495	0.984	0.996	0.99
availability	481	1	15	503	18.23	(4000, 1000)	518	0.971	0.998	0.984
availability	512	2	7	479	18.23	(4000, 1000)	486	0.986	0.996	0.991
availability	490	3	6	501	18.23	(4000, 1000)	507	0.988	0.994	0.991
availability	506	0	6	488	18.23	(4000, 1000)	494	0.988	1	0.994
integrity	503	2	0	495	17.59	(4000, 1000)	495	1	0.996	0.998
integrity	506	1	1	492	17.59	(4000, 1000)	493	0.998	0.998	0.998
integrity	497	0	0	503	17.59	(4000, 1000)	503	1	1	1
integrity	499	0	0	501	17.59	(4000, 1000)	501	1	1	1
integrity	492	0	0	508	17.59	(4000, 1000)	508	1	1	1

*Confidentiality data was scaled to (4000, 1000) to not imbalance the hypothesis test