To the Graduate Council:
I am submitting herewith a dissertation written by Tyler Andrew Cultice entitled "Cyber-Physical System Security in a Quantum-Enabled Era." I have examined the final electronic copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Computer Engineering.

Himanshu Thapliyal, Major Professor

We have read this dissertation
and recommend its acceptance:

Himanshu Thapliyal

Qing Cao

Jinyuan Sun

Annarita Giani

Accepted for the Council:

Marieke Van Puymbroeck

Vice Provost and Dean of the Graduate School

(Original signatures are on file with official student records.)

Cyber-Physical System Security in a Quantum-Enabled Era

A Dissertation Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Tyler Andrew Cultice

August 2025

© by Tyler Andrew Cultice, 2025
All Rights Reserved.

To my ever-supportive parents, whose patience and unwavering love lit every step of this path, and to my brother, whose example and encouragement gave me someone to follow and something to strive toward — this dissertation is dedicated with heartfelt gratitude.

Acknowledgements

I have been fortunate to receive incredible support throughout my graduate school journey—support that played a crucial role in both my successes and in helping me rise from setbacks.

First and foremost, I express my deepest gratitude to my advisor and friend, Dr. Himanshu Thapliyal, who invited me to the University of Tennessee and fostered my growth as both a researcher and a teacher. His mentorship has been foundational to my development.

I am also deeply thankful to my lab mates, Milad Nasab, JC Chin, Bhaskar Gaur, Wu Yang, Joseph Clark, Sounak Bhowmik, and Saif Onim, for making our lab an environment of collaboration, innovation, and productivity. Many of my publications include these talented individuals as co-authors, and their insight and creativity helped shape new ideas and applications in CPS security.

My family has played an instrumental role in shaping who I am today. The love and support of my mother, Constance Cultice; father, Steven Cultice; and aunt, Kathleen DeLong, instilled in me a drive for knowledge, practicality, and creative problem-solving. I am also grateful to my brother, Brian Cultice, whose example and strong leadership continues to inspire me as a powerful role model. I consider him one of the best people and researchers I know.

I would be remiss not to thank my greatest source of support and motivation, my wife, Jessica Cultice. Her presence throughout this journey has meant more than I can express. This dissertation is as much hers as it is mine. From patiently listening to

hours of brainstorming, to reminding me to rest, to believing in me when I struggled to believe in myself—her encouragement has been invaluable.

Lastly, I am grateful for the friendships that have sustained me through this process. From childhood friends to colleagues and friends I met along the way, your support in both the highs and lows of graduate life has made this journey more meaningful. And finally, to all the professors, friends, collaborators, and loved ones who played a role in this achievement: thank you. I dedicate this dissertation to each of you.

Abstract

The rapidly growing field of quantum computers has brought forth a revolution in the world of computing for the past decade. However, with this comes immense potential for change that breaks down the very foundation of what was previously considered computationally infeasible. In a quantum-enabled world, the fundamentals of cybersecurity change immensely, from now-unsafe cryptography to quantum-secured data. This work aims to investigate both by harnessing the power of quantum computing in one of the most high-risk, vulnerable domains in modern society: cyber-physical systems. To achieve this, we take a look at the taxonomy and structure of a typical CPS ecosystem to determine the potential weaknesses and gains from quantum computing.

First, we analyze the physical-level security from the perspective of securing Controller Area Network-based (CAN) electronic control units, essential components of modern vehicles and machinery, to develop an efficient, quantum-safe framework for trusted device authentication. Our authentication scheme featuring post-quantum cryptography (PQC) uses 4× fewer messages than leading vehicular security frameworks. Next, we extended this work to inter-device CPS communication in the context of additive manufacturing networks using the Controller Area Network (CAN) bus. Our proposed design provides quantum-safe, root-of-trust tree networks to 3D printing farm networks, allowing plug-and-play connections without fear of trojan-horse or man-in-the-middle attacks. Using 12× fewer messages for authentication compared to other PQC systems, our framework defends against most forms of

man-in-the-middle attacks and effectively mitigates most denial-of-service attacks. Furthermore, we created a supervisory-level anomaly detection model utilizing quantum fidelity kernel support vector machines, providing up to a 13% boost in classification performance of real-world industrial control system data and an 11% boost in 3D printer anomaly data. Utilizing the robust geometric distance metric, we found a clear, quantifiable advantage to quantum machine learning for securing CPS data and feedback systems. Finally, we discuss the future directions of research that could build upon this multi-level framework. Our work successfully bridges the gap between quantum computing in cyber-physical systems, both as a potential threat and strength to the future of critical infrastructure, supply chain trust, and manufacturing safety.

Table of Contents

1 Introduction **1**

1.1 Outline 6

2 Background **7**

2.1 Cyber-Physical Systems 7

2.1.1 Vulnerabilities of the CPS Hierarchy 8

2.1.2 Controller Area Network (CAN) 10

2.1.3 3D Printing (Additive Manufacturing) 13

2.2 Quantum Computing 16

2.2.1 Quantum Gates 17

2.2.2 Embedding Classical Data to Quantum 19

2.2.3 Noisy Intermediate-Scale Quantum (NISQ) Computers 19

2.3 Quantum Computing as an Attacker 20

2.3.1 Post-Quantum Cryptography 21

2.4 Quantum Computing as an Attack Detector 22

2.4.1 Quantum-Hybrid Support Vector Machines 23

2.4.2 Quantum Fidelity Kernels 24

3 Quantum-Safe Intra-Vehicular Communication **26**

3.1 Motivation 27

3.1.1 Contribution 28

3.2 Background 28

3.2.1 Post Quantum Cryptography . . . 30
3.2.2 Related Work . . . 30
3.3 Proposed PUF-Based CAN-FD Post Quantum Framework . . . 32
3.3.1 Design . . . 32
3.3.2 Post-Quantum Cryptography . . . 36
3.4 Security Analysis . . . 36
3.4.1 Attack Protection and Security . . . 37
3.4.2 Post-Quantum Authentication . . . 39
3.4.3 Area and Computational Cost . . . 39
3.5 Discussion . . . 41
3.5.1 Replacing Supersingular Isogeny Diffie-Hellman . . . 41
3.5.2 External and Inter-Network CAN Communication . . . 44
3.6 Conclusions and Future Work . . . 46

4 Quantum-Safe Manufacturing Networks **48**
4.1 Visions and Goals for a Secure 3D Printing Framework . . . 53
4.2 Background . . . 58
4.2.1 Vulnerability Analysis of CAN 3D Printers . . . 58
4.2.2 Prior Related Work in CAN and 3D Printing Security . . . 58
4.3 Design of Our Proposed Hierarchical CAN Framework . . . 60
4.3.1 Discovery and Addressing . . . 63
4.3.2 Authentication and Cryptography Establishment . . . 64
4.3.3 Normal Communication . . . 68
4.4 Analysis and Discussion of Proposed Framework . . . 71
4.4.1 Normal Communication and Routing . . . 73
4.4.2 Post-Quantum Cryptographic Schemes . . . 73
4.4.3 Threat Mitigation . . . 75
4.4.4 Comparison to Existing Security Frameworks . . . 77
4.5 Experimentation Results . . . 79

4.5.1 Authentication . . . 81
4.5.2 Normal Communication Latency . . . 83
4.6 Discussion and Concluding Remarks . . . 83

5 Quantum Anomaly Detection in Industrial Control Systems 85
5.1 Background . . . 88
5.1.1 Classical Anomaly Detection . . . 89
5.1.2 One-Class Support Vector Machines . . . 89
5.1.3 Kernel Alignment as a Similarity Metric . . . 91
5.1.4 Related Works in QML-based Anomaly Detection . . . 91
5.2 Cyber-physical System Datasets . . . 92
5.2.1 iTrust Secure Water Treatment . . . 94
5.2.2 HIL-based Augmented ICS . . . 94
5.2.3 iTrust Water Distribution . . . 96
5.2.4 Data Visualization and UMAP Projection . . . 97
5.3 Methodology . . . 99
5.3.1 Data pre-processing and Feature Reduction . . . 101
5.3.2 Quantum Fidelity Kernel . . . 102
5.4 Experimentation Results . . . 105
5.4.1 Testing Setup . . . 105
5.4.2 Dataset Results: SWaT . . . 105
5.4.3 Dataset Results: HAI . . . 107
5.4.4 Dataset Results: WADI . . . 109
5.4.5 Classical Results . . . 109
5.5 Quantum Error and Advantage Metrics . . . 112
5.5.1 Noise Effects in Kernel Error . . . 115
5.5.2 Identifying a Quantum Advantage . . . 117
5.6 Cyberphysical Monitor for 3D Printing . . . 119

5.6.1 Proposed Conceptualization of a Cyberphysical Monitor for 3D Printing . . . 119
5.6.2 3D Printing Cyberattack Testing Setup . . . 120
5.6.3 Cyberattack Dataset . . . 122
5.6.4 Data Processing . . . 125
5.6.5 Experimental Results . . . 125
5.7 Discussion, Conclusion, and Future Work . . . 128
5.7.1 Discussion . . . 130
5.7.2 Conclusion . . . 130

6 Conclusions and the Future of Quantum CPS Security Research 134
6.1 Future Work . . . 136
6.1.1 Device Fingerprinting and Key Material Generation . . . 136
6.1.2 Privacy Preservation in Post-Quantum CPS Frameworks . . . 137
6.1.3 Anomaly Detection using Quantum Machine Learning . . . 137

Bibliography 139

Vita 160

List of Tables

2.1 Attacks and risks on cyber-physical 3D printer based on CAN vulnerabilities in Hartzel et al. [1]. 12

3.1 Security/Feature overview comparison of proposed vs. existing [2, 3, 4] frameworks. As our framework is designed around providing adequate security against all fronts of attacks, we are capable of defending against most modern and future CAN threats. 38

3.2 Estimated time frameworks would spend transmitting authentication messages: This work vs. Existing Pre-Quantum [2, 3] and Post-Quantum [4] frameworks. Time estimations are based from their message values using their specified CAN configurations and only includes transmission time. 40

3.3 Post-Quantum Cryptography Key and Ciphertext Sizes 43

3.4 Key storage comparison across schemes in Bytes. Here, n is the number of nodes in the system. 43

4.1 CAN-based 3D-printer data vulnerable to CAN attacks based on category of critical data/functionality. 59

4.2 Comparison of our framework to existing CAN frameworks. As the target application is different, only typical CAN security criteria and security requirements are considered to ensure fair comparison. 78

5.1 Recent Works in Anomaly Detection with Quantum Machine Learning. 93

5.2 Median errors and decoherence times in IBM Quantum computers captured at the time of noise model simulation. 114
5.3 Average error and percent decrease of classification metrics with Belis and 2DoF kernels when using noisy kernel simulations between 8, 12, and 16 features. 114
5.4 Reported error in noisy simulation kernels of N features using dissimilarity metric: $1 - KA(K_{ideal}, K_{noisy})$. 116
5.5 Kernel Target Alignment of quantum and classical kernels to each preprocessed dataset. 118
5.6 Summary of attacks evaluated on the security network dataset. 124
5.7 Performance metrics across different feature maps on the gathered Cube 3D Printer dataset. 127
5.8 Summary of best classification metrics for quantum kernels on ICS/CPS and 3D Printer data. 131

List of Figures

1.1 An example conceptualization of a Quantum-Safe, Quantum-Enabled cyber-physical industrial ecosystem using the concepts within this work. 4

2.1 A comprehensive taxonomy of CPS security threats relative to each abstraction layer of the generalized hierarchy structure of a full-fledged CPS. The categories highlighted in red are the primary focuses of this work. 9

2.2 CAN and CAN-FD high-speed, standard data frame structure. (**a**) CAN frames consist of 111 bits to transmit. (**b**) CAN-FD frames consist of 572 total bits to transmit but at higher speed and efficiency. 12

2.3 Depiction of an example CAN-based 3D printer structure. The components connect to a CAN bus to be directly addressable by the main board. Each printer in a printer network (or Farm) is connected to the Controller PC through various protocols, the CAN bus being a notable example. 15

2.4 Bloch sphere showing a state $|\psi\rangle$ in its polar form. The variables θ and φ can be parametrized for encoding information. 18

2.5 Quantum Gate Symbols. 18

2.6 Fidelity quantum kernel, $k(x_i, x_j) = \left|\langle 0|U^\dagger(x_i)U(x_j)|0\rangle\right|^2$. x_i and x_j denote N-sized feature vectors of data x. A unitary transformation $U(x_i)$ is performed, followed by its inverse $U^\dagger(x_j)$ and the measurement. S_{x_i} refers to the encoding circuit, while $G(x_i)$ refers to any additional gates, such as CNOTs, used to encourage qubit interaction. 25

3.1 Injection and snooping attacks protected with Post-Quantum Vehicular Security on the CAN-FD Bus. By securing the vehicle's CAN bus and authenticating nodes within it, attackers who connect to the exposed bus are protected against it. 29

3.2 PUF-Based Post-Quantum Cryptographic CAN-FD Framework's generalized structure. SIDH key exchange is used to authenticate and transmit session keys that are then used by ECUs for normal node-to-node traffic encryption. All information is transmitted over a single CAN bus. 33

3.3 Time required to transmit all authentication information over the CAN network in $log_{10}(milliseconds)$ between PUF-PQC-CANFD, Labrado 2021 [2], Siddiqui 2017 [3], and LASAN_M 2020 [4]. The values are estimated on each framework's message costs, meaning more time is taken to transmit all required information between nodes. 40

3.4 Figure depicting the server's role as a firewall and gateway to vehicle-to-vehicle, vehicle-to-infrastructure, and vehicle-to-network communication. The server can also secure, isolate, and bridge other legacy CAN devices, infotainment protocols like Ethernet, and more. 45

4.1 Structure of our CAN-based 3D printing framework based on a hierarchical root-of-trust structure. This provides a high-security network for 3D printers to utilize internally and externally to connect a large number of untrusted components. 52

4.2 The 3D printing security paradigm and taxonomy of each category of security. Our framework targets the Network Security and Internal Communication Security aspects to secure CAN communications inside and outside the 3D printer and network. 57
4.3 Pipeline design of our framework from initialization to normal operation for a singular subnet. The three distinct phases of our design are discovery, authentication, and communication (including capability and address requesting). Each subnet should perform these steps upon startup. 62
4.4 Addresses and routing between a pair of network nodes. The left side shows the route from the orange node (bottom left) to the green node (bottom center), while the right shows the reverse route. Each route is composed from a series of local network address "hops". The current address is replaced with the return address after each hop, allowing the return route to be calculated. 65
4.5 General flow of standard communication depicting a sender transmitting through a router to reach a destination. Source (blue) creates the route and encrypted data to send to the first "hop" in the series. The router(s) (orange) will decrypt with the previous bus' session key, validate, and re-encrypt the data using the next unique CAN bus parent session key. The router process is repeated until the data reaches the destination (green). Addresses and TTL are modified after each hop. 70
4.6 Full test bed of our framework implementation, with attached 3D printer components. Network nodes are implemented by SAM C21 micro-controllers with servo, temperature sensor, and fan components attached to the endpoints. 80

4.7 Results of time to complete authentication, in Log(y), for the number of nodes in subnet k. Data points extrapolated by line of best fit for high values of k. Post-quantum performance gives a $10\times$ overhead, but both are well within acceptable timing ranges for expected values of k. 80

4.8 Number of messages (cost) needed to perform authentication in our PQC frameworks compared to existing CAN frameworks. Our PQC framework has noticeably different message costs, but still performs better than most other CAN frameworks at large numbers of nodes; [5, 4, 6]. 82

5.1 System depiction of Secure Water Treatment and Water Distribution from iTrust Cybersecurity Research Center, Singapore University of Technology. This full cycle water treatment and distribution testbench spans two separate CPS highlighted subsystems, SWAT and WADI, gathered in two separate datasets. 95

5.2 HIL-based Augmented ICS (HAI) structure for steam-turbine and hydro-power. P1 (Boiler), P2 (Turbine), and P3 (Water Treatment) are all connected by P4, the overall controller that simulates a power grid model. 95

5.3 2-Dimensional Uniform Manifold Approximation and Projection (UMAP) of the three cyberphysical datasets: SWAT, HAI, and WADI. This visualization can provide estimations of the difficulty of hyperplane separation between anomaly and normal data. 98

5.4 Pipeline of the hybrid-quantum SVM anomaly detector. Pre-processing occurs classically, is turned into a fidelity inner-product kernel using a quantum computer, and then utilized for SVM anomaly detection. Data kernels can then be passed into the trained SVM for the detection of anomalies. 100

5.5 2-DoF Belis et al. [7] Feature Map. S_x and S'_x angular encode N features onto $N/2$ qubits, and $G(x)$ performs NN entanglement. . . . 104
5.6 2-DoF version of Circuit 10 from Sakhnenko et al. [8] 104
5.7 Second-Order Pauli-Z Evolution Feature Map (ZZ Feature Map). [9] . 104
5.8 Results for Quantum SVM Anomaly Classification on SWaT for all feature maps. 106
5.9 Results for Quantum SVM Anomaly Classification on HAI for all feature maps. 108
5.10 Results for Quantum SVM Anomaly Classification on WADI for all feature maps. 110
5.11 Best Results for Classical Kernels' SVM Classification on all datasets. 111
5.12 A cyberattack testbench and implementation of the data acquisition framework. Information is gathered by monitoring the network, low-level buses, and physical devices. The testbench utilizes TCP/IP LAN for Level 1 and CAN bus for Level 0. The attacker has access to LAN and CAN through trojan devices on both networks. 121
5.13 2-Dimensional Uniform Manifold Approximation and Projection (UMAP) of the 3D Printer datasets. 126
5.14 Prediction vs. Ground Truth classifications of the Cube STL data anomalies. 129

Chapter 1

Introduction

Since the beginning of computing, society has integrated technology with many physical aspects of everyday life. From automation and optimization of the manufacturing industry to electronic start and autonomous driving in the vehicular industry, this relationship between computers and physical processes has become the backbone of the world's economy, social welfare, and overall evolution. This close interaction between physical and software components defines a Cyber-Physical System or CPS.

In particular, a full-fledged CPS system maintains algorithmic control of physical inputs and outputs, such as robots, sensors, actuators, and more. It holds a desired balance within a system, using networked embedded systems to adjust feedback loops and maintain efficient control. The industry of cyber-physical systems is assimilated within almost all aspects of human life, which can easily explain its evaluation of $9.56 billion in 2024 and expansion of 9.7% each year [10]. This extreme growth is also largely due to the fourth industrial revolution, or "Industry 4.0", integrating smart devices within homes, logistics, and manufacturing powerhouses, creating "smart cities" [11].

Ongoing engineering advancements in CPS include improving its adaptability, autonomy, efficiency, safety, reliability, and more [12]. However, with these

advancements and the widespread integration of Internet-of-Things in critical infrastructure [13], the CPS ecosystem only grows more susceptible to malicious, external interference. Thus, one domain of CPS research remains one of the most important: cyber-physical security.

Protecting these systems is of utmost priority due to their direct, physical interaction with us and the environment. Cyberattacks on CPS can not only deny reliability in times of need, but they can also risk the lives of both the operators and the public. However, securing these systems remains a steep, uphill battle, as the complexity of CPS and the computing power of adversaries continues to grow rapidly [14]. The world of computing is evolving and ever-changing, and many emerging technologies continue to become available for both our social welfare/safety and for adversaries to exploit for malicious gain.

One such highly emerging topic in the past two decades is a paradigm that could shift the entire balance of the computing space. In 1986, researchers like Richard Feynman could only envision computers capable of simulating quantum mechanics with high accuracy [15]. Today, we are getting closer and closer to realizing that vision with readily available quantum computers from IBM [16], IonQ [17], and Xanadu [18] consisting of hundreds of superpositioned qubits. With public APIs, like IBM's Qiskit [19], one can readily access cloud-based quantum computation to calculate and solve high-complexity problems.

In the world of cybersecurity, this explosion in quantum computing has brought forth concern and urgency, realizing that the theories of the famous mathematician, Peter Shor, could very well be feasible in the future. While Shor's Algorithm [20] is just a generalization to solving the integer factorization problem in polynomial time, the world's most essential cryptographic systems happen to utilize large, hard-to-factor prime integers for security. These same cryptosystems secure our water treatment systems, power plants, and even modern vehicles [14]. Currently, Shor's Algorithm remains infeasible on modern NISQ computers due to resources and noise, but this can easily change as technology progresses.

Nonetheless, quantum computing is a tool like any other technology, and can also be harnessed to provide benefits to CPS security. Cyberattacks on critical infrastructures, such as power plants and water treatment facilities, often result in changes in physical behavior and denial of service. Data collected by sensors or network logs within these systems can be a key to identifying if a cyberattack has occurred and manipulated the system [21]. As CPS systems continue to grow, quantum machine learning can find patterns in data that are too computationally expensive to find with classical algorithms [22].

Rebuilding our cybersecurity from the foundation is critical to protecting cyber-physical systems in a quantum-enabled future. In this work, we focus on fully establishing a quantum-safe and quantum-enhanced CPS ecosystem, from physical devices/communication to Supervisory Control and Data Acquisition (SCADA) monitoring systems. By breaking the topology structure of CPS into multiple layers of abstraction [23], we target security solutions for each level to create a proposed solution in each aspect of CPS security. Also, by evaluating existing security practices with a post-quantum mindset, we propose new frameworks in these layers that balance the same CPS speed and efficiency with higher, quantum-era standards.

Furthermore, we chose to embrace and draw on the computational power of quantum, enhancing CPS security systems with high-dimensional machine learning models for identifying malicious activity within even the most intricate CPS environments. Our ground-up approach to industrial cyber-physical system security can stop threats before they can cause immense damage to life and infrastructure. Overall, an example of our vision of a multi-layered secure industrial CPS system can be seen in Figure 1.1.

To achieve this vision, this work provides the following contributions to the field of CPS security:

- **Post-Quantum Framework for Intra-Vehicular Communication:** We first propose a post-quantum solution to vehicular-based CPS attacks via

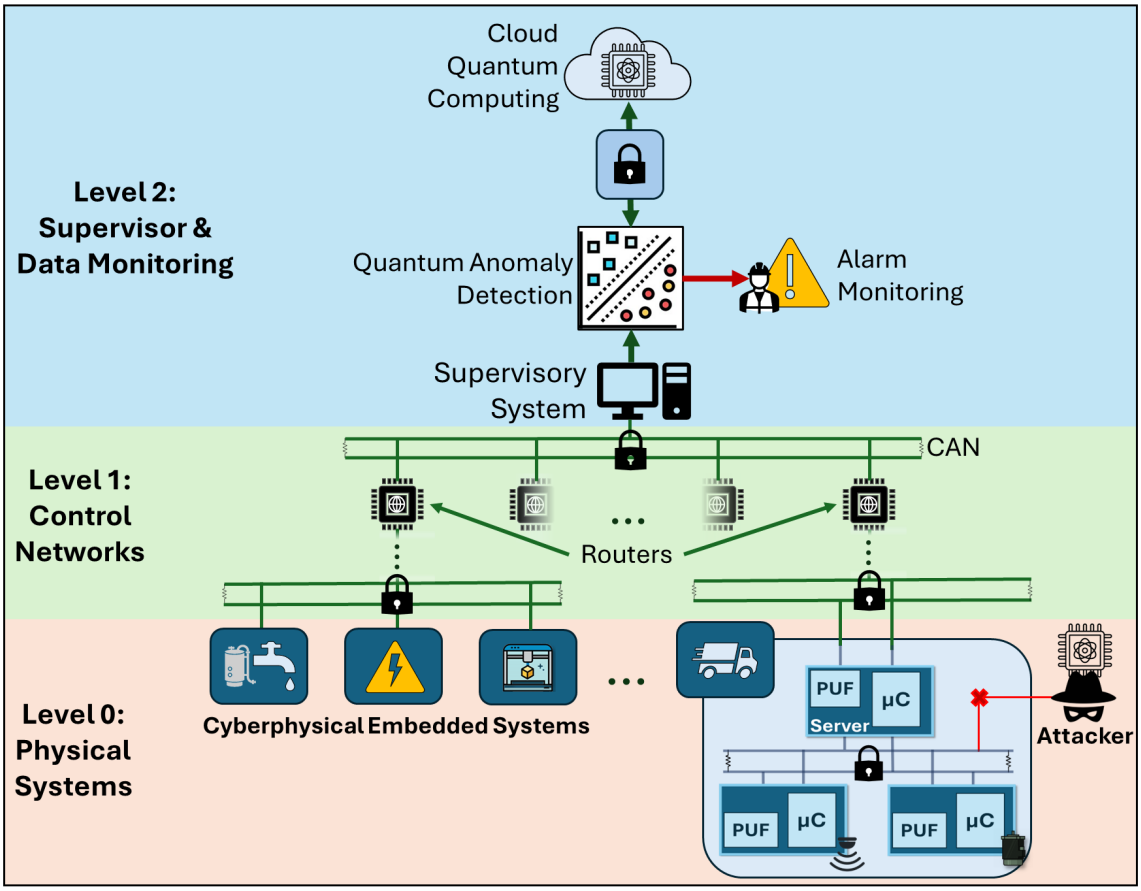


Figure 1.1: An example conceptualization of a Quantum-Safe, Quantum-Enabled cyber-physical industrial ecosystem using the concepts within this work.

the Controller Area Network (CAN) bus, shown in "Level 0" in Figure 1.1. Using specifically chosen post-quantum cryptographic suites and a Physically Unclonable Function (PUF) trust scheme, we have minimized the impact and load on the CAN bus with PQC by orders of magnitude compared to existing post-quantum frameworks.

- **Post-Quantum, Root-of-Trust Framework for CAN-based 3D Printing Inter-Communication:** Continuing from the previous contribution, we also propose a novel post-quantum tree-based network design for Controller Area Network-based 3D Printers (or any CPS device), shown as "Level 1" in Figure 1.1. Our unique approach significantly improves the security of the CAN 3D printing farm network, protecting against more attacks than both CAN-focused and 3D-printing-focused existing works have achieved. Furthermore, we also allow the staple plug-and-play capabilities of 3D printers while maintaining integrity and privacy.

- **CPS Anomaly Detection using Quantum-Hybrid Support Vector Machines:** While much of this work focuses on defenses from quantum attacks, we also propose novel solutions to protect against and detect cyberattacks from traditional vulnerabilities in cyber-physical systems. Using the quantum fidelity algorithm as a support vector machine kernel, we thoroughly investigate three industrial control system datasets with multiple hyperparameters. Our results demonstrate a clear advantage, including an average of 13% performance gain, with quantum kernels over even the best traditional SVM kernel functions. We also implemented a real-world additive manufacturing testbench to test cyberattacks on, showing an 11% increase in accuracy for 3D printing data over classical kernels. This is shown in "Level 2" in Figure 1.1.

- **Investigation of Future Work Directions and Possibilities:** Lastly, we also discuss the future work needed in this direction to fully implement our vision in Figure 1.1. This includes privacy preserving post-quantum, device

fingerprinting and strong PUF implementations, reducing QML workloads for cloud-based quantum computing, and further investigation into quantum anomaly detection in the CPS domain.

1.1 Outline

To summarize, the outline of this dissertation is as follows: Chapter 2 provides a contextual overview of CPS ecosystems/architectures, quantum mechanics, and a taxonomy of security topics related to this domain. Chapter 3 covers physical and low-level post-quantum security in the context of CAN-based vehicular control systems. These concepts are extended into Chapter 4 to develop tree-based, root-of-trust CAN networks for highly efficient and secure inter-device communication within additive manufacturing ecosystems. In Chapter 5, we move toward enhancing security in industrial supervisory data systems by developing high-performance Quantum-Hybrid Support Vector Machines (QSVMs) for anomaly detection. Finally, Chapter 6 discusses limitations related to this work, future avenues for research, and the potential implications this work has on the applicability and effects of quantum computing in the CPS domain.

Chapter 2

Background

In this section, we will provide a background on cyber-physical systems, relevant topics in CPS, and quantum computing/mechanics. We will also cover how the classical computing domain has already been affected by quantum computing as an adversary, a defense, and a victim.

2.1 Cyber-Physical Systems

Cyber-Physical Systems (CPS) are mechanisms controlled by computer algorithms that provide a highly efficient integration between the user and a specified physical environment/system. CPS devices can include smart grids, smart vehicles, industrial control systems, robotics, and avionics. From the security system of a home to the autopilot of a commercial airliner, cyber-physical systems span as one of the largest industries in electronics of the modern age [10].

CPS are made up of embedded sensors and actuators, usually connected to microcontrollers and Programmable Logic Controllers (PLC). These embedded devices are connected through physical communication networks to create vast arrays of sensors, actuators, and other low-level devices, all controlled through algorithms performed by a central computing and data collection system, or "master" [12].

2.1.1 Vulnerabilities of the CPS Hierarchy

While the design and anatomy of a CPS can vary greatly between domains, a really strong example of a good generalized CPS architecture can be based on the ISA-95 [23]. A taxonomy of CPS threats relative to each level of the CPS architecture is presented in Figure 2.1.

At the lowest level exists physical interaction with the environment using I/O, devices, and sensors connected to simple embedded systems. This concept can include process device networks, such as vehicular intra-communication networks. This data is vulnerable to snooping and modification, especially on the Controller Area Network (CAN) bus, a widely used protocol in modern vehicles, robotics, and 3D printers. However, these networks can be attacked by hardware trojans and other malicious man-in-the-middle (MITM) attacks. Securing these devices by providing wide-scale intra-device trust is the first step in creating a reliable, secure CPS system. We cover a new security framework related to this level with the context of the CAN bus in Chapter 3.

The next level of the CPS ecosystem is Level 1, which makes up the automation systems that create a mass network of interconnected embedded systems via physical and wireless connections. This "level" is highly broad, taking on various methods and structures depending on the CPS. Generally speaking, this level congregates the many physical devices, like embedded systems, into a singular network for automation/monitoring purposes. Thus, many networking-related attacks, like spoofing, snooping, denial of service, and root-device spoofing are prevalent. These attacks could impact smart vehicle communication (i.e., Vehicle-to-Vehicle or Vehicle-to-Infrastructure), or mass production automation in 3D printing farms. Particularly, Chapter 4 proposes a root-of-trust, tree-based security framework utilizing the CAN bus for mass-scale additive manufacturing networks.

The next level is Level 2, which consists of the master controller of the aforementioned Level 1 network. This usually involves various monitoring and

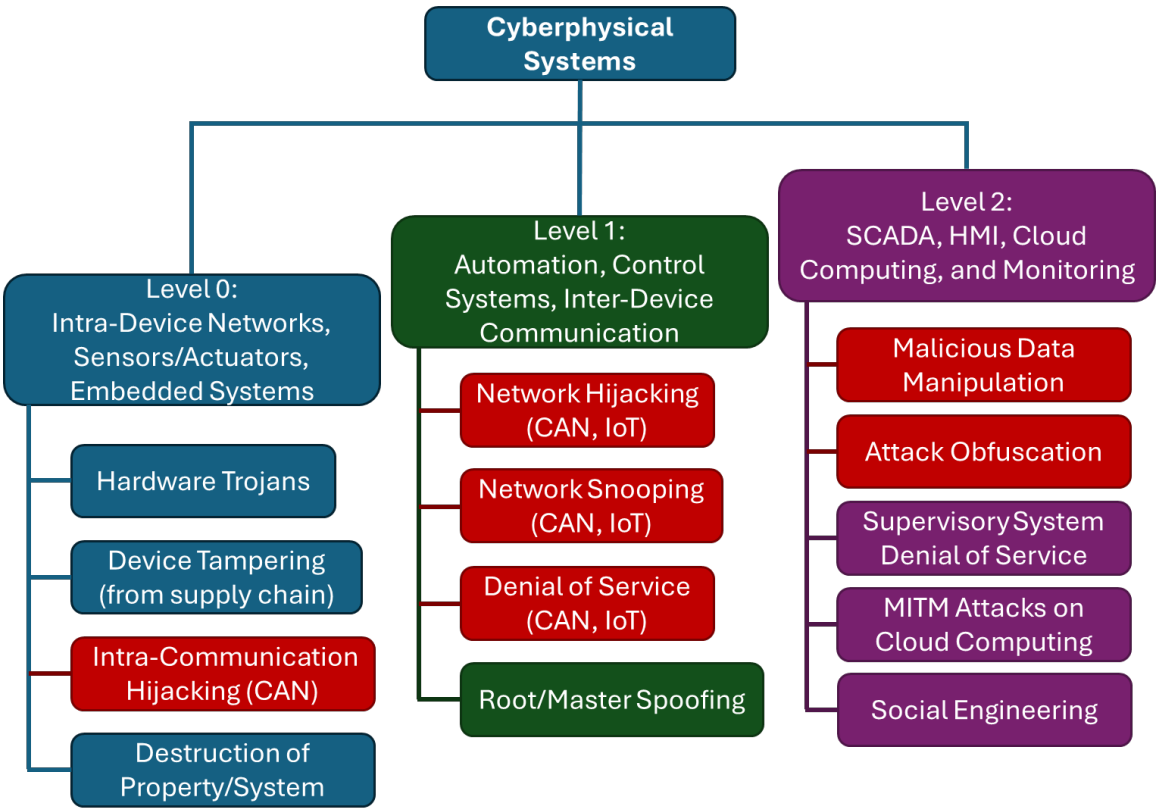


Figure 2.1: A comprehensive taxonomy of CPS security threats relative to each abstraction layer of the generalized hierarchy structure of a full-fledged CPS. The categories highlighted in red are the primary focuses of this work.

control computers, such as Supervisory Control and Data Acquisition (SCADA) systems, Human-Machine Interface (HMI), and more. This level provides the CPS ecosystem with data collection, macro-level control and automation, and anomaly/alarm monitoring services. As a result, attacks on this focus on data manipulation/theft, obfuscation of attacks on previous levels (such as turning off or tricking alarms/monitors), and social engineering attacks on the operators [24]. Level 2 can be considered the root, or master, controller system where human-machine interaction occurs. In this work, Chapter 5 discusses an advanced, quantum-assisted anomaly detection system for use in SCADA industrial control systems (ICS) to assist in cyberattack and sabotage detection.

2.1.2 Controller Area Network (CAN)

The Controller Area Network, or CAN, is a bus protocol widely used in high inter-connectivity environments where many sensitive input and output sensors and drivers communicate [25]. This includes many CPS environments, such as various robotics, vehicular electronic systems, and sensor/actuator systems of industrial control systems. These nodes transmit and receive on the same bus, using identifiers of 11 bits contained within the message payload to select the recipient(s). The source-independent messages allow for control of physical functions to be easily given to additional systems, such as deep-learning neural networks or even adversaries. All CAN devices continually listen/transmit to the bus at all times, but only messages with certain destination IDs (usually bit-masked) will be received and stored by the interested recipients.

The CAN bus' architecture and features were expanded in 2012 by Bosch [26] with Controller Area Network Flexible Data-Rate (CAN-FD), boasting 5 times higher data transmit rates. This newer design provides up to 8 times the payload size of CAN each frame, with a 29-bit identifier. CAN frames consist of 111 bits to transmit, with only 64 bits of payload data. CAN-FD frames consist of similar-sized arbitration with

much larger data frames. The CAN frame and its successor, CAN-FD, can be seen in Figure 2.2. Additionally, CAN-FD can transmit information at a significantly higher speed than standard CAN. Thus, CAN-FD's efficiency is significantly higher than its standard predecessor. As a result, this work will feature CAN-FD in Chapters 3 and 4.

However, the security of the CAN bus is minimal, which provides adversaries with sensitive information, control, and the ability to spoof all other nodes within the network with ease. Because all messages are essentially broadcast on the same unencrypted communication bus, attackers only need physical access to carry out a wide range of man-in-the-middle attacks, eavesdropping, impersonation, and denial of service [27]. Thus, given a matching transmission ID, any device on the bus may communicate as any ID regardless of authenticity.

The categories of threats that attackers may use against CAN are well detailed in previous literature [28] and laid out in Table 2.1. Eavesdropping attacks focus on stealing data and analyzing the bus for future attacks, and data insertion/spoofing attacks focus on generating unauthorized CAN frames in the network by spoofing the destination ID and payload structure of an established CAN communication. Security blindspots provide access to the insecure CAN bus from an external source, such as wireless or debug ports [29], making the risk of vulnerable systems, like IoT in industries, a huge attack vector. As CAN provides bus-based modularity, the information needed to understand the system, such as the assignment of CAN IDs or encryption engines, can be easily captured and reverse-engineered during the device initialization and registration period.

Nodes may also accidentally or purposefully fuzz/denial-of-service (DoS) communication between all devices by overloading the bus or altering the bits/voltage of the bus. DoS attacks prevent any particular node (or whole network) from providing service to other nodes [30]. There are three possible attacker profiles in CAN networks: a thief injecting software to steal data or alter control systems, a manufacturer

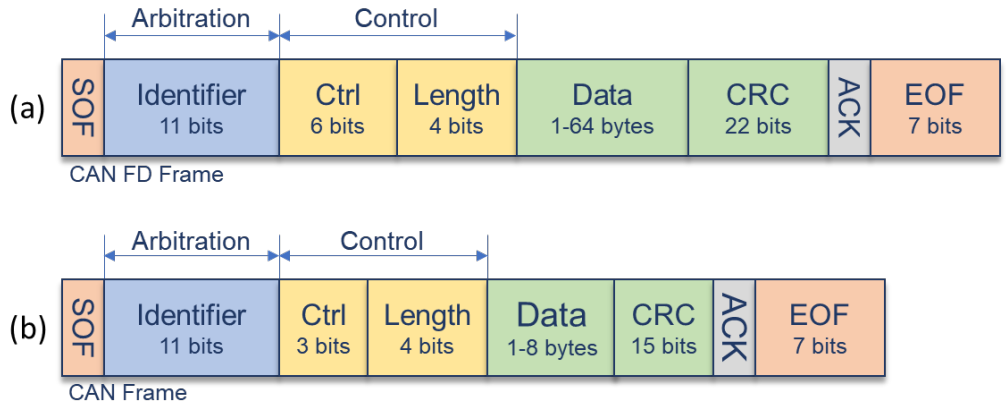


Figure 2.2: CAN and CAN-FD high-speed, standard data frame structure. (**a**) CAN frames consist of 111 bits to transmit. (**b**) CAN-FD frames consist of 572 total bits to transmit but at higher speed and efficiency.

Table 2.1: Attacks and risks on cyber-physical 3D printer based on CAN vulnerabilities in Hartzel et al. [1].

Vulnerability	Capabilities on CAN Bus
Multi-cast Messaging	Undetectable, passive listening to all messages can occur regardless of the recipient(s).
No Source Authentication/Addressing	Spoofed messages can come from any device or source.
Shared Communication Bus	Adversary can gain access to all nodes from a single exploit.
Lack of Encryption and Data Isolation	Information is not private, can be easily read and fabricated by any device on the bus.
Multi-system Integration	Various different systems integrate onto the CAN bus, extending the attack surface.
Limited Bandwidth	Easy to deny service to safety and security modules.

injecting harmful data/firmware into the bus affecting safe operation, and the system owner accidentally transmitting incorrect data or unsafe devices.

Prior work has highlighted the vulnerabilities of CAN security in robotics and vehicular applications, especially with the rise of Internet of Things (IoT) features and devices. As the CAN protocol was not designed for security, many security issues persist that require a higher-level protocol/framework to account for. The CAN protocol uses unencrypted messaging with only a destination/ID field for message filtering and arbitration purposes. This restriction causes CAN to lack the ability to authenticate message sources, making all nodes blindly accept any packet that matches their filtered ID (or filter mask). Reconstruction of command structures from eavesdropped CAN packets can be used to perform replay or impersonation attacks as any component on the 3D printer CAN bus. The message collision and arbitration methods of CAN establish a priority system that has been exploited by attackers to perform simple denial of service attacks on CAN buses by constant transmission of high-priority messages [31].

2.1.3 3D Printing (Additive Manufacturing)

Additive Manufacturing (AM) is the process of developing 3D objects through an additive, layer-by-layer approach [32]. This process, also known as 3D printing, provides fast prototyping of complex parts and can utilize food, polymers, metals, and living tissue or matter [33]. The models these printers are designed to physically construct are specified by computer-scripted g-code, which defines how the system should move [34]. These printers utilize multiple cyber-physical devices to help in this process, from thermocouples and resistive heaters to servo motors with positional awareness. Marlin [35], a popular open-source framework designed around the RepRap movement, forms the foundation of many commercial 3D printer firmware and software packages.

Furthermore, commercial printers are beginning to advertise their use of the CAN bus to connect plug-and-play components for its low-cost, highly modular features [36]. A recent survey on CAN in 3D printing detailed the different additive manufacturing applications in which the CAN bus has been utilized [37]. Commercial 3D printing solutions have also begun to include one or more IoT-enabled devices within the printers, allowing a PC to control the 3D printer remotely via the internet. Furthermore, 3D printing farms have been popularized in large-scale manufacturing and cloud printing services due to their ability to generate complex structures. These mass arrays of 3D printers all connect to large servers capable of controlling multiple printers at the same time. The CAN protocol has been utilized in previous literature to network 3D printer farms and peripheral devices, like cameras, together in place of an Ethernet LAN [38]. Figure 2.3 depicts an example of a CAN-based 3D printer structure for both component communications and secure networking to the controller PC.

Prior work in discovering vulnerabilities in popular 3D printers and their common firmware, Marlin, has been an ongoing topic among researchers. Popularity in 3D printer security began with a study focusing on discretely modifying model g-code using malware [39]. This attack raised concerns when they were able to inject a weakness in the 3D-printed drone propeller's model, causing the part to fail shortly after printing. Furthermore, attacks on 3D printing files and networks have been demonstrated which capture/steal model information and inject subtle, malicious changes to the data to cause printer malfunctions or product failures [40, 41, 42]. Other attacks focus on using machine learning and other methods to reconstruct g-code by gathering sensor data and monitoring the physical properties of the printer [43]. Lastly, the CAN bus has previously been exploited in commercial 3D printers to modify target values and feedback loops through replay attacks and impersonation. The study demonstrates the malicious modification of target and current temperatures of the printing material and bed, end-stop limit switches, movement axes, and more to create extremely dangerous conditions [44].

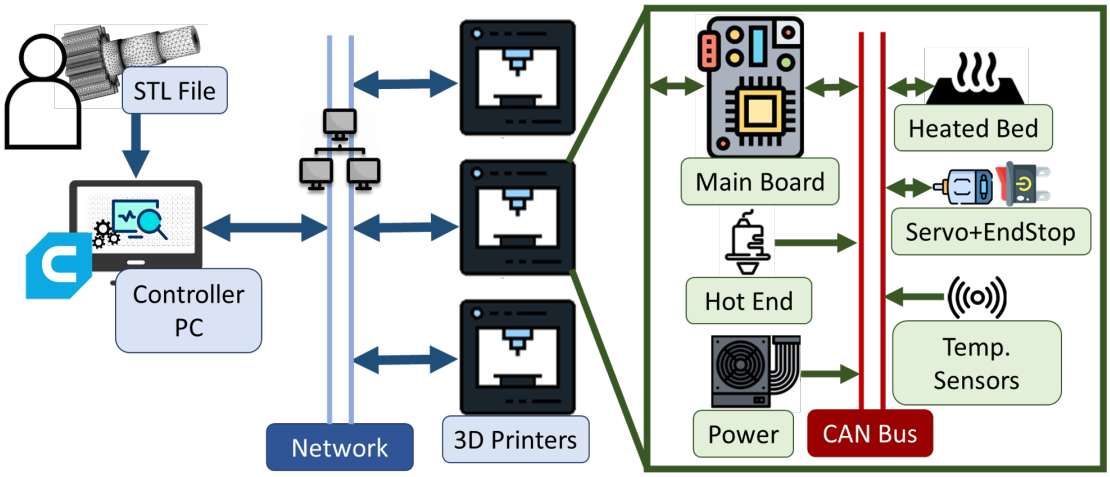


Figure 2.3: Depiction of an example CAN-based 3D printer structure. The components connect to a CAN bus to be directly addressable by the main board. Each printer in a printer network (or Farm) is connected to the Controller PC through various protocols, the CAN bus being a notable example.

2.2 Quantum Computing

As quantum computing is a core aspect of this work, it is appropriate to provide an explanation of the relevant quantum background to understand the remaining chapters.

In traditional computing, information is represented in a binary (two-state) digit known as a "bit" that can only represent a 0 or 1 state. The entire world of electronic computing has been based on this phenomenon, providing a deterministic, finite number of representable values to perform logic operations with. However, pushing the boundaries of sequential, classical computing has pushed us to explore horizons that allow better ways to represent and process our information. Quantum computing is a newly expanding paradigm of computing that takes advantage of quantum mechanics. It utilizes what's known as a quantum bit, or "qubit", which encodes a two-state quantum system built from a probability of an outcome state. Thus, we can consider a qubit like a vector in a complex space, it can be described (with quantum nomenclature) as:

$$|\psi\rangle = c_0|0\rangle + c_1|1\rangle \tag{2.1}$$

where $c_0, c_1 \in C$. In this case, $|c_0|^2$ and $|c_1|^2$ are the probabilities of finding the qubit in state $|0\rangle$ and $|1\rangle$, respectively. Thus, the two probabilities must add to 1, or $|c_0|^2 + |c_1|^2 = 1$. These can be built by many incredible technologies, such as superconducting qubits [45] and ion-trap qubits [17].

Using combinations of these qubits, we can create highly complex systems, outputting a possibility of 2^N states (where N is the number of qubits), all with their own probability amplitudes. While classical bits can be represented with just as many states, quantum qubits can represent all 2^N simultaneously through a concept called "superpositioning". These principles allow us to perform incredible tasks like integer factoring and 2^N-dimensional matrix distance calculations, such as NP-hard problems. Additionally, just as quantum state coefficients can be represented

as probability amplitudes c_i, we can represent information in polar form and can represent information as a point on a unitary Bloch sphere, as depicted in Figure 2.4.

2.2.1 Quantum Gates

In quantum, operations performed on these qubits are called "quantum gates", which are represented by unitary transformations. These unitaries can operate on one or more qubits to alter the output probabilities, but can also be used to intertwine the outcomes of multiple qubits together, called "entanglement". The most common gates related to this work are shown in Figure 2.5.

Controlled-Not Gate (C-NOT)

The "Controlled-Not" gate, or C-NOT, is an essential operation for creating entanglement, or dependency, between two qubits. The C-NOT's circuit diagram is shown in Figure 2.5c. This entanglement is important in this work for extracting/expressing relationships between qubit states is necessary. The C-NOT gate has a control qubit that will flip the target qubit if the control qubit is $|1\rangle$. The matrix is defined as:

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{2.2}$$

Hadamard Gates (H)

Depicted in Figure 2.5a, the Hadamard Gate (H) is a single, non-parametric qubit gate that is usually used at the beginning of a circuit to induce superposition into a qubit by establishing equal probabilities of $|0\rangle$ and $|1\rangle$. The matrix form is depicted as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 \\ 1 & -1 \end{bmatrix} \tag{2.3}$$

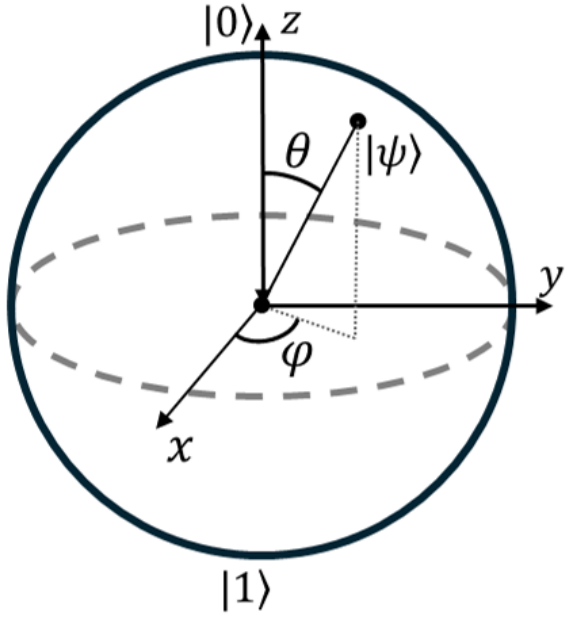
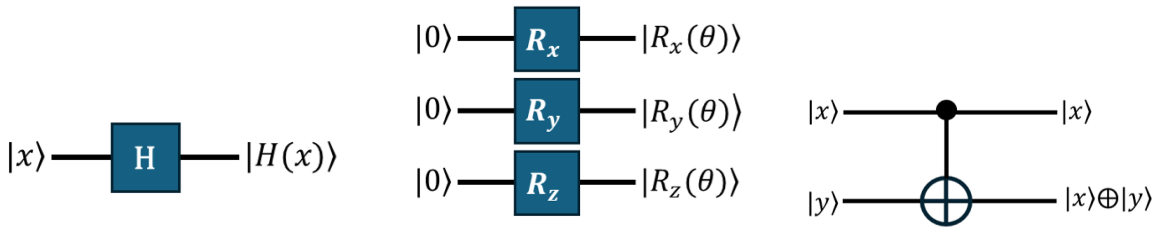


Figure 2.4: Bloch sphere showing a state $|\psi\rangle$ in its polar form. The variables θ and φ can be parametrized for encoding information.



(a) Hadamard Gate (H)

(b) Rotation Gates (R_x, R_y, R_z)

(c) Controlled-NOT Gate (C-NOT)

Figure 2.5: Quantum Gate Symbols.

Rotation Gates

Rotation gates, or R_* gates, are single-qubit gates that can be parametrized by classical rotation values. Thus, using the equation

$$|\psi\rangle = cos\theta|0\rangle + sin\theta e^{i*\varphi}|1\rangle, \tag{2.4}$$

we can use parameters θ and φ to represent our quantum state as a vector rotated about the central axes, x,y, and z. These general rotation gates ($R_x/R_y/R_z$) allow for us to rotate the quantum state about the Bloch sphere at an angle, θ, that we can parametrize before computation. Thus, these gates are commonly used for encoding classical data, such as machine learning data. The rotation gate notation is shown in Figure 2.5b.

2.2.2 Embedding Classical Data to Quantum

To represent classical values, such as sensor data points, encoding circuits are used to map data features on each qubit, resulting in a quantum state representing all N features. By far, Angular encoding remains the most popular method due to its simplicity, which utilizes parametrized rotation gates to encode data into superpositioned qubit states. Alternatively, dense-angle encoding represents two features per qubit by taking advantage of the relative phase degree of freedom [46], resulting in N features expressed by $n = N/2$ qubits. We generalized this operation as a unitary S_x, which maps the feature vector $x = [x_1, ..., x_N]^T \in \mathbb{R}^N$ to $N/2$ qubits.

2.2.3 Noisy Intermediate-Scale Quantum (NISQ) Computers

While great strides have been made in the fields of quantum computing, one confounding variable has held back the applicability of quantum algorithms in real-world uses: decoherence. Decoherence can be thought of as a form of error, where the delicate quantum state breaks down its quantum properties upon interaction with

the environment. Thus, as long as error persists within a model, the results achieved from a quantum computer can never be regarded as purely deterministic or certain [47].

In this age of quantum computing, the term for modern quantum machines with noisy qubits and limited resources was coined "Noisy Intermediate-Scale Quantum" computing, or NISQ. These computers currently struggle to perform real-world applied quantum algorithms that need high precision. However, many machines have begun to lower their error, from newer technologies like Ion-Q Forte [17] having 0.4% Error-per-Layered-Gate (EPLG) to new IBM Heron R2 0.348% EPLG [19]. These machines have been able to maintain fragile quantum states longer by utilizing quantum error mitigation models [48]. Thankfully, many machine learning concepts do not require high precision and are acceptable with various approximate methods, such as Clark et al. [49].

2.3 Quantum Computing as an Attacker

Quantum computing's advantage is well known in the community for being able to reduce the exponential time complexity of many known NP-hard problems in the world. One such problem that relates to security research is the Integer Factorization problem, which usually is an exponential growth in complexity.

Shor's Algorithm, proposed by Peter Shor [20], suggests that using quantum parallelism allows us to perform an exponential number of operations in one step. This can reduce the time complexity of the integer factorization problem from sub-exponential time to polynomial time in $log(n)$, where n is the integer to factor. To factor an integer n, Shor's Algorithm uses the Quantum Fourier Transform (QFT) to find the period of the function $F(a) = x^a \ mod \ n$ when x is an integer coprime to n. This helps us find the period $F(a) = r$, which is used to find the factor classically. In other words, using a superposition of all possible values of a, we can calculate $x^a \ mod \ n$ an exponential number of times.

This has major implications for the cybersecurity sector as the world's most popular cryptographic algorithms rely on the immense difficulty of factorizing numbers like n. With this, a quantum attacker could theoretically break popular asymmetric algorithms (like Diffie-Hellman, RSA, and more) [50] by determining the prime factors of the public key system.

Grover's Search Algorithm [51] similarly utilizes quantum mechanics to speed up the brute force search needed to crack symmetric encryption, like AES. Normally, it would take $O(N)$ time to search through all N elements of a possible key. However, with Grover's Algorithm, it would take merely $O(\sqrt{(N)})$, suggesting major implications for small key sizes. This algorithm creates a superposition of all N elements, performing the diffusion unitary operator $U = \frac{1}{2}I - \frac{1}{2}|\psi\rangle\langle\psi|$, where I is the identity operator, approximately $O(\sqrt{(N)})$ times.

Upon measurement, an ideal quantum computation would return the marked state as the highest probability amplitude. This, however, requires an absurd number of qubits to properly achieve a significant quadratic time reduction. This resource requirement is currently well beyond our current NISQ computing limitations.

2.3.1 Post-Quantum Cryptography

Most modern cryptosystems are ill-equipped for a quantum threat, as many still operate from quantum-unsafe asymmetric cryptography, such as RSA. These schemes had been deemed to be quantum-insecure as they are likely breakable in polynomial time with quantum computation. Furthermore, Grover's Algorithm [51] even risks most small or lightweight cryptography used in CPS. Thus, the NIST proposed a change to cryptography safe from both classical and quantum threats [52].

Post-quantum cryptography, or PQC, is a classification of cryptographic systems that are secure from cryptanalytic attacks by quantum computers. The concept of a quantum-safe cryptosystem came from a direct response to the potential approach of Shor's and Grover's Algorithms. There are many forms of PQC, primarily in

the categories of lattice-based, hash-based, isogeny-based, and multivariate-based cryptography.

Lattice-based, the most popular and NIST-recommended, involves a computationally exhaustive "Shortest Vector Problem", which approximates the Euclidean length of a lattice vector. The NIST has selected the CRYSTALS-Kyber, a lattice-based method [52], for security-critical applications needing key establishment, including manufacturing. This PQC utilizes a variant of the "learning with errors" lattice problem, another quantum-safe NP-hard problem. We utilize CRYSTALS-KYBER later in Chapter 4.

In this work, we also utilize isogeny-based post-quantum cryptography using Supersingular Isogeny graphs. This method uses Supersingular elliptic curves over finite fields to establish a Diffie-Hellman-like key exchange system. In particular, we utilized Supersingular Isogeny Diffie-Hellman [53], or SIDH/SIKE, in Chapter 3 due to its extremely low key size and portability compared to other methods.

As per the NIST, classical symmetric schemes (i.e., AES256) are also considered secure as long as their targeted bit-security level exceeds double the security requirement, as Grover's Algorithm can only quadratically speed up exhaustive key searching [54]. As these algorithms are computationally way more expensive, careful consideration is required to efficiently apply them to resource-limited domains, such as CPS embedded hardware.

2.4 Quantum Computing as an Attack Detector

Quantum computing can also be utilized to perform exponentially difficult computations when detecting and defending against cyberattacks. Quantum Machine Learning (QML) is a concept that utilizes quantum computing for either fully adapted or hybrid machine learning algorithms [55], especially classification tasks. This includes examples such as quantum support vector machines [56], quantum neural networks [57], and more. These models are designed to classify when a data

point is much different than the trained, expected behavior of the system, which is called an "anomaly". These anomalies are clear signs of system malfunction or data tampering, a likely result of cyberattacks on a CPS supervisory system. Thus, we can employ quantum in these detection systems to find highly complex, deep-relationship abnormalities that low-interaction, classical methods would be unable to find.

2.4.1 Quantum-Hybrid Support Vector Machines

The One-Class Support Vector Machine (OCSVM) is a robust and widely used method for anomaly detection. It is a variant of the Support Vector Machines (SVM) originally proposed by Cortes and Vapnik [58], where they first talked about the optimal hyperplane in high-dimensional feature space [59]. This is particularly useful where the training data consists predominantly of baseline or normal observations with few anomaly data points. The algorithm tries to find the best enclosed spaces or decision boundaries that cover the normal data keeping the anomalous data outside the boundary. Using a "kernel mapping function", an inner-product matrix called a "kernel" maps data to allow training and classification on this decision boundary. Thus, the data is classified by its existence within/outside the bounds of a hyperplane drawn in a high-dimensional feature space represented by the "kernel".

One extremely emerging variant of the OCSVM for detecting anomalies is using a hybrid approach between classical SVM classification boundaries with data processed and dimensionally flattened with quantum [9]. In a Quantum-Hybrid SVM (QSVM), the traditional kernel function is replaced with the robust quantum fidelity algorithm operating on information and eventually projecting it back to a classical representation. These measured quantum quasiprobabilities become a "quantum kernel".

2.4.2 Quantum Fidelity Kernels

Quantum-Hybrid Support Vector Machines utilize projected quantum kernel functions to transform the information into a higher-dimensional space for the classical SVM. Quantum kernel functions derive from the state fidelity, or overlap, between two quantum states produced by the parameterized "feature map", $U(x)$. This state fidelity is defined in Equation 2.5, which is effectively an internal dot product between the image of two vectors. Quantum feature maps [60] encode each feature to a higher-dimensional n-qubit Hilbert Space $\mathcal{H}$, resulting in a separable, dimensionally flattened state. Additional quantum operations, such as C-NOTs as entanglement gates, transform the encoded information to provide interaction between qubits, which we have defined as $G(x)$. Generally, a feature map is performed on an initialized known basis-state, such as $|0\rangle^{\otimes n}$. The resulting output state of $U(x)$, defined as $\phi(x)$, is used for the fidelity kernel.

$$K(x_i, x_j) = |\langle \phi(x_i) | \phi(x_j) \rangle|^2 \tag{2.5}$$

The efficient method of computing state fidelity using only n qubits uses the Hilbert-Schmidt inner product, as described by Havlíček et al. [9]. A visual representation is included in Figure 2.6. This process applies the feature map unitary $U(x_i)$ followed by its inverse unitary $U^\dagger(x_j)$ and a measurement. The frequency of measuring the base state $|0\rangle$ determines the state fidelity or overlap of x_i and x_j, which directly characterizes the similarity between the two data points in x. For example, if $x_i = x_j$, the output state from $U(x_i)$ would be perfectly inverted by $U^\dagger(x_j)$ resulting in the original, pure basal state. When this fidelity operation is repeated for all x_i and x_j, the result is an inner-product matrix of all fidelities bounded by $K(x_i, x_j) \rightarrow [0, 1]$. This kernel is utilized in the One-Class Support Vector Machine for unsupervised classification.

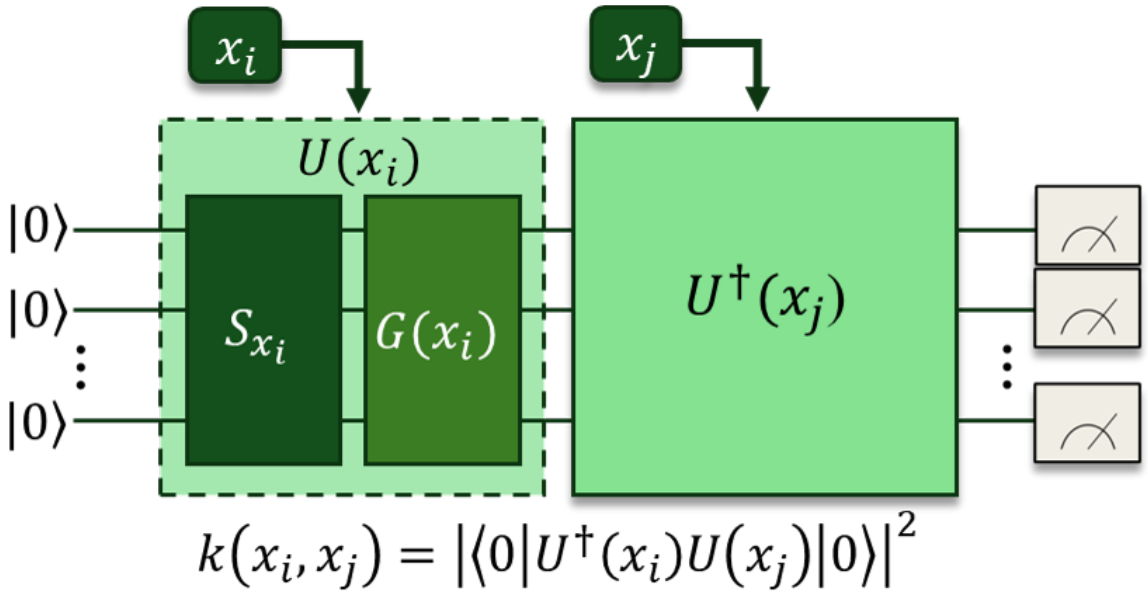


Figure 2.6: Fidelity quantum kernel, $k(x_i, x_j) = \left|\langle 0|U^\dagger(x_i)U(x_j)|0\rangle\right|^2$. x_i and x_j denote N-sized feature vectors of data x. A unitary transformation $U(x_i)$ is performed, followed by its inverse $U^\dagger(x_j)$ and the measurement. S_{x_i} refers to the encoding circuit, while $G(x_i)$ refers to any additional gates, such as CNOTs, used to encourage qubit interaction.

Chapter 3

Quantum-Safe Intra-Vehicular Communication

In this chapter, we discuss the adaptation of post-quantum cryptography within Controller Area Network (CAN), a leading communication protocol in cyberphysical systems. One of the primary applications of the Controller Area Network is within vehicles, where the majority of functionalities of the vehicle's electronic systems, called Electronic Control Units (ECUs), communicate through the differential bus. For this chapter, we will discuss CAN security in the context of vehicular ECU security, however, the principles applied here can apply to any CAN-based CPS [61].

Security of the multi-master, fully-connected CAN systems has challenged researchers for decades to develop frameworks capable of producing sufficient protection on extremely low resources. Vehicular CAN security focuses on three principles, as explained by [2]:

- Minimizing performance and size cost.
- Maintaining the CAN protocol and standards.
- Minimize overhead in message transmission.

The resource constraints of the ECUs cause vehicular security frameworks to only consider low-security cryptography for smaller ciphertext and resource usage. However, with the rapid adaptation of a newer standard of the Controller Area Network, known as CAN-FD (Flexible Data-rate), higher efficiency that spans up to eight times larger payloads can be used for cryptography and information transfer.

3.1 Motivation

Due to the rise in wireless and IoT-based technologies, Controller Area Network's insecurities have become more exposed to remote attacks providing full control of sensitive information. Accompanying this, the development of quantum technology using Shor's Algorithm is estimated to break most existing asymmetric cryptography, as described in Chapter 2. Furthermore, non-volatile storage of private keys poses a huge risk to cryptography, as most non-volatile memory is insecure and vulnerable. Security frameworks centered around a reliably generated and unclonable signature are essential to providing a form of identification or verification of authenticity in cryptosystems. Physically Unclonable Functions, or PUFs, generate authenticity and cryptographic material with high-entropy, consistent responses and have observed strong usage in various existing pre-quantum frameworks [2]. However, existing frameworks utilizing lightweight, low-security algorithms are not quantum-secure despite utilizing strong PUF technology. Additionally, post-quantum cryptography usually requires large memory and payload requirements, causing the need for CAN-FD and deep optimization. With these concerns in mind, the design of post-quantum CAN-FD frameworks is essential to minimizing performance overhead and mitigating future quantum security concerns.

3.1.1 Contribution

To mitigate these threats on the bus, we propose the use of post-quantum cryptography within a framework scheme utilizing optimized, high-speed CAN-FD frames. Our framework, "PUF-PQC-CANFD" or PUF-Based Post-Quantum Cryptographic CAN-FD Framework, utilizes post-quantum SIDH (197B) [53] and AES-256-GCM to validate the authenticity of servers and nodes through challenge-responses. We demonstrate high-security requirements while keeping memory and transmission requirements small for low-resource CAN ECUs. The framework provides for security from quantum attacks and other CAN threats, as shown in Figure 3.1. We also propose an authentication algorithm that verifies both the authenticity of the server and the node based on pre-existing, locally stored public keys. Our message and storage requirements for authentication and standard communication are competitive with existing pre-quantum and post-quantum CAN frameworks. Thus, PUF-PQC-CANFD provides a storage and message-optimized post-quantum approach to CAN security within vehicles.

3.2 Background

Vehicular electronic systems contain hundreds of ECUs that control most functionalities within the vehicle. Smart vehicles also integrate with external surrounding devices, including vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), and vehicle-to-pedestrian (V2P) systems [62]. These ECUs utilize CAN, a standardized communication system discussed in Chapter 2 with control over essential functions, such as acceleration, steering, and braking. With the continued growth of CAN with wirelessly enabled ECUs, the feasibility of conducting remote attacks on the CAN bus has become a strong concern. Previous research [63, 64] has demonstrated CAN-related vehicle attacks that were capable of taking control of vital vehicular functions.

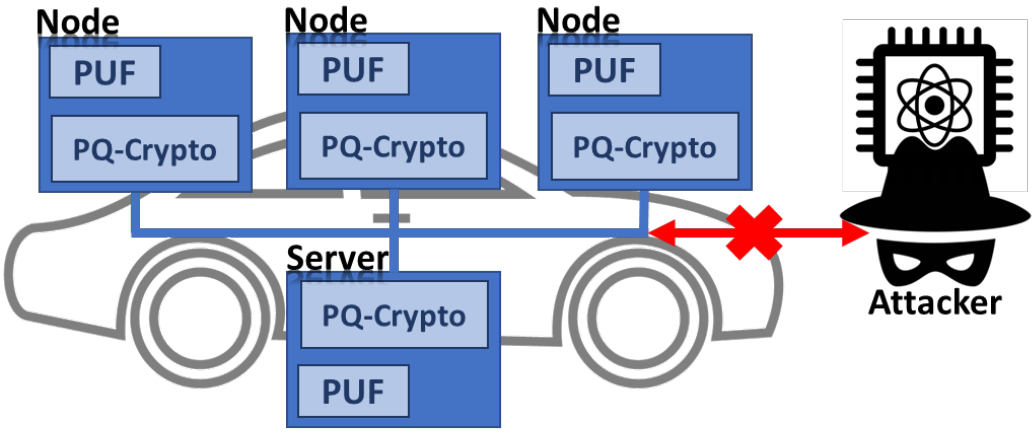


Figure 3.1: Injection and snooping attacks protected with Post-Quantum Vehicular Security on the CAN-FD Bus. By securing the vehicle's CAN bus and authenticating nodes within it, attackers who connect to the exposed bus are protected against it.

Thus, exposed and unencrypted CAN buses with no authentication systems can risk the safety of drivers.

3.2.1 Post Quantum Cryptography

The PQC used in this work is known as the Supersingular Isogeny Diffie–Hellman Key Exchange, or SIDH [53], which replaces elliptic curve mathematics with a supersingular isogeny graph. Similarly to the Elliptic Curve Diffie–Hellman (ECDH), SIDH establishes secret keys through insecure channels (such as CAN). By utilizing a private, sensitive key and another party's public key, one can generate a secure key without transmitting any vital information. Afterwards, symmetric encryption can be used to securely transmit information encrypted with the shared secret key. While key sizes of SIDH are larger than ECDH, the key sizes of sufficient security are much smaller than alternatives used in Post-Quantum Cryptographic vehicular security, such as NewHope, Kyber (800 B), and Dilithium (1312 B) [65]. However, our design allows for a flexible selection of possible key-establishment algorithms that can be used while maintaining many benefits.

3.2.2 Related Work

Many prior PUF frameworks have been proposed in pre-quantum vehicular security. Labrado et al. [2] discuss the use of PUF in a lightweight key-exchange-based authentication framework featuring a similar session key distribution system. The lightweight behavior allowed for minimal messages on standard CAN frames with sufficient security. Furthermore, Siddiqui et al. [3] provides a similar standard-encryption-based PUF framework, but suffers from extremely high message requirements due to the algorithm's "transmit all public-key" design alongside moderately sized cryptographic primitives. Cloud-reliant PUF verification frameworks, such as Easy-Sec [66], also utilize PUFs in a public-key-based authentication scheme with smart vehicle networks (i.e., V2I). These verify PUF and vehicle information through

the cloud, rather than within the vehicle. Some PUF-based CAN frameworks change the contents of the CAN packet to provide additional space [67], but these designs do not follow the ISO 11898 standard. Additionally, LASAN_M, a Post-Quantum security framework, proposed by Ravi et al. [4], discussed the use of Kyber for key exchange and Dilithium for digital signatures. LASAN_M does not use PUFs and relies on generated signatures, similar to the Transport Layer Security (TLS). The system utilizes digital certificates/signatures and a central communication or root-of-trust security module to authenticate ECUs and handle all communication. A central authority (CA) would digitally sign the vehicles that the vehicular ECUs would use to validate themselves. However, the message requirement of signature data (thousands of bytes) alongside storage requirements is LASAN_M's trade-off for extremely strong security. A popular CAN framework, TESLA [68], also repurposed its framework with post-quantum security parameters (128-bit quantum and classical security) in mind, but suffers the same trade-off as LASAN_M.

These frameworks provide specific features that focus on different applications and optimizations. Labrado et al. [2] focuses on providing lightweight, but sufficient security (PRESENT-80) to small, fast CAN systems. Additionally, Siddiqui et al.'s [3] design provides a framework more "plug-and-play" with more common 128-bit security. In LASAN_M, smaller cryptographic data is traded away for strong, fast post-quantum security for CAN-FD. However, some applications may require post-quantum security with a significantly lower amount of messages, as larger counts of nodes can become extremely vulnerable to self-denial-of-service (or significant transmission slowdowns). Thus, there is a need for a post-quantum secure framework that transmits fewer messages but stores smaller cryptographic information. PUF-PQC-CANFD aims to fill this gap in PUF-based CAN frameworks by providing a focus on smaller post-quantum cryptographic information size and, as a result, required message count.

3.3 Proposed PUF-Based CAN-FD Post Quantum Framework

We propose a design for a vehicular security framework designed for use in CAN-FD utilizing PUF responses and post-quantum security. The structure of the CAN network is shown in Figure 3.2. While the size of most cryptographic ciphertext is overwhelming for standard CAN, our design demonstrates that a CAN-FD framework is capable of post-quantum security with performance comparable to other designs. While some CAN security designs repurpose arbitration fields in the data frame [67], our proposal adheres to the CAN-FD standards and uses only the data field for ciphertext information.

3.3.1 Design

The design of PUF-PQC-CANFD features a challenge-response validation system with an additional ECU defined as a "server". The server ECU maintains a large database of all valid node public keys (loaded when an ECU is installed in the system) to generate the shared keys used in authentication upon initialization. The vehicle ECUs, defined as "nodes", will generate a shared key using their PUF response and the locally stored public key material of the valid server. This shrinks storage costs in resource-limited nodes quadratically compared to storing all public keys observed in [3]. While the public keys are safe to store in non-volatile memory, the PUF provides consistent private key material without requiring storage in insecure, non-volatile memory. These keys are unique to their nodes and will be used for authentication purposes to the server through dedicated message IDs. A two-way authentication through challenge responses is performed by the server. The server transmits a uniquely encrypted challenge to nodes and the nodes reply with both the response and a reverse-challenge. The server must generate and transmit the correct response for the node to accept the session key and whitelist from the server. To minimize

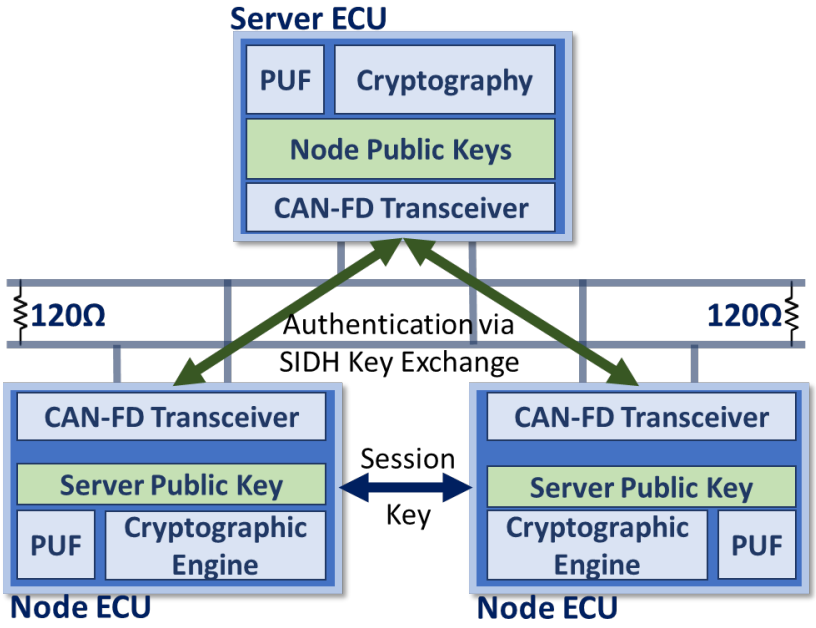


Figure 3.2: PUF-Based Post-Quantum Cryptographic CAN-FD Framework's generalized structure. SIDH key exchange is used to authenticate and transmit session keys that are then used by ECUs for normal node-to-node traffic encryption. All information is transmitted over a single CAN bus.

message costs, the challenge-responses are conducted alongside increasingly sensitive cryptographic information (i.e., session key, whitelist, etc.). Responses should be known on the challenger side for validation and can be generated from any type of function with a sufficient value space (i.e., hash, PUF response, or even a simple addition). Afterward, valid nodes are provided a temporary session key for node-to-node encryption and a whitelist containing verified ECUs. If the server's response to the challenge is invalid, the session key will not be used and an error will be issued. Therefore, the SIDH key is only used periodically for communication with the server and authentication while the session keys are used for communication between nodes. Furthermore, the exposure of the SIDH key is much less frequent than the temporary ephemeral session key, thus providing strong security alongside the PQC key. A more detailed algorithm of the authentication scheme is defined in Algorithm 1.

Normal communication should utilize the inherent AES-GMAC/CTR of GCM mode for all communication to secure and prevent replay attacks and random/garbage injection attacks. The system should redistribute new symmetric session keys for normal communication of every M messages, where M is selected per application. Additionally, a new challenge-response validation should occur with each session key by using the previous response as the next challenge. Our scheme scales linearly with the node count, as it uses significantly less messages than other PQC frameworks. Furthermore, parallelization may provide additional performance benefits, and the hardware acceleration of cryptography may remove the computational overhead of larger cryptographic primitives. Additionally, new public keys should be re-flashed into relevant ECUs when nodes/servers are replaced. The process of reflashing is implementation-specific and should be secure and tailored to the CAN system utilized. This decision to tie servers and nodes together allows local public key storage and provides additional security by using the public key as a signature for the correct server or node.

Algorithm 1 PUF-PQC-CANFD: Server-Node Authentication Scheme

Require: Server S, $NodeCount > 0$
  $S : Private_S \leftarrow PUFResponse(PUF_S)$
  for $\{N : Public_N\}$ in NodeList **do**
    $N : Private_N \leftarrow PUFResponse(PUF_N)$
    $S : Shared_N \leftarrow SIDHShared(Private_S, Public_N)$
    $N : Shared_S \leftarrow SIDHShared(Private_N, Public_S)$
  end for
  while $N_{Validated} < N_{Total}$ or not TimedOut **do**
    for N in $N_{Remaining}$ **do**
      S: Create Challenge C_S
      S: Transmit $Encrypt(C_S)$ to N
      N: $R_S \leftarrow Func(Decrypt(C_S))$
      N: Create Challenge C_N
      N: Transmit $Encrypt(R_S||C_N)$ to S
      if $R_S == R_{Expected}$ **then**
        S: N added to $N_{Validated}$
      end if
    end for
  end while
  while True **do** ▷ Repeated every M messages
    if $numMessagesSent > M$ **then**
      Generate Random 32-byte Session Key K_S
      for N in $N_{Validated}$ **do**
        S: $R_N \leftarrow Func(Decrypt(C_N))$
        S: $C_N \leftarrow R_N$
        S: $List_N \leftarrow N_{Validated}$
        S: Transmit $Encrypt(K_S||List_N||R_N)$ to N
        if $R_N == R_{Expected}$ **then**
          N: Uses K_S for Node-To-Node Encryption
        end if
      end for
    end if
  end while

3.3.2 Post-Quantum Cryptography

The selection of cryptography in this framework aims to provide a sufficient level of security against quantum attacks while limiting the message and memory cost of the cryptography. Thus, we selected a compressed SIDH [53] key exchange and AES-256 with GCM mode (for replay prevention and authentication). SIDH provides a relatively small key compared to other NIST post-quantum standard finalists with long-term usage in mind. Additionally, a secure hashing algorithm, such as SHA-256, should be used to hash the generated shared key. For additional security, one may also decide to hash responses from the PUF.

AES-256 is also considered quantum safe with 128-bits of security, as per Grover's Algorithm. Additionally, message requirements of PUF-PQC-CANFD account for the additional IV and other information for AES256-GCM required. The IV should be randomly selected and included alongside the messages, as the available space in each CAN message is generally more than enough to transmit in a single frame. Furthermore, the GMAC feature of GCM mode should be utilized as well for message source authenticity.

3.4 Security Analysis

In this section, a qualitative and quantitative analysis of CAN-message performance and security is discussed. Additionally, a discussion of the impact of this framework compared to others on the transmission time over the CAN bus is provided. These comparisons are theoretical and calculated and do not include the overhead time of cryptographic computation. These comparisons frame the benefits of our proposed design and the message cost of transmitting such large amounts of information. These comparisons help to demonstrate our alternative solution to CAN security, focusing on post-quantum protection and low-storage/transmission requirements. The results assume standard ID and high-speed CAN-FD. While the max bitrate of CAN-FD can

be up to 15Mbits/s (or higher), beyond 5Mbits/s is likely prone to issues [26]. This shrinks the data bitrate to approximately 4.2Mbits/s when factoring in overhead.

3.4.1 Attack Protection and Security

PUF-PQC-CANFD focuses on providing additional security and post-quantum cryptography without significantly impacting message cost. An overview of the defenses and security features compared to existing pre-quantum and post-quantum frameworks are shown in Table 3.1. The use of GCM mode prevents replay and random injection attacks in our framework. Counters (AES-CTR), an inherent feature of AES-GCM, ensure that no two messages are the same, even with the same plaintext information (stopping replay attacks). By maintaining the public key of the server or nodes internally, the ECUs can verify that the server/node has not been replaced by another seemingly "valid" ECU, a feature missing in some compared frameworks. By using a session key system, our framework does not require all communication to be passed through the security module such as [4]. Additionally, by frequently refreshing session keys, we can avoid creating point-to-point keys between nodes such as in [3]. Furthermore, our system uses post-quantum cryptography, including post-quantum Diffie–Hellman keys for authentication and session information. Individual keys for point-to-point encryption would provide stronger security; however, the security of the post-quantum authentication and ephemeral session keys is adequate to provide long-term safety to the CAN network. While all the mentioned frameworks protect against many cryptography-based attacks, they do not protect against denial-of-service (DoS) attacks. DoS attacks may selectively/indefinitely delay ECUs from receiving certain information. Overall, this proposal provides a strong defense against attacks and other security features that sufficiently protect against quantum and traditional threats.

Table 3.1: Security/Feature overview comparison of proposed vs. existing [2, 3, 4] frameworks. As our framework is designed around providing adequate security against all fronts of attacks, we are capable of defending against most modern and future CAN threats.

	CAN/CAN-FD Frameworks			
Criteria	**This Work**	**Labrado [2]**	**Siddiqui [3]**	**LASAN [4]**
Replay Defense	✔	✔	✘	✔
Sender Authenticity	✔	✘	✘	✔
Snooping Defense	✔	✔	✔	✔
Spoofing Defense	✔	✔	✔	✔
Node Blacklists	✔	✘	✔	✔
Post-Quantum	✔	✘	✘	✔
Server Verification	✔	✔	✘	✘
DoS Defense	✘	✘	✘	✘

3.4.2 Post-Quantum Authentication

CAN-FD frames provide much larger payloads to transmit more encrypted information without the need to split data into multiple messages. Alongside this, our system stores the server's public key within the local data of each ECU. This is for validation purposes and removes the need to transmit any public keys during authentication. These benefits combined provide results with similar message counts to lightweight pre-quantum algorithms despite the difference in key sizes. Additionally, the data transmission requires only $3n$ messages for authentication, where n is the number of nodes in the system. This is comparable to Labrado et al.'s [2] $4n$, and much lower than Siddiqui et al.'s [3] $3n^2 + 2n$ messages. Moreover, it is significantly lower than post-quantum framework LASAN_M's [4] $321n$ messages.

The estimated times taken to transmit the frames based on message costs using CAN-FD @ 5Mbps for these frameworks are shown in Table 3.2 and Figure 3.3. PUF-PQC-CANFD takes approximately 75% , 4%, and 1% of the time required to transmit authentication frames at 25 ECUs compared to [2], [3], and [4], respectively. Comparison with 50, 75, and 100 ECUs are also provided to compare more to real world ECU counts of 50 to 100 ECUs [69]. This time difference continues to grow as the number of ECUs increases. The time taken during authentication should be accounted for upon vehicle start up and reauthentication/re-keying. During this period, this stage will disable electronics in the vehicle temporarily. In addition, this framework provides 128-bits of post-quantum security with comparable results to lightweight CAN frameworks. Our proposal can send larger ciphertext with less risk of self-inflicting a DoS attack, a problem frequently discussed for standard CAN with large counts of ECUs.

3.4.3 Area and Computational Cost

It is important to also account for computation time and area of the SIDH algorithm. While a software implementation of SIDH would include a large time overhead during

Table 3.2: Estimated time frameworks would spend transmitting authentication messages: This work vs. Existing Pre-Quantum [2, 3] and Post-Quantum [4] frameworks. Time estimations are based from their message values using their specified CAN configurations and only includes transmission time.

	Number of ECUs to Authenticate			
Framework	**5**	**10**	**15**	**20**
This Work	1.68 ms	3.36 ms	5.04 ms	6.72 ms
Labrado [2]	2.24 ms	4.48 ms	6.72 ms	8.96 ms
Siddiqui [3]	9.52 ms	35.84 ms	78.96 ms	138.87 ms
LASAN_M [4]	179.62 ms	359.24 ms	538.86 ms	718.48 ms
Framework	**25**	**50**	**75**	**100**
This Work	8.4 ms	16.8 ms	25.2 ms	33.6 ms
Labrado [2]	11.2 ms	22.4 ms	33.6 ms	44.8 ms
Siddiqui [3]	215.6 ms	851.2 ms	1906.8 ms	3382.4 ms
LASAN_M [4]	898.1 ms	1796.2 ms	2694.3 ms	3592.4 ms

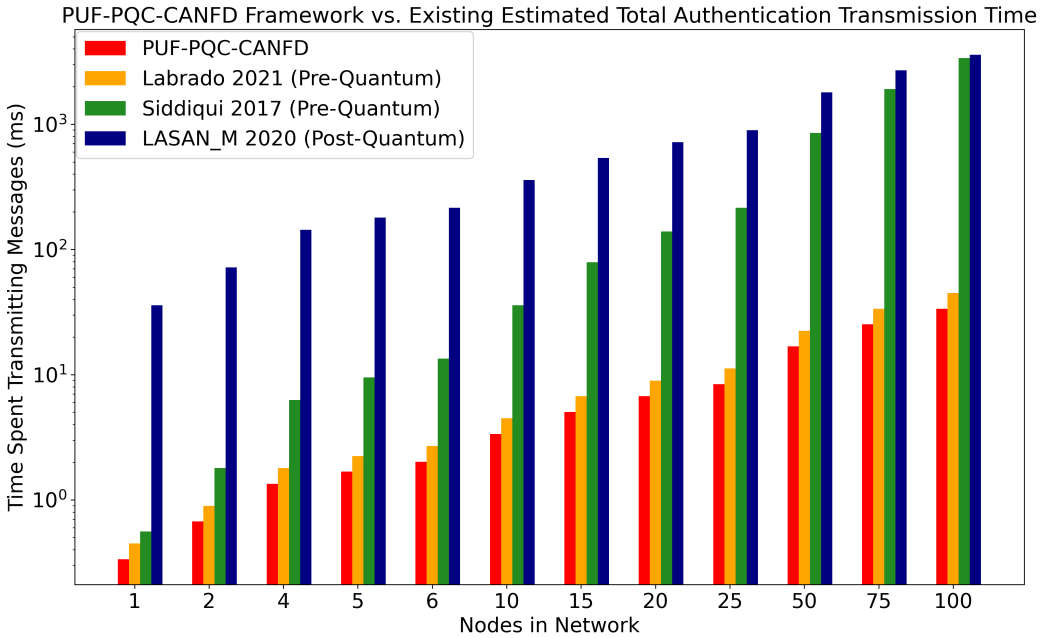


Figure 3.3: Time required to transmit all authentication information over the CAN network in $log_{10}(milliseconds)$ between PUF-PQC-CANFD, Labrado 2021 [2], Siddiqui 2017 [3], and LASAN_M 2020 [4]. The values are estimated on each framework's message costs, meaning more time is taken to transmit all required information between nodes.

the key-exchange computation, recent research in FPGA-based implementations of SIDH demonstrated key-exchanges in only 31.6ms with fewer resources than some other post-quantum algorithms [70]. Computations of SIDH are comparatively slower than cryptography used by other frameworks (i.e., ECDH, Kyber). However, lower key sizes with post-quantum allow for smaller non-volatile/volatile storage hardware requirements on low-resource nodes than other implementations. This trade-off allows for more SIDH hardware optimizations in the future to greatly benefit our framework, as the storage requirements and number of messages transmitted remain generally constant. Additionally, a design of an AES256-GCM hardware acceleration crypto-core was proposed by [71] that is sized and optimized for IoT/embedded security applications such as this framework.

3.5 Discussion

In this section, we will discuss two optional modifications that would severely improve the applicability and security of this secure CAN framework. Particularly, this section focuses on replacing the now insecure "Supersingular Isogeny Diffie-Hellman" and discussing CAN communication bridging with external devices and networks.

3.5.1 Replacing Supersingular Isogeny Diffie-Hellman

As explained earlier, SIDH was greatly impacted by a trivial attack [72] related to the isogenies that its cryptosystem relied upon. Based on our design, selecting a new encryption scheme would be trivial as the message cost is irrespective of the size of the public key or private key cryptography. Thus, the same transmission times as shown in Table 3.2 would be correct for any PQC scheme used. However, we do have two major constraints that linger in the background of consideration: storage and computational costs. Which is more important is highly situational to the constraints of the application.

In this chapter, we primarily avoided discussion of the computational and spatial complexity of the algorithms as hardware acceleration and non-volatile storage could be assumed for infrastructural devices. Furthermore, the key establishment code is only run once at a non-critical time of a vehicle's operation, making the application less reliant on extremely fast transmissions. The data size, however, can be more important to reduce the cost of non-volatile storage in the hundreds of CAN devices in a vehicle network. Thus, key size is our focus target in this discussion for selecting a proper replacement to SIDH.

The round 4 finalists for NIST Post-Quantum Cryptography [73] are shown in Table 3.3. This includes Kyber, BIKE, HQC, and McEliece. As stated, we are primarily looking for an encapsulation or establishment scheme that provides the smallest key and ciphertext sizes to replace SIDH. By this definition, the NIST-selected standard for asymmetric keys, KYBER-512, best fits our requirements. This will also play a significant role in the framework presented in the next chapter.

In comparison to the existing work from before, Table 3.4 is provided that discusses the typical key storage requirements necessary for each framework as a function of n client nodes. It is immediately apparent that our encryption keys are much larger with KYBER than they were with SIDH. Thus, the amount of space needed to perform our handshake has more than tripled by swapping to the new system, mostly in part of the significantly large private key. However, the public key still maintains a moderate size of 800 bytes, making our work vastly outperform the Dilithium-based signature method LASAN [4] uses.

However, when comparing it to the storage requirements of non-post-quantum solutions, the size of our solution begins to show the trade-offs necessary to migrate to a fully quantum-safe environment. While in modern hardware this spatial overhead is trivial to solve, cost-constrained embedded hardware may require additional storage for servers. Thankfully, as the public key is not sensitive material, this information can be stored in non-secure, non-volatile storage, like flash memory.

Table 3.3: Post-Quantum Cryptography Key and Ciphertext Sizes

Type	**Public Key Size (B)**	**Secret Key Size (B)**	**Ciphertext Size (B)**	**Category**
Kyber512	800	1,632	768	Lattice (LWE)
BIKE128	1,541	3,114	1,573	Code-based
HQC128	2,249	2,289	4,497	Code-based
McEliece348864	261,120	6,492	196	Code-based
SIDH128 (Unsafe)	330	374	346	Isogeny-based

Table 3.4: Key storage comparison across schemes in Bytes. Here, n is the number of nodes in the system.

Scheme	**Post-Quantum?**	**Cryptography Suite**	**Node Storage (B)**	**Server Storage (B)**
This Work	Yes	SIDH	736	$362n + 736$
This Work	Yes	Kyber	2,464	$800n + 2,464$
LASAN [4]	Yes	Kyber/Dilithium	20,528	$20{,}528n + 20{,}528$
Labrado [2]	No	ECDH	148	$74n + 74$
Siddiqui [3]	No	ECDH	$80n$	$80n + 80$

3.5.2 External and Inter-Network CAN Communication

In this chapter, we also abstracted away the concept of device communication outside of the CAN bus, such as Vehicle-to-Infrastructure IoT communication, device re-enrollment (during maintenance), and even inter-network communication between isolated CAN buses. However, addressing these topics would greatly strengthen the real-world applicability of this proposed design in modern, modular vehicles.

As the vehicular network paradigm shifts more toward having autonomous and vehicle-to-Infrastructure smart communications [74], many external subsystems, such as OBD-II ports [75], will require direct access to information on the CAN bus. Thus, it should be expected that anything from multimedia systems to autonomous driving vehicles will gain access to the CAN network. By treating the server as both an authentication tool at startup and a networking firewall in run-time, we can manage access to the CAN bus in a sophisticated and verified manner.

Furthermore, the public keys within the server ECUs are transferred and stored during the vehicle's manufacturing process. However, if components of the vehicle need to be exchanged or replaced, this would result in a tricky situation with re-enrolling that device into the CAN network without also reflashing or replacing the server. Thus, having a secure channel for re-enrollment via the server would allow for secure removal and addition of device ECUs using a signed flashing device.

Lastly, one of the strongest pillars of security is mitigating the damage that can be done by a single attacker as much as possible. Thus, vehicular networks may choose to isolate critical components on separate CAN buses from external-facing networks. Similar to the WAN/LAN subnetting paradigm in typical IP networking, servers can be treated as firewall "routers" for isolating critical devices from non-critical ones. The server would then, at run-time, act as the bridge between the two separate CAN networks.

A framework depiction of this is shown in Figure 3.4. In this depiction, we see the servers acting both as authentication controllers in startup and as gateway devices

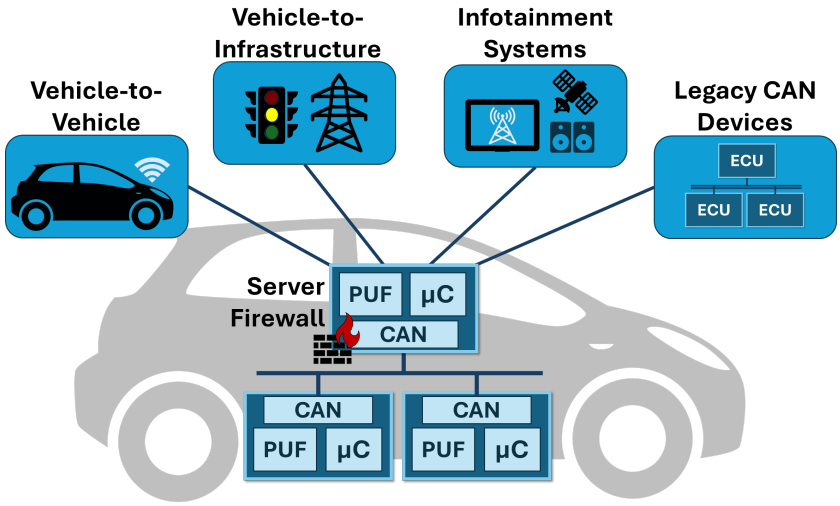


Figure 3.4: Figure depicting the server's role as a firewall and gateway to vehicle-to-vehicle, vehicle-to-infrastructure, and vehicle-to-network communication. The server can also secure, isolate, and bridge other legacy CAN devices, infotainment protocols like Ethernet, and more.

for managing, logging, and securing traffic to and from the vehicle's most critical data path. The external side of the server from the CAN network can then bridge to any other network, such as Lin, Ethernet, IoT, and others. This concept will be utilized in the next chapter for 3D printer networks.

3.6 Conclusions and Future Work

The use of CAN-FD is rising as more vehicles require faster and larger buses to transmit more data, such as smart vehicles. Thus, vehicle security must also utilize CAN-FD to protect against rising threats by utilizing these improved protocols. In this chapter, we proposed PUF-PQC-CANFD, a PUF-utilizing CAN-FD security framework with post-quantum security. By optimizing traffic and carefully considering local public key storage, our design performs similarly or better over the bus than both existing pre-quantum and post-quantum frameworks.

In this work, we utilized SIDH for post-quantum cryptography and authentication. However, due to recently demonstrated attacks on SIDH [72], we have also presented an ideal substitution to the cryptosystems. As this framework's design is independent of the post-quantum cryptosystem, replacement of SIDH to a larger cryptosystem will still provide low message cost and high security. We identified the NIST standardization winner, CRYSTALS-Kyber [76], could replace SIDH, as it would provide strong, standardized security with the cost of larger keys. Our design is optimized for small storage requirements and low traffic, especially in authentication. As standards for communication and ECU systems progress, vehicular security must adapt and evolve to mitigate new threats.

Finally, we also extended upon the CAN network to discuss the application of our design in a real-world, IoT-enabled vehicular system. As the CAN devices are critical communications that talk to many other subsystems, we additionally proposed the use of the server as a real-time firewall and message forwarding system for communication with non-CAN intranet devices.

Exploration into DoS protection would prove useful in preventing adversaries from selectively disabling certain features of the vehicle by flooding the CAN bus. By layering denial-of-service protection alongside this CAN-FD framework, authentication can be used to validate and deny CAN bus access to malicious nodes potentially performing a DoS attack. We further explore denial-of-service protection in the CAN framework in the next chapter.

This work is also versatile, capable of being applied to much more than just vehicular CPS. The use of CAN spreads well beyond intra-system communication, including macro-level interaction with other CPS in a large-scale network. In the next chapter, we will cover a macro-level extension of this work into a popular industrial control system-based CPS architecture, 3D printing farms using CAN buses.

Chapter 4

Quantum-Safe Manufacturing Networks

In this chapter, we propose a new tree-based, root-of-trust framework for quantum-safe manufacturing networks using the Controller Area Network bus [77]. As such, we will approach this on the topic of mass-connected, parallelized 3D printing farms. 3D printing is a rapidly growing industry, receiving massive support from consumers and industries alike. The demand for additive manufacturing services and goods has brought forth huge revolutions in 3D printing technologies as industries develop networks of 3D printing systems, all centered around the ideas of the RepRap movement.

Modern 3D printing systems highlight detailed networking capabilities, complex operating systems and automation, and a significant number of interconnected embedded systems and cyber–physical devices. The internal networks connecting various micro-controllers, sensors, etc., transmit critical information, such as coordinates and safety statuses, through physical communication protocols and interconnections. One such protocol that has seen growth in the market for 3D printers is the Controller Area Network (CAN) [37], which provides an error-correcting, high-interconnectivity

method for fast transmission between several internal and external components of the printers.

Despite this incredibly sensitive information being transmitted on the bus, there are many ongoing vulnerabilities with CAN: its multicast messaging, lack of message authentication, lack of addressing, common point of entry, lack of encryption, multi-system integration, and limited bandwidth. Similarly to vehicular CAN attacks, the structure of 3D printing CAN attacks presents itself in the form of compromising an existing node or including malicious hardware within a new/replacement node. In 3D printers, these compromised devices come from unregulated third-party modules, Wi-Fi based monitoring software, network connectivity to the central controller, and other remote modules.

However, the CAN protocol (and many others) provides only a foundation for communication without considerations for the security or integrity of the data transmitted; while much work has gone into the networking of PC-to-printer (external) communications, little security is typically provided to secure the internal structure of the printer from compromised modules, hardware Trojans, or other adversarial connections on the CAN bus. Because these buses connect the peripherals to the heart of the printer, the transmission of malicious data risks the operability, safety, and privacy of the systems and the users operating them. Modularity and compatibility have contributed significantly to the success of 3D printing as the AM community and industries utilize modified or custom-constructed parts to suit the needs of their applications. This opens up a wide attack surface for adversaries to exploit compromised/hacked 3D printing modules and connections which would provide unrestricted access to unsecured communication. This is a well-known issue for CAN bus applications in the vehicular industry [78], and recently has also been exploited and demonstrated on commercial additive manufacturing systems [44]. Attackers on CAN buses can eavesdrop, manipulate, block, and fabricate messages pertaining to safety-critical, telemetric, and intellectual property (or other private) information to and from all devices utilizing the CAN bus. For example, an attacking

compromised node may at any time transmit a command to overheat the heater cartridge to critically unsafe temperatures. It has been shown in other applications how malicious alterations to commands and data like this attack can result in severe damage and safety risks to both users and property through other forms of attack [27]. Furthermore, denial-of-service (DoS) of the CAN bus would result in disabling most, if not all, devices' connectivity to the central CPU as the bus controls all communication between the devices.

This issue is likely to become more significant in the coming years as 3D printing farms and systems rely on an increasingly many interconnected, variably sourced parts with unknown supply chain security to operate in parallel and keep up with product demand. Many software packages, including the popular Ultimaker Cura, are capable of remotely operating and monitoring sensors, cameras, and other functions of multiple printers simultaneously, which requires more internal components to transmit even more data over the insecure communication protocols. Each new function and degree of connectivity increases the potential attack surface of Industrial Internet of Things (I-IoT)-enabled systems. CAN has also been used prior in communication systems for printer-to-PC connections, operating cameras, safety equipment, and more [38], widening the scope of these vulnerabilities and concerns. Additionally, the development of Shor's Algorithm [50] and Grover's Algorithm [51] have begun a theoretical countdown timer on the modern asymmetric and symmetric encryption and key generation schemes used in many typical or lightweight applications. This post-quantum future should also be factored into the security of critical infrastructure, especially when static, long-term keys are utilized for encryption and authentication purposes (such as server keys).

Three-dimensional printing security is difficult to design for as attacks can be launched using many attack surfaces both internal and external to the printer. A singular security solution, such as adding encryption to external connections of the 3D printer central CPU, is not sufficient to successfully cover the cyberphysical devices and independent systems within the printer that can become compromised

or hold Trojan payloads. Successful attacks on these components can also easily jeopardize all other genuine devices operating within that system. Furthermore, as a core value of 3D printing is compatibility and modularity, it cannot be assumed that the supply chain for 3D printer components is always trustworthy. Thus, these systems must focus their efforts on securing the critical communication that connects, interprets, and uses messages potentially transmitted by adversary-compromised parts. However, manufacturers do not need to turn away from using CAN or networking capabilities within 3D printing and thus from the associated benefits. Instead, strong cyberphysical and data security systems must be designed and implemented while avoiding any major deviations from the CAN protocol to allow for maximum compatibility. To overcome these challenges, we propose a novel solution for 3D printing security with CAN.

Contribution

In this work, we propose a novel, low-overhead security framework that restructures the CAN network to divide and isolate critical components, authenticate and validate messages, authenticate connected devices, and encrypt and obfuscate data from attackers. Unlike existing solutions, our framework is based on a hierarchical tree structure with root-of-trust originating from the root node (or PC), improving our previous CAN 3D printer work [79]. To realize this design, strictly authenticated devices called "routers" forward authenticated, encrypted messages from untrusted 3D printer devices, or "endpoints". Existing security solutions revolve around the need for full trust to be established in devices before allowing them to connect at all to the system. Our framework allows a large array of untrusted 3D printer components to be connected while mitigating the potential damage and impact of adversarial devices, including denial of service attacks. This design is versatile and can support a range of cryptographic suites, from lightweight (LWC) to post-quantum cryptography (PQC). Figure 4.1 demonstrates the overall structure of the framework

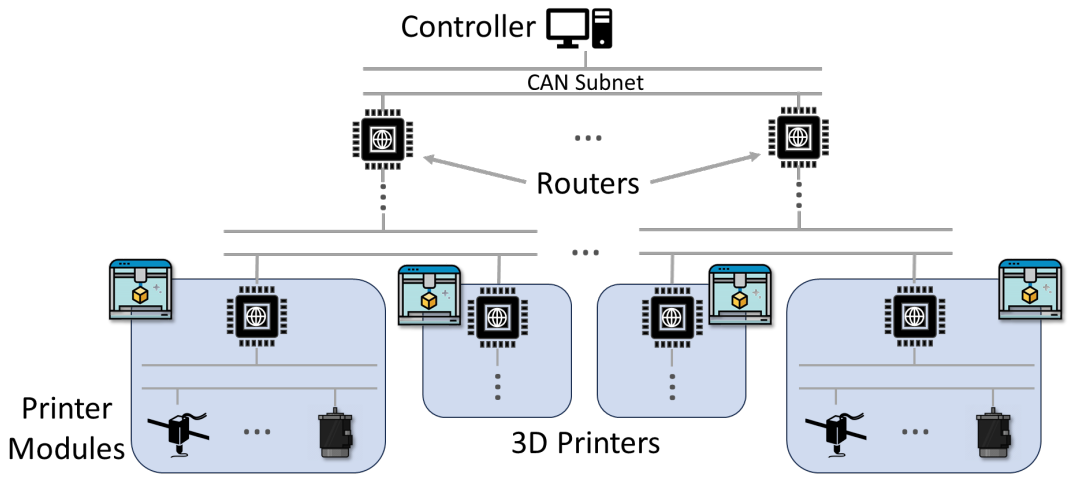


Figure 4.1: Structure of our CAN-based 3D printing framework based on a hierarchical root-of-trust structure. This provides a high-security network for 3D printers to utilize internally and externally to connect a large number of untrusted components.

utilized within an example 3D printer setup. The novel contributions of our design can be summarized as follows:

- A novel root-of-trust-based 3D printing security scheme at the cyberphysical communications level.
- Proven efficient and effective post-quantum security for 3D printers with better performance than many existing CAN solutions.
- Support for a highly expandable array of 3D printers and 3D printer components with strong mitigation of attacks correlated with the supply chain and hardware Trojans (untrustworthy 3D printing components).
- Mitigation of denial-of-service (DoS) attacks via bus filtering methods on individual, isolated buses organized in a flexible tree design.

The organization of Chapter 4 is as follows: Section 4.1 explains our vision and goals for hierarchical networks and cryptography as an AM security solution; Section 4.2 details background information and prior related work on CAN and 3D printing security; Section 4.3 details the proposed framework and its sub-components and operations; Section 4.4 analyzes the security, compatibility, and other considerations of the framework; Section 4.5 provides results from an implementation of the framework and compares them to existing CAN security solutions; finally, Section 4.6 concludes this chapter and provides final thoughts.

4.1 Visions and Goals for a Secure 3D Printing Framework

Typical approaches to CAN security center around acceptance of only strongly authenticated endpoints and additional infrastructure, such as a "server"; while these conventional approaches work within their respective domains, 3D printers

also include frequent component swapping and 3rd party supply chains, which severely restrict the ability for keys and authentication methods to be consistently reprogrammed within each device. For example, reprogramming the root node's secure memory with new authentication keys for each endpoint each time new devices are connected may seriously weaken security against a malicious man-in-the-middle actor. Additionally, typical 3D printer security designs target the connections outside of the printer, such as data transfers or insecure storage of prepared 3D models; while these approaches are very effective at securing the external connections to the printer, the firmware and devices within the printer remain a huge target for adversaries.

Alongside strong security, 3D printer communications must also remain fast enough to continuously stream commands, safety information, and sensor data without severe bottlenecks; while algorithmic speed is not of critical importance, security solutions must take care to keep message transmission reactive and fast. Thus, there are three goals to focus on when designing 3D printer security for the internalized communication modules:

1. Maintain high security while upholding modularity and compatibility;
2. Minimize communication overhead;
3. Minimize changes to existing protocols.

First, a strong level of security that encompasses and protects as much of the system as possible is key to any security solution. Minimizing the damage that an adversary can do is an important goal for 3D printing security frameworks, including protection from denial-of-service attacks. Furthermore, post-quantum cryptography is becoming a necessity to protect modern critical infrastructure, like high-risk manufacturing, from a potential future quantum-enhanced world. Additionally, 3D printing security must also consider upholding component and machinery swap capabilities and compatibility with legacy or other devices.

Second, we must minimize the overhead of the security framework to avoid causing significant bottlenecks which affect safety and vital 3D printing components and equipment; while the timing constraints of 3D printers are far less strict than other CAN applications, speed must still be considered to ensure the protocol does not significantly impact operational speed compared to the original, insecure variant. Controller Area Network Flexible Data-Rate (CAN-FD) provides larger, 64-byte payloads at a faster rate and higher packet efficiency compared to CAN, but still requires multiple packets-per-message to send cryptographic materials like public keys and ciphertext. Depending on the environment's security criticality or resource constraints, the selection of proper encryption schemes can change the number of messages and the amount of overhead imposed on the system by the framework. Thus, minimizing unnecessary packet usage and maximizing the efficiency of messages is also vital in creating incentives for security in devices.

Lastly, security solutions should avoid changes to protocols or damaging compatibility with the hardware and systems already utilized in 3D printing today. Modifying the CAN packet or using sections of the CAN packet for unintended purposes may result in a reduction of the compatibility and feasibility of the system as the CAN back-end and hardware would likely require changes to accommodate for this. Furthermore, changing to a larger, more space-consuming protocol will result in higher costs per part and lessen the incentive for users to adopt security frameworks into their system. One of the most significant benefits of CAN is its ability to maintain a huge number of connections with very few required ports or dedicated hardware components.

Many existing CAN security solutions introduce huge message overhead and/or focus on authenticity checks of all devices within applications which expect few hardware changes, such as within a vehicle's ECU system [80, 5, 4]. Some of these frameworks even alter the structure and intention for multiple fields in the CAN packet, resulting in changes to the protocol and possibly the back-end software and hardware that handle such packets [27]. These solutions would thus result in violating

the modularity of the system and do not fit the 3D printing system constraints. Similarly to the aforementioned frameworks, our design would require a method of generating secure, consistent private keys, as discussed in Section 4.3. As mentioned in prior literature, the use of a Physically Unclonable Function—PUF—in this design would provide node authenticity while being a cheaper alternative compared to costly secure key storage, trusted platform modules (TPMs), etc. [6].

Furthermore, many security frameworks for 3D printers apply solutions that require an external device to monitor cyber–physical sensor/device manipulation and command alteration through adversarial attacks, such as machine learning or block-chain applications [81, 82]; while these methods offer strong protection, they also require significantly higher computational resources, additional hardware, alterations to protocol, and structural changes, and can be easily manipulated by false sensor readings through man-in-the-middle attacks. This becomes an even larger concern when applying these methods in a printer farm of potentially hundreds of 3D printers with individual security needs. Integration of security within the local communication between printer MCUs would lower the risk of a breach and mitigate the damage that a compromised device could do using man-in-the-middle attacks on itself and neighboring devices. Furthermore, it could be integrated with other work to fully secure the 3D printing pipeline and supply chain by design.

Where this work stands in the 3D printing security paradigm is shown by Figure 4.2, which demonstrates our system secures two critical structures of the AM taxonomy. As stated before, much existing 3D printer research work focuses on the security of the IoT and controlling server/PC, such as securing against modification using malware or STL file attacks [39, 83], supply chain and other related attacks to weaken the printer's structural integrity [84, 85], and attacks on vulnerable firewalls from the greater internet on the local LAN communication [86]. Additionally, some work explores improving the network security of the system using various unique encryption or authentication methods [82, 87]. These security systems are discussed further in Section 4.2. Our design provides novel security in networking to other

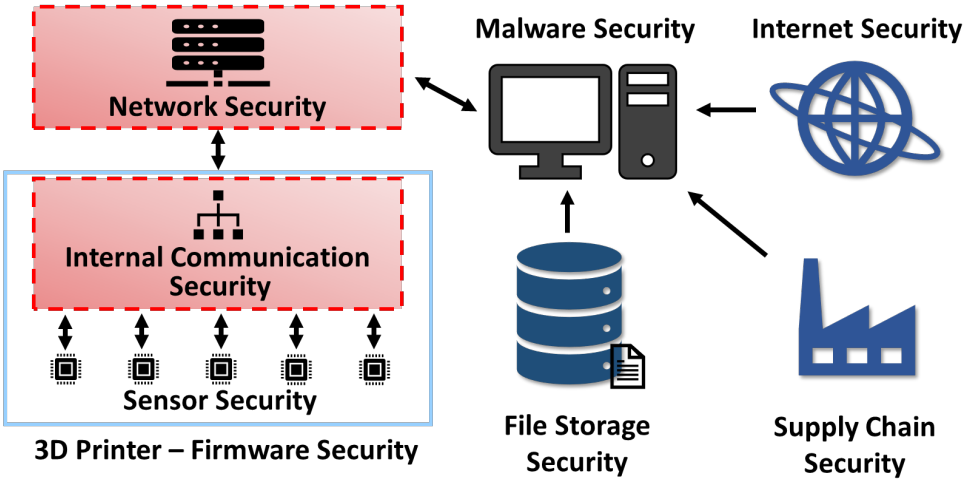


Figure 4.2: The 3D printing security paradigm and taxonomy of each category of security. Our framework targets the Network Security and Internal Communication Security aspects to secure CAN communications inside and outside the 3D printer and network.

printers (or devices), and internal communication security using a secured Controller Area Network bus. Little existing work focuses on this important aspect of security in 3D printers despite their huge emphasis on interchangeable parts. Thus, using our design in conjunction with security targeting other taxonomy topics is greatly recommended.

4.2 Background

4.2.1 Vulnerability Analysis of CAN 3D Printers

A background on additive manufacturing, or 3D printing, and CAN's use in it is provided in Chapter 2. As described in a technical report by Hartzel et al. [1], there are primarily three categories of CAN message assets: (1) private data, (2) safety critical data, and (3) device security data. Private data includes private information of the system's owner that could violate privacy or security, namely printer status information, usage habits, the model being printed, and more. Safety critical data includes data critical to the operation of the system, such as movement or temperature data. Lastly, security data is data used to control security features of the device including 3D printer limit switches, safety switches, air purifiers, etc. The types of data vulnerable in CAN 3D printing devices are given in Table 4.1. The table is sorted by category of CAN data, ranging from user's private data (STL files, settings, user accounts, etc.) to even the security data used to monitor other types of attacks (i.e., security cameras, security updates within firmware, etc.).

4.2.2 Prior Related Work in CAN and 3D Printing Security

Security solutions for 3D printing systems have focused on utilizing the power and capability of the controller PC, such as a server desktop, to perform intrusion detection and encryption of g-code or other command and model data. Existing efforts in this direction include digital audio signatures captured from on-site monitoring [84],

Table 4.1: CAN-based 3D-printer data vulnerable to CAN attacks based on category of critical data/functionality.

Data Category	Type of Data	Proposed Method of Attack
Private User	STL/G-Code Data	Eavesdropping on positional data sent to motor or printer head modules via CAN bus.
	Printer Status	Eavesdropping/Fabricating status information transmitted to controlling or monitoring system.
	User Preference/Settings	Eavesdropping on transmitted user settings, such as Wi-Fi networks, account data, or device identifiable information that may be sent via CAN.
	Device Calibration	Spoofing and DoS during calibration can result in miscalibrated and out-of-sync modules.
Safety Critical	Temperature/Sensor	Manipulating reported sensor or temperature readings sent to central controller.
	Target Temperature	Spoofing temperature target commands sent to heating modules (i.e., printer head).
	Limit Switches/Position	Fuzzing/Spoofing limit and positional data to misalign, miscalibrate, or control device manually.
	Safety Device	Spoofing/blocking safety features (i.e., air filters, kill switches, fire alarms)
	General Data/Commands	Overwhelming/Manipulating the physical bus via DoS to block vital commands or data.
Security	Cameras/Monitoring	Spoofing or DoS attacks on printer farm monitoring devices, such as cameras or sensors in [88].
	Module Firmware	Software updates allow firmware flashing through CAN [89].
	Settings/Config Data	Spoofing/Sniffing device data (i.e., device settings, security data).

and machine learning anomaly detection of physical defects and cyberattacks [85, 81]. Additionally, encryption, hardware-rooted trust, and embedded watermarking of 3D printed parts have been addressed in prior work [87, 83], including the use of blockchain for cryptographic security [82]. An extensive survey was performed that covers existing vulnerabilities and security measures, including the physical communication protocol vulnerabilities we address in this work [86]. However, little work has been conducted to mitigate these vulnerabilities, which demonstrates the novelty of our work in the field of AM security.

Methods of protecting the CAN bus range from server ECU-based authentication of every CAN device to intrusion detection on the CAN bus. As this is similar work to Chapter 2, this rehashing of previous literature will be in context of 3D printing CAN networks. ECU authentication methods usually focus on statically defining and storing cryptography or identification material that can provide authenticity, such as digital certificates [4], PUFs [6], firmware digests [90], and co-channel RF watermarking [91]. Furthermore, some work has been done on Post-Quantum Cryptography for CAN encryption in vehicles [4, 61]. An in depth survey on intrusion detection using machine learning was previously proposed that studied detection methods for DoS attacks, fuzzing attacks (injection), impersonation attacks, and more [92]; while these methods are powerful detection schemes, preventing intrusion and malicious attacks before they can happen would be preferable, especially for alternative CAN bus applications like 3D printing.

4.3 Design of Our Proposed Hierarchical CAN Framework

Our proposed framework for both internal and external/networked 3D printer connectivity utilizes a hierarchical, tree-based design for security with root-of-trust authentication and encryption. The source of trust originates from the controller

PC and is negotiated using two-way authentication down each layer/depth of the tree by "routers", providing a strong, highly secured, and authenticated design for infrastructural devices. We recommend authenticity be provided using Physically Unclonable Functions (PUFs), but the choice is abstracted away in this framework to support any form of secure storage/generation method (i.e., TPM, secure storage); while other communication schemes are compatible with our design, we focus on and recommend the adoption of CAN-FD (which we will generically refer to as CAN) for communication within the 3D printers for its multiplexing broadcast, built-in error, collision correction, and sufficiently-sized payloads. Our design converts a singular CAN network into multiple subnets controlled using "router" devices that handle authentication and message forwarding between the various subnets along the tree's greater network. Endpoints, or connected 3D printer components, develop individual trust to the router and are isolated in groups to mitigate potential impersonation and eavesdropping of critical components of the bus. The general pipeline design is shown in Figure 4.3. All three sections of this design are detailed in the following subsections.

Our work can support any form of cryptosystem that fits the block size and security requirements for quantum-safe PQC designs. However, we will focus specifically on PQC-based designs. Our PQC framework requires 256-bit security or higher symmetric encryption with block-sizes smaller than 56 bytes, a quantum-safe asymmetric scheme with at least 128 bits of security, and a hashing algorithm capable of generating at least 256-bit hashes (for 256-bit keys). In this PQC framework, we use AES256-GCM [93], CRYSTALS-Kyber-512 [94], and SHA3-256 for symmetric, asymmetric, and hashing functionalities, respectively; while AES and SHA256 are well known and standardized cryptosystems, CRYSTALS-Kyber is the recently selected NIST standard for post-quantum asymmetric cryptography [52]. Kyber-512 utilizes public keys of 800 bytes, ciphertexts of 768 bytes, and is computationally expensive, requiring our design to carefully consider performance, storage, and message cost constraints. Additionally, AES256-GCM mode is utilized

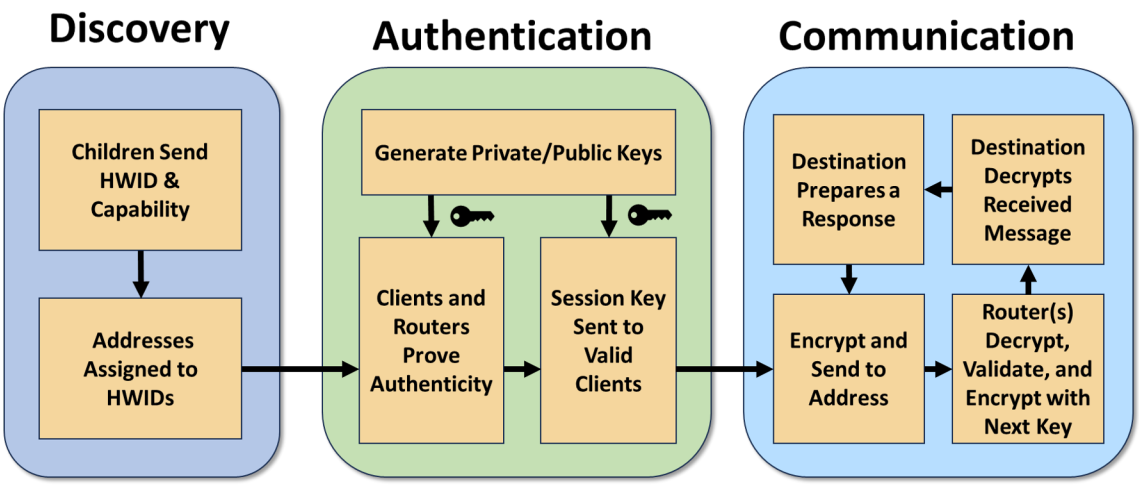


Figure 4.3: Pipeline design of our framework from initialization to normal operation for a singular subnet. The three distinct phases of our design are discovery, authentication, and communication (including capability and address requesting). Each subnet should perform these steps upon startup.

for its counter and authenticity tags. Authenticated encryption with associated data (AEAD) is utilized in both symmetric schemes to ensure credibility of the unencrypted arbitration data of CAN and our framework. These cryptographic systems may be swapped out with other post-quantum algorithms, but careful consideration of security level should be carried out when doing so.

The proposed framework has three core elements: discovery (and addressing), authentication and cryptographic material establishment, and normal communication. The discovery and authentication will be performed upon every startup of the system, generating new, high-entropy keys for normal communication each time. After startup, address requesting can occur throughout the tree to request capability and address information of all the nodes in the system. This is vital for a system with a non-deterministic structure and interchangeable parts that can have a wide variety of configurations (such as 3D printers with CNC tool capabilities). The rest of this section describes each of the phases in greater detail.

4.3.1 Discovery and Addressing

As seen in the discovery section of Figure 4.3, node discovery is the first phase of our design, and the process by which nodes in a given subnet make themselves known and are assigned an address by their parent router. The overall structure of this process is similar to that of the Dynamic Host Configuration Protocol (DHCP), but is initiated through the parent router, rather than through connecting nodes. The discovery process begins with a broadcast by the parent, which prompts all nodes to respond to be assigned an address. The response packet contains the node's Hardware ID (HID) and other fundamental information, including the primary capability of the node. Because the node has yet to be assigned an address, the nodes respond as the broadcast address, with the HID temporarily serving as a means to differentiate nodes. The parent router then sends a broadcast in response to each node, containing that node's HID and assigned address. After a configurable wait time (125 ms in the

current implementation) the parent node stops listening for discovery responses and begins the authentication process. The protocol was designed with standard CAN hardware in mind, such that devices will always have a CAN filter listening on the broadcast address, and configure an additional filter to listen for their assigned address once they receive one.

Our framework uses the 11-bit CAN ID field to perform local addressing within a subnet, alongside a forwarding system which uses variable-length addressing to allow communication between subnets. Local addresses are assigned by the parent router of each subnet during discovery (addresses are sequential in the current implementation). Because the tree structure allows only one route between each pair of nodes, our variable-length addressing scheme uses the intended route of a packet as an address. An example is shown in Figure 4.4. As shown in the figure, the variable length address is composed of the local address of each intended hop for a given packet. In addition to removing the need for routers to possess complex routing logic, this system also allows the source address for a message to be gradually constructed as the message is forwarded along the tree. After each hop, the portion of the address associated with that hop is overwritten with the return address for that hop. Performing this step after each hop and inverting the hops in the resulting address produces the return address of a given message. Calculating the return address for a message in this manner also makes spoofing a message inherently difficult: even with no additional spoofing protection, nodes can only spoof nodes in their local network, and a router can only spoof parts of a route which precede them. We also implement a Time-To-Live (TTL) field, which serves the dual purpose of preventing infinite circular routing and enabling routers to keep track of the next intended destination of a packet.

4.3.2 Authentication and Cryptography Establishment

Authenticity is a very important factor in the security of this framework's infrastructure, such as routers and root connections. After initial device discovery and

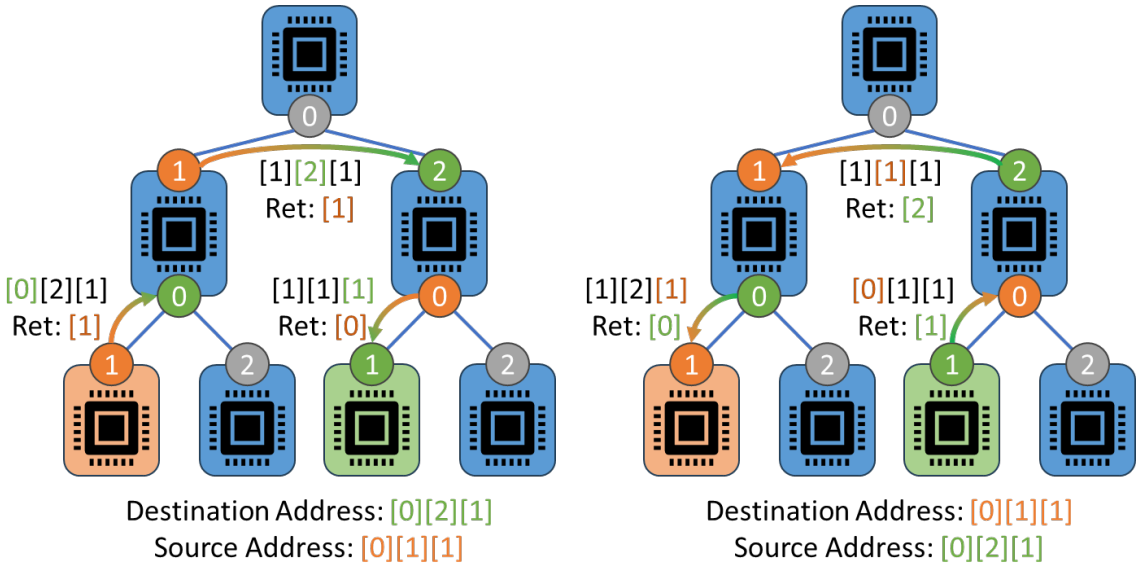


Figure 4.4: Addresses and routing between a pair of network nodes. The left side shows the route from the orange node (bottom left) to the green node (bottom center), while the right shows the reverse route. Each route is composed from a series of local network address "hops". The current address is replaced with the return address after each hop, allowing the return route to be calculated.

address allocation, the network begins the authentication phase, indicated in Figure 4.3. In this phase, the trusted PC or micro-controller acts as the controlling—or root—node. This node begins a two-way authentication process with the routers, which then validate their own child subnet (for which they are the server/parent). As this authentication process cascades down the tree (in parallel), each router will authenticate their parent and any routers within their child subnet (two-way auth), ensuring all devices trust their parent and prove trust to their children. Finally, the endpoints will perform validation of their router (one-way auth) and establish secure communication if the router's authenticity is accepted. Alternatively, endpoints can also be two-way validated in the same scheme as the routers. Trusted or accepted child nodes will receive a randomly-generated session key that will be used in subnet communications and salt the parent session key used for secure forwarding communication. Any node that is not accepted should not receive the session key and should be considered disabled. To minimize timing requirements, the session key can be transmitted safely concatenated to the response hash, as the child device (if applicable) should already be authenticated. Under no circumstances should the shared key be used for normal communication to minimize the risk of key leakage.

Post-Quantum Cryptography Framework

The entire authentication process for the clients (child) and router (parent) is secured with post-quantum cryptography. The PQC authentication steps are illustrated in Algorithm 2. This scheme uses public-key cryptography (PKC) to encrypt shared secret material with public keys and generate a shared AES-256 key of 32 bytes. To begin, after key initialization (Steps 1–4), an additional initial step of transmitting the client's public key (Step 7) can be utilized to save on precious embedded resources on the chip. This public key can be hashed and compared to a stored, smaller 32-byte hash stored in the parent/server for the two-way (or router) authentication scheme. Alternatively, these public keys can be stored within each system to reduce the number

Algorithm 2 Post-quantum cryptographic authentication (1-way and 2-way)

Input: $Clients$ **Output:** $Session\,Keys, Valid\,Nodes$
Generate private key S_C and private key S_R for Clients and Router, respectively.
Router hashes private key to generate challenge hash H_R.
Router generates a random Session Key, $K_{Session}$.
Clients store Router Public Key P_R and Router uses Clients' Public Keys P_C.
for $i \leftarrow 1, numClients$ **do**
    Client $\leftarrow Clients_i$
    Client transmits their Public Key P_{C_i} to Router, if necessary.
    if Client is also a Router (Two-Way Auth) **then**
        Router should validate hash of transmitted P_{C_i} with stored hash.
        Client hashes private key to generate challenge hash H_{C_i}.
        Client generates random 32-byte key Ch_C and encrypts with key P_R.
        Client transmits Kyber-encrypted Ch_C to Router.
        Router generates random 32-byte key Ch_R and encrypts using key P_{C_i}.
        Router transmits Kyber-encrypted Ch_R to Client.
        Router decrypts Ch_C using S_R and Client decrypts Ch_R using S_{C_i}.
        Both combine keys to make Shared Key: $SSec_i \leftarrow (Ch_C \oplus Ch_R)$.
        Client sends encrypted H_{C_i} to Client using $SSec_i$.
        Router decrypts H_{C_i} using $SSec_i$ and validates it to stored authentic hash.
    else if Client is an Endpoint (One-Way Auth) **then**
        Client generates random 32-byte Ch_C and encrypts using public key P_R.
        Client transmits Kyber-encrypted Ch_C to Router.
        Router decrypts Ch_C using S_R.
        Router uses challenge as Shared Key: $SSec_i \leftarrow Ch_C$.
    end if
    Router sends encrypted H_R to Client using $SSec_i$.
    Client decrypts H_R using $SSec_i$ and validates it to stored authentic hash.
    Router sends encrypted Session Key $K_{Session}$ to Client using $SSec_i$ shared key.
    Both generate Router Session Key by $SK_{Router} \leftarrow Hash(K_{Session}|SSec_i)$.
end for
Router signals to all Clients that authentication is complete.
Return Session Keys, Valid Nodes

of messages required. The public key transfer in Step 7 is non-optional for one-way endpoint authentication as the endpoint's public key is not known to the server.

In the two-way authentication scheme, both the client and router will encapsulate a random 32-byte shared value with Kyber using the other's public key and trade the ciphertext (Steps 8–18). Next, they decrypt the information and XOR the two generated keys together to generate a combined authentication key (Step 16). In order for this secure channel to work, both sides must be authentic (the owners of the respective stored public keys) to ensure that the XOR produces the correct value, providing a strong authenticity check. Using this secure channel, they will then trade known ciphertexts, such as PUF response hashes, before the server provides the 32-byte randomly generated session key sequence to the client (Steps 25–28). In one-way authentication (shown in Steps 19–23), only the client will transfer a 32-byte challenge ciphertext and both parties will use this for the shared key authentication (Step 23). The client will utilize the session key, $K_{Session}$, for CAN inter-network communication. Similarly, a unique router session key, SK_{Router}, is also generated using a hashed $K_{Session}$ salted with the shared key $SSec_i$ by $SK_{Router} = Hash(K_{Session}|SSec_i)$ (Step 28).

4.3.3 Normal Communication

The final phase of our design is normal communication, represented as a loop in Figure 4.3. When any device wishes to connect to a node outside of its own CAN subnet, it will first ensure they have acquired the address of the destination from an authenticated authority node, such as requesting the root. The device will then need to fill in the appropriate address, length/TTL, data, counter/nonce, etc. All packets' data fields must be encrypted (unless defined otherwise by the application) using the most up-to-date pair of session keys established in the authentication phase, including the tag for authenticity checks. For communications to routers, devices must encrypt using the router session key generated previously for secure

router-to-endpoint communication. Otherwise, local subnet communication can be encrypted with the subnet session key to talk to neighbors without the need for router forwarding. When encrypting packets before transmission, associated data (AD) or similar features (authenticated encryption with associated data—AEAD) should be utilized to prevent the CAN headers and our framework headers from being modified, as they cannot be encrypted. Routers will forward the packet information based on the address and TTL fields to the correct location, destroying/stopping any packet that does not pass the encryption and authenticity check and application-designed firewalls (if any). This general flow from sender to receiver can be seen in Figure 4.5.

In normal communication, all messages are encrypted using the run-time session keys generated previously in authentication. This encryption utilizes all the additional security features included with the chosen symmetric encryption scheme, including authenticity tags to prevent fuzzing/impersonation and counters to prevent replay attacks. The typical CAN-FD payload offers a 64-byte data field to operate with. In our normal communication, eight of those bytes are reserved for the routing system to allow for header information (counter, address, TTL, etc.) without compromising the CAN-FD protocol itself. That allows for 56 bytes to be utilized for the data, including the tag (which is usually 8–16 bytes). While our protocol takes up some of the CAN-FD data field, it remains still faster and more efficient than the 8-byte fields of CAN and is still efficient enough to maintain 3D printing payload and performance requirements as shown in Section 4.5.

In addition to basic communication, our framework provides a few simple functions to components connected to the network, most notably the ability to request addresses or capabilities of network nodes. Each node is associated with a capability which describes the function of the device; examples include servo, temperature sensing, and fan capabilities. We include a command for requesting the network address of nodes with a given capability along a subnet, which allows, for example, the address of a temperature controller on a subnet to be retrieved so that it may be queried for temperature data. We also include a command to request the capability of a node at

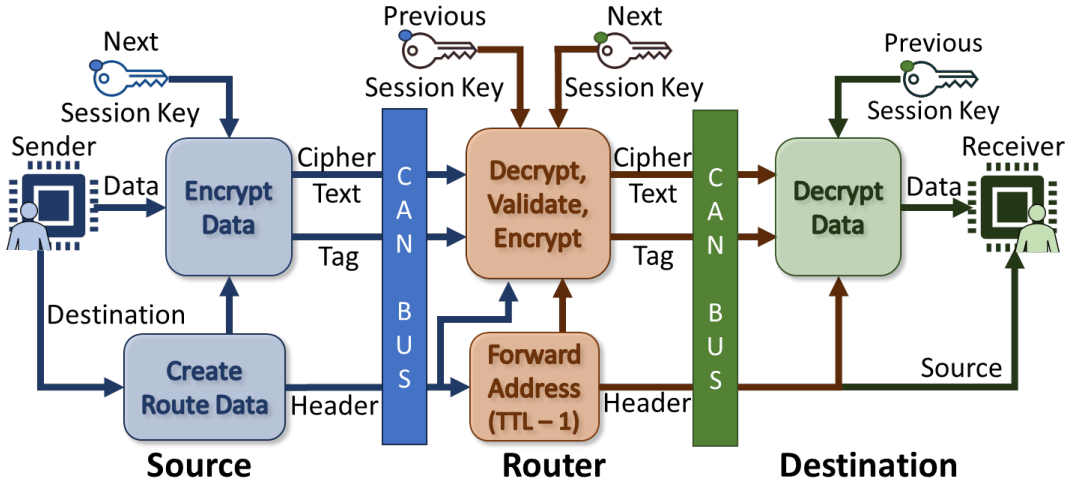


Figure 4.5: General flow of standard communication depicting a sender transmitting through a router to reach a destination. Source (blue) creates the route and encrypted data to send to the first "hop" in the series. The router(s) (orange) will decrypt with the previous bus' session key, validate, and re-encrypt the data using the next unique CAN bus parent session key. The router process is repeated until the data reaches the destination (green). Addresses and TTL are modified after each hop.

a particular network address, which enables inspection of the network by routers or the root node.

4.4 Analysis and Discussion of Proposed Framework

In this section, we discuss the security, implications, and design justifications of our proposed design, as well as compare our design to existing methodologies in both CAN and 3D printing. By implementing a hierarchical authentication structure instead of the typical CAN bus, we can protect against attacks that even many popular existing frameworks remain vulnerable to. One such example is denial-of-service mitigation, as minimizing the number of devices dependent on the disabled CAN bus allows most of the tree network to continue normal operation during attacks.

Furthermore, our root-of-trust security scheme for both crypto-suites provides robust, secure communication and trustworthy infrastructure to both 3D printing networks and components. As communication keys are ephemeral, compromising any node in the system would require physical access to the secure private key storage/generator (i.e., PUF) to retrieve the means to authenticate and generate the appropriate keys. These private keys should be retrieved from secure storage or generated at run-time as needed and should not be cached or stored in insecure non-volatile memory to avoid key leakage. As public keys are non-sensitive information, they do not need to be kept secret and can be stored within unsecured non-volatile memory, such as flash memory. Endpoint nodes' public keys do not require permanent storage as these devices can be swapped among routers and are run-time dependent. As hashes are considered irreversible operations, these too can be stored in insecure memory as adversarial theft of hashes would not compromise the system.

Use of the long-term PQC private/public keys, should be limited to authentication and ephemeral key transfer only. All encrypted communication after authentication

should utilize the temporary session keys generated. Re-authentication of nodes based on a message count or timing system can be configured to provide additional protection against attacks on session keys. This would also give the system the ability to identify devices which have been tampered with or disabled during real-time operation. This process does not require the devices to regenerate and transmit public keys or shared keys. If new routers are added to the system, devices that authenticate them (their parent and children) will require re-enrollment of their public keys, hash, and other necessary data.

Because we utilize a secure tree structure with maximum modularity, our framework is essentially infinitely expandable. This framework can also be utilized in both the internal CAN connections and expand to connect external, networked CAN 3D printers. Due to the design of the framework's header, the tree structure we utilize does not have to be balanced, nor do endpoints have to always be at the bottom level of the tree (although, we recommend only two-way authenticated endpoints for this to minimize the damage of endpoints performing DoS attacks on the subnet routers). This design means that even legacy 3D printers, with or without the CAN bus, can also be connected to the CAN network using a compatible CAN device to bridge the connections and protocol. Our design is also compatible with Marlin-based firmware. The depth and width of the tree and its branches/subnets are heavily dependent on the parameter of max devices allowed in a subnet, or k. If k is too small, then the network could suffer increased latency due to the increased tree depth required to connect a given number of devices. However, a larger value of k will increase the impact an attacker can have on the system, as more devices will be located in the same subnet as the attacker. As shown later in Section 4.5, the relationship between the number of endpoint devices n and the selected parameter k is logarithmic in nature.

Additionally, CAN-based 3D printers, such as Snapmaker [36], already utilize a discovery and capability phase in their device startup, making our design easily drop into existing CAN-based 3D printing firmwares. However, our goal is also to

encourage further use of CAN/CAN-FD in current and future 3D printing devices and applications to improve their design and capabilities. Thus, providing a security solution like this is one of the first steps in promoting such communication design choices.

4.4.1 Normal Communication and Routing

Routers in normal communication are tasked with routing packets between the CAN buses above (parent subnet) and below (child subnet) them appropriately after checking for authenticity and valid routing. Based on typical 3D printer communications, most communication in this stage will be local subnet communication, broadcast commands (i.e., g-code or requests) sent by the root, and query responses for physical sensor/status readings. By routing this way, we also substantially reduce the load on any singular CAN bus in systems with high node counts, as more information is being broadcast down or is isolated in respective subnets. When routers perform forwarding/hopping, each subnet's encryption key will be used on each hop (decrypt using previous key, encrypt using next key) to ensure that new keys or existing keys do not have to be exposed outside of subnets, strengthening the security of the system and minimizing the number of devices that contain these secure keys. As this is a low-level framework, user applications have the ability to also include an additional layer of protocol to create end-to-end encryption between the devices, such as an SSL-like protocol.

4.4.2 Post-Quantum Cryptographic Schemes

The performance impact of a pure-software PQC implementation may encourage users to find faster implementations, such as hardware acceleration which can occur in parallel alongside the initialization process. Many TPM modules are beginning to include post-quantum cryptography, including CRYSTALS-Kyber, in their hardware acceleration designs, allowing users to have both a secure method

of storing/generating private keys and of generating the necessary public-key cryptography in one package [95]. This would allows users to continue using low-cost micro-controllers while having the benefits of PQC. However, many low-cost, highly optimized implementations of CRYSTALS-Kyber have been proposed that could also be implemented alongside the PUF to provide similar benefits [96, 97]. Otherwise, as shown in Section 4.5, the pure-software implementation of the cryptography still provides high performance in normal communication and acceptable speeds in authentication, especially given that initialization steps have less constrained timing requirements.

Asymmetric Key Generation

For public key cryptography in our PQC design, we chose the NIST standard CRYSTALS-Kyber [94] which targets a 128-bit post-quantum security level for public key cryptography. This provides excellent security against classical computing as well. As this asymmetric scheme is different than Diffie–Hellman, the PQC framework was modified to accommodate the differences.

Authenticated Encryption and Decryption

As quantum experts indicate that the security level of all symmetric ciphers are essentially "halved" against quantum computers, we chose the standardized AES-256 cipher to match the security level of Kyber. However, any cipher with 256-bit security and 16-byte block sizes is acceptable. We choose to use the Galois/Counter mode (GCM) to provide replay attack and message authenticity protection. By utilizing associated data (AD), we can include the unencrypted header fields in the authentication to prevent tempering.

Hashing

Our hashing function should have at least 256-bits to ensure hashed keys have 256-bit security with AES. Comparison hashes, such as the private key and public key hashes, can be truncated to 128-bit to save memory. This also allows for the transferred challenges to be exactly 16-bytes large to match the AES block size. We chose SHA256 for this task.

4.4.3 Threat Mitigation

One of our framework's primary capabilities is protection against eavesdropped and fabricated messages sent to authority nodes, such as the root device. In standard CAN configurations, an attacker could easily fabricate messages to any destination, including impersonating other devices. The main attack surface our design's security protects is the bottom-most endpoint (or 3D printing component) layer as many components sourced from potentially insecure supply chains are often swapped around and replaced. We do not discuss concerns of the root (or controller) being compromised as this is outside the scope of this work and has already been addressed in a significant amount of the literature.

Our framework provides source and destination authentication using unique keys, challenge–response authentication upon startup, and counter/tag cryptographic systems. This allows us to prove the validity of and prevent impersonation for all messages being sent in the system. Because the data are in a secure, encrypted state with uniquely modified session keys, eavesdropping attacks are counteracted. The only information an attacker is capable of retrieving is metadata on the CAN header (i.e., destination ID) or framework header (TTL, Addr, etc.), which are all unencrypted for routing purposes. The generated tag (and associated data) secures our design from fuzzing attacks and packet manipulation attacks as secured MACs are calculated and transmitted with every packet to validate both the encrypted and header information. Replay attacks on the authentication phase would

immediately be stopped by the randomness in challenge keys and session keys of the frameworks, preventing nodes from impersonating devices during startup. In the case of transmitted public keys, these public keys are hashed and compared to stored registered device hashes. If manipulation occurs during discovery or other initialization phases, it will only result in a denial-of-service behavior that will impact only a single subnet, creating a similar scenario to the CAN subnet simply being disabled.

If re-authentication is utilized, session keys in both of the presented cryptographic schemes change rapidly enough to significantly reduce the impact of even cracked session keys. This does not eliminate the threat of eventual eavesdropping (as past messages can be decoded at anytime if a session key is cracked), it does reduce the impact of any individual key being compromised. Because the session keys are generated and transferred randomly during run time, our design provides forward secrecy (FS). Even if the private key is eventually compromised, the attacker would still be required to brute-force the session key of past communications. If the user requires it, endpoints can easily be reconfigured to perform two-way authentication with the router, requiring the endpoint to be statically defined in the router's known-authentic device table.

By having routers perform the routing of the packets, a firewall is inherently provided that blocks impersonated traffic, malformed packets, and unauthorized routing. This also means that malicious nodes cannot read traffic coming from other subnets, preventing them from extracting any information. We believe these benefits to greatly outweigh the message/traffic cost of having to redeploy messages along the bus. Denial-of-service attacks cannot breach the inherent rate-limiting and authenticity checks of the router, thus protecting our system from network-wide denial-of-service. As such, attackers are limited to a potential attack surface of only k subnet nodes, which our additional security measures protect, authenticate, and secure.

4.4.4 Comparison to Existing Security Frameworks

Our framework has a number of unique features compared to existing 3D printing and CAN security frameworks. However, we can still provide relevant points of comparison between this work and existing work to determine where our work stands in the 3D printing security and CAN security environments. The security holes which exist in these systems have produced a variety of security solutions in either domain. No single solution is able to cover all aspects of the additive manufacturing supply chain and devices. This section will compare our solution to both existing CAN frameworks (including Vehicular ones) and existing 3D printing models. These CAN comparisons are shown in Table 4.2.

Our design uniquely reshapes the CAN bus in a way that allows CAN devices to have safe, confidential transmissions to and from the central device. Existing frameworks were usually designed with different constraints (vehicular) than our work, resulting in features unsuitable for our application or a lack of vital characteristics required for 3D printing. However, many aspects of our work and existing CAN systems are comparable, including authentication and communication/security along the bus. CAN frameworks are usually structured as a singular bus which contains a server that performs the authentication of nodes, similar to ours. Two-way authentication is the only allowable option for other CAN frameworks. In our design, we must accept 3D printers into the system which could potentially have hardware Trojans and mitigate their damage.

Most designs, including this work, utilize secure private key storage or generation, such as PUFs, in place of hard-coded values or other methods. Our framework utilizes counters and tags to ensure received messages are validated before being processed to prevent fuzzing, replay, and other forms of bus manipulation. Frameworks lacking these features are vulnerable to identical and random ciphertext attacks. Finally, our design maintains a degree of denial-of-service protection, which was an important goal for our design. The tree-shaped structure of our framework prevents attackers

Table 4.2: Comparison of our framework to existing CAN frameworks. As the target application is different, only typical CAN security criteria and security requirements are considered to ensure fair comparison.

	CAN/CAN-FD Frameworks			
Criteria	**PQC Framework**	**Labrado [6]**	**Siddiqui [5]**	**LASAN [4]**
Auth Messages	$26n$	$4n$	$3n^2 + 2n$	$321n$
Encryption Type	Post-Quantum	Lightweight	Standard	Post-Quantum
CAN Bus Structure	Tree-based	Direct	Direct	Centralized Router
Secure Private Key	✔	✔	✔	✘
Forward Secrecy	✔	✔	✘	✔
Replay Defense	✔	✔	✘	✔
Plug-and-Play	✔	✘	✘	✘
Message Authenticity	✔	✘	✘	✔
Eavesdropping Defense	✔	✔	✔	✔
Fuzzing Defense	✔	✘	✘	✔
Impersonation Defense	✔	✔	✔	✔
Server Verification	✔	✔	✘	✔
DoS Defense	✦*	✘	✘	✘

* Denial-of-Service attacks limited to affecting only adversary's subnet.

from accessing other subnets, mitigating the much of the damage other frameworks would be vulnerable to.

Similarly, some comparisons of the general impact and targeted security features of 3D printing security designs can also be qualitatively performed to get a better picture of where our work stands. No solution will be able to provide complete security, making it important that these solutions be combined to provide more comprehensive coverage. As our work targets a domain others have not investigated deeply (communication), our work defends against some adversarial threats that fall out of scope for other security solutions [81, 87, 84, 82]. Our design actively protects the communications system from tampering, eavesdropping, and impersonation as described previously. As our work is directly located within the 3D printer's systems, it does not require external computers or servers to perform monitoring of audio, print abnormalities, etc. or host machine learning applications. However, this means that while our system can detect and prevent most attacks entirely, any other forms of attack will not be discovered by our system (alongside some others) and would require intrusion detection models like Beckwith et al. [81] or Belikovsky et al. [84].

4.5 Experimentation Results

To draw conclusions about the performance of our framework, we created a working, generalized implementation of our design on a CAN-FD network composed of Microchip SAM C Arm Cortex-M0+ based micro-controllers. These ARM-based MCUs have a clock frequency of 80 MHz, 256 KB of flash, and 32 KB of RAM. An example configuration of this implementation is shown in Figure 4.6. To ensure that encryption suites are accurately compared, we utilized only software implementations of official sources for every cryptographic algorithm. The tests we performed include the time taken to authenticate a subnet based on k (Figure 4.7) and the CAN message cost for authentication compared to other frameworks. For comparison of the overhead of PQC results, we have included a lightweight variant of this framework [77]

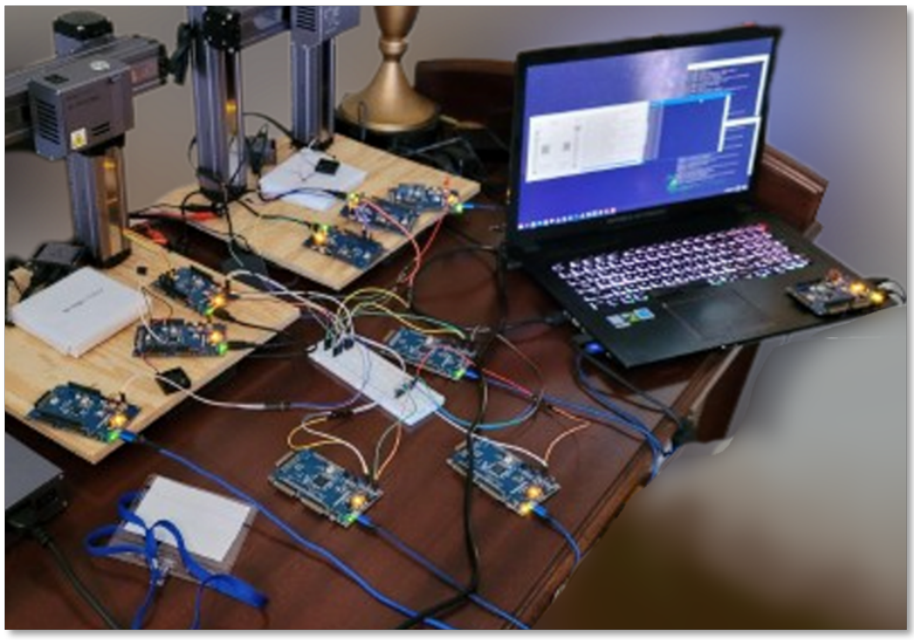


Figure 4.6: Full test bed of our framework implementation, with attached 3D printer components. Network nodes are implemented by SAM C21 micro-controllers with servo, temperature sensor, and fan components attached to the endpoints.

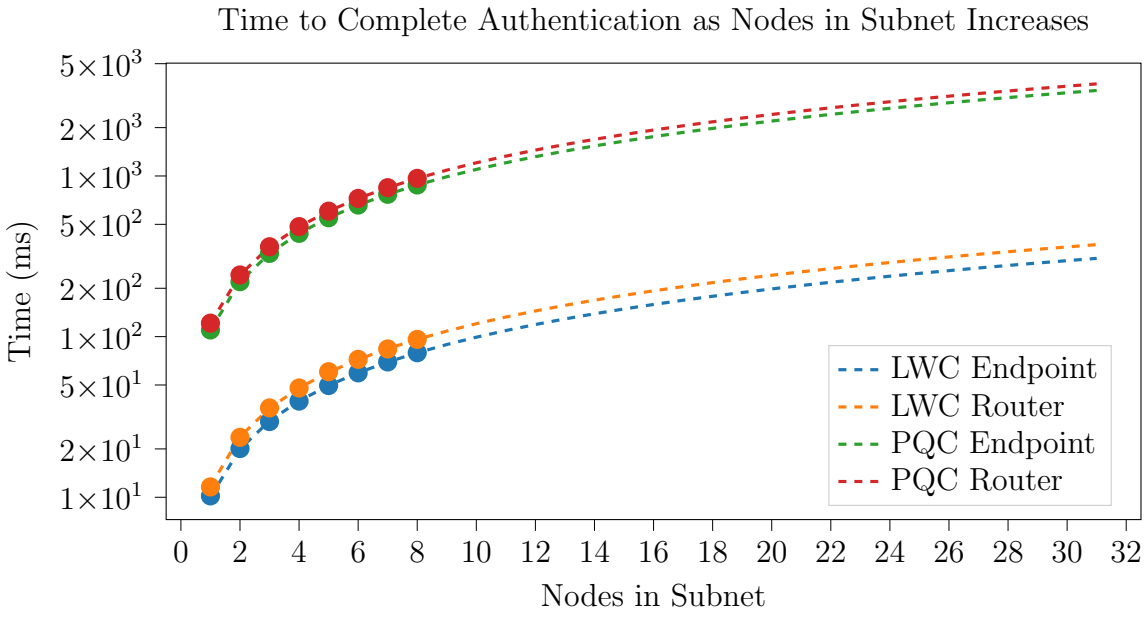


Figure 4.7: Results of time to complete authentication, in Log(y), for the number of nodes in subnet k. Data points extrapolated by line of best fit for high values of k. Post-quantum performance gives a 10× overhead, but both are well within acceptable timing ranges for expected values of k.

in the graphs, denoted with "LWC". This section will discuss these implementation test results.

4.5.1 Authentication

The authentication process which is performed on initialization is vital to the security of our design. This step provides the root-of-trust and establishes all the cryptographic keys and structures that will be used for the entire rest of the session. Thus, providing sufficient security to prevent attackers from being incorrectly validated as critical infrastructure was the core focus of our design. Since this stage is performed while the system is relatively inactive (during initialization) performance is not a crucial metric. However, a strong CAN framework should still consider optimization, message cost, and total run time to prevent authentication (and especially re-authentication) from bottlenecking the system, blocking critical safety messages, and/or removing incentive for users to adopt the framework.

Our framework attempts to minimize the numbers of transmissions that are required for authentication. To demonstrate how strong our authentication scheme is, we have compared the number of messages required to that of existing CAN security frameworks to get a general idea of how efficient each framework is. The number of messages for each value of k CAN bus nodes is located in Figure 4.8. Our PQC framework performs better than existing CAN post-quantum frameworks, such as Ravi et al. [4], requiring approximately 90% less packets. All four frameworks also outperform non-linear scaling frameworks, with the performance improvement increasing for higher node counts. Our results include the public key transfer in PQC mode for both routers and endpoints, resulting in equal message costs.

Authentication is separated into either two-way or one-way validation, as described in previous sections. We gathered timing results for both endpoints (one-way) and routers (two-way). The results can be seen in Figure 4.7. Our graph is extrapolated for values of k from our experimental results to see the trend for larger

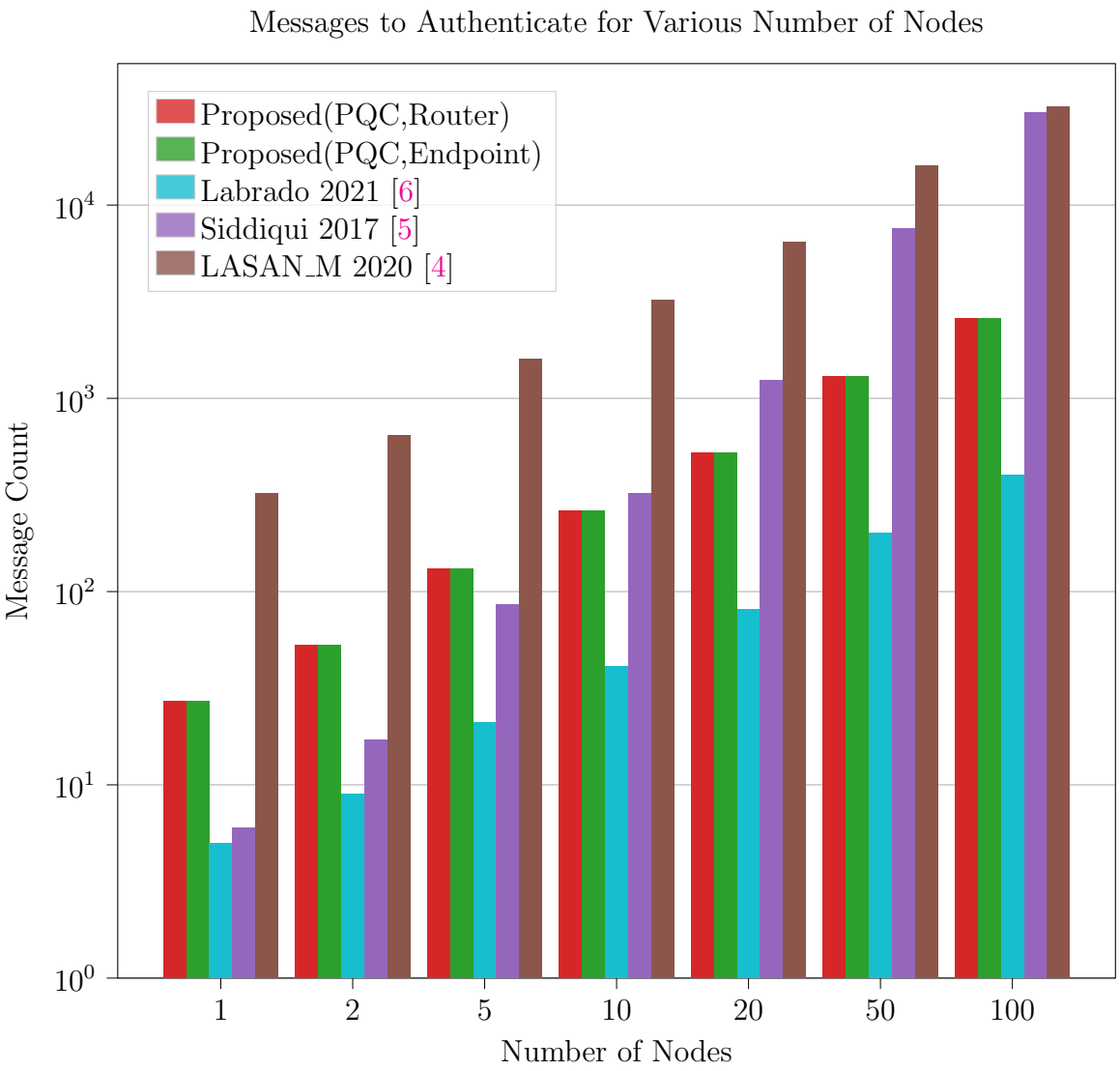


Figure 4.8: Number of messages (cost) needed to perform authentication in our PQC frameworks compared to existing CAN frameworks. Our PQC framework has noticeably different message costs, but still performs better than most other CAN frameworks at large numbers of nodes; [5, 4, 6].

subnets. As expected, the time taken for our authentication scheme is linear with little variation. Comparing it to the exact LWC variant [77], we can see a direct order of magnitude difference, showing the effect of more computationally expensive PQC algorithms. Regardless, we see acceptable run times for recommended values of k, especially given that authentication occurs during startup (and thus likely before critical communications are necessary).

4.5.2 Normal Communication Latency

After authentication, normal operation involves transmitting messages along the CAN buses, either between subnet devices or through routers to a verified destination. Routers are required to decrypt, validate, and re-encrypt packets traveling between subnets to ensure that the traffic is secure and authentic. The maximum number of hops required to reach a destination scales with the depth of the system, which is in turn influenced by the subnet size, $log(k)$. For example, a message from an endpoint node would need to hop up the full depth of the tree to reach the root node. This is a value that can be reduced by applying hardware acceleration, optimization, and other algorithm performance improvements. However, regardless of algorithmic optimizations, the performance will still follow a linear trend.

4.6 Discussion and Concluding Remarks

In this chapter, we present a novel security framework for 3D printing based on a hierarchical CAN bus structure built from root-of-trust. The proposed framework offers 3D printing applications strong security that takes plug-and-play and post-quantum protection into consideration; while many 3D printing security solutions exist, our work is one of the first to address security in the internal communication systems (i.e., CAN). Our design addresses security concerns for untrusted 3D printer components that are regularly swapped, such as hardware Trojans. Our framework

provides improvements compared to existing CAN frameworks for other applications and brings new ideas to the internal connections and networking of 3D printers. Our design is able to provide PQC encryption while utilizing at most 12% additional overhead in hardware and still maintaining fast message transmission speeds and low message costs for authentication and normal operation. Our design requires 90% and 25% less messages for authentication than existing post-quantum and lightweight CAN solutions, respectively.

Our framework provides significant versatility which can be used to vary the performance and security of the system and structure. First, our design features a customizable subnet size parameter, k, which significantly affects framework behavior. We discussed its effect on system hardware efficiency and the impact on "transmission hops". Second, authentication is performed using both two-way and one-way approaches, which we generally define as router and endpoint authentication, respectively. However, the user may also choose to utilize two-way authentication on endpoints to prove trust in those devices. Finally, we utilize only post-quantum cryptographic suites for our design, but we did not perform hardware acceleration for either scheme in our results. As a result, an avenue for future work could be testing various quantum-safe cryptography in software/hardware accelerated packages to produce a comprehensive study in resource-constrained 3D printer applications.

Chapter 5

Quantum Anomaly Detection in Industrial Control Systems

In the previous two chapters, we covered how to defend against a potential quantum threat by utilizing quantum-resilient ciphers for cryptographic CAN frameworks. However, just as quantum can be used by a potential adversary, we can also utilize quantum mechanics to strengthen the security and response against cyberattacks conducted on CPS data collection systems. In this chapter, we discuss an investigative approach to using the quantum fidelity algorithm with hybrid support vector machines to detect malicious or anomalous data in industrial control systems.

Cyberphysical Systems, also known as CPS, remain one of the critical pillars of many of the fastest-growing cybernetic domains in recent times, especially with the inclusion of Industry 2.0 "smart" systems. In 2025, it is expected that the CPS market will grow from $124.1 billion to 255.3 billion from 2024 to 2029 [98]. Industrial Control Systems (ICS), a type of cyberphysical system, brilliantly use sensor-based data and tightly monitored feedback loops controlled by computing to keep industrial infrastructures operational and highly efficient. The integration of Industry 2.0 concepts, such as remote monitoring and data-driven automation, has

become a cornerstone of modern operations in industries like power, water treatment, and manufacturing.

The use of cyberattacks to target ICS infrastructure has been of great concern for security experts. Devastating attacks like the Stuxnet worm [99] show the motivation and resources many malicious threat actors have to effectively cripple industrial plants and facilities. Studies have shown that in H2 2022, there were 115 published CPS vulnerabilities per month, with over 82% of organizations with ICS systems reporting that they have suffered cyber-attacks targeting their IoT-enabled systems [100]. Many of these vulnerabilities lead malicious parties to a very critical target: the data.

Anomaly Detection, or AD, in data identifies when sensor data and logs contain anomalous samples from malfunctioning devices, likely victims of malicious interference. Most ICS systems utilize the Supervisory Control and Data Acquisition (SCADA) architecture, which contains monitoring alarms based on rules built in or user-created. However, many researchers have previously determined that SCADA alarms alone are inadequate for modern security threats due to noisy behavior and high frequency of false positives [101]. Thus, more sophisticated methods for AD monitoring have been utilized in the past decade, including machine learning (ML) models to dynamically classify anomalous data and behaviors. However, just as modern ICS systems continue to rapidly grow in complexity and connectivity, the number of attack vectors and potential targets expands as well. Solutions for AD must keep up with the growing industry and looming threats that await them.

Quantum Machine Learning (QML) can be a pioneer solution to this ever-growing control cybersecurity problem. Using the highly expressive Hilbert spaces of quantum kernels, Quantum-Hybrid Support Vector Machines (QSVMs) can efficiently calculate the dissimilarities of anomalies in real-world data. However, there are many deciding factors to training an effective diagnostic model, from various proposed "feature maps", pre/post-processing methods, and parameterization approaches.

Machine limitations of older quantum systems immensely restricted the maximum feature count and sample size, limiting the capability of using a highly-dimensional

dataset. Furthermore, modern Noisy Intermediate-Scale Quantum (NISQ) computing realizes the risk of noise in the computed SVM kernels, which can greatly impact experimentation results. These issues contribute to a research gap in modern QML literature and drive isolation between the quantum and CPS research fields. To objectively select and judge a quantum approach to a high-risk domain like real-world ICS, in-depth analysis and conceptualization using modern, updated quantum systems is mandatory to identify any quantifiable quantum advantage over traditional kernel functions.

This study investigates the application of Quantum Hybrid Support Vector Machines as an anomaly detection technique for identifying unusual patterns in real-world datasets from ICS and Cyber-Physical Systems CPS. By pairing deeply parameterized, state-of-the-art feature maps and unique pre-processing methods, this work aims to thoroughly bridge the gap between theory and practice and identify an advantage of quantum kernels in the modern NISQ era. Our QSVM analysis is conducted using three independent ICS datasets from critical water infrastructure, specifically: (1) water treatment, (2) water distribution, and (3) hydro-power generation. We also extend beyond the water-specific ICS datasets by directly applying our model to 3D printing networking infrastructure and information to evaluate the effectiveness of our model implemented directly on the previous chapters of this dissertation.

A quantum advantage is demonstrated, with quantum computing showing improvements over classical computing, highlighting the benefits of quantum capabilities. Our contributions can be described as follows:

- Implementation of AD-focused QSVM designs using multiple highly effective feature maps from existing literature with an average classification F1 Score of 0.9.

- Investigation into quantum kernel performance using three popular CPS datasets with up to 24 features and 2000 windowed and processed data points each, larger than previous literature today.
- Comparison of the best performing quantum models to traditional kernel methods, resulting in an average 13.3% improvement in F1 score over even the best classical kernel solutions across all three datasets.
- Identified a clear quantum advantage of using quantum feature maps by observing the highly accepted geometric kernel distance metric and more.
- Implemented and performed experimentation on a robust, unique proof-of-concept for detecting cyberattacks on a real 3D printing network to determine the effectiveness of our model.

The organization of this chapter is as follows: Section 5.1 provides background on topics and literature in CPS security and QSVMs for anomaly detection. Section 5.2 provides a breakdown of the CPS dataset used in this work: HAI. Section 5.3 discusses the generalized framework and methodology. Section 5.4 discusses results and comparisons of the quantum kernel QSVMs, classical SVMs, and existing literature in the context of three ICS datasets. Section 5.5 investigates quantum noise, and "quantum advantage". Section 5.6 provides a system design and experimentation of anomaly detection in our real-world case study, 3D printers. Section 5.7 concludes and discusses the effectiveness of a QSVM for ICS anomaly detection, including discussing possible future works.

5.1 Background

A background on quantum fidelity kernels, quantum-hybrid models, and other quantum mechanics is provided in Chapter 2. This background specifically focuses on the properties of this proposed model using One-Class SVMs.

5.1.1 Classical Anomaly Detection

Cyberattacks against cyber-physical control systems utilizing zero-day threats are a growing threat, as documented in previous security risk assessments [102, 103]. Similarly, cyber-related vulnerabilities and attack models in critical infrastructures have been surveyed in existing literature [104, 105, 106]. Many of these works have shown that the slightest alteration of the data using these cyberattacks can result in catastrophic effects on the CPS feedback loops [107].

Classification of CPS anomalies in the classical domain is well documented in many systems and CPS literature. Deep-convolution neural network models have previously been applied to solve these complex CPS anomalies [108, 109, 110]. Kim et al. [111] investigated the feature importance and statistical characteristics of CPS datasets, such as SWaT and HAI. Typically, OCSVMs are well known for being strong anomaly detectors when applied correctly. However, recent results demonstrated classical SVMs to be very weak detection models for these aforementioned CPS datasets [112]. Thus, other classical methods are generally applied.

5.1.2 One-Class Support Vector Machines

The kernel trick is a powerful method to handle nonlinearly separable data by implicitly transforming it into a higher-dimensional space where it becomes more linearly separable. Instead of explicitly computing the transformation, the kernel trick uses a kernel function to compute the transformed vectors directly. For example, common classical kernel functions include the linear, polynomial, and radial basis function kernels as shown in Equation 5.1 with offset b and standard deviation σ.

$$\begin{aligned} &Linear\ Kernel, K(x_i, x_j) = x_i \cdot x_j \\ &Polynomial\ Kernel, K(x_i, x_j) = (x_i \cdot x_j + b)^p \\ &RBF\ Kernel, K(x_i, x_j) = exp\left(\frac{\|x_i - x_j\|^2}{2 \times \sigma^2}\right) \end{aligned} \tag{5.1}$$

The OCSVM learns the decision function $f(x)$ in such a way that most of the normal class data points lie within the decision boundary. It uses the "kernel trick" to elevate the data points and map them in high-dimensional feature space. With the increase in dimension, it becomes easier for the classifier to distinguish between classes. The optimization function is:

$$\underset{w,\rho,\zeta}{min}\left[\frac{1}{2}\|w\|^2 + \frac{1}{b\times n}\times\sum_{i=1}^{N}\zeta_i - \rho\right] \tag{5.2}$$

with the constraint:

$$(w\times\phi(x)) \leq \rho - \zeta_i, where\ \zeta_i > 0,\ and\ i = 1, 2, ..., n \tag{5.3}$$

The final decision function is:

$$f(x) = (w\times\phi(x)) - \rho \tag{5.4}$$

Here, $\phi(x)$ is the feature mapped data, w is the normal vector to the hyperplane, ϕ defines the offset for the hyperplane, ζ ensures the soft decision boundary by allowing a certain number of anomalous points, and ν is the parameter that controls the outliers by setting an upper bound.

The training process solves the following optimization problem by taking the kernel $k(x_i, x_j)$ into account, where $k(x_i, x_j) = \phi(x_i)\cdot\phi(x_j)$.

$$\underset{\alpha}{max}\left[-\frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\times\alpha_j\times k(x_i, x_j)\right] \tag{5.5}$$

Given, $0 \leq \alpha \leq \frac{1}{\nu\times n}$ and also, $\sum_{i=1}^{n}\alpha_i = 1$, where, ν is the Lagrange multiplier. In this work, the kernel function $k(x_i, x_j)$ is computed by the feature map fidelity algorithm defined earlier.

5.1.3 Kernel Alignment as a Similarity Metric

The best way to quantify the similarities between two kernels is to calculate the inner product between two kernels using Kernel Alignment [113], defined as

$$\mathrm{KA}(K_1, K_2) = \frac{\mathrm{Tr}(K_1 K_2)}{\sqrt{\mathrm{Tr}(K_1^2)\,\mathrm{Tr}(K_2^2)}}. \tag{5.6}$$

This equation calculates a normalized Frobenius Inner Product between the two matrices. This equation is just the cosine angle between the two kernels K_1 and K_2 if seen as vectors in the space of matrices. Thus, it is also commonly referred to as the "geometric distance" (GD). As it is normalized, the alignment is bounded between 0 and 1, or $KA(K_1, K_2) \rightarrow [0, 1]$. This concept can be further specialized to calculate the alignment between the kernel, K_1, and the data labels it is classifying, y. The kernel-target alignment function [114] is defined as follows:

$$\mathrm{KTA}_{\boldsymbol{y}}(K, y) = \frac{\mathrm{Tr}(K\boldsymbol{y}\boldsymbol{y}^T)}{\sqrt{\mathrm{Tr}(K^2)\,\mathrm{Tr}((\boldsymbol{y}\boldsymbol{y}^T)^2)}} = \frac{\boldsymbol{y}^T K \boldsymbol{y}}{\sqrt{\mathrm{Tr}(K^2)}N} \tag{5.7}$$

This function performs the kernel alignment of Equation 5.6 with an outer product of the labels, $K_2 = yy^T$. The higher the target alignment, the better the kernel fits the dataset vector in the feature space. If there is sufficient positive separation between the quantum and classical kernel target alignments, the kernel likely has a "quantum advantage" [115].

5.1.4 Related Works in QML-based Anomaly Detection

Existing QML literature supports quantum anomaly detection models for other domains. Quantum autoencoders [116], Quantum "Circuit Learning" [117], and density estimation [118] have been used to identify anomalies in new physics data. Hyperparameter studies for quantum kernel methods have been conducted for supervised learning models using Hamiltonian-evolution feature maps [119]. However, multiple works have investigated the use of quantum kernel SVMs for high-energy

physics data anomalies [7, 120, 121] and financial fraud [122]. Unfortunately, many of these works were limited by the number of qubits or shots in the quantum system at the time, resulting in heavy limitations on feature count and dataset scale.

Quantum CPS-specific anomaly work remains scarce, with only a few works mentioning CPS data in their proposed solutions. In recent work [123], simple QSVM models were compared using different feature reduction techniques on SCADA wind turbine systems. Furthermore, a gas power plant dataset was used in recent work to propose an autoencoder-reduced, parametrized QSVM feature map [8].

An overview of recent research on anomaly detection with Quantum Machine Learning (QML) is given in Table 5.1 categorized by methods. Several research studies used QSVM-based approaches, including those by Gouveia et al. [124], Correa-Jullia et al. [123], and Tscharke et al. [125], with datasets such as NF-UNSW-NB15, WTS, and KDD. Wang et al. [114] investigated hybrid models (QHDNN) on the Fashion MNIST and MNIST datasets, demonstrating competitive AUC values. QNN and QOC-SVM were employed in more recent research by Kukliansky et al. [126] and Cultice et al. [127] respectively, and achieved good F1-scores on the NF-UNSW-NB15 and HAI datasets. The table illustrates how QML techniques are becoming more varied and efficient in anomaly detection tasks.

5.2 Cyber-physical System Datasets

An essential challenge to identifying the suitability of quantum machine learning on data from thousands of cyber-physical sensors is to select suitable real-world datasets. This work features three very respected datasets used in previous literature: Secure Water Treatment Testbed (SWaT), HIL-based Augmented ICS (HAI), and WADI. These datasets all focus on industrial control systems related to water-based critical infrastructure. This includes water treatment, water distribution, and hydro-power turbine generation.

Table 5.1: Recent Works in Anomaly Detection with Quantum Machine Learning.

Author and Year	**QML Algorithm**	**Dataset**	**# of Qubits**	**Feature**	**Metrics (best)**
Gouveia et al. [124] 2020	QSVM	NF-UNSW-NB15	16	16	Acc: 64
Jullia et al. [123] 2022	QSVM	WTS	8	16	Acc: 88.8% F1: 0.893
Tscharke et al. [125] 2023	QSVR	KDD	5	5	Acc: 82.0% F1: 0.78
Wang et al. [130] 2023	QHDNN	Fashion MNIST	16	16	AUC: 88.24
Wang et al. [130] 2023	QHDNN	MNIST	16	16	AUC: 89.41
Kukliansky et al. [126] 2024	QNN	NF-UNSW-NB15	16	16	F1: 0.86
Cultice et al. [127] 2024	QOC-SVM	HAI	8	16	F1: 0.86 ACC: 87%

5.2.1 iTrust Secure Water Treatment

The first dataset of this work is the Secure Water Treatment Testbed, or SWaT, by the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [128]. This 2015 dataset has been used in many existing literatures [129] and is described as a modern scaled-down, industry-compliant water treatment plant. This system consists of a six-stage process configuration of Water Extraction (P1), Pre-Treatment (P2), Ultrafiltration Unit (P3), Dechlorination Ultraviolet Lamps (P4), Reverse Osmosis (P5), and Backwash Process (P6). In usual SCADA systems, there exist four levels of protocol: Physical, Basic Control, Supervisory, and Operations Systems, including actuators/sensors on Level 0 up to high-speed Ethernet traffic on Level 2. This is all depicted in Figure 5.1.

The dataset consists of collected network traffic and raw values of 25 sensors and 26 actuators located throughout all processes, totaling 51 features. SWaT consists of 11 days at a one-second resolution of continuous operation of the emulation testbed, with 4 days of attacks. 41 unique attacks were launched during the 4 days, including combinations of said attacks. These attacks were thoroughly crafted through attack models designed by the iTrust research team. This data was labeled based on the author's interpretation of normal and abnormal behaviors of the system.

5.2.2 HIL-based Augmented ICS

We have also selected the Hardware-in-Loop-based (HIL) Augmented Industrial Control System (ICS) Security Dataset, or HAI 20.07 [131]. The HAI dataset is a realistic ICS testbed that physically emulates hydro-power and steam-turbine power generation using Programmable Logic Controllers (PLCs). The system contains four critical processes that are extremely essential to safe operation: the Boiler process (P1), the Turbine process (P2), the Water Treatment process (P3), and the Hardware-in-Loop Simulation (P4). How these processes are interconnected can be seen in Figure 5.2. Input parameters are set for feedback control loops which alter the

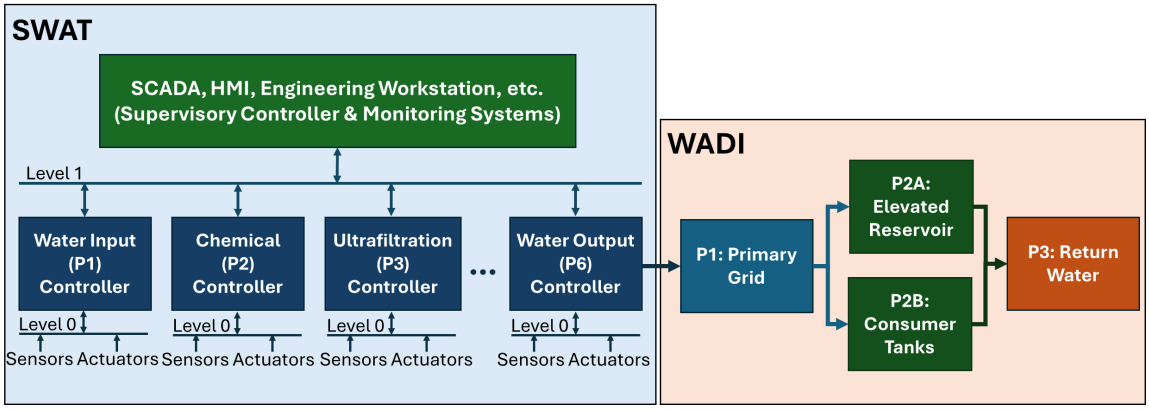


Figure 5.1: System depiction of Secure Water Treatment and Water Distribution from iTrust Cybersecurity Research Center, Singapore University of Technology. This full cycle water treatment and distribution testbench spans two separate CPS highlighted subsystems, SWAT and WADI, gathered in two separate datasets.

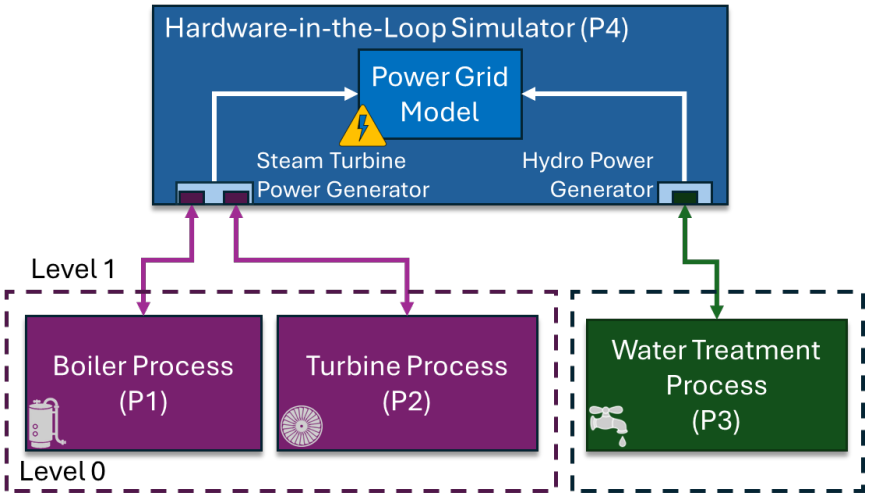


Figure 5.2: HIL-based Augmented ICS (HAI) structure for steam-turbine and hydropower. P1 (Boiler), P2 (Turbine), and P3 (Water Treatment) are all connected by P4, the overall controller that simulates a power grid model.

behavior of the system and PID controllers, including water pressure, flow rate, boiling temperature, and expected power output.

In this dataset, there are 59 features (including the input parameters) spread across all four processes. These features are generally measured points within each process that directly impact how the system will respond, react, and possibly warn the operator. Training data was captured from normal operation with no anomalies for 177 hours, with various set-points throughout the 24-hour period. The test data contains both normal and anomalous data captured over 123 hours with similar inputs. In total, 38 attacks were conducted, including 14 different attack primitives and combinations of attacks. Anomalies are defined as attacker-modified data showing significant deviation from typical patterns between features. The resolution of the dataset is at 1 second each. The percentage of anomalies in the test data we used is 4% [112], requiring additional considerations for the class imbalance.

5.2.3 iTrust Water Distribution

The third and final dataset we used is the Water Distribution dataset, or WADI, by the iTrust, Centre for Research in Cyber Security, Singapore University of Technology and Design [132]. This dataset from 2017 recorded attacks on an operational testbed simulating a scaled-down version of a city water distribution network. The testbed contained three major processes controlled by PLCs: the primary grid (P1), the secondary grid (P2), and the return water grid (3). It used three levels of communication protocol, including actuators/sensors on L0, up to high-speed Ethernet traffic on L2. As shown in Figure 5.1, this dataset is based on a water distribution extension to the SWaT testbed, simulating the effects of consumer tanks of a city water network. Thus, many of the input signals of P1 directly come from SWaT, demonstrating the overlap and distinction between the Level 0 and Level 1 signals across different CPS domains.

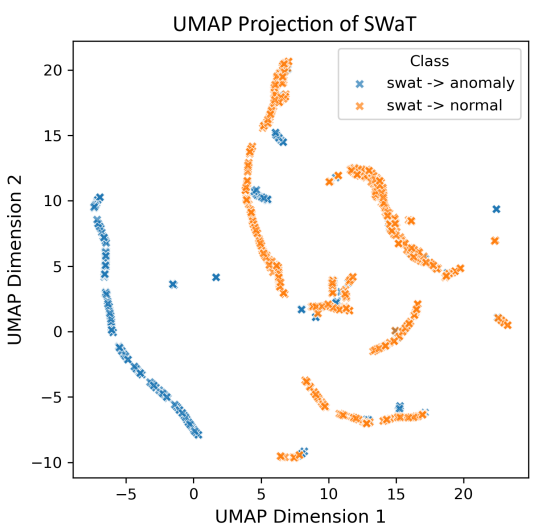
WADI contains 123 sensors and actuators between the three processes connected to the network. The dataset consists of 16 days of continuous operation at one-second resolution, where only two days had attack scenarios occur. In those two days, 15 attacks were launched including combinations of attack models. Similarly to SWaT, attack scenarios were derived from attack models developed by the iTrust researchers, focused on the intent of a potential attacker. Labels were provided by the authors explicitly defining whether an attack was occurring or not. Any NaN values found were replaced with 0's.

5.2.4 Data Visualization and UMAP Projection

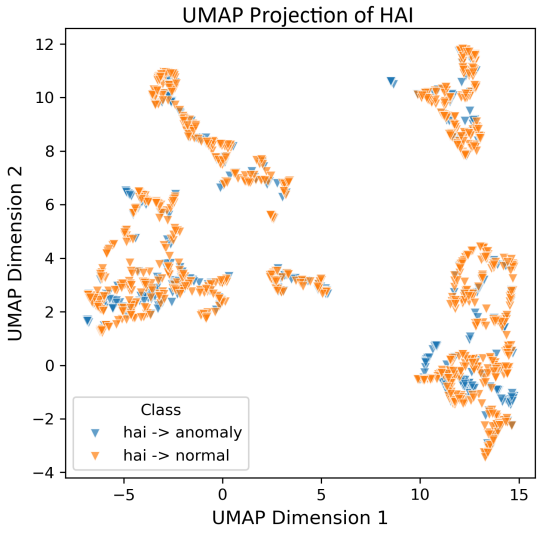
A UMAP (Uniform Manifold Approximation and Projection) of each dataset is shown in Figure 5.3. UMAP transforms high-dimensional data into a lower-dimensional space while preserving its underlying structure which helps visualize the dataset. In this case, we visualize our data as a 2-dimensional UMAP, which provides some extremely useful insight into the separability and difficulty of classification.

First, the separability of anomalies and normal data in SWaT is very clearly visible in the projection in Figure 5.3a. Anomalies are pulled far away from the normal cluster, with only some small clusters of anomalies showing up in harder-to-separate areas. This likely suggests that a certain attack, or combination of attacks, is harder to detect in a linear decision boundary, suggesting the need for higher-dimensional spaces like our QSVM.

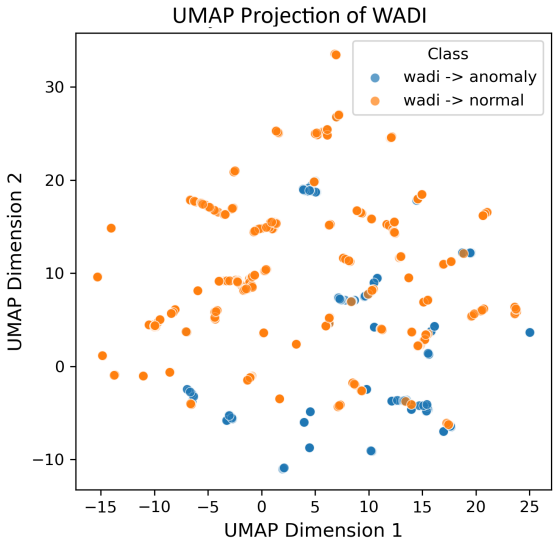
In the HAI UMAP presented in Figure 5.3b, the boundary between anomaly and normal data is a lot less obvious. There are clear clusters of information, suggesting datapoints collected during different states of the HAI power system (as described earlier). However, any further distinction of the classes will require the power of further dimensional extraction through the kernel trick, and might likely still struggle the most of all three datasets in classification tasks.



(a) SWAT UMAP Projection



(b) HAI UMAP Projection



(c) WADI UMAP Projection

Figure 5.3: 2-Dimensional Uniform Manifold Approximation and Projection (UMAP) of the three cyberphysical datasets: SWAT, HAI, and WADI. This visualization can provide estimations of the difficulty of hyperplane separation between anomaly and normal data.

Finally, the UMAP for the WADI dataset shown in Figure 5.3c demonstrates no strong clustering of the datapoints in anomaly and non-anomaly data. While this may seem at first like the data is of poor quality, the UMAP projection will emphasize small differences between data when the data contains a singular large cluster, such as this one. Thus, the main focus of predicting dataset behavior using a UMAP is to identify if there is a meaningful way to draw an N-dimensional "line", or boundary, between the two datasets to maximally separate the two classes. In that context, the WADI dataset is of intermediate complexity compared to the other two datasets.

5.3 Methodology

While there are many components of the Quantum-Hybrid Support Vector Machine (QSVM) that can be parametrized and interchanged, the overall algorithmic flow remains the same. It can be described as follows: Prepare the data and create train/test sets, calculate the quantum fidelity by performing the Hilbert-Schmidt inner product between the train data and current data vectors, and use the resulting similarity matrix to train and classify on the SVM's hyperplane. This process flowchart can be seen in Figure 5.4.

The SVM's kernel takes advantage of quantum feature maps within the kernel function to identify hidden similarities or differences in data patterns that may be too computationally expensive for classical computing. To reduce the noise and resource consumption, the data pre-processing and SVM components are performed classically.

In previous work [127], this topic expanded upon only a single method of feature selection and one well-defined feature map. To get a clearer picture of the advantages, our current methodology parametrizes multiple aspects of the QSVM with known high-performing alternatives to identify the best.

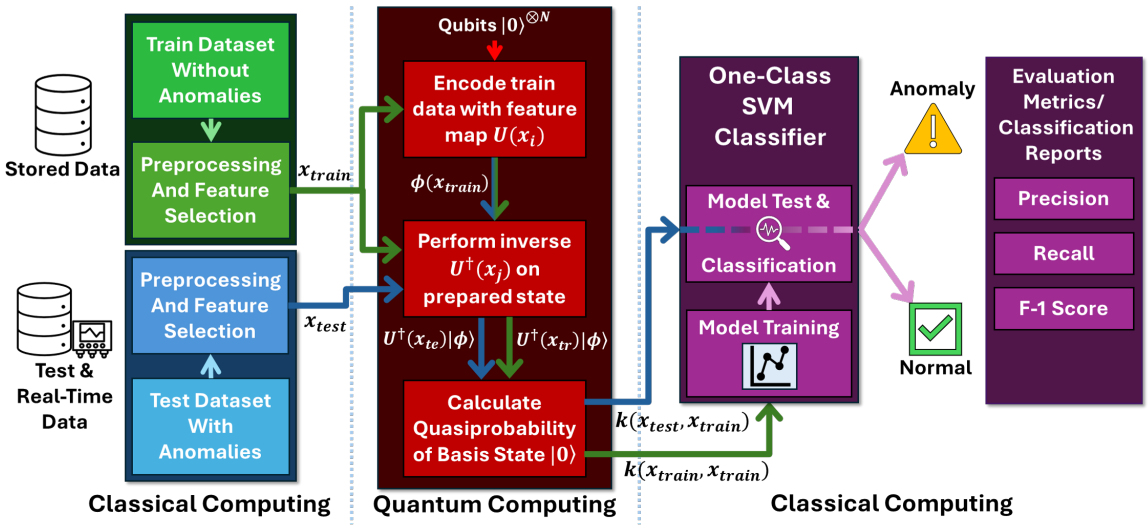


Figure 5.4: Pipeline of the hybrid-quantum SVM anomaly detector. Pre-processing occurs classically, is turned into a fidelity inner-product kernel using a quantum computer, and then utilized for SVM anomaly detection. Data kernels can then be passed into the trained SVM for the detection of anomalies.

5.3.1 Data pre-processing and Feature Reduction

Each dataset is prepared for the feature selection using some consecutive pre-processing techniques. Firstly, the dataset is scanned with a moving average of 60 overlapping sample windows. This helps reduce the noise and artifacts by capturing the time series trends. Equation (5.8) is used for the moving average where w is the window length and x_i is the sample at time i.

$$ma_t = \frac{1}{w} \times \sum_{i=t-w+1}^{t} x_i \tag{5.8}$$

Next, the dataset is standardized with standard-scaler transformation or Z-score normalization with Equation (5.9). Here, μ is the sample mean and σ is the standard deviation. This moves the mean of the distribution to zero and the standard deviation to 1, which helps to equalize the scales of different units. Furthermore, standardization tends to bring faster convergence for optimization techniques like gradient descent.

$$x_{std} = \frac{x_o - \mu}{\sigma} \tag{5.9}$$

The pre-processing steps end with the feature reduction techniques. We experimented with three types of feature reduction techniques: a) Decision Tree, b) Principal Component Analysis (PCA), and c) Non-Negative Matrix Factorization(NMF). The number of features is chosen based on the number of qubits.

a) Decision Tree: Tree-based methods calculate the feature importance rank from the Gini impurity. Based on the relative decrease in the impurity it can sort the features from most important to least.

b) Principal Component Analysis: The PCA-based methods calculate the principal component from the dataset in eigenvalue representation. The eigenvalue defines the weight of that feature and can help rank it based on the relative weights.

c) Non-Negative Matrix Factorization: NMF-based methods help reduce the number of features by decomposing a high-dimensional non-negative data matrix into two lower-dimensional non-negative matrices.

The pre-processing techniques were applied separately to the training set and test set to avoid data leakage.

5.3.2 Quantum Fidelity Kernel

The kernel function is the most important component of a quantum-hybrid SVM, as it directly takes advantage of quantum computation. To create a quantum kernel, we use the quantum fidelity algorithm paired with a strong data-encoding Feature Map (FM) circuit. Proper parameter selection of the feature map is required for optimal class separation, including varying the repetitions, scale, and interactivity.

Each feature map encodes N features in $n/2$ qubits to the Hilbert space using two degrees of freedom. Feature maps are also repeated 1-3 times to drive more interaction between features. Additionally, data is scaled between $[-\pi, \pi]$, with careful consideration of the encoded data's kernel bandwidth [133]. A kernel inner-product matrix is created by fidelity testing each datapoint to one another, or $O(|x_{input}| \times |x_{train}|)$ shots. Finally, the resulting kernels are post-processed by exponentiating the values, $k_p(x_i, x_j) = e^{k(x_i, x_j)}$.

In this work, we benchmark four separate feature maps from popular existing literature: Belis et al. [7], Sakhnenko et al. [8], the 2nd-Order Pauli-Z Feature Map [9], and a simple U2-gate 2DoF feature map (for reference). These feature maps have been modified to take advantage of two degrees of freedom.

Belis 2DoF Feature Map

This feature map proposed by Belis et al. [7] features encoding of two degrees of rotational freedom followed by nearest neighbor (NN) entanglement, $G(x)$, and a final rotation S'_x on different axes. The gate $U(\pi/2, \phi, \lambda) \in SU(2)$ is a universal

1-qubit gate rotating about two axes. The circuit diagram is presented in Figure 5.5. We repeat the gates of this feature map up to three times in the fidelity calculation.

Basic 2DoF Rotation Feature Map

As a ground state to determine how much the entanglements actually change the outcome, we have included a basic two-degree-of-freedom quantum feature map. This circuit consists of just universal 1-qubit $U(\pi/2, \phi, \lambda)$ gates which encode two features each onto $N/2$ qubits. There is no $G(x)$ entangling/interaction circuit, making this map extremely simple.

Sakhnenko Parametrized Feature Map

Feature maps based on parametrized circuit structures were proposed in Sakhnenko et al. [8] demonstrating strong results in gas turbine synthetic data. Particularly, their "Circuit 10" performed very strongly, made up entirely of rotate x (R_x), rotate y (R_y), and CNOT gates. The angle parameter, θ, was set to $\theta = \pi/2$ for all cases. We have created a 2-DoF variant inspired by the original Circuit 10 that is not repeated. This two-rotational-degree version of this circuit is provided in Figure 5.6.

Second-Order Pauli-Z Evolution Feature Map

The ZZ Feature Map (ZZFM) is the most popular method of data encoding which uses 2nd-order Pauli-Z evolution with linear entanglement [9]. This feature map presented in Figure 5.7 utilizes a classical nonlinear function, $\varphi(x, y) = (\pi - x)(\pi - y)$, as parameters in Pauli gates. Unlike the others, ZZFM utilizes 1DoF, or N qubits for N features. It also does not contain separated encoding and entanglement operations, choosing to interweave them instead. This was implemented using Qiskit's *ZZFeatureMap* [19] construct.

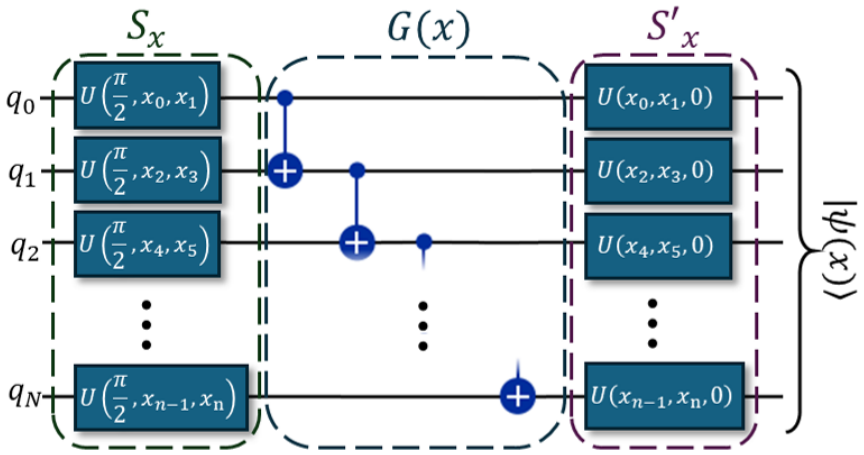


Figure 5.5: 2-DoF Belis et al. [7] Feature Map. S_x and S'_x angular encode N features onto $N/2$ qubits, and $G(x)$ performs NN entanglement.

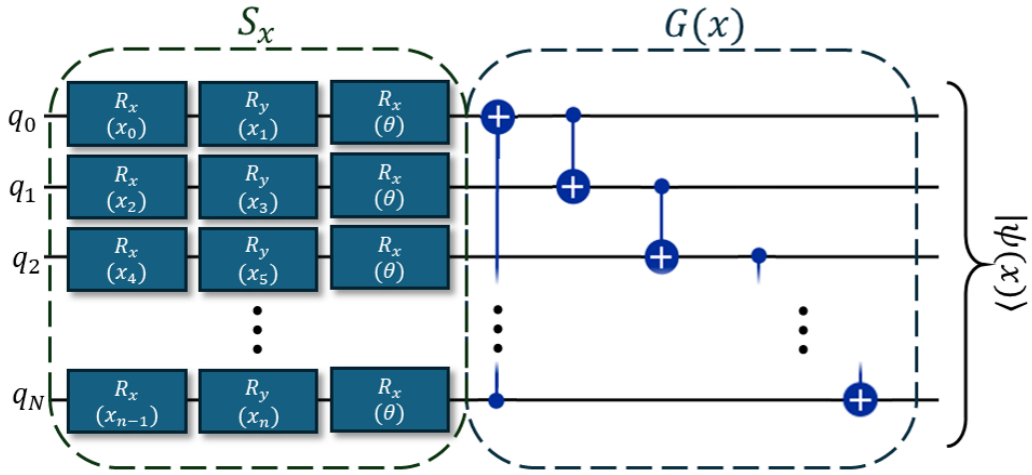


Figure 5.6: 2-DoF version of Circuit 10 from Sakhnenko et al. [8]

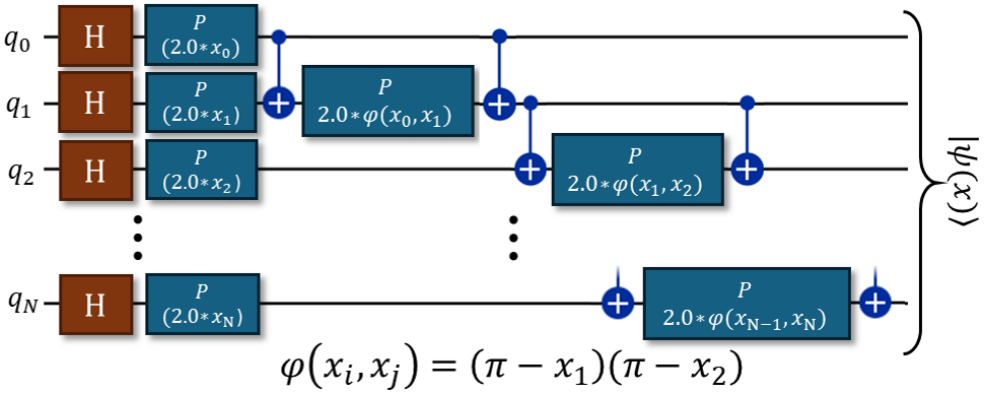


Figure 5.7: Second-Order Pauli-Z Evolution Feature Map (ZZ Feature Map). [9]

5.4 Experimentation Results

5.4.1 Testing Setup

To determine the effectiveness of quantum kernels in the ICS ecosystem, thoughtful experimentation was conducted to quantify the benefits of the highly nonlinear behaviors of quantum computing. This experimentation investigates different pre-processing methods, feature maps, and features. The SVM parameters less relevant to concluding our work, such as the nu and tolerance, were swept and the best results selected to ensure all classical and quantum results were selected fairly and optimally.

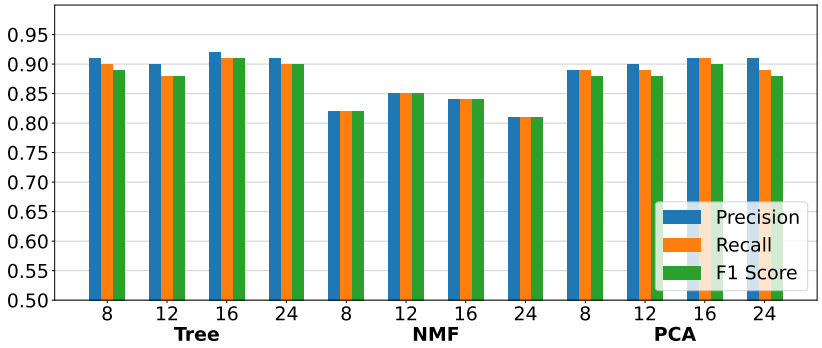
The Belis, Basic 2DoF, and Sakhnenko feature maps were parametrized between three pre-processing methods (Decision Tree, NMF, and PCA), and four feature counts ($N = 8, 12, 16, 24$). Due to the exponential computing complexity, the ZZ Feature Map was only used for $N = 8, 12, 16$.

5.4.2 Dataset Results: SWaT

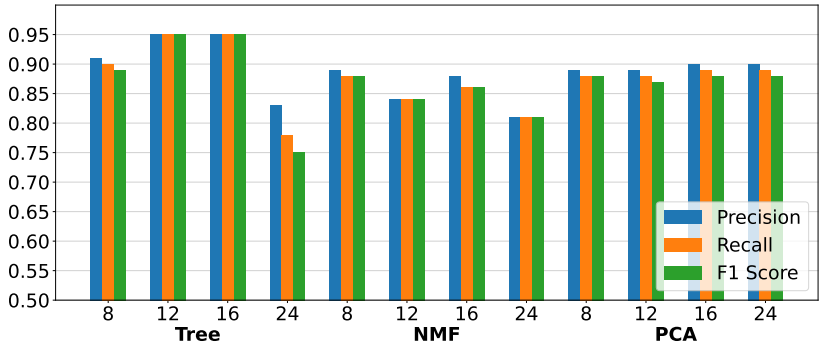
The results, as shown in Figure 5.8, start off strong with very good classification performance from quantum kernels on the SWaT dataset. The three 2DoF feature maps all reach F1 metrics of over 0.9 with multiple configurations. While the feature maps had some degraded performance on the NMF feature reduction method, the performance remains acceptable for fairly reliable classification tasks.

The basic 2DoF feature map had the best performance of 0.95 F1 score with the Decision Tree feature selection of 16 features. Thus, we can theorize that the entanglement structures of the other feature maps did not provide any meaningful improvement over simply encoding with universal rotation gates. In fact, the ZZ Feature Map circuit struggled with consistency, varying in performance over the different pre-processing methods.

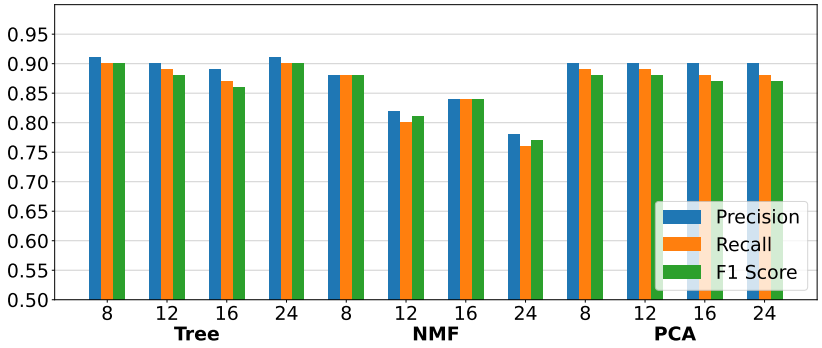
This observation aligns with our previous UMAP hypotheses. The SWaT dataset was already proven to be the easiest of the three datasets due to its relatively large



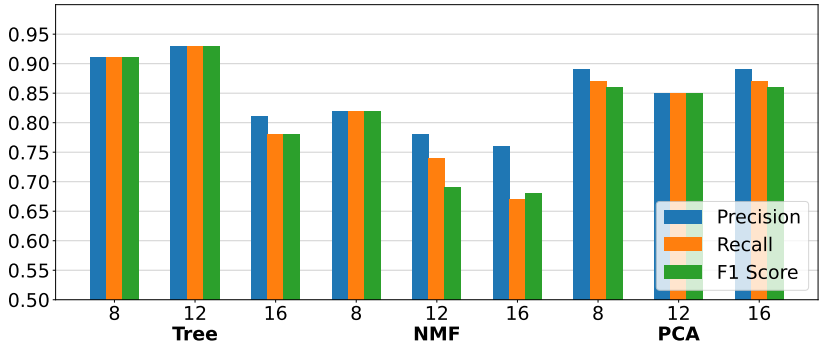
(a) Belis Feature Map Ideal Performance



(b) Basic 2DoF Feature Map Performance



(c) Sakhnenko Feature Map Performance



(d) ZZ Feature Map Performance

Figure 5.8: Results for Quantum SVM Anomaly Classification on SWaT for all feature maps.

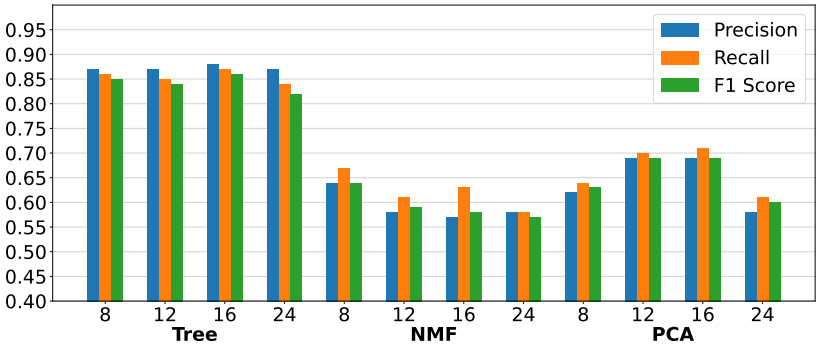
disparity between anomalous and normal states. However, this can help us conclude that the QSVM does not seemingly struggle with many overfitting issues from being a highly complex model.

5.4.3 Dataset Results: HAI

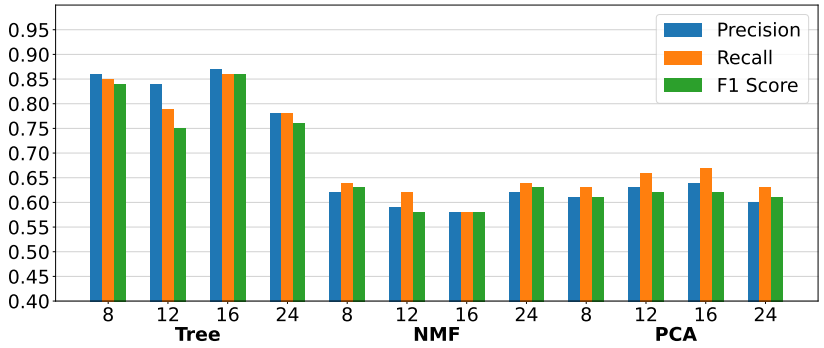
The HAI dataset, shown in Figure 5.9, was arguably the most difficult to classify due to its very complex, nonlinear relationships between features and noisy data. Similarly to previous work [127], the selection of features using a Gini impurity decision tree was the best method of pre-processing for quantum kernels. This was likely due to many inconsequential features that have very low correlation with each other or trends in the data.

All feature maps averaged above 0.82 in all three classification scores with the decision tree, which is well above the average of the other pre-processing methods. The best results showed in Belis kernel, 16 tree-reduced features with 0.88/0.87/0.86 in precision, recall, and F1. In the Belis, 2DoF, and Sakhnenko kernels, PCA performed very poorly at an average of 0.63 in the classification metrics. However, the ZZ Feature Map's highly nonlinear design was the only feature map that could find good decision boundaries in the PCA-reduced HAI data. This resulted in performance similar to the decision tree at 0.8 precision, 0.8 recall, and 0.79 F1. The NMF results were also very low, almost averaging close to a purely random chance of 50-50. This further suggests that the features pruned by the tree add unnecessary relational feature complexity with no benefit in classification.

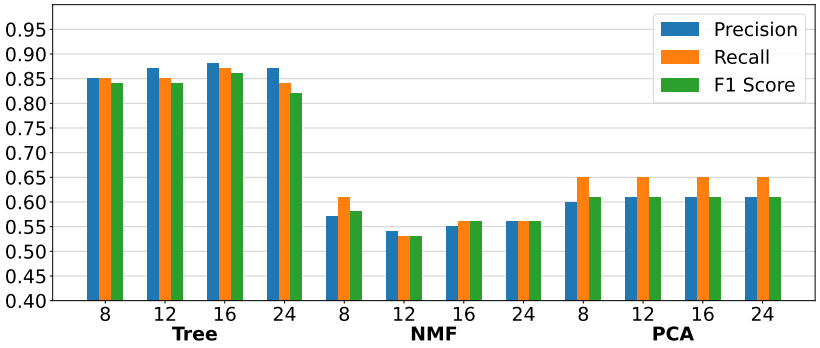
Ultimately, the HAI dataset has shown that the method of reducing features and condensing data plays a huge role in classifiability, especially in quantum kernel bandwidth. Furthermore, this also further strengthens the usage of UMAP to visually predict data separability, as the HAI dataset was projected to be the hardest to classify.



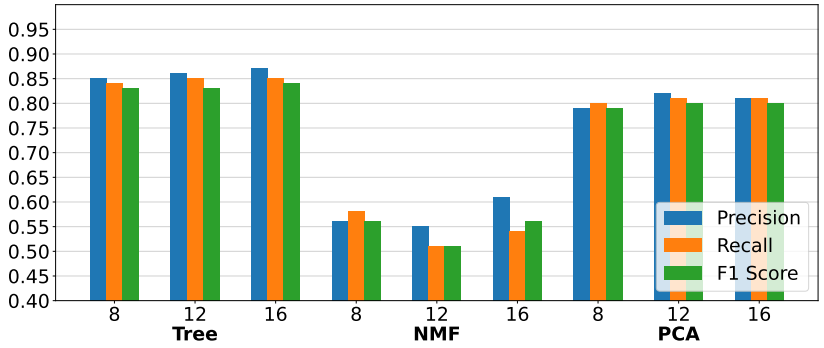
(a) Belis Feature Map Ideal Performance



(b) Basic 2DoF Feature Map Performance



(c) Sakhnenko Feature Map Performance



(d) ZZ Feature Map Performance

Figure 5.9: Results for Quantum SVM Anomaly Classification on HAI for all feature maps.

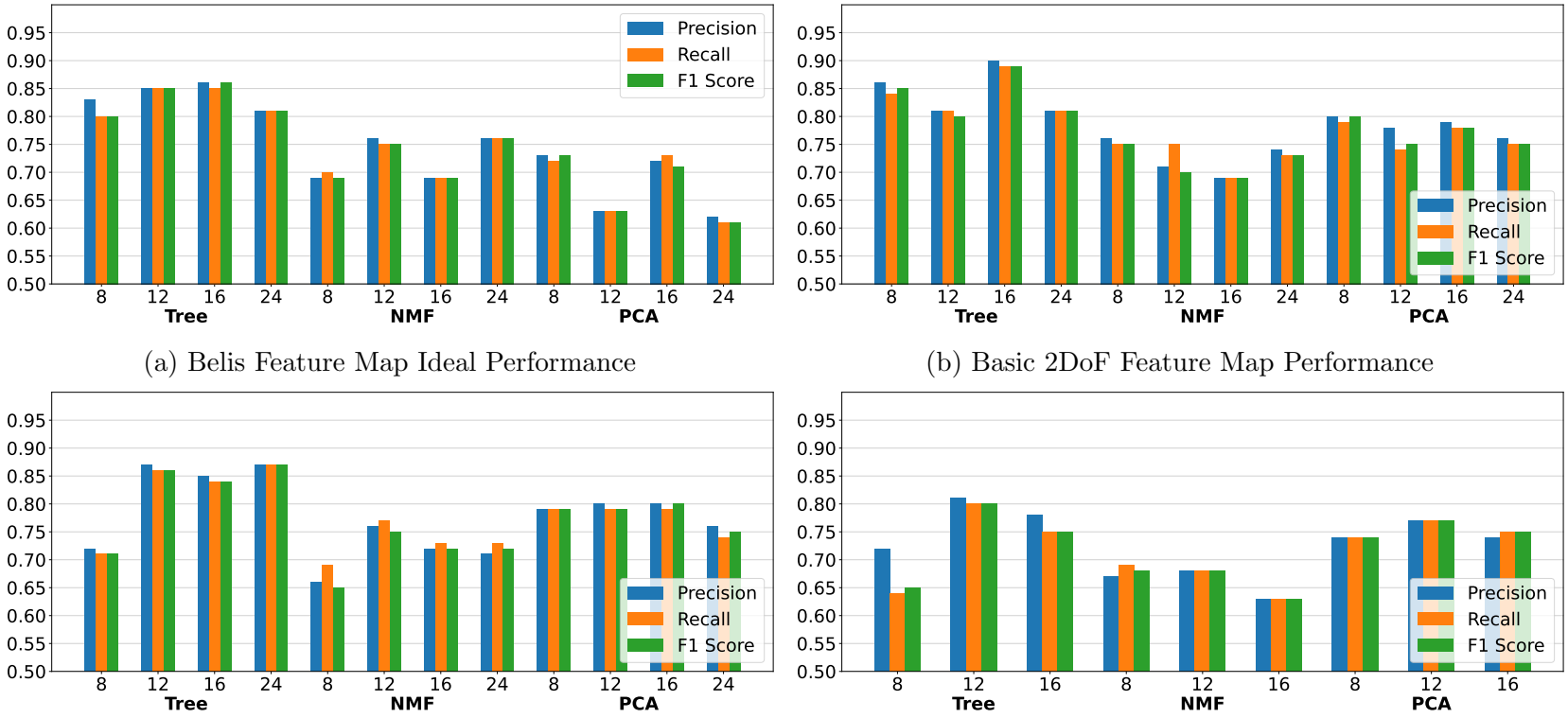
5.4.4 Dataset Results: WADI

The WADI dataset, shown in Figure 5.10, was seemingly average in classification complexity and difficulty based on the results. Out of the three pre-processing methods, the Decision Tree selection method performed the best, likely due to many confounding/noisy features of the least importance being pruned by the selection algorithm. Similarly to before, the NMF pre-processing method was devastating for both quantum and classical methods, likely due to the dataset's particular distribution.

Most quantum feature maps thrived with the data, performing consistently above 0.7 Precision/Recall/F1 Score with all configurations. The maximum performing model was the simple 2DoF model using Decision Tree selection of $N = 16$ features, reporting a Precision of 0.9, Recall of 0.89, and an F1 of 0.89. It is highly likely that qubit entanglement did not greatly impact the kernel or SVM trained hyperplane, as the simple $U2$-gate feature map performed just as well as the entangled circuits. Furthermore, the ZZ Feature Map underperformed on this task, further reinforcing the idea that simple quantum models may have an advantage with this specific data.

5.4.5 Classical Results

To identify any improvement to classification, classical kernel function results are presented for Polynomial, RBF, and Linear kernels in Figure 5.11. Overall, classical provided consistently lower results with more variation than quantum for all three datasets. For the SWaT dataset, we see a similar performance between the two, with an average of 6% difference in metrics. The best classical result for SWaT was the RBF kernel using a decision tree, which achieved a recall of 0.91 and F1 of 0.89. However, classical averaged 0.88 F1 score across all kernel and preprocessing configurations, leading us to conclude that the SWaT dataset is overall just easier to classify compared to the other CPS datasets.



(a) Belis Feature Map Ideal Performance

(b) Basic 2DoF Feature Map Performance

(c) Sakhnenko Feature Map Performance

(d) ZZ Feature Map Performance

Figure 5.10: Results for Quantum SVM Anomaly Classification on WADI for all feature maps.

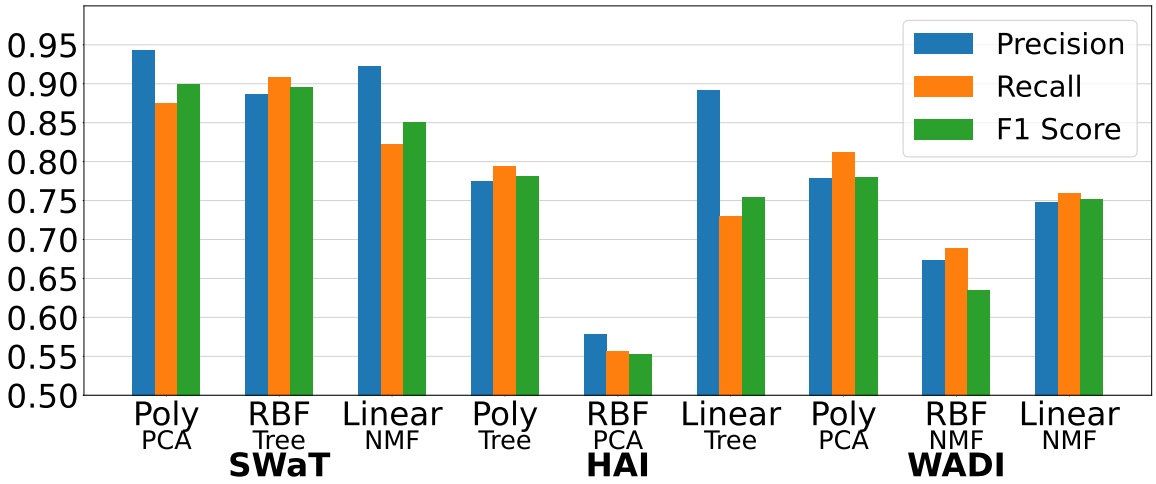


Figure 5.11: Best Results for Classical Kernels' SVM Classification on all datasets.

For HAI, classical methods struggled with NMF and PCA data, quite similar to the quantum kernels. However, the models based on polynomial/linear dot products were able to produce acceptable results using the decision tree method, with the best result of 0.77/0.79/0.78 in precision, recall, and F1. Supported by previous work [112], this leads us to believe that HAI has many confounding features as the decision tree would remove low-quality information. However, this is still over 10% lower than the best quantum kernel. Quantum kernels still significantly outmatched the classical, averaging approximately 4.6% higher in precision, 9.8% higher in recall, and 11.8% higher in F1 score. Thus, we can conclude that using a quantum kernel has made significant improvements over classical in HAI.

The WADI dataset also resulted in favor of the quantum kernel methods. The best quantum feature map configurations performed 13% better over all classical functions in the classification performance of WADI. For this dataset, the NMF feature reduction method seemed to be optimal for classical kernels. On average, classical kernels presented a modest F1 score of 0.722. However, when factoring in noise, quantum outperformed classical by only 4.2% on average, lowering the experimental certainty of a quantum advantage.

Overall, quantum models performed an average of 13.3% better than their classical counterparts across each preprocessed dataset and kernel. These results help us conclude that quantum kernels can perform consistently above classical methods in these particular CPS datasets. Common trends between classical and quantum suggest that certain datasets and kernel types favor specific pre-processing methods, such as the importance decision tree.

5.5 Quantum Error and Advantage Metrics

Quantum noise was estimated from real IBMQ hardware in the quantum fidelity circuits. This includes three popular, modern machines: IBM Kyiv, IBM Sherbrooke, and IBM Torino. These machines are all based on IBMQ's latest Eagle R3 and Heron

R1 processor families. To simulate noise, we used Qiskit AER [19] density matrix simulations with estimated noise snapshots from the time of experimentation, which are presented in Table 5.2. Due to the computational complexity of simulating noisy density matrices, noisy results were computed only for feature count $N = 8, 12, 16$ with the two best-performing feature maps: Belis et al. [7] and Simple 2DoF. The results, represented as percent decrease in classification metric between statevector and noisy simulations, are presented in Table 5.3.

Noise did not strongly impact the classification of SWaT. Both kernels retained consistent results for 8 and 12 features, suffering at most 2.27% change reduction in classification F-1. However, the Belis kernel experienced up to 5.49% reduction in F1 score when using 16 features, likely due to the noisy disturbances pushing a significant cluster of datapoints beyond the hyperplane boundary. However, as this is only showing percent change, we can determine that the kernels were resilient to disturbances to noise and still retain an advantage over their classical counterparts.

For the HAI dataset, the effects of noise were even less impactful on our experimental conclusions. The simple benchmark 2DoF kernel was extremely resilient to change, with a 1.15% decrease in results on 12-feature recall. Similarly, the Belis kernel barely fluctuated, with less than 2.35% reduction in classification quality across all feature counts. This lack of classification degradation leads to the conclusion that the HAI dataset clusters are fairly robust and changes in the kernel don't meaningfully change the resulting hyperplane.

The effect of noise was somewhat drastic for the WADI dataset. For the Belis Feature Map, the performance drops anywhere between 1% and 15% with error. For the basic 2DoF feature map, the negative effect of noise is similar, with up to 11.25% decrease in F1 for 8 features. However, the overall average error of 2DoF on WADI is 3.57% less than Belis, suggesting again that decoherence of longer circuits, especially repeated ones, definitely plays a role in kernel variation. This coincides with WADI having the highest dissimilarity error in Section 5.5.1 out of all three datasets.

Table 5.2: Median errors and decoherence times in IBM Quantum computers captured at the time of noise model simulation.

Instance Name	# of Qubits	Best 2Q EPLG	SX Gate	CZ/ECR Gate	Readout Asgmt.	T1 (us)	T2 (us)
Torino	133	1.08e-3	3.149e-4	2.936e-3	2.6e-2	177.93	137.79
Sherbrooke	127	2.39e-3	2.582e-4	7.726e-3	1.39e-2	262.43	164.33
Kyiv	127	4.33e-3	2.779e-4	1.208e-2	8.2e-3	265.84	101.59

Table 5.3: Average error and percent decrease of classification metrics with Belis and 2DoF kernels when using noisy kernel simulations between 8, 12, and 16 features.

(a) 8 Features

		% ↓ in Performance		
	Kernel	Prec	Rec	F1
SWaT	Belis [7]	1.12%	2.20%	1.11%
	2DOF	0.00%	1.09%	0.00%
HAI	Belis [7]	2.35%	2.30%	2.33%
	2DOF	0.00%	0.00%	0.00%
WADI	Belis [7]	2.50%	3.61%	2.50%
	2DOF	1.25%	9.64%	11.25%

(b) 12 Features

		% ↓ in Performance		
	Kernel	Prec	Rec	F1
SWaT	Belis [7]	0.00%	0.00%	0.00%
	2DOF	2.27%	1.11%	2.27%
HAI	Belis [7]	0.00%	0.00%	0.00%
	2DOF	0.00%	1.15%	0.00%
WADI	Belis [7]	8.23%	9.41%	9.41%
	2DOF	1.18%	4.71%	3.53%

(c) 16 Features

		% ↓ in Performance		
	Kernel	Prec	Rec	F1
SWaT	Belis [7]	3.30%	4.35%	5.49%
	2DOF	1.10%	1.09%	1.10%
HAI	Belis [7]	1.16%	1.14%	1.15%
	2DOF	0.00%	0.00%	0.00%
WADI	Belis [7]	15.12%	12.88%	14.11%
	2DOF	3.49%	4.65%	5.88%

5.5.1 Noise Effects in Kernel Error

To understand the effects of noise on the quantum kernel, we define kernel error using the kernel geometric difference function described in Section 5.1. This allows us to measure the geometric similarities between the two kernels. However, as the goal is to capture the difference between the statevector (ideal) and noisy matrices, we represent our results as the complement of the normalized alignment. Thus, we define $D_{Error} = 1 - KA(K_{ideal}, K_{noisy})$, where D_{Error} is the dissimilarity between the matrices. This means a lower value ($D_{Error} \to 0$) is better, as the noisy kernel would have less deviation from the ideal model.

Based on the results presented in Table 5.4, we can see that the geometric dissimilarity is relatively low, allowing our noisy results to closely resemble the ideal results. This resilience to quantum noise can be both in part by mitigated error from IBM circuit compilation and our low-depth circuits. Despite this, though, even the smallest of kernel perturbations can cause immense classification troubles if the hyperplane is unstable, as shown with the noisy results of WADI.

Interestingly, the effect of the Readout Assignment Error from Table 5.2 seems to have a correlation to the impact on the results. For example, IBM Torino's Readout error at the time of experimentation was 2.6×10^{-2}, which was 2.78 and 1.87 times greater than IBM Kyiv and IBM Sherbrooke, respectively. We can see a very similar ratio in the IBM machine's dissimilarity error of Table 5.4, the differences increasing rapidly for larger values of N features. This effect is also reflected in the error bars of the classification performance.

Furthermore, we can see the expected effect of gate and relaxation errors on longer circuits as the dissimilarity of the Belis feature map is often more deviated/erroneous than the basic 2DoF feature map. Longer circuits mean a greater chance of gate and decoherence issues, which are present in the dissimilarity. This most assuredly should be explored further in future work.

Table 5.4: Reported error in noisy simulation kernels of N features using dissimilarity metric: $1 - KA(K_{ideal}, K_{noisy})$.

Dataset	N	Torino		Sherbrooke		Kyiv	
		Belis [7]	2DOF	Belis [7]	2DOF	Belis [7]	2DOF
SWaT	8	3.67e-03	3.71e-03	1.46e-03	1.07e-03	9.73e-04	7.04e-04
	12	6.98e-03	1.06e-03	3.79e-03	6.34e-04	2.36e-03	3.41e-04
	16	8.60e-03	1.09e-03	6.91e-03	1.02e-03	3.07e-03	3.66e-04
HAI	8	1.02e-03	8.68e-04	5.59e-06	1.82e-06	5.92e-04	8.98e-05
	12	6.50e-03	1.12e-03	7.06e-03	5.62e-04	3.61e-03	2.25e-04
	16	1.25e-03	1.07e-03	1.07e-05	2.11e-06	6.32e-06	1.30e-06
WADI	8	4.71e-03	5.32e-03	1.28e-03	1.07e-03	9.84e-04	8.43e-04
	12	5.62e-03	6.44e-03	5.99e-03	3.00e-03	7.77e-03	4.00e-04
	16	3.59e-03	9.80e-03	3.26e-03	4.67e-03	4.12e-03	2.61e-04

5.5.2 Identifying a Quantum Advantage

"Quantum advantage" is a term used in literature to describe if there is substantial evidence to claim the benefits of using Quantum over classical ML. To find this, we must determine how much benefit performing high-dimensional feature mapping in quantum provides over well-known classical methods. The best way to identify this quantum advantage is to quantify the "target-kernel alignment" defined in Section 5.1 between the validation set and kernel matrices [115]. We used the PennyLane [134] implementation of the target alignment to calculate this similarity metric and presented it in Table 5.5.

The expectation is that if there is a quantifiable advantage to quantum kernels, there should be a significant difference between the classical and quantum KTAs. Interestingly, the target alignments reflect just that; there is a significant positive difference of almost 4.8% KTA.

In HAI, Belis Feature Map maintains an order of magnitude higher kernel alignment in all pre-processing methods, followed by Sakhnenko and the simple 2DoF. The classical kernels have distinctly lower KTAs, and the classification results described before reflect this. The HAI dataset was very nonlinear and complex, which can possibly explain this difference.

For SWaT, the comparison is not as clear. The Belis and Sakhnenko kernels had the best target alignment out of all, but the polynomial and linear kernel functions were reportedly better than the simple 2DoF model. However, this allows us to see that even some low-performing models may have higher geometric distance than seemingly better classifiers. This further emphasizes that kernel alignment is important, but not the only aspect of a good-performing model.

Finally, WADI represents an equal comparison of kernel alignment between quantum feature maps and classical functions and how it can affect classification. Quantum performed better with the Decision Tree feature selection, but the classical methods achieved similar results in NMF and PCA. Looking at the WADI section in

Table 5.5: Kernel Target Alignment of quantum and classical kernels to each pre-processed dataset.

Method		Belis [7]	2DOF	Sakh. [8]	RBF	Poly	Linear
HAI	Tree	2.71e-02	3.58e-03	1.90e-02	2.78e-03	2.11e-03	6.80e-04
	NMF	1.74e-04	1.94e-05	6.56e-05	1.01e-05	1.00e-05	5.02e-06
	PCA	5.46e-03	1.89e-03	3.96e-03	8.92e-04	7.81e-04	4.28e-04
SWaT	Tree	2.31e-01	4.37e-02	2.10e-01	5.01e-02	5.62e-02	2.49e-02
	NMF	1.95e-01	1.07e-01	2.04e-01	1.24e-01	1.03e-01	6.64e-02
	PCA	2.10e-01	6.29e-02	1.81e-01	4.20e-02	4.66e-02	2.16e-02
WADI	Tree	1.39e-01	2.64e-01	1.80e-01	1.88e-01	1.91e-01	1.00e-01
	NMF	9.65e-02	5.91e-02	1.05e-01	9.08e-02	7.89e-02	6.19e-02
	PCA	1.10e-01	1.13e-01	1.50e-01	6.92e-02	6.90e-02	3.39e-02

Table 5.5, a similar trend appears in the kernel target alignment. In "Tree", 2DoF achieves much better classification and alignment results; however, in all other preprocessing methods, the classical and quantum were very much alike in classification and alignment.

On average, there is a 0.048 increase for quantum kernels in kernel alignment across all the datasets. This presents a 91.023% change over classical, demonstrating a higher kernel alignment for quantum over classical kernel functions.

5.6 Cyberphysical Monitor for 3D Printing

Next, we will look into real-world applicability using a robust data monitoring framework for 3D printing. Here we will demonstrate the generalized approach to monitoring the printing network, detail our data gathering methods and implementation, and show the results of applying this chapter's QSVM model to the data collected by the design.

5.6.1 Proposed Conceptualization of a Cyberphysical Monitor for 3D Printing

So far this chapter has designed a machine learning model assuming already gathered train and test CPS datasets. However, it is equally as important to lay out the framework of a fully established anomaly-detecting tool, from data gathering to quantum kernel SVMs. Particularly, we will focus on capturing data and processing it to detect anomalies in additive manufacturing ICS systems.

In the CPS paradigm discussed in Figure 1.1, Level 2 contains the supervisor and data monitoring systems. Typically, SCADA systems utilize information transferred to them through Level 1. However, this opens a very dangerous vulnerability, as network information can be modified by man-in-the-middle attacks.

Thus, our supervisory system should be able to gather the information through both the Level 1 control network and directly from the physical Level 0 CAN-based devices. This includes using extra infrastructure to gather the telemetry information. Additionally, gathering input and control information, such as the g-code submitted by the 3D printer, should provide context. The supervisory system has direct access to separate, trusted devices that directly probe the devices and CAN buses. This information is aggregated at the supervisory system, which hands it over to the trained QSVM anomaly detector model detailed in Section 5.3.

5.6.2 3D Printing Cyberattack Testing Setup

In order to prove the efficacy of this machine learning and data acquisition model, we fully implemented a 3D printing cyberattack testbench using the Snapmaker 2.0 All-in-One Printer [36]. We chose this particular printer due to its use of an insecure CAN bus and highly vulnerable firmware and network stack. As a result, we gathered cyberattack data for the physical sensor data, network metadata, and CAN bus data. A depiction of our testing apparatus is shown in Figure 5.12.

The scenario of the testing apparatus is as follows: The attacker(s) have performed man-in-the-middle attacks to acquire access to both the physical CAN bus and the TCP connection between the printer and supervisory computer. The selection of HTTP via LAN instead of CAN for Level 1 networking was based on the existing commercial network stack of the Snapmaker and Marlin frameworks, which provides a more realistic impression of a modern vulnerability. This also provides us with interesting data, as CAN bus-related data is already being captured at Level 0.

The structure of the testbench features multiple physical and virtual machines located on the same subnet. The client computer is connected to the 3D printer via consistent TCP heartbeats and streams g-code data through individual commands to the 3D printer's Marlin API using HTTP POSTs. This device streams the g-code to the printer continuously with HTTP.

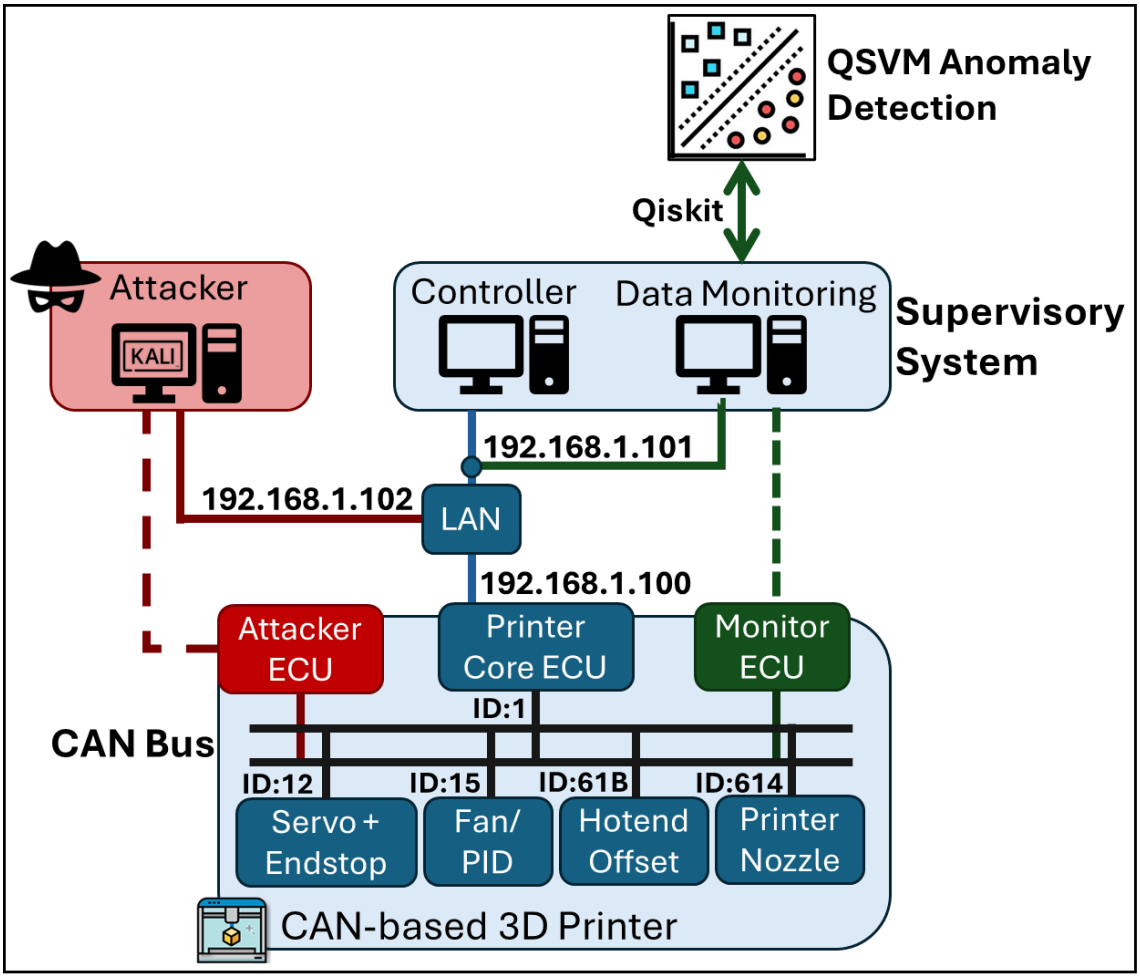


Figure 5.12: A cyberattack testbench and implementation of the data acquisition framework. Information is gathered by monitoring the network, low-level buses, and physical devices. The testbench utilizes TCP/IP LAN for Level 1 and CAN bus for Level 0. The attacker has access to LAN and CAN through trojan devices on both networks.

The data monitoring system runs on the client PC to take advantage of network monitoring techniques. Using Python's Scapy [135] package, it performs sniffing on the network interface card to passively gather network data. Furthermore, it connects to a custom firmware SAM C21 microcontroller via PySerial [136] to monitor the CAN bus and directly acquire information from the printer's internals. This is the main component of this case study.

Finally, the attacker is comprised of both a separate physical machine running Kali Linux and a custom CAN-spoofing microcontroller that takes commands from the physical machine via PySerial. Additionally, Scapy is used by the attacker to achieve man-in-the-middle via ARP spoofing, modify packets in transit, and spoof handshakes/packets. This ultimately allows the attacker to modify and fabricate Level 1 sensor status messages, g-code transmissions, and Level 0 CAN bus messages containing cyberphysical sensor/actuator control information.

5.6.3 Cyberattack Dataset

The collected data spans approximately one hour of normal printing and one hour of printing with periodic cyberattacks for two separate models, or STLs. Simple cube and cylinder 3D models were selected to be able to continue "normal" data collection even after devastating CPS attacks. This helped to make the data less trivial to detect anomalies and to reflect realistic printer farm recovery method standards more closely.

Normal Training Data

The training data resembles the structure of the previous three CPS datasets. The cube and cylinder prints consist of 57 minutes and 75 minutes respectively of non-anomalous data for unsupervised learning. As before, we emphasize unsupervised learning due to the severe unpredictability of attacks. Data is gathered asynchronously due to the nature of collecting from two independent subsystems.

However, the firmware of the printer conveniently spaces them out in approximately half-second and two-second intervals for CAN and Network data, respectively, resulting in an overall resolution of 2 seconds. The data is windowed as before, making the number of samples approximately equal to the other datasets.

Anomalous Testing Data

The testing data consists of the same cube-STL and cylinder-STL prints but the attacker's scripts have been enabled. For simplicity, the same attack intervals and times are used on both print datasets. Individual and combinations of attacks were performed on the 3D printing system, with gaps of time to allow the device to return to normal working status before another attack is performed. This helped to allow for a balance of anomalous and non-anomalous data within the prints.

The attacks performed are shown in Table 5.6. We focused on performing attacks that manipulated g-code commands, manipulated returning network temperature and positional data, denied service to the LAN and CAN communications, and performed CAN injection attacks to modify target temperature, current temperature, and fan speeds. This helped provide a compact, yet comprehensive, set of attacks that would resemble the ones presented from SCADA attacks like Stuxnet [99]. Spoofing g-code entailed sending HTTP POSTs to */api/execute_code* in the printer, while "g-code modification" involved using MITM to modify victim-transmitted packets to include deviation and error in the movements and temperature.

Unlike the previous data where the anomalies were purposefully situated at the end of the testing set, we have included them throughout the data in time-series order for this data. This allows us to retain the order of the anomalies and get a better insight into the behavior of a "recovering CPS system" after a devastating cyberattack, and how it could affect our anomaly detection results.

Table 5.6: Summary of attacks evaluated on the security network dataset.

ID	Name	Description	Target	Start	End
N1	G-code Spoof	Sends random g-code commands to printer	LAN	5:50	14:10
N2	G-code Modify	Modifies victim's g-code packets	LAN	27:30	39:11
N3	Status Modify	Modifies printer's printer status packets	LAN	30:51	41:31
C1	Temperature Spoof	Overloads the CAN bus with incorrect temperature values	CAN	44:51	50:35
C2	Fan Speed Denial of Service	Overloads the CAN bus with dangerous fan speed values	CAN	50:41	54:50
C3	Target Temperature Spoof	Overloads the CAN bus with dangerous target temperature values	CAN	54:50	57:00

5.6.4 Data Processing

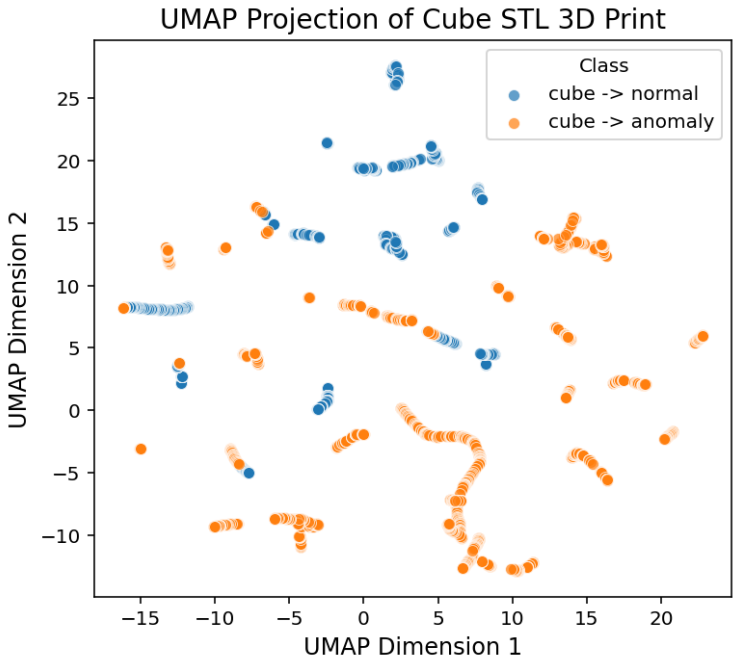
Typical procedures for feature removal were used to prune away unimportant fields and information from the system. For example, fields like the "Restricted" or "CAN-FD Enabled" field will never be set in a valid CAN frame. Additionally, we did not utilize components like an air filtration system, so our dataset removes this unnecessary field for our particular application.

Unlike the previous CPS datasets, this data was in a raw form and required additional processing to extract important features. Thus, we began by performing feature extraction on the data to derive features related to the axis movement and nozzle and bed temperatures. This provided us with statistically derived features for the temperatures and derivative-based features for the movement. All non-zero features unrelated to these two categories were kept for optimal detection. Finally, we reduce the features using PCA as described in Section 5.3.1 down to a final count of 12 features.

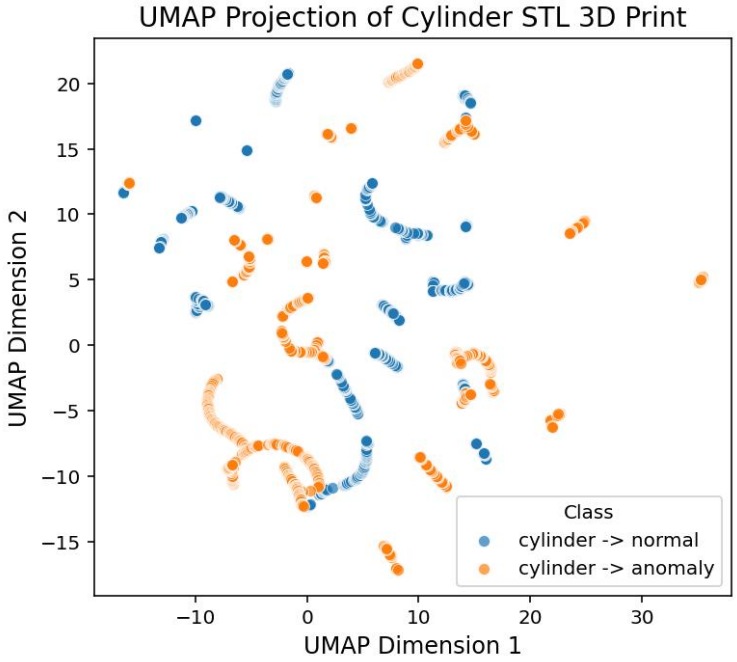
The distribution of the datasets is presented in a UMAP 2-D projection in Figure 5.13. From visual analysis alone of both Figures 5.13a and 5.13b, we can infer a severe need of a high-dimensional feature map to be able to separate the anomalies using an SVM hyperplane. However, this demonstrates a fairly strong potential for acceptable results as the anomalies remain in their clusters, possibly suggesting the use of K-means clustering. This provides us with our hypothesis that only the highly expressive and complex feature maps, like Belis' Feature Map and ZZ Feature Map, would be able to properly handle this data.

5.6.5 Experimental Results

In our experimentation, we utilized the two degree of freedom feature maps: Belis [7], Sakhnenko's Circuit 10 [8], and the basic 2DoF rotation map. We presented the performances for both datasets in Table 5.7 for each of the feature maps. Additionally, we report the best classical kernel function for comparison.



(a) Cube STL Dataset.



(b) Cylinder STL Dataset.

Figure 5.13: 2-Dimensional Uniform Manifold Approximation and Projection (UMAP) of the 3D Printer datasets.

Table 5.7: Performance metrics across different feature maps on the gathered Cube 3D Printer dataset.

Dataset	Feature Map	Precision	Recall	F1	Accuracy
Cube	**Belis**	**0.76**	**0.77**	**0.76**	**77%**
	SakhC10	0.75	0.72	0.72	72%
	2DoF	0.49	0.57	0.52	57%
	Classical RBF	0.72	0.66	0.67	66%
Cylinder	**Belis**	**0.79**	**0.77**	**0.73**	**77%**
	SakhC10	0.81	0.76	0.72	76%
	2DoF	0.66	0.68	0.64	68%
	Classical RBF	0.55	0.52	0.52	52%

Upon initial inspection, we can clearly see that our hypothesis derived from the UMAP data was correct. We achieved up to 77% accuracy on the Cube and Cylinder datasets with the Belis [7] feature map. The lower complexity models, like 2DoF and classical feature maps, performed fairly poorly on this data, while the models with entanglement and quantum superposition performed acceptably. The results are similar to those of the HAI dataset, suggesting that this dataset too contains a large amount of relationships and time-series information that can often trick our model into misclassifying in tricky scenarios.

Furthermore, we present the classification graph for Cube STL across all samples in Figure 5.14 to show the detection performance "over time". We can see that the beginning of the print is classified as an anomaly, which likely means it struggles greatly with the "build-up" of the print speed from the slow beginning. Additionally, we can see that the overall "shape" of the graph is preserved. This trend means that the overall behavior of the anomaly detector is working, as a real-world scenario only needs some anomalies in each "group" of attacks to trigger a human-machine interaction response. This also shows that the "gradual" time-series trends are lost in the SVM kernel, which is one downside of using OCSVMs.

Overall, we can see a really strong detection model with plenty of room to grow. Our 3-D printer detection model clearly picks up on multiple high-risk anomalies, even when subtly masked by other sensor changes.

5.7 Discussion, Conclusion, and Future Work

In this work, we introduced a method of anomaly detection in CPS systems based on the quantum advantage for high-dimensional computation of data. By utilizing a kernel prepared from the fidelity of parameterized quantum circuits, this work flattens the higher dimensional feature space of real-world data for effective SVM classification. We have demonstrated a fairly clear quantitative advantage to quantum

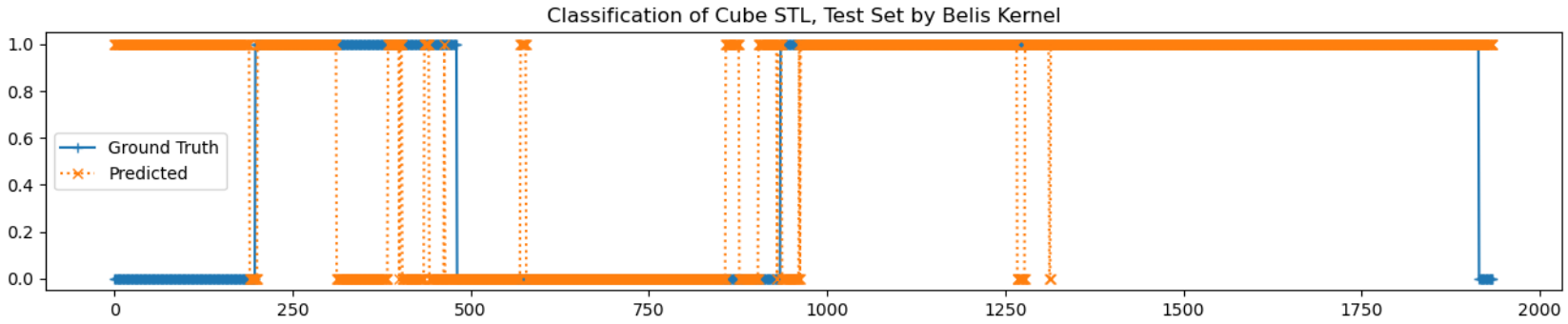


Figure 5.14: Prediction vs. Ground Truth classifications of the Cube STL data anomalies.

anomaly detection methods in our experiments, but classification performance is hardly the only deciding factor for safe and effective real-world implementation.

5.7.1 Discussion

Quantum has shown superiority in the classification of all three datasets, demonstrating a fairly clear quantitative advantage to quantum anomaly detection methods in our experiments. Our quantum experiments demonstrated a 13.3% increase in F1 classification and 91.023% stronger kernel-target alignment than classical methods. However, classification and KTA performance are hardly the only deciding factors to safe and effective real-world implementation. A summarization of the best kernel configurations and feature counts for all three datasets are provided in Table 5.8.

We can see that the best two feature maps were the simplest: Belis et al.'s kernel and the simple U2-gate "2DoF" kernel, both presenting over 0.86 F1 score over all three datasets. This shows that the simple models thrive better as they avoid overfitting models and still provide quantum efficiency to the kernel calculations. Additionally, we can see that tree selection preprocessing to 12 and 16 features (6 and 8 qubits) presented the best results, likely due to feature importance playing a role.

Furthermore, we presented a case study of 3D printing anomaly detection using our quantum machine learning model. Our highly complex datasets resemble real-world commercial printing conditions and further demonstrate the effect of a "quantum advantage" in CPS data anomaly detection. The best configurations of the 3D printer data are shown in Table 5.8. Our best performing model achieved 77% accuracy, approximately 11% higher than the best classical feature map.

5.7.2 Conclusion

In conclusion, we have effectively investigated the use of Quantum Hybrid Support Vector Machines in the CPS anomaly detection domain and shown a clear quantitative

Table 5.8: Summary of best classification metrics for quantum kernels on ICS/CPS and 3D Printer data.

	Best Configuration:			Results:		
Dataset	Kernel	Feature	Method	Precision	Recall	F1 Score
SWaT	2DoF	12	Tree	0.95	0.95	0.95
HAI	Belis [7]	16	Tree	0.88	0.87	0.86
WADI	2DoF	16	Tree	0.90	0.88	0.88
Cube	Belis [7]	12	PCA	0.76	0.77	0.76
Cylinder	Belis [7]	12	PCA	0.79	0.77	0.73

advantage to using them over their classical counterparts for ICS data. With a consistently significant improvement in both classification and alignment in all three real-world datasets, our quantum kernels demonstrated strong potential for applicability, even in a noisy environment such as NISQ computing. As shown in this work, deep parametrization and experimentation are essential for QAD. For example, if a feature map is too simple, the fundamental concepts are extremely simplified, resulting in possibly little to no quantum advantage and easy classical reproducibility [9]. Similarly, quantum fidelity error can increase possibly exponentially with qubit count, a trend analogous to the "curse of dimensionality" [133]. This can cause a model to quickly overfit to the highly exponential system and destroy classification performance. These concerns vary across application domains, requiring strong consideration.

Additionally, longer circuits are often noiser and can easily drop classification performance, as shown in previous QML literature. Thus, feature maps with more gate operations and complexity create more noise. While noise-mitigated algorithms are beginning to surface in quantum-cloud computing [137], the effect of noise is still a confounding variable that must be considered and alleviated.

Furthermore, NISQ computing is highly limited in availability and practicality resulting in lengthy queues for resources and very limited computing time [138]. This is currently infeasible for large datasets as the number of total shots would be upper bounded by $O(|x_{input}| \times |x_{train}| \times shots)$. Thus, past literature usually uses fewer than 500 data samples for QML in their experimentation. For kernel applications, large feature maps with many hyperparametrized gates still remain a theory with no current feasible application.

There are many practicality and availability problems in modern quantum computing that greatly restrict the immediate commercialization of a responsive quantum-enabled AD system. However, with the continuing trend of rapid technical advances in Quantum [137, 139], the results presented here present a clear potential in the advantages of quantum computing.

This work pulls together quantum machine learning with real-world applications through a thorough study. Future work can build upon this investigation by performing deeper analyses with future quantum computing resources, kernel algorithms, and feature maps. With better computing resources, the highly exhaustive number of quantum shots for kernels can be optimized, more qubits can enable more features, and intricate, low-noise feature maps can be better implemented. On a related note, our work only features quantum simulations of real computer noise models. While these noise models are highly realistic to current IBM systems, this study must be extended to actual shots on real hardware, including alternative quantum technologies like ion-trap machines.

As the realm of quantum kernel computing only continues to grow in applicability and accessibility in the real world, research should continue to determine the true impact the quantum kernel domain can have on the cyberphysical computing space surrounding us every day.

Chapter 6

Conclusions and the Future of Quantum CPS Security Research

In this work, we developed a fully-implemented cyberphysical security framework that protects each layer of the CPS paradigm shown in Figure 1.1. Our design improves upon existing pre-quantum work in each category by applying a quantum-safe and quantum-enhanced approach to each carefully designed solution. By comprehensively tackling the problems layer-by-layer, we were able to secure from all levels of attacks from within CPS and ICS ecosystems.

In Chapter 3, we discussed the use of post-quantum cryptographic suites in resource-constrained vehicular Controller Area Networks (CAN). With the use of the upgraded protocol, CAN Flexible Data-rate (CAN-FD), we were able to efficiently secure against rising threats from smart vehicle communication with infrastructure, network, infotainment, and other vehicles. This vehicular framework utilizes post-quantum authentication schemes while taking over 75% less transmission time and messages than existing PQC CAN solutions. Our scheme's transmission time performance is irrespective of the cryptographic suite, allowing us to propose the use of CRYSTALS-KYBER [52] instead of the now-insecure SIDH cryptography [72]. Furthermore, we discussed the use of the server as a firewall to monitor and gate

transmissions going in and out of the CAN bus to the various aforementioned smart vehicle infrastructures. This protects the CAN system from exposure to external attack vectors and ensures a uniform level of security across the entire CAN bus.

Secondly, in Chapter 4, we expanded to post-quantum security of inter-device CAN networks in the context of additive manufacturing farms. Unlike the intra-vehicular networks from before, these CAN networks were designed to support the ability to "plug-and-play" 3D printer parts while providing a root-of-trust security framework against hardware trojans. We require approximately 90% fewer messages for authentication than existing post-quantum solutions, while utilizing only 12% additional overhead in hardware for creating a root-of-trust, tree-based network. Additionally, our framework provides significant versatility that can uniquely customize the CAN network physically and virtually to provide varying levels of isolation and denial-of-service security.

Lastly, in Chapter 5, we secured the supervisory and control layer of the CPS paradigm by applying quantum machine learning models to the telemetry and operational data collected by a typical SCADA system in a cyberphysical system. We experimented on three highly popular water-infrastructure CPS datasets, demonstrating a 13.3% increase in F1 score classification and 91.023% stronger kernel-target alignment compared to classical methods. Additionally, our work demonstrates a very clear quantitative advantage to quantum anomaly detection using state-of-the-art metrics while demonstrating the potential nuances of utilizing a too highly complex model. We further expanded the machine learning model to include a framework for gathering information on the CPS Level 0 and Level 1 networks of CAN-based 3D printers by implementing a comprehensive cyberattack testbench. Using highly complex attack models, we were able to achieve up to 77% accuracy using quantum machine learning, providing an advantage of up to 19% over classical kernel functions.

The frameworks proposed in this work are designed to be implemented in tandem as a comprehensive security suite. While many of the concepts focused on CAN-based

3D printing and vehicular networks, the concepts proposed here can also expand well beyond these focused topics, including other forms of CPS networking protocols and applications.

6.1 Future Work

Quantum should not be regarded as only a potential danger to the cybersecurity field. While attackers can utilize the exponential speedup, we have also shown in this work how that exponential speedup can assist defenders as well. Various other security concepts have emerged from recent literature, such as quantum symmetric encryption [140], quantum Physically Unclonable Functions (PUFs) [141], and quantum key distribution via "quantum internet" [142].

Whether used for good or bad, quantum computation is likely going to be a pillar of emerging technologies in the future. Combining the content of this work, one can create a quantum-safe and quantum-enabled cybersecurity ecosystem for essentially any CPS system. However, just as stronger threats continue to emerge, cybersecurity remains an uphill battle as security solutions can never provide an indefinite and fully secure framework for security in any domain. Thus, it is highly important to analyze the future work and approaches researchers can take to advance the field of secure computing in cyber-physical environments.

6.1.1 Device Fingerprinting and Key Material Generation

In Chapters 3 and 4, we discuss the use of Physically Unclonable Functions, or PUFs, for generating authentication and key materials for proving identities of secure devices. In this work, we abstract away the PUF by assuming the implementation provides a strong, high-entropy device capable of generating consistent keys. A few examples of PUFs that could work well are sensor-based PUFs [143, 144], MRAM-based PUFs [145], and arbiter-PUFs [146].

Implementation of the post-quantum frameworks in hardware should also be completed to further explore the aging/implementation of the PUF, along with energy, computational time, and area costs of hardware-accelerated modules. Physical implementation using proposed PUF designs will provide further insight into the effectiveness of minimizing traffic and key sizes. For example, previous work implements both the PUF and the hardware primitives of their design, allowing for more metric data to be gathered from their design [3]. Exploration into other post-quantum systems in the frameworks presented in this dissertation will also demonstrate the robustness and flexibility in real-world applications.

6.1.2 Privacy Preservation in Post-Quantum CPS Frameworks

Similarly, one major topic of cyberphysical system security is how to balance the security methods with privacy-preserving methods. For example, in Chapter 5, we suggest applying a data gathering model to identify anomalies in the cloud, likely applied directly over the top of the framework in Chapter 4. Research should be conducted to be able to safely gather this information while anonymizing the driver and avoiding severe data leakage from the gathered data from the models.

Privacy-preserving post-quantum is a strong topic in modern cybersecurity. A few examples of strong ideas that could be applied to the CPS domain are Post-Quantum Homomorphic Encryption [147], post-quantum anonymous credential protocols [148], and zero-knowledge proofs of lattice-based post-quantum cryptography suites [149]. Additionally, one could also include pseudonym certificates when communicating using external V2X messages [150], also briefly described in Chapter 3.

6.1.3 Anomaly Detection using Quantum Machine Learning

In Chapter 5, we utilized quantum support vector machines (QSVMs) for detecting anomalies in cyberphysical systems. While we proved the extremely effective use

of QSVMs, many cyberphysical systems may not work best with support vector machines. For example, extremely time-series-reliant datasets may prefer Long-Short-Term-Memory (LSTM) models [151]. Quantum neural networks [152], quantum Boltzmann machines [153], and quantum autoencoders [116]. In future work, expansion on utilizing these other models, whether as competing models or as ensemble learning [154], should be the next step toward continuing the work presented in this dissertation.

Furthermore, Chapter 5 discussed feasibility of applying quantum SVMs. Particularly, quantum SVM models cause up to $O(|x_{input}| \times |x_{train}|)$ runs of a quantum computer to calculate. This is currently infeasible for the modern-day quantum cloud infrastructure, but this can change in the future. Future work could focus on continuing experimentation with QML on future quantum cloud computing, providing more opportunities for heavier workloads.

Furthermore, optimization can be performed to reduce the workload required on the quantum computer for performing anomaly detection. Further improvements could also be made, such as approximate computing models [155], distributed computing [156], and high-performance computing classical models for emulating quantum behaviors [157]. Providing a solution to the incredible overhead of QSVMs would greatly improve its real-world applicability in resource-limited environments. Overall, these ideas would effectively bridge the security "gap" between the classical and quantum spaces.

Bibliography

[1] S. Hartzell, C. Stubel, and T. Bonaci, "Security analysis of an automobile controller area network bus," *IEEE Potentials*, vol. 39, no. 3, pp. 19–24, 2020. xii, 12, 58

[2] C. Labrado, H. Thapliyal, and S. P. Mohanty, "Fortifying vehicular security through low overhead physically unclonable functions," *J. Emerg. Technol. Comput. Syst.*, vol. 18, sep 2021. xii, xv, 26, 27, 30, 31, 38, 39, 40, 43

[3] A. S. Siddiqui, Y. Gui, J. Plusquellic, and F. Saqib, "A secure communication framework for ecus," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, pp. 1307–1313, 08 2017. xii, xv, 30, 31, 32, 37, 38, 39, 40, 43, 137

[4] P. Ravi, V. K. Sundar, A. Chattopadhyay, S. Bhasin, and A. Easwaran, "Authentication protocol for secure automotive systems: Benchmarking post-quantum cryptography," in *2020 IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1–5, 2020. xii, xv, xvii, 31, 37, 38, 39, 40, 42, 43, 55, 60, 78, 81, 82

[5] A. S. Siddiqui, Y. Gui, J. Plusquellic, and F. Saqib, "A secure communication framework for ecus," *Advances in Science, Technology and Engineering Systems Journal*, vol. 2, no. 3, pp. 1307–1313, 2017. xvii, 55, 78, 82

[6] C. Labrado, H. Thapliyal, and S. P. Mohanty, "Fortifying vehicular security through low overhead physically unclonable functions," *J. Emerg. Technol. Comput. Syst.*, vol. 18, sep 2021. xvii, 56, 60, 78, 82

[7] V. Belis, K. A. Woźniak, and E. Puljak, "Quantum anomaly detection in the latent space of proton collision events at the lhc," *Communications Physics*, vol. 7, p. 334, Oct 2024. xviii, 92, 102, 104, 113, 114, 116, 118, 125, 128, 131

[8] A. Sakhnenko, C. O'Meara, K. J. B. Ghosh, C. B. Mendl, G. Cortiana, and J. Bernabé-Moreno, "Hybrid classical-quantum autoencoder for anomaly detection," *Quantum Machine Intelligence*, vol. 4, p. 27, Sep 2022. xviii, 92, 102, 103, 104, 118, 125

[9] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, pp. 209–212, Mar 2019. xviii, 23, 24, 102, 103, 104, 132

[10] O. Research, "Global cyber physical system market 2020 by company, regions, type and application, forecast to 2025." https://www.orbisresearch.com/reports/index/2015-2025-global-cyber-physical-system-cps-market-research-by-type-end-use-and-region, 2020. 1, 7

[11] M. Lom, O. Pribyl, and M. Svitek, "Industry 4.0 as a part of smart cities," in *2016 Smart Cities Symposium Prague (SCSP)*, pp. 1–6, 2016. 1

[12] C. Alippi, *Intelligence for Embedded Systems*, vol. 89. Springer International Publishing, 2014. 1, 7

[13] C. Greer, M. Burns, D. Wollman, and E. Griffor, "Cyber-physical systems and internet of things," *NIST Special Publication*, vol. 1900, p. 202, 2019. 2

[14] G. Wen, W. Yu, and X. Yu, "Complex cyber-physical networks: From cybersecurity to security control," *Journal of Systems Science and Complexity*, vol. 30, pp. 46–67, Feb 2017. 2

[15] R. P. Feynman, "Quantum mechanical computers," *Foundations of Physics*, vol. 16, no. 6, pp. 507–531, 1986. 2

[16] Q. contributors., "Qiskit: An open-source framework for quantum computing," 2023. 2

[17] C. Monroe, "Ionq quantum computers: clear to scale," in *APS March Meeting Abstracts*, vol. 2021, pp. P10–002, 2021. 2, 16, 20

[18] J. M. Arrazola, V. Bergholm, and K. Brádler, "Quantum circuits with many photons on a programmable nanophotonic chip," *Nature*, vol. 591, pp. 54–60, Mar 2021. 2

[19] A. Javadi-Abhari, M. Treinish, and K. Krsulich, "Quantum computing with Qiskit," 2024. 2, 20, 103, 113

[20] H. Y. Wong, *Shor's Algorithm*, pp. 289–298. Cham: Springer International Publishing, 2024. 2, 20

[21] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," vol. 41, July 2009. 3

[22] N. Liu and P. Rebentrost, "Quantum machine learning for quantum anomaly detection," *Phys. Rev. A*, vol. 97, p. 042315, Apr 2018. 3

[23] K. A. Coulibaly, A. Llaria, and M. Hamlich, "Industry 4.0: Internet of things and cyber-physical systems for the implementation of pedagogical factory digital twin," in *Smart Applications and Data Analysis*, (Cham), pp. 16–29, Springer Nature Switzerland, 2024. 3, 8

[24] M. Alanazi, A. Mahmood, and M. J. M. Chowdhury, "Scada vulnerabilities and attacks: A review of the state-of-the-art and open issues," *Computers & Security*, vol. 125, p. 103028, 2023. 10

[25] R. I. Davis, A. Burns, R. J. Bril, and J. J. Lukkien, "Controller area network (can) schedulability analysis: Refuted, revisited and revised," *Real-Time Systems*, vol. 35, pp. 239–272, Apr 2007. 10

[26] M. Schreiner, L. Donat, and S. Köngeter, "Introduction of can fd into the next generation of vehicle e/e architectures," *IEEE International Conference on Communications*, 2017. 10, 37

[27] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015. 11, 50, 55

[28] M. Bozdal, M. Samie, and I. Jennions, "A survey on can bus protocol: Attacks, challenges, and potential solutions," in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 201–205, IEEE, 2018. 11

[29] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 6123–6141, 2022. 11

[30] S. Mukherjee, H. Shirazi, I. Ray, J. Daily, and R. Gamble, "Practical dos attacks on embedded networks in commercial vehicles," in *International Conference on Information Systems Security*, pp. 23–42, Springer, 2016. 11

[31] R. Buttigieg, M. Farrugia, and C. Meli, "Security issues in controller area networks in automobiles," in *2017 18th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*, pp. 93–98, 2017. 13

[32] I. Gibson, D. W. Rosen, B. Stucker, M. Khorasani, D. Rosen, B. Stucker, and M. Khorasani, *Additive manufacturing technologies*, vol. 17. Springer, 2021. 13

[33] T. D. Ngo, A. Kashani, G. Imbalzano, K. T. Nguyen, and D. Hui, "Additive manufacturing (3d printing): A review of materials, methods, applications and challenges," *Composites Part B: Engineering*, vol. 143, pp. 172–196, 2018. 13

[34] I. Gibson, D. Rosen, B. Stucker, and M. Khorasani, *Introduction and Basic Principles*, pp. 1–21. Cham: Springer International Publishing, 2021. 13

[35] J. Krüger, W. Gu, H. Shen, M. Mukelabai, R. Hebig, and T. Berger, "Towards a better understanding of software features and their characteristics: A case study of marlin," in *Proceedings of the 12th International Workshop on Variability Modelling of Software-Intensive Systems*, VAMOS '18, (New York, NY, USA), p. 105–112, Association for Computing Machinery, 2018. 13

[36] "Snapmaker 2.0 all-in-one 3d printer," 2022. 14, 72, 120

[37] J.-C. Chin, H. Thapliyal, and T. Cultice, "Can bus: The future of additive manufacturing (3d printing)," *IEEE Consumer Electronics Magazine*, pp. 1–6, 2022. 14, 48

[38] J. Wu, Y. Tan, and G. Ma, "3d printing monitoring platform based on the internet of things," *IET Conference Proceedings*, pp. 5 .–5 .(1), January 2015. 14, 50

[39] S. Belikovetsky, M. Yampolskiy, J. Toh, J. Gatlin, and Y. Elovici, "dr0wned – Cyber-Physical attack with additive manufacturing," in *11th USENIX Workshop on Offensive Technologies (WOOT 17)*, (Vancouver, BC), USENIX Association, aug 2017. 14, 56

[40] M. Yampolskiy, L. Graves, J. Gatlin, A. Skjellum, and M. Yung, "What did you add to my additive manufacturing data?: Steganographic attacks on 3d printing

files," in *Proceedings of the 24th International Symposium on Research in Attacks, Intrusions and Defenses*, RAID '21, (New York, NY, USA), p. 266–281, Association for Computing Machinery, 2021. 14

[41] M. McCormack, S. Chandrasekaran, G. Liu, T. Yu, S. DeVincent Wolf, and V. Sekar, "Security analysis of networked 3d printers," in *2020 IEEE Security and Privacy Workshops (SPW)*, pp. 118–125, 2020. 14

[42] S. Moore, P. Armstrong, T. McDonald, and M. Yampolskiy, "Vulnerability analysis of desktop 3d printer software," in *2016 Resilience Week (RWS)*, pp. 46–51, 2016. 14

[43] C. Song, F. Lin, Z. Ba, K. Ren, C. Zhou, and W. Xu, "My smartphone knows what you print: Exploring smartphone-based side-channel attacks against 3d printers," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, (New York, NY, USA), p. 895–907, Association for Computing Machinery, 2016. 14

[44] T. Cultice and H. Thapliyal, "Vulnerabilities and attacks on can-based 3d printing/additive manufacturing," *IEEE Consumer Electronics Magazine*, pp. 1–7, 2023. 14, 49

[45] M. Kjaergaard, M. E. Schwartz, J. Braumüller, P. Krantz, J. I.-J. Wang, S. Gustavsson, and W. D. Oliver, "Superconducting qubits: Current state of play," *Annual Review of Condensed Matter Physics*, vol. 11, no. Volume 11, 2020, pp. 369–395, 2020. 16

[46] R. LaRose and B. Coyle, "Robust data encodings for quantum classifiers," *Phys. Rev. A*, vol. 102, p. 032420, Sep 2020. 19

[47] S. Resch and U. R. Karpuzcu, "Benchmarking quantum computers and the impact of quantum noise," *ACM Comput. Surv.*, vol. 54, July 2021. 20

[48] Z. Cai, R. Babbush, S. C. Benjamin, S. Endo, W. J. Huggins, Y. Li, J. R. McClean, and T. E. O'Brien, "Quantum error mitigation," *Rev. Mod. Phys.*, vol. 95, p. 045005, Dec 2023. 20

[49] J. Clark and H. Thapliyal, "Peephole optimization for quantum approximate synthesis," in *2024 25th International Symposium on Quality Electronic Design (ISQED)*, pp. 1–8, 2024. 20

[50] R. LaPierre, *Shor Algorithm*, pp. 177–192. Cham: Springer International Publishing, 2021. 21, 50

[51] M. Grassl, B. Langenberg, M. Roetteler, and R. Steinwandt, "Applying grover's algorithm to aes: Quantum resource estimates," in *Post-Quantum Cryptography* (T. Takagi, ed.), (Cham), pp. 29–43, Springer International Publishing, 2016. 21, 50

[52] T. Hasija, K. R. Ramkumar, A. Kaur, S. Mittal, and B. Singh, "A survey on nist selected third round candidates for post quantum cryptography," in *2022 7th International Conference on Communication and Electronics Systems (ICCES)*, pp. 737–743, 2022. 21, 22, 61, 134

[53] D. Moody, G. Alagic, D. Apon, D. Cooper, Q. Dang, J. Kelsey, Y.-K. Liu, C. Miller, R. Peralta, R. Perlner, A. Robinson, D. Smith-Tone, and J. Alperin-Sheriff, "Status report on the second round of the nist post-quantum cryptography standardization process," 2020-07-22 2020. 22, 28, 30, 36

[54] A. Chailloux, M. Naya-Plasencia, and A. Schrottenloher, "An efficient quantum collision search algorithm and implications on symmetric cryptography," in *Advances in Cryptology – ASIACRYPT 2017* (T. Takagi and T. Peyrin, eds.), (Cham), pp. 211–240, Springer International Publishing, 2017. 22

[55] K. A. Tychola, T. Kalampokas, and G. A. Papakostas, "Quantum machine learning—an overview," *Electronics*, vol. 12, no. 11, 2023. 22

[56] J. Li, Y. Li, J. Song, J. Zhang, and S. Zhang, "Quantum support vector machine for classifying noisy data," *IEEE Transactions on Computers*, vol. 73, no. 9, pp. 2233–2247, 2024. 22

[57] F. V. Massoli, L. Vadicamo, G. Amato, and F. Falchi, "A leap among quantum computing and quantum neural networks: A survey," *ACM Comput. Surv.*, vol. 55, Dec. 2022. 22

[58] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, 1995. 23

[59] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings of the fifth annual workshop on Computational learning theory*, pp. 144–152, 1992. 23

[60] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," *Phys. Rev. Lett.*, vol. 122, p. 040504, Feb 2019. 24

[61] T. Cultice and H. Thapliyal, "Puf-based post-quantum can-fd framework for vehicular security," *Information*, vol. 13, no. 8, 2022. 26, 60

[62] S. Chen, J. Hu, Y. Shi, Y. Peng, J. Fang, R. Zhao, and L. Zhao, "Vehicle-to-everything (v2x) services supported by lte-based systems and 5g," *IEEE Communications Standards Magazine*, vol. 1, no. 2, pp. 70–76, 2017. 28

[63] M. R. Moore, R. A. Bridges, F. L. Combs, and A. L. Anderson, "Data-driven extraction of vehicle states from can bus traffic for cyberprotection and safety," *IEEE Consumer Electronics Magazine*, vol. 8, no. 6, pp. 104–110, 2019. 28

[64] H. J. Jo and W. Choi, "A survey of attacks on controller area networks and corresponding countermeasures," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–19, 2021. 28

[65] T. George, J. Li, A. P. Fournaris, R. K. Zhao, A. Sakzad, and R. Steinfeld, "Performance evaluation of post-quantum tls 1.3 on embedded systems." Cryptology ePrint Archive, Paper 2021/1553, 2021. https://eprint.iacr.org/2021/1553. 30

[66] P. K. Sadhu, V. P. Yanambaka, S. P. Mohanty, and E. Kougianos, "Easy-sec: Puf-based rapid and robust authentication framework for the internet of vehicles," 2022. 30

[67] S. Woo, H. J. Jo, and D. H. Lee, "A practical wireless attack on the connected car and security protocol for in-vehicle can," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 993–1006, 2015. 31, 32

[68] E. Alkim, N. Bindel, J. Buchmann, Ö. Dagdelen, E. Eaton, G. Gutoski, J. Krämer, and F. Pawlega, "Revisiting tesla in the quantum random oracle model," in *Post-Quantum Cryptography* (T. Lange and T. Takagi, eds.), (Cham), pp. 143–162, Springer International Publishing, 2017. 31

[69] D. P. Möller and R. E. Haas, *Guide to automotive connectivity and cybersecurity*. Springer, 2019. 39

[70] B. Koziel, R. Azarderakhsh, and M. M. Kermani, "A high-performance and scalable hardware architecture for isogeny-based cryptography," *IEEE Transactions on Computers*, vol. 67, no. 11, pp. 1594–1609, 2018. 41

[71] B.-Y. Sung, K.-B. Kim, and K.-W. Shin, "An aes-gcm authenticated encryption crypto-core for iot security," in *2018 International Conference on Electronics, Information, and Communication (ICEIC)*, pp. 1–3, 2018. 41

[72] W. Castryck and T. Decru, "An efficient key recovery attack on sidh (preliminary version)." Cryptology ePrint Archive, Paper 2022/975, 2022. https://eprint.iacr.org/2022/975. 41, 46, 134

[73] D. Moody, D. Smith-Tone, T. Farah, and A. Robinson, "Status report on the fourth round of the nist post-quantum cryptography standardization process," Tech. Rep. NISTIR 8413, National Institute of Standards and Technology, Apr. 2024. 42

[74] J. Zhao, Y. Chen, and Y. Gong, "Study of connectivity probability of vehicle-to-vehicle and vehicle-to-infrastructure communication systems," in *2016 IEEE 83rd vehicular technology conference (VTC Spring)*, pp. 1–4, IEEE, 2016. 44

[75] H. Wen, Q. A. Chen, and Z. Lin, "{Plug-N-Pwned}: Comprehensive vulnerability analysis of {OBD-II} dongles as a new {Over-the-Air} attack surface in automotive {IoT}," in *29th USENIX security symposium (USENIX Security 20)*, pp. 949–965, 2020. 44

[76] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehlé, "Crystals-kyber: a cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, IEEE, 2018. 46

[77] T. Cultice, J. Clark, W. Yang, and H. Thapliyal, "A novel hierarchical security solution for controller-area-network-based 3d printing in a post-quantum world," *Sensors*, vol. 23, no. 24, 2023. 48, 79, 83

[78] M. Bozdal, M. Samie, and I. Jennions, "A survey on can bus protocol: Attacks, challenges, and potential solutions," in *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pp. 201–205, 2018. 49

[79] T. Cultice, J. Clark, and H. Thapliyal, "Lightweight hierarchical root-of-trust framework for can-based 3d printing security," in *Proceedings of the Great Lakes Symposium on VLSI 2023*, GLSVLSI '23, (New York, NY, USA), p. 215–216, Association for Computing Machinery, 2023. 51

[80] Q. Wang and S. Sawhney, "Vecure: A practical security framework to protect the can bus of vehicles," in *2014 International Conference on the Internet of Things (IOT)*, pp. 13–18, 2014. 55

[81] C. Beckwith, H. S. Naicker, S. Mehta, V. R. Udupa, N. T. Nim, V. Gadre, H. Pearce, G. Mac, and N. Gupta, "Needle in a haystack: Detecting subtle malicious edits to additive manufacturing g-code files," *IEEE Embedded Systems Letters*, vol. 14, no. 3, pp. 111–114, 2022. 56, 60, 79

[82] Z. Shi, C. Kan, W. Tian, and C. Liu, "A Blockchain-Based G-Code Protection Approach for Cyber-Physical Security in Additive Manufacturing," *Journal of Computing and Information Science in Engineering*, vol. 21, p. 041007, 02 2021. 56, 60, 79

[83] F. Chen, G. Mac, and N. Gupta, "Security features embedded in computer aided design (cad) solid models for additive manufacturing," *Materials & Design*, vol. 128, pp. 182–194, 2017. 56, 60

[84] S. Belikovetsky, Y. A. Solewicz, M. Yampolskiy, J. Toh, and Y. Elovici, "Digital audio signature for 3d printing integrity," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 5, pp. 1127–1141, 2019. 56, 58, 79

[85] M. Wu, Z. Song, and Y. B. Moon, "Detecting cyber-physical attacks in cybermanufacturing systems with machine learning methods," *Journal of Intelligent Manufacturing*, vol. 30, pp. 1111–1123, Mar 2019. 56, 60

[86] M. Yampolskiy, W. E. King, J. Gatlin, S. Belikovetsky, A. Brown, A. Skjellum, and Y. Elovici, "Security of additive manufacturing: Attack taxonomy and survey," *Additive Manufacturing*, vol. 21, pp. 431–457, 2018. 56, 60

[87] D. R. Safford and M. Wiseman, "Hardware rooted trust for additive manufacturing," *IEEE Access*, vol. 7, pp. 79211–79215, 2019. 56, 60, 79

[88] J. Wu, Y. Tan, and G. Ma, "3d printing monitoring platform based on the internet of things," in *Fifth Asia International Symposium on Mechatronics (AISM 2015)*, pp. 1–5, 2015. 59

[89] R. McKuckie and E. Kilbride, "Flash programming via can," *Application note AN1828/D rev*, vol. 1, pp. 4–5, 2004. 59

[90] B. Palaniswamy, S. Camtepe, E. Foo, and J. Pieprzyk, "An efficient authentication scheme for intra-vehicular controller area network," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3107–3122, 2020. 60

[91] A. J. Michaels, V. S. S. Palukuru, M. J. Fletcher, C. Henshaw, S. Williams, T. Krauss, J. Lawlis, and J. J. Moore, "Can bus message authentication via co-channel rf watermark," *IEEE Transactions on Vehicular Technology*, vol. 71, no. 4, pp. 3670–3686, 2022. 60

[92] B. S. Bari, K. Yelamarthi, and S. Ghafoor, "Intrusion detection in vehicle controller area network (can) bus using machine learning: A comparative performance study," *Sensors*, vol. 23, no. 7, 2023. 60

[93] D. McGrew and J. Viega, "The galois/counter mode of operation (gcm)," *submission to NIST Modes of Operation Process*, vol. 20, pp. 0278–0070, 2004. 61

[94] J. Bos, L. Ducas, E. Kiltz, T. Lepoint, V. Lyubashevsky, J. M. Schanck, P. Schwabe, G. Seiler, and D. Stehle, "Crystals - kyber: A cca-secure module-lattice-based kem," in *2018 IEEE European Symposium on Security and Privacy (EuroS&P)*, pp. 353–367, 2018. 61, 74

[95] S. Paul, F. Schick, and J. Seedorf, "Tpm-based post-quantum cryptography: A case study on quantum-resistant and mutually authenticated tls for iot environments," in *Proceedings of the 16th International Conference on*

Availability, Reliability and Security, ARES '21, (New York, NY, USA), Association for Computing Machinery, 2021. 74

[96] Y. Huang, M. Huang, Z. Lei, and J. Wu, "A pure hardware implementation of crystals-kyber pqc algorithm through resource reuse," *IEICE Electronics Express*, vol. 17, no. 17, pp. 20200234–20200234, 2020. 74

[97] Y. Xing and S. Li, "A compact hardware implementation of cca-secure key exchange mechanism crystals-kyber on fpga," *IACR Transactions on Cryptographic Hardware and Embedded Systems*, vol. 2021, p. 328–356, Feb. 2021. 74

[98] Markets and Markets, "Cyber-physical systems (cps) market size." https://www.marketsandmarkets.com/Market-Reports/cyber-physical-systems-cps-market-150375126.html, September 2024. 85

[99] S. Al-Rabiaah, "The "stuxnet" virus of 2010 as an example of a "apt" and its "recent" variances," in *2018 21st Saudi Computer Society National Computer Conference (NCC)*, pp. 1–5, 2018. 86, 123

[100] B. Ofner, R. Mesika, N. Erez, S. Brizinov, and C. Fradkin, "State of XIoT Security: 2H 2022," tech. rep., Claroty, H2 2022. 86

[101] C. Velandia-Cardenas, Y. Vidal, and F. Pozo, "Wind turbine fault detection using highly imbalanced real scada data," *Energies*, vol. 14, no. 6, 2021. 86

[102] A. A. Cárdenas, S. Amin, and Z.-S. Lin, "Attacks against process control systems: risk assessment, detection, and response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*, p. 355–366, 2011. 89

[103] C.-S. Cho, W.-H. Chung, and S.-Y. Kuo, "Cyberphysical security and dependability analysis of digital control systems in nuclear power plants,"

IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 46, no. 3, pp. 356–369, 2016. 89

[104] D. Ding, Q.-L. Han, X. Ge, and J. Wang, "Secure state estimation and control of cyber-physical systems: A survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 176–190, 2021. 89

[105] Y.-C. Chen, V. Mooney, and S. Grijalva, "A survey of attack models for cyber-physical security assessment in electricity grid," in *2019 IFIP/IEEE 27th International Conference on Very Large Scale Integration (VLSI-SoC)*, pp. 242–243, 2019. 89

[106] C.-W. Ten, G. Manimaran, and C.-C. Liu, "Cybersecurity for critical infrastructures: Attack and defense modeling," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 4, pp. 853–865, 2010. 89

[107] S. Talukder, M. Ibrahim, and R. Kumar, "Resilience indices for power/cyberphysical systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 4, pp. 2159–2172, 2021. 89

[108] T. Cultice, D. Ionel, and H. Thapliyal, "Smart home sensor anomaly detection using convolutional autoencoder neural network," in *2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pp. 67–70, 2020. 89

[109] C. Yin, S. Zhang, J. Wang, and N. N. Xiong, "Anomaly detection based on convolutional recurrent autoencoder for iot time series," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 1, pp. 112–122, 2022. 89

[110] M. R. G. Raman and A. P. Mathur, "A hybrid physics-based data-driven framework for anomaly detection in industrial control systems," *IEEE*

Transactions on Systems, Man, and Cybernetics: Systems, vol. 52, no. 9, pp. 6003–6014, 2022. 89

[111] B. Kim, M. A. Alawami, and E. Kim, "A comparative study of time series anomaly detection models for industrial control systems," *Sensors*, vol. 23, no. 3, 2023. 89

[112] O. Tushkanova, D. Levshun, A. Branitskiy, E. Fedorchenko, E. Novikova, and I. Kotenko, "Detection of cyberattacks and anomalies in cyber-physical systems: Approaches, data sources, evaluation," *Algorithms*, vol. 16, no. 2, 2023. 89, 96, 112

[113] N. Cristianini, J. Shawe-Taylor, A. Elisseeff, and J. Kandola, "On kernel-target alignment," in *Advances in Neural Information Processing Systems* (T. Dietterich, S. Becker, and Z. Ghahramani, eds.), vol. 14, MIT Press, 2001. 91

[114] T. Wang, D. Zhao, and S. Tian, "An overview of kernel alignment and its applications," *Artificial Intelligence Review*, vol. 43, pp. 179–192, Feb 2015. 91, 92

[115] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, S. Boixo, H. Neven, and J. R. McClean, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, p. 2631, May 2021. 91, 117

[116] V. S. Ngairangbam, M. Spannowsky, and M. Takeuchi, "Anomaly detection in high-energy physics using a quantum autoencoder," *Phys. Rev. D*, vol. 105, p. 095004, May 2022. 91, 138

[117] S. Alvi, C. W. Bauer, and B. Nachman, "Quantum anomaly detection for collider physics," *Journal of High Energy Physics*, vol. 2023, p. 220, Feb 2023. 91

[118] J.-M. Liang, S.-Q. Shen, M. Li, and L. Li, "Quantum anomaly detection with density estimation and multivariate gaussian distribution," *Phys. Rev. A*, vol. 99, p. 052310, May 2019. 91

[119] S. Egginger, A. Sakhnenko, and J. M. Lorenz, "A hyperparameter study for quantum kernel methods," *Quantum Machine Intelligence*, vol. 6, p. 44, Jul 2024. 91

[120] J. Schuhmacher, L. Boggia, and V. Belis, "Unravelling physics beyond the standard model with classical and quantum anomaly detection," *Machine Learning: Science and Technology*, vol. 4, p. 045031, nov 2023. 92

[121] S. L. Wu, S. Sun, W. Guan, and C. Zhou, "Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the lhc," *Phys. Rev. Res.*, vol. 3, p. 033221, Sep 2021. 92

[122] D. Pranjić, F. Knäble, P. Kunst, D. Kutzias, D. Klau, C. Tutschku, L. Simon, M. Kraus, and A. Abedi, "Unsupervised quantum anomaly detection on noisy quantum processors," 2024. 92

[123] C. Correa-Jullian, S. Cofre-Martel, and G. San Martin, "Exploring quantum machine learning and feature reduction techniques for wind turbine pitch fault detection," *Energies*, vol. 15, no. 8, 2022. 92, 93

[124] A. Gouveia and M. Correia, "Towards quantum-enhanced machine learning for network intrusion detection," in *2020 IEEE 19th International Symposium on Network Computing and Applications (NCA)*, pp. 1–8, IEEE, 2020. 92, 93

[125] K. Tscharke, S. Issel, and P. Debus, "Semisupervised anomaly detection using support vector regression with quantum kernel," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, vol. 01, pp. 611–620, 2023. 92, 93

[126] A. Kukliansky, M. Orescanin, C. Bollmann, and T. Huffmire, "Network anomaly detection using quantum neural networks on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 5, pp. 1–11, 2024. 92, 93

[127] T. Cultice, M. S. Hassan Onim, A. Giani, and H. Thapliyal, "Anomaly detection for real-world cyber-physical security using quantum hybrid support vector machines," in *2024 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 619–624, 2024. 92, 93, 99, 107

[128] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Critical Information Infrastructures Security* (G. Havarneanu, R. Setola, H. Nassopoulos, and S. Wolthusen, eds.), (Cham), pp. 88–99, Springer International Publishing, 2017. 94

[129] Q. Lin, S. Verwer, R. Kooij, and A. Mathur, "Using datasets from industrial control systems for cyber security research and education," in *Critical Information Infrastructures Security* (S. Nadjm-Tehrani, ed.), (Cham), pp. 122–133, Springer International Publishing, 2020. 94

[130] M. Wang, A. Huang, Y. Liu, X. Yi, J. Wu, and S. Wang, "A quantum-classical hybrid solution for deep anomaly detection," *Entropy*, vol. 25, no. 3, 2023. 93

[131] H.-K. Shin, W. Lee, J.-H. Yun, and H. Kim, *HAI 1.0: HIL-Based Augmented ICS Security Dataset*. USA: USENIX Association, 2020. 94

[132] C. M. Ahmed, V. R. Palleti, and A. P. Mathur, "Wadi: a water distribution testbed for research in the design of secure cyber physical systems," in *Proceedings of the 3rd International Workshop on Cyber-Physical Systems for Smart Water Networks*, CySWATER '17, p. 25–28, Association for Computing Machinery, 2017. 96

[133] R. Shaydulin and S. M. Wild, "Importance of kernel bandwidth in quantum machine learning," *Phys. Rev. A*, vol. 106, p. 042407, Oct 2022. 102, 132

[134] V. Bergholm, J. Izaac, M. Schuld, and C. Gogolin, "Pennylane: Automatic differentiation of hybrid quantum-classical computations," 2022. 117

[135] P. Biondi and the Scapy contributors, "Scapy: Packet crafting for python." https://scapy.net, 2024. Version 2.6.1. 122

[136] C. Liechti and the pySerial contributors, "pyserial: Python serial port extension." https://pyserial.readthedocs.io/, 2024. Version 3.5. 122

[137] G. Q. AI and Collaborators, "Quantum error correction below the surface code threshold," *Nature*, Dec 2024. 132

[138] M. Fellous-Asiani, J. H. Chai, R. S. Whitney, A. Auffèves, and H. K. Ng, "Limitations in quantum computing from resource constraints," *PRX Quantum*, vol. 2, p. 040335, Nov 2021. 132

[139] E. Lötstedt and K. Yamanouchi, "Comparison of current quantum devices for quantum computing of heisenberg spin chain dynamics," *Chemical Physics Letters*, vol. 836, p. 140975, 2024. 132

[140] G. Alagic, T. Gagliardoni, and C. Majenz, "Unforgeable quantum encryption," in *Advances in Cryptology – EUROCRYPT 2018* (J. B. Nielsen and V. Rijmen, eds.), (Cham), pp. 489–519, Springer International Publishing, 2018. 136

[141] K. Phalak, A. A. Saki, M. Alam, R. O. Topaloglu, and S. Ghosh, "Quantum puf for security and trust in quantum computing," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 11, no. 2, pp. 333–342, 2021. 136

[142] Y. Cao, Y. Zhao, Q. Wang, J. Zhang, S. X. Ng, and L. Hanzo, "The evolution of quantum key distribution networks: On the road to the qinternet," *IEEE Communications Surveys & Tutorials*, vol. 24, no. 2, pp. 839–894, 2022. 136

[143] C. Labrado, H. Thapliyal, S. Prowell, and T. Kuruganti, "Use of thermistor temperature sensors for cyber-physical system security," *Sensors*, vol. 19, no. 18, 2019. 136

[144] A. Degada and H. Thapliyal, "An integrated trng-puf architecture based on photovoltaic solar cells," *IEEE Consumer Electronics Magazine*, vol. 10, no. 4, pp. 99–105, 2021. 136

[145] J. Kang, D. Han, K. Lee, S. Ko, D. Koh, C. Park, J. Ahn, M. Yu, M. Pakala, S. Noh, H. Lee, J. Kwon, K.-J. Kim, J. Park, S. Lee, J. Lee, and B.-G. Park, "Highly reliable magnetic memory-based physical unclonable functions," *ACS Nano*, vol. 18, no. 20, pp. 12853–12860, 2024. PMID: 38718347. 136

[146] D. P. Sahoo, D. Mukhopadhyay, R. S. Chakraborty, and P. H. Nguyen, "A multiplexer-based arbiter puf composition with enhanced reliability and security," *IEEE Transactions on Computers*, vol. 67, no. 3, pp. 403–417, 2017. 136

[147] K. Lauter, "Postquantum opportunities: Lattices, homomorphic encryption, and supersingular isogeny graphs," *IEEE Security and Privacy*, vol. 15, no. 4, pp. 22–27, 2017. 137

[148] O. Blazy, C. Chevalier, G. Renaut, T. Ricosset, E. Sageloli, and H. Senet, "Efficient implementation of a post-quantum anonymous credential protocol," in *Proceedings of the 18th International Conference on Availability, Reliability and Security*, ARES '23, (New York, NY, USA), Association for Computing Machinery, 2023. 137

[149] R. Steinfeld, "Post-quantum zero-knowledge proofs and applications," in *Proceedings of the 10th ACM Asia Public-Key Cryptography Workshop*, APKC '23, (New York, NY, USA), p. 1, Association for Computing Machinery, 2023. 137

[150] T. Yoshizawa and B. Preneel, "Post-quantum impacts on v2x certificates – already at the end of the road," in *2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring)*, pp. 1–6, 2023. 137

[151] O. I. Provotar, Y. M. Linder, and M. M. Veres, "Unsupervised anomaly detection in time series using lstm-based autoencoders," in *2019 IEEE International Conference on Advanced Trends in Information Theory (ATIT)*, pp. 513–517, IEEE, 2019. 138

[152] A. Kukliansky, M. Orescanin, C. Bollmann, and T. Huffmire, "Network anomaly detection using quantum neural networks on noisy quantum computers," *IEEE Transactions on Quantum Engineering*, vol. 5, pp. 1–11, 2024. 138

[153] L. Moro and E. Prati, "Anomaly detection speed-up by quantum restricted boltzmann machines," *Communications Physics*, vol. 6, no. 1, p. 269, 2023. 138

[154] J. Vanerio and P. Casas, "Ensemble-learning approaches for network security and anomaly detection," in *Proceedings of the workshop on big data analytics and machine learning for data communication networks*, pp. 1–6, 2017. 138

[155] J. Clark and H. Thapliyal, "Peephole optimization for quantum approximate synthesis," in *2024 25th International Symposium on Quality Electronic Design (ISQED)*, pp. 1–8, 2024. 138

[156] D. Barral, F. J. Cardama, G. Diaz-Camacho, D. Faílde, I. F. Llovo, M. Mussa-Juane, J. Vázquez-Pérez, J. Villasuso, C. Piñeiro, N. Costas, *et al.*, "Review of distributed quantum computing: from single qpu to high performance quantum computing," *Computer Science Review*, vol. 57, p. 100747, 2025. 138

[157] S. Corli, L. Moro, D. Dragoni, M. Dispenza, and E. Prati, "Quantum machine learning algorithms for anomaly detection: A review," *Future Generation Computer Systems*, p. 107632, 2024. 138

Vita

Tyler Andrew Cultice was born in 1999 in Springfield, Ohio, where he spent most of his life. He earned a Bachelor of Science in Computer Engineering from the University of Kentucky in 2021. Following this, Tyler enrolled in the University of Tennessee Knoxville, where he pursued doctoral studies in Computer Engineering under the supervision of Professor Himanshu Thapliyal. Tyler's doctoral research has focused on post-quantum cryptography for IoT and embedded firmware and anomaly detection models using emerging technologies, such as cloud-based quantum machine learning. At the start of his doctoral career, his research was focused around cyberphysical system security in 3D printing and vehicular communication in collaboration with the Southeastern Advanced Machine Tools Network. In 2022, he was awarded the prestigious "NSF Graduate Research Fellowship" for his role as a national student leader in embedded and quantum cybersecurity research. He won the Best Paper Award at the ISVLSI 2024 conference for his work in using quantum machine learning methods to secure cyberphysical data. Tyler also performed duties as a graduate teaching assistant at the University of Tennessee Knoxville since 2021, where he won the UTK Gonzalez Outstanding Graduate Teaching Assistant award for involvement in developing HDL and embedded system courses and guiding undergraduate students through both coursework and independent research projects.

Upon graduation in August 2025, Tyler plans to work as a firmware engineer at Google, where he will apply his expertise in the development of firmware, driver, and hardware design.