HARDWARE INTRUSION DETECTION FOR SUPPLY-CHAIN THREATS TO CRITICAL INFRASTRUCTURE EMBEDDED SYSTEMS

BY

NATHAN J. EDWARDS

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2012

Urbana, Illinois

Adviser:

Professor David M. Nicol

ABSTRACT

Along with an increase in cyber security concerns for critical infrastructure applications, there is a growing concern and lack of solutions for cyber-based supply chain and device life-cycle threats. The challenge for this application space is that cost-driven engineering and market viability requires the use of commercially available off-the-shelf (COTS) components or just-in-time (JIT) manufacturing processes for sub-assemblies most of which originate from unsecured foreign facilities. In addition, many of the deployed embedded system devices are easily accessible (i.e. poor physical security) and can easily be tampered with or altered during their life-cycle such that the authentication or integrity of the devices cannot be assured. In this research I propose the foundations of a new technology that helps address these growing issues with a hardware-based intrusion detection system. This technology combines the use of an analog signal response from a resistor-capacitor circuit and machine learning techniques to not only identify the presence of a hardware Trojan on an inter-chip communication bus at 100% accuracy for the dataset of over 2000 measurements, but which also correctly distinguishes between several types of implanted Trojans at 89% accuracy. And while this research has focused on the security of inter-chip communication, it demonstrates the possibility of using low-power analog signals for device-level information assurance.

To my wife and three children, for their love, support, and understanding of time spent away from home to build electronic gadgets and participate in a greater science.

Jesus looked at them and said, "With man this is impossible, but with God all things are possible."

Matthew 19:26 (NIV)

ACKNOWLEDGMENTS

Some of the materials contained herein are subject to pending patent application(s)

This material is based upon work supported by United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000, and by United States Department of Energy's Office of Electricity Delivery and Energy Reliability contract award number DE-OE0000097 (Trustworthy Cyber Infrastructure for the Power Grid).

The author acknowledges the support and guidance of many great research staff and faculty at the University of Illinois at Urbana-Champaign and at Sandia National Laboratories.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ADC	Analog-to-digital converter
AIC	Akaike Information Criterion
AMI	Advanced metering infrastructure
ANN	Artificial neural networks
ANOVA	Analysis of variance
ANSI	American National Standards Institute
ASIC	Application specific integrated circuit
AUC	Area under curve
BJT	Bipolar junction transistor
CMOS	Complimentary metal–oxide–semiconductor
COTS	Commercially available off-the-shelf
DIP	Dual inline pins
DMZ	Demilitarized zone for computer networks
DoD	U.S. Department of Defense
DUT	Device under test
eCIS	Enterprise Component Information System
FDA	U.S. Food and Drug Administration
FPR	False positive rate
GLM	Generalized linear model
I^2C	Inter-integrated circuit [communication protocol]
IDPS	Intrusion detection and prevention systems
IDS	Intrusion detection system
IPS	Intrusion prevention system
ISO	International Organization for Standardization
JIT	Just-in-time [manufacturing]
KCL	Kirchhoff’s current law
KVL	Kirchhoff’s voltage law
KNN	K^{th} nearest neighbor
LRM	Logistic regression model
MNLM	Multinomial logit model
MOS	Metal–oxide–semiconductor
MPW	Multi-project wafer
MSE	Mean square error

NARUC	National Association of Regulatory Utility Commissioners
NCER	National Center for Electronics Recycling
NOINTR	No intruder present
OEM	Original equipment manufacturer
PCA	Principal component analysis
PPV	Positive predictive value
PUF	Physical unclonable function(s)
ROC	Receiver operating characteristic
SAR	Successive approximation register
SCRM	Supply chain risk mitigation
SPDT	Single-pole double-throw
SVM	Support vector machine
THD	Total harmonic distortion
TPM	Trusted platform module
TPR	True positive rate
XOR	Exclusive-OR (logical)

CHAPTER 1 INTRODUCTION

Critical infrastructure such as energy delivery systems are characterized by reliability, availability, and dependability. The infrastructures are expected to provide the services necessary for daily function of our society regardless of unpredictable changes in their operating environment. And despite only moderate expansion of the infrastructure, our dependencies on them are increasing. We expect them to operate more efficiently, at a lower cost, and also to provide the end-users with accurate information on the system's current status. All these expectations seem to be a driving force behind the smart technologies. Consequently, the societal expectations have caused rapid growth in the communication links between technologies (e.g. smart phones, smart appliances, smart meters, etc.).

Yet the interconnectivity also significantly increases the cybersecurity risks for our nation's critical infrastructures. For example, customer devices linked to the Internet are also allowed, by design, to connect to the advanced metering infrastructure (AMI) which can alter a local power-grid's load distribution through remote meter disconnects. So significant cybersecurity questions emerge: "Can a customer's device also control the local energy delivery system?" or the inverse "Can a smart meter control customer's home appliances and resultantly affect the load on the power grid?" A recent report the National Association of Regulatory Utility Commissioners (NARUC) states that "Cybersecurity must encompass not only utility-owned systems, but some aspects of customer and third party components that interact with the grid"[1]. So the need for more cybersecurity solutions comes with an increasing number of interconnections.

In addition, the societal expectations of lower costs have helped transition the electronics industry's supply chain to use components that originate from countries that might have higher potentials for cyber crime and known histories of producing counterfeit parts. These same components have made it into safety critical applications and likely into energy delivery systems [2]. One primary reason is the challenge of detecting such threats to an embedded system device. While much security research in academia focuses on the upper levels shown in Fig. 1.1, it can be stated that the number of trust assumptions is large. This can basically be summarized as the software trusts the hardware to perform as expected. Yet the fallacy is that there may be extra hardware or side channels that leak critical information, none of which can be easily

detected by state-of-the art intrusion detection systems that are primarily focused on the software/protocol layers. Ultimately the cyber attack landscapes along with the challenges to defend systems are rapidly increasing in complexity.

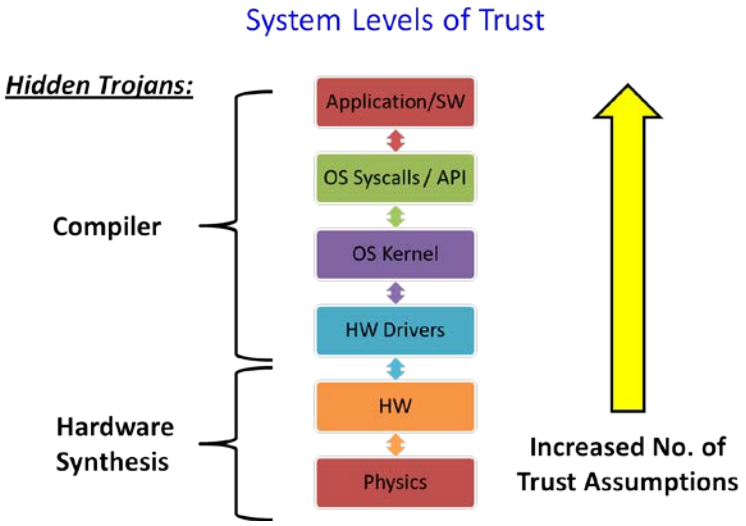


Figure 1.1: System levels of trust

1.1 The Importance of Security Research on Advanced Metering Infrastructure

The primary goal of this research is to help address these growing cybersecurity risks to energy delivery systems and in particular those of the AMI domain. Although there might be debate on the criticality of an AMI system, for example the failure of one device might not cause catastrophic effects like that experienced in the 2003 Blackout, the importance of the AMI system has a much broader impact on society. The viability of the smart grid depends on the AMI system as it is the public face of the smart grid. It is attached to their homes, the operation of an AMI system affects their finances, and it is the mechanism for customers to interact with and directly see the promised benefits of the smart grid. Customers have taken ownership and a vested interest in AMI domain without regard for other major improvements to transmission and distribution automation equipment. If an AMI system fails due to reliability issues, business management (i.e. demand response pricing), or cyber-attack, then the whole Smart Grid fails in the eyes of a customer.

The AMI domain is also of high technical interest and has significant cybersecurity challenges. The devices are easily accessible with relatively low risk to an attacker (compared to high-energy substations). AMI meters are also networked to both utility resources and customer

owned resources connected to the Internet, essentially becoming a public gateway into a utility control system. And while the implemented ZigBee Smart Energy communication standards provide some cybersecurity safeguards on the wireless communication link, these low-power embedded devices likely cannot support industry standard DMZ technology that protects corporate networks from Internet-based intrusions. Similarly, the AMI system is designed to allow connection of many smart appliances or heating and cooling control systems, which also means that a cyber attacker might be able to gain access to thousands or millions of homes if long-lasting protection mechanisms are not designed into the AMI system. Consequently, this large-scale potential violation of privacy means that public trust and confidence of the smart grid depends on a secure AMI system.

Along with an increase in cybersecurity concerns for energy delivery applications, there is a growing risk and lack of solutions for cyber-based threats to supply chain and the device lifecycle. The challenge for the AMI application space is that cost-driven engineering and market viability requires the use of commercially available off-the-shelf (COTS) components or just-in-time (JIT) manufacturing processes for sub-assemblies, most of which originate from unsecured foreign facilities. In addition, many of the deployed embedded system devices are easily accessible (i.e. have poor physical security) and can easily be tampered with or altered during their lifecycle such that the authenticity or integrity of the devices cannot be assured.

1.2 Summary of Key Issues in Supply Chain and Product Lifecycle

- Backdoors can easily be inserted during just-in-time manufacturing processes or device lifecycle.
- Hardware/software attacks might be latent or intermittent.
- Hardware attacks may not be visible to software or network IDS (side-channel communication).

1.3 Research Goals

This research seeks to help solve these growing cybersecurity challenges for critical infrastructure systems with innovative solutions that are forward compatible with the increased risks and future vulnerabilities. This project will not only address the current issues of AMI system components, but also will help explore solutions that can be pragmatic and cost-effectively deployed by AMI technology vendors. And while I focus on advanced metering

infrastructure, the results of the work can be broadly applied to energy delivery systems and ultimately ensure public trust, confidence and the security of smart grid technologies.

1.4 Research Objectives

- Model hardware-intruder based attacks using dynamic response.
- Create proof-of-concept for low-level intrusion detection system that can identify embedded system device hardware eavesdropping and intruders.
- Perform statistical analysis to determine the optimal model to detect hardware-intruders.
- Determine several design considerations with regard to IDS sensitivity, accuracy and manufacturability.

1.5 Fundamental Questions

- Can we detect hardware Trojans?
- Can our detection mechanisms provide any additional information that helps characterize the hardware Trojan?
- Can we distinguish between Trojan classes? How do we do this?
- Will this concept system detect real hardware Trojans?
- How well does it perform?
- How do we test this concept: Simulation vs. hardware?
- How do we analyze the data?

1.6 Design Challenges

- Hardware-based IDS solution must be low-cost in order for industry to use technology on AMI devices.
- Hardware-based IDS solution must not have significant impact on the AMI device or network performance.
- Analog circuit components degrade independently over time and are also susceptible to changes in the operating environment.

CHAPTER 2 CYBERSECURITY ISSUES IN CRITICAL INFRASTRUCTURE SUPPLY CHAIN AND DEVICE LIFECYCLE

2.1 Lifecycle of an Embedded System Device

In order to comprehend the complexities and vulnerabilities of an embedded system device that will become part of our nation's critical infrastructure, we must first understand the engineering process for a typical device from inception of idea through end-of-life disposal. This process is commonly known as a product lifecycle. Although there are many different models used by industry and business development experts to characterize and manage a product lifecycle (e.g waterfall, agile development, etc.), it can be generalized into six phases: 1) concept, 2) design, 3) supply chain management, 4) production, 5) operation, and 6) disposal as shown in Fig. 2.1.

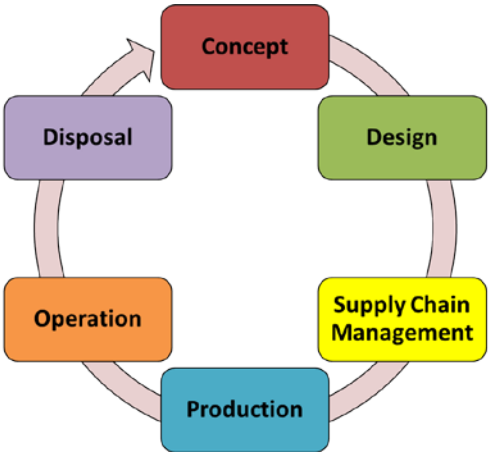


Figure 2.1: Typical device lifecycle

Although in many business practices, the order of the product lifecycle phases are not strictly adhered to, the generalization in Fig. 2.1 shows some sequential dependencies between phases. The concept phase includes the initial development of the product idea and a system requirements assessment along with a preliminary market survey. The design phase includes all the hardware and software design and the selection of specific components to use. The design phase is usually iteratively active through product release so that it can support or redesign any

issues that arise from fielding the devices. The design phase usually includes many cycles of validation and verification testing of the product and may even include a controlled pre-release of the product to get real usage information and to help identify unforeseeable design flaws so they can be corrected before a general release of the product.

Once the components are specified in the design phase, supply chain management activities begin to evaluate and vet potential suppliers and to secure longer-term contracts as a just-in-time (JIT) supplier or for a product lifetime purchase of components. In FDA regulated medical devices, manufacturers are required to support their products approximately seven years beyond the end of production, so contracts with riders for lifetime purchase of parts is common. Critical infrastructure devices currently do not have this stringent requirement of maintenance support, although many vendors offer long-term or lifetime warranties of their products given the initial cost of investment for infrastructure owners. Although it is not well characterized in Fig. 2.1, the supply chain management of a product, or at least the procurement records, may exist many years after the end of production and makes this phase the most complex of a product lifecycle.

The production phase can be described by months to years of preparation, tool modification, process validation and verification, statistical process control, quality management, and many other aspects that are traditionally incorporated into the discipline of industrial engineering. Much planning on the aforementioned topics occurs before the first full production run. Quality management in particular drives many of the common practices for manufacturing production as technology vendors seek to have a highly repeatable production process and to maintain their ISO certifications (ISO 9001, ISO 14001, ISO 27001). Much like supply chain management, the production phase is an on-going effort throughout the lifecycle of the product.

The operation phase includes shipping and receiving of new devices, installation and deployment of the infrastructure, and continuous operation by the asset owners. Many devices may actually be fielded for more than 20 years. A major challenge in the operation phase is deploying a new system. The infrastructure owners need to deploy the equipment and bring it online without disrupting existing services to their customers. Additionally, if an early device failure occurs, the owners must put into effect their contingency plans. During this phase, devices are exposed to unpredictable environmental changes and system stressors including

cyber events that are usually not well modeled during the device design, despite best efforts. This operation phase is a true test of a system's reliability.

The final phase in a product lifecycle, disposal, begins when a device reaches the end of its life, needs to be replaced with newer technology, or becomes obsolete. Current practice is to remove devices from service, sometimes send them to the manufacturer for a recycling discount on new equipment, or send them to general electronics recycling centers after marring or removing outer marking such as utility names, serial numbers, etc. It is unknown what becomes of these devices after they are sent for recycling, although there is a strong possibility that the circuit boards will be extracted for the metals while the components end up in the secondary markets [3].

Each phase in a product lifecycle has its own cybersecurity challenges and may require different mitigation techniques. This is why cybersecurity of the supply chain is very complex and very costly to manufacturers. For example, the disposal phase is typically overlooked during the development of a cybersecurity program. The challenge of the disposal phase is that the devices operating in critical infrastructure environments may still contain sensitive information in their non-volatile memory. Yet utility operators may not have the appropriate equipment or budget to access individual ICs within a device and erase all non-volatile memories. Another challenge is that administrators and public utilities will have difficulty justifying large costs related to specialized disposal of devices primarily because it does not increase the level of service to their constituents. This is the same struggle for cybersecurity programs in general.

While many manufacturers accept a "trade-in" of the old devices, even ones from other manufacturers, there are no established guidelines of how an asset owner should sanitize the old equipment of sensitive information. Even new standardizations with extensive guidelines and security controls such as the recent NIST Special Publication 800-53 do not account for proper disposal of devices that might contain sensitive information [4]. Furthermore, the U.S. Department of Commerce, Bureau of Industry and Security recommends that government entities need to "Establish guidance for the proper destruction, recycling, and/or disposal of electronic parts and systems" [2], yet utilities will run into similar budgetary justification issues to implement guidelines without the notion of regulatory compliance and the associated fines for non-compliance.

2.2 Supply Chain Concerns: A Real Problem

> "Last May, customs officials at the Port of Long Beach, Calif., intercepted a shipment of almost $1 million worth of fake SanDisk memory chips stashed inside nearly 2,000 karaoke machines, shipped in a container from a foreign country" [5].

Cybersecurity for the supply chain is a real problem. As demonstrated by this section's leading quote describing a 2011 event, supply chain risk mitigation (SCRM) is not solely a matter of protecting the financial information, contracts, design documents (including software and hardware) or even configuration data. There are many documented cases where counterfeit parts have infiltrated the supply chain of critical applications. Yet the generalized problem is much greater: if counterfeit parts can enter the supply chain either through forward (upstream) or backward (product returns or buy back) channels, what other well-crafted cyber vulnerabilities can be introduced through these same channels? This problem is not specific to military applications. A 2010 industry assessment reports that industrial/commercial and high reliability–industrial, which encompass critical infrastructure and energy delivery systems, were among the top categories in counterfeits incidents [2].

Consequently, supply chain risk mitigation must focus on prevention and detection, although neither of which are trivial. Prevention through stringent supplier evaluation, component traceability and legislative oversight may be the best defense against counterfeit parts [5], yet the cost to maintain such a program can be prohibitive. On the other hand, the detection of counterfeit parts or hardware and software Trojans could potentially be automated within a manufacturer's validation and verification testing process, yet technical solutions scarcely exist. An alarming fact in the 2010 report "Defense Industrial Base Assessment: Counterfeit Electronics" [2] states that circuit board assemblers who manufacture device subsystems, did not discover the counterfeit parts – they were primarily discovered after the parts had been returned as defective (see Fig. IV-7 and Fig. IV-8 shown in the report). It is possible that circuit board assemblers simply do not have the capability (staff, equipment or budget) to detect counterfeits, as similar testing of reliability, qualification, and verification requires a large effort [6][7]. Perhaps the counterfeit parts could have been detected sooner if inexpensive, automated and proven technical solutions existed.

Parts obsolescence and availability are also major concerns for supply chain risk management. For example, the technical and security risks increase if a power utility procures highly specialized equipment for specific systems, yet the "technology manufacturer ceases to provide replacement parts because technology has advanced since that equipment was fielded" [6]. The same issues can arise when a device manufacturer contracts a semiconductor company to design and fabricate application specific ICs, although in some cases the device manufacturer will opt for a lifetime buy based on the expected life of their devices. For this reason some technical reports state that availability rather than cost is the major driver toward the use of COTS components and ICs [7].

2.3 Supplier Evaluation and Vetting

In a supply chain domain, the evaluation and selection of suppliers are an important first step to mitigate risks. An evaluation of supplier business practices along with process control information and quality engineering documents help identify risks or supply chain vulnerabilities. Mitigation of the risks can be enforced through contracts, but the challenge still resides in the detection of low-quality, counterfeit, or subverted parts to show a breach of the contract. Supplier evaluation processes have evolved over the last several decades with efforts focused on critical government systems and are slowly surfacing in commercial markets that furnish our nation's critical infrastructure such as the telecom industry or energy delivery systems. Out of these extended efforts, several guidelines on best practices have been published by DoD, DOE, and the U.S. Department of Commerce. A good example of this is the 1996 DoD guidelines for evaluating original equipment manufacturers (OEM) shown below. And while it is dated, the general recommendations still hold true.

<u>DoD OEMs Supplier Evaluation Guidelines (circa 1996):</u> [7]

- Use qualified suppliers and parts (previously qualified to MIL-SPEC).
- Use silicon die characterized for mil/industrial grades.
- Assure design margin (the subject of several best practices).
- Use circuit simulation and modeling.
- Ask supplier if parts will work in your application.
- Talk to other users.

- Record all information on a given part.
- Take advantage of new technology.
- Ring out and eliminate problems with environmental stress screening approach.

A DOE report from 2008 describes more complete process for supplier selection and COTS components into critical government systems [8]. One of the core technologies is the Enterprise Component Information System (eCIS) which tracks detailed component information, testing results, and failure analysis information. The eCIS also provides a framework to track the COTS component insertion process shown in Fig. 2.2 and helps ensure the rigorous engineering practices needed to reduce supply chain risks [8].

Figure 2.2: Typical supply chain component insertion process, highlighting the higher risks that occur with inconsistent supply chain surveillance and monitoring

While most manufacturers follow a similar process for supply chain management during initial product development, there usually is a significant lack in surveillance and monitoring activities of the incoming components (i.e. random audits or destructive testing). In terms of cybersecurity this means that counterfeit or subverted parts can easily be inserted into the supply chain after a manufacturer begins a full production run. Conclusively, surveillance and monitoring of the supply chain is crucial to ensure the security of critical infrastructure and energy delivery systems despite the significant costs. This re-emphasizes one goal of this research: to create cybersecurity solutions that are pragmatic and cost-effective.

2.4 Existing Practices for Functional Verification Testing

In a manufacturing environment there are several forms of verification testing that can occur as a device is being assembled with the primary goal to ensure the system is functional and meets the design specifications. Verification testing is usually performed in phases depending on the predetermined test plan and the specific industry. For example, medical devices which are regulated by the FDA are required to go through visual inspection, automated testing of the electronics and software interfaces, burn-in or other environmental testing (similar to that

specified by MIL-STD-810 [9]), and testing on the final assembly. The testing typically includes electrical continuity tests and digital interface testing at a minimum.

At each phase the data is typically recorded and maintained as specified by the manufacturer's records and retention schedule, although in some cases it might be a simple pass/fail rather than details of the measurement results. If a device passes all the testing, it is packaged and sent to the distributors or end-users. If a device fails any manufacturing test or inspection, it is typically routed to an internal "rework" team who review the test data, determine the possible cause, and manually replace a faulty component. These repaired devices are again sent through the verification testing process which hopefully results in successful results, although sometimes it results in one or two more rework cycles before the faulty device is destroyed. In the event of a fielded device failure, a manufacturer will usually initiate a formal investigation that analyzes the verification test data, procurement information, and possibly perform additional tests to help determine a root cause. If the device was a lower-cost device not designed for a critical application, a manufacturer may forego failure analysis because of a cost-benefit determination.

While many of the generic testing process steps are listed in a number of quality management systems or standards like ISO 9001, it is up to the test engineer to design a test plan that fits the guidelines to a specific device and industry. Some industries, such as avionics flight control systems, require rigorous testing on software and hardware to ensure coverage as well as system interactions and timing. In contrast, consumer products may only go through a minimal set of final verification testing, primarily to ensure basic functionality, and will have strong reliance on the initial validation, reliability and qualification testing performed before the start of production. Again there is much dependence on decisions of the responsible test engineer(s).

In theory the test engineer of a robust system will test all specified interfaces and operational states of the device, taking into account the combinatorial possibilities of each input (e.g. a brute-force approach), but in practice that is too costly and time consuming. However, there has been much research in the testing domain examining the best approaches to identify critical parameters or combinations that are likely to have a large effect on the system. These include well-established statistical design of experiment techniques like response surface methods [10] and robust parameter design (i.e. Taguchi methodology) using orthogonal arrays of test parameters [11]. Yet, current engineering hiring practices do not support the use of such

testing methods, by hiring less experienced engineers (with only an undergraduate degree) to design and perform the system test while allocating the senior engineers to design the system or safety critical software. The challenge is that junior engineers may not have the academic background to fully understand the advanced statistical testing methods and typically lack the experience or integrity gained through years of mentorship to perform the rigorous testing required by critical applications in avionics, aerospace, telecommunications, and energy delivery systems that have an up-time of many years.

2.5 Just-in-Time Manufacturing

One of the most problematic areas in supply chain is the trichotomy between cost, schedule, and reliability. This tension is well enumerated by Shaw, Speyerer, and Sandborn's 2010 commissioned report to Defense Microelectronics Activity (the U.S government's authority on microelectronics obsolescence focused on DoD applications) which assigns cost and time metrics associated with resolving supply chain issues [12]. The comprehensive report indicates that the mean cost for a manufacturer to select an alternate source for a part is $41,000 and only takes an average of 11 weeks to resolve. In contrast the mean cost for a design change required by selecting another comparable COTS component is $1,118,000 and can take an average of 42 weeks to resolve. Simply stated, using alternate suppliers (including those who have not been thoroughly evaluated) significantly reduces cost, time, and ultimately lost revenue.

In order to help alleviate the pressures of cost, schedule, and reliability many manufacturers have adopted a just-in-time manufacturing (JIT) business model and use lean or six-sigma techniques to further increase quality and reduce manufacturing costs [13]. Taiichi Ōno first established the JIT concept in his post-World War II development of the Toyota production system: "Just-in-time means that, in a flow process, the right parts needed in assembly reach the assembly line at the time they are needed and only in the amount needed," [14][15]. And while terms like "kanban" still depict the original system, the JIT model has evolved to a business model with goals to approach a "stockless production" and reduce the amount of stored inventory [16]. For modern manufacturers this means they only maintain small or moderate sized lots from their suppliers, only enough to buffer the unpredictability in shipping or upstream supply chain.

To give a specific example, one manufacturer of critical infrastructure devices uses an off-site supplier to build sub-assembly circuit boards and maintains enough stock to buffer a two-week delay in parts. The primary supplier of the circuit boards then uses secondary suppliers or brokers of individual components such as transistors, ICs, capacitors, and power regulators and similarly maintain minimal parts inventory. Each supplier might have a two-week supply of already assembled parts ready to ship to their customer, the device manufacturer, again to accommodate a minimal interruption of the supply chain. The device manufacturer will perform a final assembly of all parts from their suppliers, and then perform function verification testing on the final product. The testing usually occurs immediately prior to placing the products in a temporary storage before shipping to the end user, although in some cases of a longer shelf time the manufacturer will test the devices again before shipping to make sure a current verification test is recorded.

2.6 Vulnerabilities in the Supply Chain

In terms of supply chain vulnerabilities for our nation's critical infrastructure, if a device manufacturer's normal supply chain becomes disrupted most will likely select an alternate source due to the lower cost and less down-time. By using an alternate supplier who is not part of the regular JIT value stream, there are significant risks of delay and cost if the normal supplier evaluation and vetting process were required. Consequently, a manufacturer might decide to use a less comprehensive supplier evaluation, despite the additional risks of component reliability or counterfeit part insertion. A mitigation for this issue, and one that is in common practice for FDA regulated medical devices, is to randomly or selectively use components or subsystems from multiple suppliers during a regular manufacturing run.

While the methodology to have supplier redundancy provides diversity and less dependency on a single point of failure in the supply chain, it still does not mitigate issues or subversive actions that occur further upstream. For example, a primary supplier who builds a subsystem circuit board might also make the same risky decision of using an unevaluated alternate supplier of components due to similar pressures of cost and time. In addition, if one of the primary suppliers is a parts broker offering similar redundancy in secondary suppliers, the supply chain risk increases; parts brokers, independent distributors, and Internet-exclusive sources were identified as the top three supply chain sources of counterfeit parts [5][2]. One

particular example of this danger is the 2010 federal prosecution of the integrated circuit broker, VisionTech, who was convicted in a conspiracy that sold hundreds of thousands of counterfeit ICs to the U.S. Navy, defense contractors, and others [17][18][19].

It is expected that a critical infrastructure device manufacturer evaluate the secondary suppliers at the beginning of a new device's design lifecycle, but reoccurring audits of the upstream supply chain are very costly and are likely not to have occurred with enough frequency to ensure the authenticity and reliability of the final assembly. The supply chain risk for a JIT manufacturing environment can be modeled as shown in Fig. 2.3, although it does not characterize all the security implications. One such example not shown is the vulnerability of a temporarily shelved device that has undergone verification testing, yet has not been shipped to the customer. Within the given lag time between final verification test and shipment, the hardware or software could be altered.

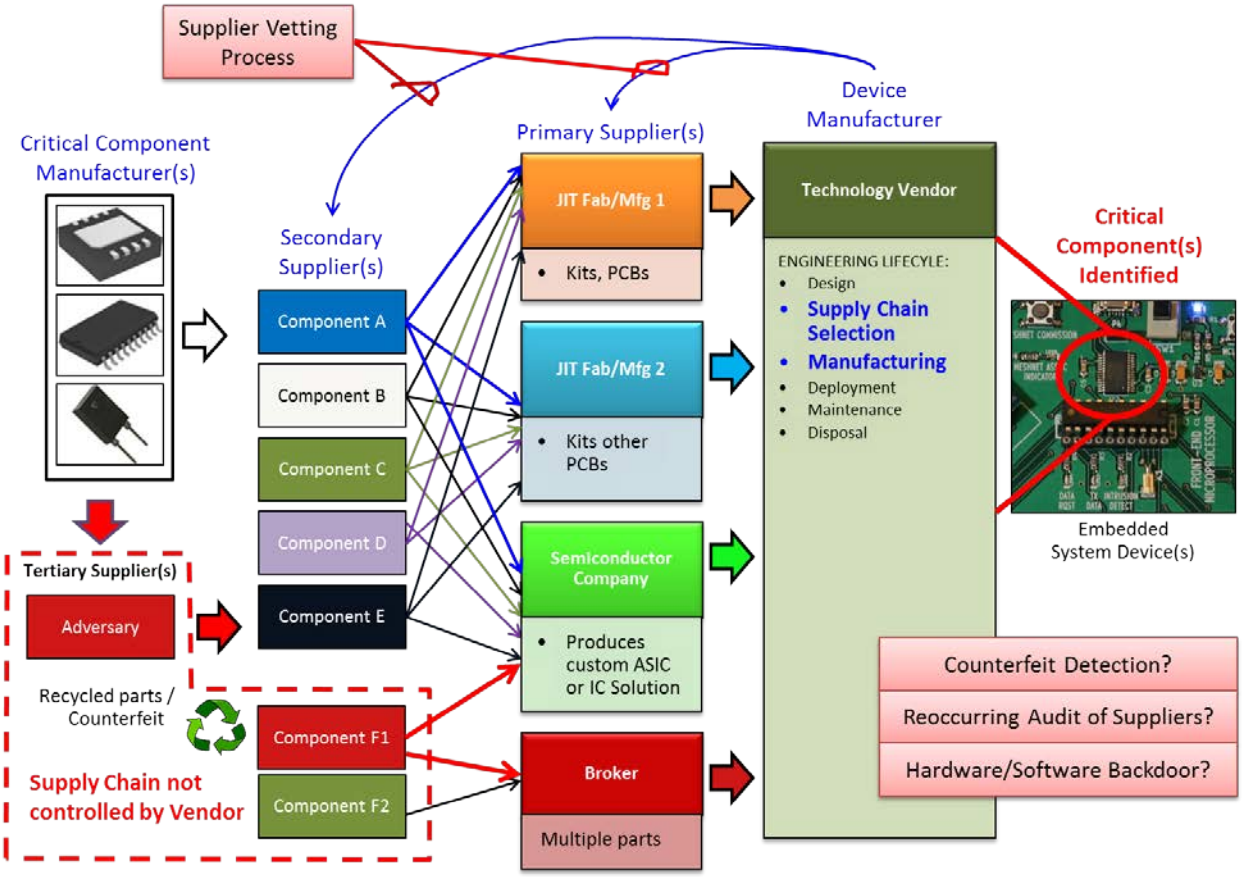


Figure 2.3: JIT supply chain risk model (does not show the vulnerabilities introduced by product returns or excess inventory buy back practices)

A similar vulnerability can occur between the receiving time and contracted installation of a new smart grid deployment, such as an AMI system, where the supplier or device manufacturer will contract with a regional service to warehouse devices and perform the installation so that regular infrastructure or utility staff are not overwhelmed with the deployment. Ultimately this means that due to cost and time constraints there are more opportunities for a sophisticated cyber adversary to insert a hardware or software backdoor into the supply chain without being detected. This is a significant problem for cybersecurity.

Another alarming vulnerability in the supply chain and perhaps the easiest to overlook are the risks introduced by returned parts or excess inventory buy-back programs. It is standard business practice for suppliers and parts distributors to accept returns from the device manufacturers or circuit board assemblers or to buy back their unused inventory. Part of this practice is encouraged for reasons of good customer service and long-term business relationships. Although there are usually strict guidelines on the returned product such as maintenance of the appropriate environmental controls (e.g. proper humidity or electrostatic discharge protection) and sometimes requirements for unopened packaging, it is still possible for "customers to purchase counterfeit parts from another source and, knowingly or unknowingly, return those parts to the original component manufacturer" [2]. Not only does this highlight an alternate channel for counterfeit parts to enter the supply chain, it also emphasizes the importance of proper testing and screening of returned parts before placing them back into a supplier's inventory.

CHAPTER 3 EMBEDDED SYSTEMS HARDWARE THREAT MODEL

3.1 Overview

While the discussion on cybersecurity issues with the supply chain highlight potential paths or vectors of inserting a hardware Trojan (used synonymously with hardware intruder), it now becomes necessary to describe specifics of the hardware intrusion problem which this research attempts to address. To summarize, this chapter will discuss previous work on taxonomies of hardware-based Trojans, fully describe this research's specific threat model to embedded systems, and discuss several approaches to detect hardware intruders.

3.2 Taxonomy of Hardware-Based Cyber Attacks

Although there has been much previous work on hardware Trojan taxonomy [20–23] with respect to integrated circuits, they must be reduced to a simplistic form in order to describe a hardware-based attack on an embedded system. In the reduced description a hardware-based intruder can be classified by three main categories summarized in Fig. 3.1: 1) physical characteristics, 2) trigger or activation mechanisms, and 3) action characteristics [21]. The physical characteristics are described by circuit size, distribution, layout structure, and type – parametric or functional. It is expected that the physical characteristics will also have some effect on the electrical behaviors of the system. A Trojan's activation mechanism or trigger is perhaps the most difficult to uncover during forensic analysis. The trigger can be through internal state-based monitoring or from external activation over covert radio communication. The final category describes the actions taken by the Trojan which could be to transmit data or modify the system's functionality.

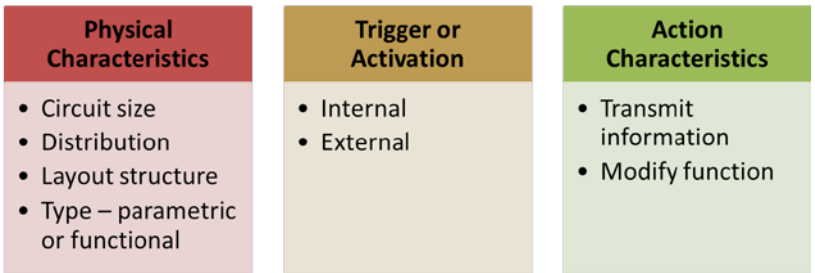


Figure 3.1: Simplified taxonomy of hardware Trojans

3.3 Threat Model for Embedded Systems Hardware

Despite similarities with much of the existing research on hardware Trojans, hardware-based cyber-attacks on embedded system devices are more achievable by a lesser capable adversary. For example an extra COTS IC with integrated RF communication can be added with minimal effort to an embedded system device by soldering a few wires. This inclusion can be performed at any time in the device lifecycle, especially during final assembly or post-deployment. The threats of such an attack, and the primary focus of this research, are centered on unprotected inter-chip communication, data that can be extracted from it, and the possibility of controlling a system through the communication bus (Fig. 3.2). Examples of this include acquiring power usage data, activating remote disconnects, using the mesh network to corrupt a utility's enterprise computing system, or even controlling customer-owned home area networked devices and smart appliances. These attacks might focus on a target facility or might propagate through the AMI network. Even worse, if the hardware Trojan was installed during manufacturing, then it might be possible to capture extremely sensitive information such as cryptographic variables or configuration data used during the enrollment of a cryptographic-IC, such as a trusted platform module (TPM).

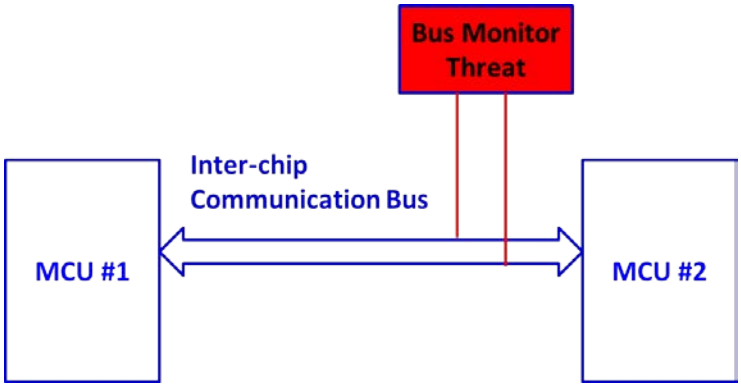


Figure 3.2: Inter-chip communication bus monitor threat

The action characteristics of an embedded system hardware Trojan primarily fit into two types of attacks: passive eavesdropping or active command or control of the communication bus. In terms of the previously discussed taxonomy, these would fall under transmit information (passive attack) and modify function (active attack). Both forms of attack can be implemented with COTS hardware with multiple activation mechanisms, giving it much flexibility and also highlighting the danger of such a threat. First, the Trojan device might capture information

based on active inter-chip communication using an edge trigger transistor configuration. The second activation would occur from an external trigger over a covert RF communication link which then transmits the captured data. The same RF communication link could be used to initiate a carefully designed active attack that propagates from the inter-chip communication bus to the front-end microprocessor and out through the AMI network.

Hardware-based cyber attacks can be modeled in three domains: digital, analog and temporal (Fig. 3.3). Each can provide characteristic information about the attack mechanisms and associated hardware, but none can fully represent an attack. The digital domain is perhaps the easiest to model since the information will be in a discrete binary or state-based form. The problem of modeling hardware attacks solely within the digital domain is that certain attacks might be implemented using noisy analog signals or electromagnetic waveforms. The analog domain characterizes an attack through analysis of signals, electrical characteristics, power and heat dissipation, and most continuous values that can be measured on an embedded system. The temporal domain looks at the timing characteristics such as latencies or clock signals in both the digital and analog domains and the associated timing attacks. This research primarily focuses on the analog domain.

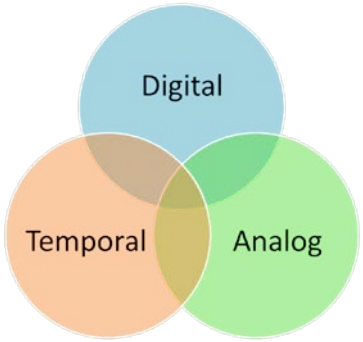


Figure 3.3: Hardware attack modeling domains

From an engineering and implementation perspective, the type of attack and the physical insertion point will determine the type of hardware circuit used by the adversary and consequently the attack model. The easiest example is a hardware Trojan implemented using a COTS microcontroller that has I/O and analog sampling capabilities. An active attack as previously described will use the I/O capabilities to sink or source the inter-chip communication bus to initiate the attack. To drive an I/O pin, a microcontroller's internal hardware typically

uses a CMOS inverter as shown in Fig. 3.4 (a) which consists of paired PMOS and NMOS transistors along with a capacitor to maintain the pin's output voltage level.

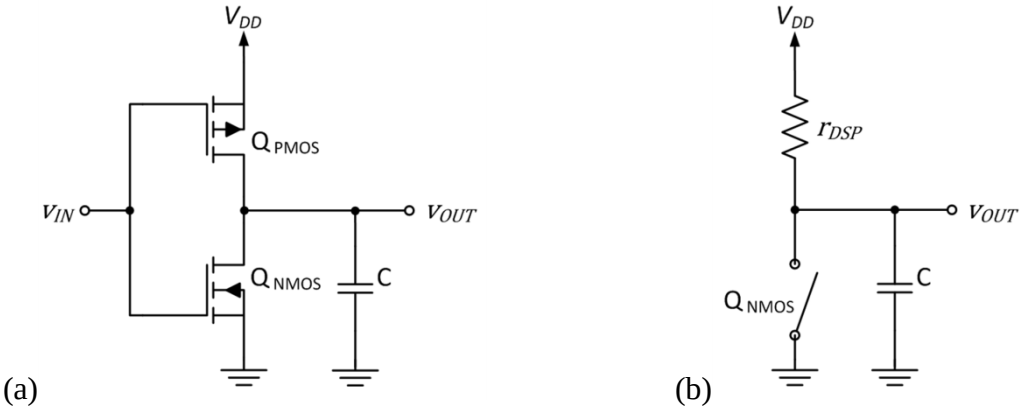


Figure 3.4: CMOS inverter (a) and equivalent circuit (b)

In the context of observing the electrical characteristics of such an intruder it is more practical to analyze the equivalent circuit model shown in Fig. 3.4 (b) and the dynamic power dissipation. The total energy dissipation (dynamic power) of the inverter has dependencies of transistor switching rate, output capacitance and the drain voltage [24]:

$$\omega_Q = fCV_{DD}^2, \quad f = \textit{transistor switching rate} \tag{3.1}$$

So when considering hardware detection techniques, it is expected that a CMOS inverter Trojan might have observable characteristics similar (not identical) to standard resistor-capacitor (RC) circuit.

A hardware Trojan circuit might also use a unity-gain voltage-follower configuration operational amplifier (op amp) to reduce its electrical effects on the circuit under measurement. The voltage-follower configuration shown in Fig. 3.5 is also known as an output buffer and is very common in sensor designs such as medical devices, audio applications, or other signal processing applications. The high input impedance of the op amp reduces the amount of current drawn from the circuit, and combines with a current gain stage so the output can drive a system where the output has a large capacitance or resistive load. In other words, the Trojan will be able to sense voltage or logic values on a communication bus possibly without having a large electrical footprint. Fortunately there are several parasitic electrical characteristics of commercial op amps that can affect the communication bus, and can possibly be used to identify its presence. Although there are many transistor-level implementations of an op amp, including MOS-based [25] and monolithic bipolar junction transistor (BJT) op amps [26], each input node

has transistor junction characteristics such as the input resistance and capacitance [27] that are externally measurable.

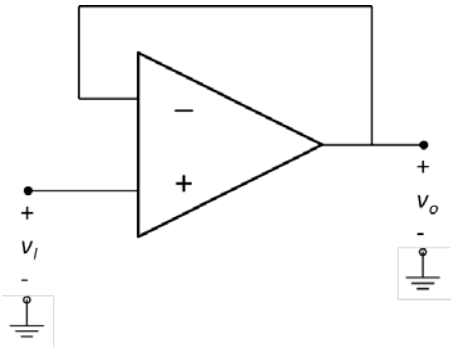


Figure 3.5: Op amp in a voltage-follower configuration

3.4 Approaches to Hardware Intrusion Detection and Prevention

In general there are several well-established principles to intrusion detection and prevention such as those discussed by NIST Special Publication 800-94 Guide to Intrusion Detection and Prevention Systems (IDPS) [28]. The primary goal is for the system to automatically respond to a detected threat by attempting to prevent it from succeeding. The detection methodologies typically fit into or a combination of three main classes: signature-based, anomaly-based, and stateful protocol analysis. The cyber response will vary based primarily on the mission of the monitored system and reliability aspects of the architecture. For example, blocking all enterprise network traffic from a general purpose desktop computer with suspected malicious activity may be of no consequence. In contrast, blocking all traffic from an industrial controller that regulates a high consequence process within a chemical plant could be disastrous.

NIST Special Publication 800-94 further defines the detection methodologies. Signature-based detection compare known threat signatures (usually static definitions) to observed events. Anomaly-based detection compare definitions or signatures of normal activity against observed events to identify significant deviations. Stateful protocol analysis compares deviations of observed events to predetermined profiles of benign protocol activity for each protocol state. In addition, a promising newer approach called specification-based detection consisting of identifying deviations from a correct behavior profile predefined using logical specifications has been successfully applied to an AMI emulation environment or proposed for Home Area Networks [29–31]. The distinction between stateful protocol analysis and specification-based

detection is that one focuses strictly on communication protocol, while the later works by building a state machine of a process, and then monitoring activity to check whether anything escapes from the system boundaries previously specified.

But regardless of the detection methodology, the overall challenge is to apply the IDPS principles to a complex hardware threat model that captures the supply chain insertion, analog signal characteristics, digital logic, and timing characteristics. A recently presented approach proposes the use of ring oscillator circuits within an IC to capture timing characteristics of a probing attack on an input pin [32]. While this approach focuses on attacks to an IC, it is very similar to the hardware threat model on an inter-chip communication bus. Yet the detection system requires more internal hardware on a custom ASIC and only looks at the binary result of comparing phase delays (delay or no delay) rather than additional information like wave shape perturbations, peak voltages, etc.

More promising techniques for hardware Trojan detection include a signature-based technique that uses noise modeling to construct fingerprints for an IC family utilizing side-channel information such as power, temperature, and electromagnetic (EM) profiles [33]. Another technique that can help identify device cloning or counterfeit ICs uses passive monitors and exploits unintentional RF emissions to build statistical fingerprints and the associated classifier system [34][35]. A third research area is focused on using process variations and measurable properties of physical unclonable functions (PUF) which offer many techniques and applications such as authentication, integrity testing, counterfeit identification, and cryptographic functions without the vulnerabilities of preprogrammed signatures in non-volatile memory [36]. Due to complex manufacturing process variations and the extreme difficulty to replicate them, the physical function becomes unclonable. An optical beam, electrical signal, or other challenge is induced on the system which then produces an unpredictable physical response and yet is repeatable (with some noise compensation). Depending on the possible challenges, a PUF has the capability to produce a high-order number of signatures called challenge-response pairs which identify unique characteristics of a specific system.

To summarize, there are many approaches to intrusion detection and not one of them provides a complete solution. Ultimately the best solution will account for possible insertions into the supply chain and span the analog, digital, and temporal domains to provide a much more complete view of the system's security posture.

CHAPTER 4 HARDWARE-BASED INTRUSION DETECTION USING RESISTOR-CAPACITOR CIRCUITS: A NEW APPROACH

4.1 Description of Hardware

A novel approach for intrusion detection on an inter-chip communication bus is to use resistive-capacitive (RC) circuits which have been designed to give a dynamic response to a binary challenge (single-bit or multi-bit square wave). Statistical perturbations to the waveform indicate that an intruder is present. In particular the circuit is an appropriately sized two-stage resistor-capacitor filter circuit as described by [37]. The RC circuit is connected to the communication bus via a low-resistance analog switch which is controlled by one of the embedded system's microprocessor. Signal perturbations of the dynamic response waveforms are monitored at the positive nodes of each capacitor, V_1 and V_2, using a two-channel analog-to-digital converter (ADC) module that is part of the microprocessor's peripheral I/O capabilities. The intrusion detection hardware configuration is shown in Fig. 4.1

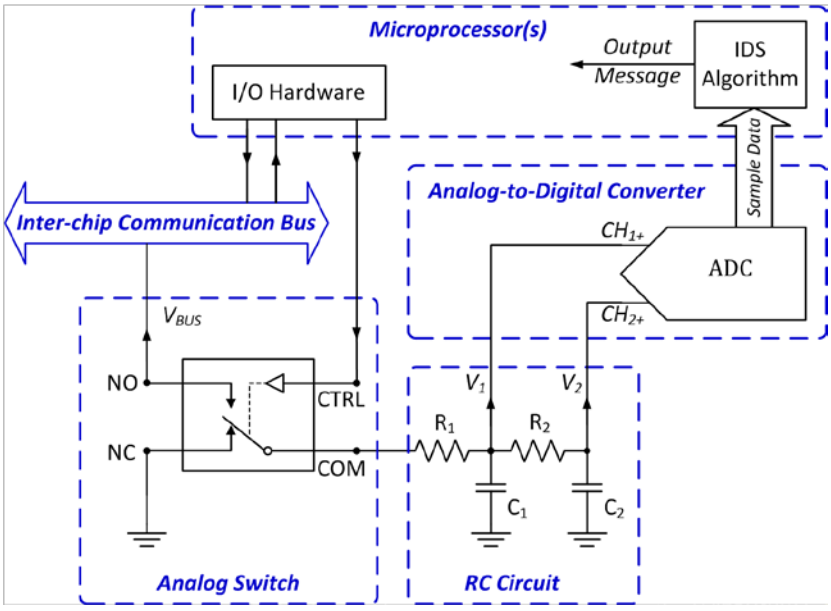


Figure 4.1: Intrusion detection system hardware

The single-pole double-throw (SPDT) analog switch has low charge injection and low total harmonic distortion (THD) which allows signals from the inter-chip communication bus to

propagate through the switch with minimal signal loss. The sampling rate of the ADC hardware must be fast enough to capture the metrics as discussed in upcoming sections without decreasing the performance of the intrusion detection model. The initial system was tested using low resolution (8-bit), so it is expected that any ADC with higher resolution (10-bit, 12-bit, etc.) will offer at least the same performance as long as the sampling rate meets the minimum requirements for the detection model. This research has not studied the effects of using different ADC technologies such as successive approximation register (SAR) or sigma-delta, although final system integration needs to select the ADC technology that performs best with the specific embedded system and intrusion detection model [38].

4.2 Concept of Operation

This intrusion detection system is centered on the concept of challenge-response authentication which is prevalent in modern trusted computing systems. To summarize the concept, a verifier will send a challenge to the device under test (DUT) and the system should respond correctly within an appropriate tolerance. Many of the computer security applications which use a challenge-response concept focus on digital responses or cryptographic functions. In contrast, this intrusion detection system induces a challenge using the physical properties of resistor-capacitor circuits along with principles of energy conservation, and measures electrical characteristics of the system's dynamic power response. Of course the current system being studied is the inter-chip communication bus, but the concept of this intrusion detection system can be applied to nearly all low-power electronic devices.

During normal operation, the analog switch will connect the RC circuit to the communication bus,[1] allowing the capacitors to charge at a rate determined by the time constant $\tau = RC$ = (equivalent resistance)*(equivalent capacitance). V_1 and V_2 are monitored during the connected state, and the response signals are then compared to the disconnected state where the capacitors are discharging with a natural exponential decay response. Due to well-understood physical properties of an RC circuit and by selecting a balanced design where $R1 \approx R2$ and $C1 \approx C2$, the charge and discharge cycles without an intruder have similar rates of decay (which can be shown with a simple linear transformation of the signal about the horizontal axis). It is expected that the per-cycle (charge-discharge cycle) comparisons along with a sliding window

[1] This research primarily uses the I^2C communication bus which is at V_{CC} during its idle state, but the concepts presented herein are applicable to other communication protocols and buses.

comparison of the previous *n*-cycles will indicate any significant signal perturbations caused by a hardware intruder.

4.3 Intrusion Detection Measurements

While both channels of the ADC are used to monitor and capture the voltage-time data of the capacitors in the RC circuit, there are a number of metrics both measured and derived to indicate the presence of an intruder. Some measurements are voltage or time data, others include first-derivative functions (interval slopes) or integral functions (areas under the curve). In addition, some of the metrics used are simple incremental counts. Table 4.1 describes examples of metrics used in this system to help build and refine an accurate IDS model.

A simpler approach to detecting intruders on a communication bus would be to sample the bus for unexpected logic level changes during idle periods or wait until the data packets are delivered to the application/protocol layer. Yet this approach has several disadvantages. The first of which is that detection using this approach has a strong dependency on a hardware Trojan to actively use the communication bus. If an intruder were passively "eavesdropping" and collecting system data as described in Chapter 3, then the logic-level monitoring approach would fail. So this simpler detection approach is only effective for active bus usage, which might only occur as a zero-day attack and could be too late. A second disadvantage is that the logic-level monitoring is scheduled, which, if it is periodic, an adversary could simply decide to initiate an active attack on the system during an unscheduled time period. Randomizing the logic-level monitoring would help address this problem, but still is susceptible to the first disadvantage mentioned. The intrusion detection system must use dynamic response characteristics to capture both the passive and active attacks.

4.4 System Integration

From systems perspective, the integration of this RC-circuit-based intrusion detection system is well suited for existing critical infrastructure embedded devices. In particular, this IDS technology can easily be engineered into an existing AMI smart meter's design. Most microprocessors that are used for embedded systems are COTS devices with many general purpose I/O functions, one of which is ADC peripheral hardware. In addition the footprint of the other components (resistors, capacitors, and analog switch) have minimal space requirements on

a printed circuit board or can be included as a basic block in an application-specific integrated circuit (ASIC).

Table 4.1: Description of metrics used for intrusion detection

Type of Measurement	Name	Units	Description
Discrete Components	"Cap"	farad	Nominal capacitance of C1 & C2 on the IDS module.
	"Res"	ohm	Nominal resistance of R1 & R2 on the IDS module.
Voltage Measurements	"V1pk"	V	Peak voltage of C1 during a measurement cycle.
	"V2pk"	V	Peak voltage of C2 during a measurement cycle.
Time Measurements	"V1pkToIDSoff"	second	Time difference between peak voltage of C1 and the transition edge from charge cycle to discharge cycle (as monitored by the analog switch's control signal).
	"V2pkToIDSoff"	second	Same as "V1pkToIDSoff" except using C2 peak voltage.
Interval Slopes (First-derivative)	"SDslopeV1_OnOff"	V/10µs	Standard deviation on the difference between interval slope *i* of the charge cycle and the transform of interval slope *i* of the discharge cycle. Interval slopes are acquired with a 300 kHz sampling rate.
	"SDslopeV2_OnOff"	V/10µs	Same as "SDslopeV1_OnOff" except using C2 voltage-time measurements.
	"SlopeV1qty"	integer	Count of the number of interval slopes less than a predetermined threshold (initial threshold set to 0)
	"SlopeV2qty"	integer	Same as "SlopeV1qty" except using C2 interval slopes.
Area Under Curves (Integral function)	"AreaV1on"	V·s	Area under the curve during capacitor C1 charge cycle
	"AreaV2on"	V·s	Area under the curve during capacitor C2 charge cycle
	"AreaV1V2on"	V·s	Difference between the area under curves V1 and V2 during the charge cycle.
	"AreaV1off"	V·s	Area under the curve during capacitor C1 discharge cycle
	"AreaV2off"	V·s	Area under the curve during capacitor C2 discharge cycle
	"AreaV1V2off"	V·s	Difference between the area under curves V1 and V2 during the discharge cycle.
	"AreaV1V2_OnOff"	V·s	Difference between "AreaV1V2on" and "AreaV2off"
	"AreaV1_OnOff"	V·s	Difference between "AreaV1on" and "AreaV1off"
	"AreaV2_OnOff"	V·s	Difference between "AreaV2on" and "AreaV2off"

The intrusion detection algorithm resides on one of the system's microprocessors and uses the voltage-time data from the two-channel ADC module to detect a hardware intruder. In some cases a reduced algorithm may be run to identify the presence of an intruder, then a more costly algorithm is run to characterize the intruder for attribution or to verify a false positive (if the specificity of the system is tuned to allow more false positives while reducing false negatives). Upon detection, an intruder flag is raised and informs the rest of the system, such as other on-board microprocessors or software services, through a simple message passing interface (serial, parallel, or internal interface). In the case of an AMI smart meter, the message of a hardware intruder would be propagated through the AMI mesh network radios using the ANSI C12.22 protocol and inform the power utility operators of a cyber event (Fig. 4.2). The message propagation would be similar to a smart meter's existing tamper detection or tilt-sensor message [39]. Upon detection of an intruder, the embedded device's software can be configured to execute under probation [40] such that only a minimal set of operations on the device are permitted like sampling and transmitting power usage data. In this reduced state, operations like firmware upgrades, encryption key updates, or device enrollment in the mesh network would not be allowed to execute.

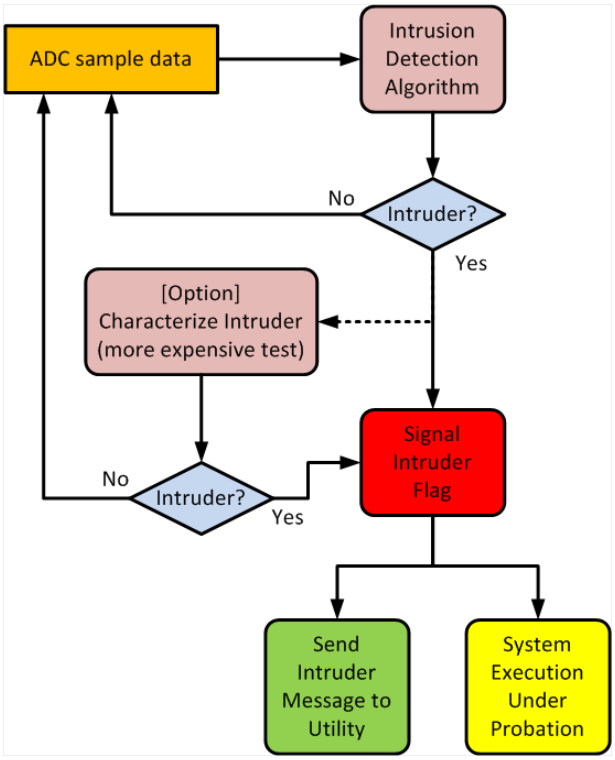


Figure 4.2: Intrusion detection data and control flow

4.5 Example Use Cases

One use case for this IDS technology is to build an external IDS circuit and create the accompanying test routine which becomes part of the functional verification testing at the device manufacturing facility. Although this scheme will not provide on-going integrity checks of the device, it can be useful to help reduce supply chain risk with minimal cost to a manufacturing system. The IDS technology would reside on the manufacturer's benchtest apparatus rather than as part of the embedded system device (Fig. 4.3), and thus eliminate a significant cost of engineering design change-orders for the actual device. Another major benefit to this modality of intrusion detection is that it can be combined with other off-line Trojan detection techniques like side-channel noise modeling [33], RF fingerprinting [34][35], and physical unclonable functions [36][41] to provide a full suite of risk reducing supply-chain security mitigations.

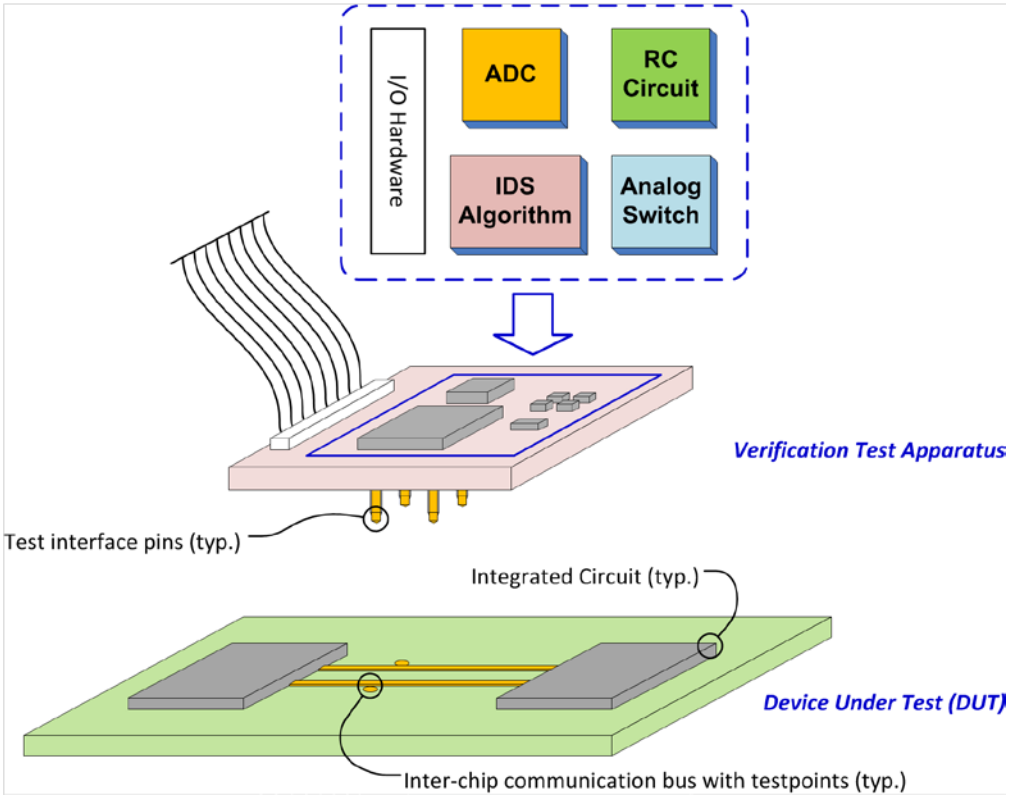


Figure 4.3: IDS system used for verification testing of existing and legacy devices

One particular challenge to this scenario is that a hardware Trojan trigger event is seldom known until after a cyber attack. So it then becomes critical to tune the sensitivity of this IDS system to detect slight changes to the dynamic electrical characteristics of an inter-chip

communication bus that might be observed during a passive eavesdropping attack. The test routine must also include electrical test patterns or software induced patterns to catch Trojan introduce anomalies to the analog signals of this IDS system.

Alternately, and the primary intended use, this IDS technology can be included it within the device's design. This option offers the benefit of intrusion detection during functional verification testing as well as on-going intrusion detection after the device has been deployed in the field. By using Bayesian techniques, the intrusion detection model can also be automatically tuned to the device's operating environment or will stay consistent with the degradation from device aging, both of which change the electrical characteristics of analog components. Although including the IDS technology in the device design has a slightly larger space requirement for physical layout, ultimately it helps address upstream supply chain risks and also provides a longer term solution to the hardware Trojan problem.

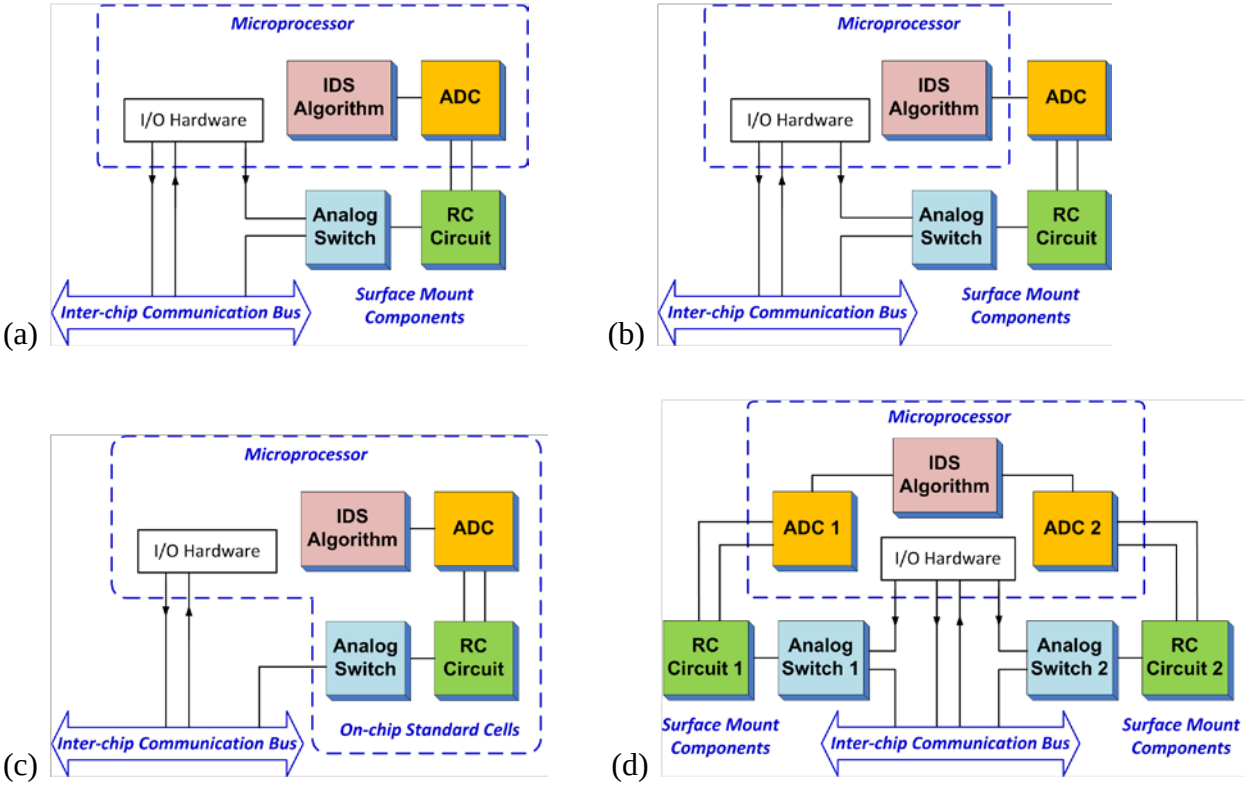


Figure 4.4: Several configurations of the IDS system built into embedded system device: Using COTS components with built-in ADC (a); using COTS components (b); IDS system internal to ASIC (c); dual mode configuration (d)

With the built-in IDS technology, there are four primary configurations as shown in Fig. 4.4. It is expected that configuration (a) will be the most common choice for device manufacturers since it uses COTS components, does not require the expense of designing and ASIC, and unused pins of a microprocessor in an existing design might be able to accommodate the new I/O requirements. One anticipated challenge is that the internal ADC hardware must meet the sampling rate requirements of the IDS model for a specific communication bus. If it does not, the microprocessor of an existing design must be replaced with a suitable one. An alternate is to use configuration (b) which has an external ADC module, giving the designer the flexibility to specify one that meets the IDS system requirements. A disadvantage of this configuration is that an adversary could easily subvert the IDS system by cutting the ADC data output lines and injecting false information.

Perhaps the most secure built-in IDS configuration is that shown in Fig. 4.4 (c). By packaging all IDS components within an application-specific IC, it is not as susceptible to subversive physical attacks without destroying the microprocessor. The downside of using an ASIC is increased cost for an end-device manufacturer to create or sub-contract the creation of a custom IC. Additionally, the physical placement of internal resistors and capacitors within the ASIC and which are appropriately sized for the IDS model has a large footprint. For example, adding a 100 kΩ resistor within an IC will use approximately 1250 μm^2 of space (based on 80 Ω resistance per μm^2). The ASIC design is still achievable within some cost-constraints by using a multi-project wafer (MPW) fabrication run. Of course the added security benefit of using an ASIC may far outweigh the higher cost when compared to surface-mount COTS components.

Given that a hardware Trojan might be one of several hardware configurations as discussed in Chapter 4, it can be beneficial to have more than one IDS circuit installed in the embedded system as shown in Fig. 4.4 (d). Since each of the hardware threat models has different equivalent circuits, the energy transfer characteristics are expected to be different and might respond better or worse to a variety of induced challenges. Consequently, it becomes necessary to install several orthogonal IDS circuits to provide more intrusion detection coverage to the hardware threat.

4.6 Theoretical Basis on Principles of Energy Conservation: KVL and KCL

The proposed solution relies primarily on the physics principle of conservation of energy as described by Kirchhoff's voltage law (KVL) and Kirchhoff's current law (KCL) which state that the sum of electrical potentials in a closed network is zero (KVL) and that the sum of all current flowing into or out of a node is zero (KCL). In terms of Kirchhoff's laws, the inter-chip communication bus is the closed network, and the point of attachment for a hardware Trojan and the IDS circuit is the node described by KCL. Normally an intruder actively using the bus, i.e. "active attack," will source energy (pull up the voltage potential to V_{DD}) or sink energy (pull down the voltage potential close to 0) resulting in a square wave which is then decoded by target devices as a binary 1 or 0. The energy or dynamic power of an intruder hardware based on the CMOS inverter is described by Eq. 3.1. With respect to energy conservation, a significant amount of the energy used by a hardware intruder to sink or source activity will be transferred to the RC components of the IDS circuit when they are connected to the bus. The power dissipation for a resistor in an RC circuit is described by Eq. 4.1 while that of the capacitor is described by Eq. 4.2:

$$\omega_R(t) = \frac{1}{2}CV_0^2(1 - e^{-2t/\tau}), \quad \tau = RC \tag{4.1}$$

$$\omega_C(t) = \frac{1}{2}CV_t^2 \tag{4.2}$$

It is easily observable that the dynamic power of the intruder, resistors and capacitors are all proportional to subsystem capacitance and the square of a voltage with respect to time. The relationship of subsystem energy transfer for two-stage low-pass filters is further described by early research on using passive components for signal filtering [42].

By using an appropriately sized two-stage resistor-capacitor filter circuit, we are able to slow down the voltage response caused by an intruder. Essentially the capacitor charging current is limited by the resistors. The intruder's square wave becomes a gradual curve with an exponential decay determined by time-constant τ. This exaggerates any signal perturbations and allows several electrical characteristics and intrusion detection metrics to be more easily measured.

4.7 Voltage Response of Two-Stage RC Circuit

In order to fully understand how the IDS circuit works, we first look at the physics of a basic resistor-capacitor circuit. The voltage response of a single-stage RC circuit is well understood to be proportional to the initial voltage V_0 and the exponential of time t over the time constant τ (which has dependencies on the resistance and capacitance):

$$v(t) = V_0 e^{-\frac{t}{\tau}}, \quad \tau = RC \tag{4.3}$$

The voltage measurement of the first capacitor C_1 then becomes:

$$v_1(t) = V_0 e^{-\frac{t}{R_1 C_1}}, \quad V_0 = v_{switch}(t) \tag{4.4}$$

where V_0 is equal to the initial voltage of the pin at the analog switch. Since this IDS system uses a two-stage RC filter circuit as shown in Fig. 4.1, the initial voltage of the second stage is that of the first stage based on principles from KVL and KCL:

$$\begin{aligned} v_2(t) &= V_0 e^{-\frac{t}{\tau}}, \quad \tau = R_2 C_2, \;\; V_0 = v_1(t) \\ &= \left(V_0 e^{-\frac{t}{\tau_1}} \right) e^{-\frac{t}{\tau_2}} \\ &= V_0 e^{(-\frac{t}{\tau_1} - \frac{t}{\tau_2})} \\ v_2(t) &= V_0 e^{(-\frac{t}{R_1 C_1} - \frac{t}{R_2 C_2})} \end{aligned} \tag{4.5}$$

Because this particular solution to hardware Trojan detection proposes to use a comparison of interval slopes and area under curves between the RC charging cycle and the discharging cycle, it is important to have similar rates of decay between each cycle. In this case we set $R1 \approx R2$ and $C1 \approx C2$, keeping in mind that manufacturing process variations will dictate the actual values of each component, and the resulting ideal voltage response for *V2* depends on two times *t* divided by the nominal values of R and C:

$$v_2(t) = V_0 e^{(-2\frac{t}{R_n C_n})} \tag{4.6}$$

During the discharge cycle, the voltage response of *V1* will follow the same form as Eq. 4.4, and the voltage response of *V2* will be similar to Eq. 4.2 (except with dependencies on *R2* and *C2*). If the two resistors and capacitors were not reasonably balanced, then the discharge voltage response of *V1* becomes similar to Eq. 4.3 and the charge and discharge cycles will be too different to apply a linear transformation of the signals to compare the area under curves or interval slopes. In addition, a balanced two-stage RC circuit also works well with different input

pulse shapes [37], thus showing its use for the intrusion detection circuit is not limited to specific intruder waveforms. Hence, a balanced RC design seems like the optimal configuration.

Due to the nature of a two-stage filter circuit, signal perturbations are filtered by the first capacitor which dampens the effect prior to the second capacitor. What this means for intrusion detection is that if a signal perturbation is detected while monitoring the second capacitor, the cause of the perturbation must have significant energy transfer capabilities such as a hardware Trojan attempting to use the communication bus rather than system noise (which is normally filtered out). In other words this IDS circuit helps reduce false positives.

Within a multistage RC network we begin to notice that there are dependencies of the charging and discharging characteristics on the neighboring stages. In the case of an embedded device such as an AMI smart meter, the subsystem under test along with the IDS analog switch will have a small signal equivalent circuit with resistive and capacitive elements. For example the analog switch used in this IDS circuit to connect the RC components to the inter-chip communication bus has an on-state resistance, charge injection, and a pin capacitance that depend on the switch's connection state [43]. Similarly, each embedded system microprocessor or IC connected to the communication bus will also have a resistive and capacitive characteristic that affects the IDS initial voltage V_0. In fact the pin capacitance is a known design limitation of buses and supporting hardware like that of I^2C protocol which is limited to 400 pF (an upper bound of approximately 20 ICs).

4.8 Environment, Aging Degradation, and Effect of Temperature on the IDS Circuit

One major concern of using passive components for the intrusion detection circuit is the environmental effect on the system, especially of temperature. AMI smart meters and other devices will be installed in environments that significantly vary in temperature. This includes locations with extreme heat such as Phoenix, Arizona, or locations with extreme cold such as the Province of Ontario, Canada. Of particular concern is the temperature effect on the dielectric characteristics of multilayer ceramic chip capacitors which are used for the IDS circuit. It is well established that inexpensive dielectrics such as Z5U and Z5V are highly susceptible to changes in temperature: X7R is moderately affected, and NP0 is not affected [44][45][46]. Some hardware designer blogs have jokingly stated that some ceramic chip capacitors could be used as

a thermometer. Consequently, the voltage responses of this IDS system are expected to vary with the changes in operating environment.

Accordingly, hardware-based cybersecurity solutions must accommodate the environmental effects on the circuitry. They must also account for the analog component degradation over its lifetime. The life expectancy in ceramic capacitors is affected by chemical changes related to temperature and with some dielectric chemistries the degradation rate doubles every 10 degrees C [44]. One approach to handle these transient and cumulative effects is to implement an algorithmic compensation on previous voltage response signatures when comparing them to the measured voltage responses, much like error correcting code techniques. In contrast, a better solution would be to use a Bayesian approach with previous signatures so that they evolve with the specific device degradation and its environment. This results in the added benefit of tying a device to its specific environment and therefore increasing the unclonable naturally occurring variation to the integrity measurement (i.e. intrusion detection).

A final side-effect of this IDS circuit is the added capacitance to the communication bus. In terms of intrusion prevention (IPS), the IDS circuit capacitance may physically limit an intruder's active use of the communication bus. As mentioned before, communication protocols like I^2C protocol are limited to 400 pF, otherwise the square wave edge transitions develop an oscillatory ringing that cannot be interpreted by the supporting transistor hardware. This means that if the IDS circuit's equivalent capacitance were large enough, then attempts by a Trojan to use or control other ICs through the communication bus, would be rendered nonfunctional by the oscillatory ringing.

CHAPTER 5 SMART METER RESEARCH PLATFORM

5.1 General Capabilities

In order to conduct realistic experiments on hardware intrusion detection, a real embedded system device must be used. Since the availability of AMI smart meters are tightly controlled by the manufacturers due to security concerns, an emulated environment had to be created for the empirical data collection of this research. Thus, the creation of the smart meter research platform which enables researchers to study embedded system questions within a realistic context. The platform provides the basic functionalities of the smart meter without energy usage sampling capability. This alleviates some of the dangers of working with the high energy of a real meter. The smart meter research platform also provides flexibility to interact with the device over USB or over radios of different technologies.

This system, shown in Fig. 5.1, makes use of four reprogrammable microprocessors which allow for software emulation of the basic internal functions of a smart meter: front-end microprocessor, metering engine, optical port IC, and data storage IC. The device also allows for custom software to be installed on the microprocessors, which can enable specific research experiments. Additionally there are several connection pins that allow for electrical characterization of the inter-chip communication and device power usage.

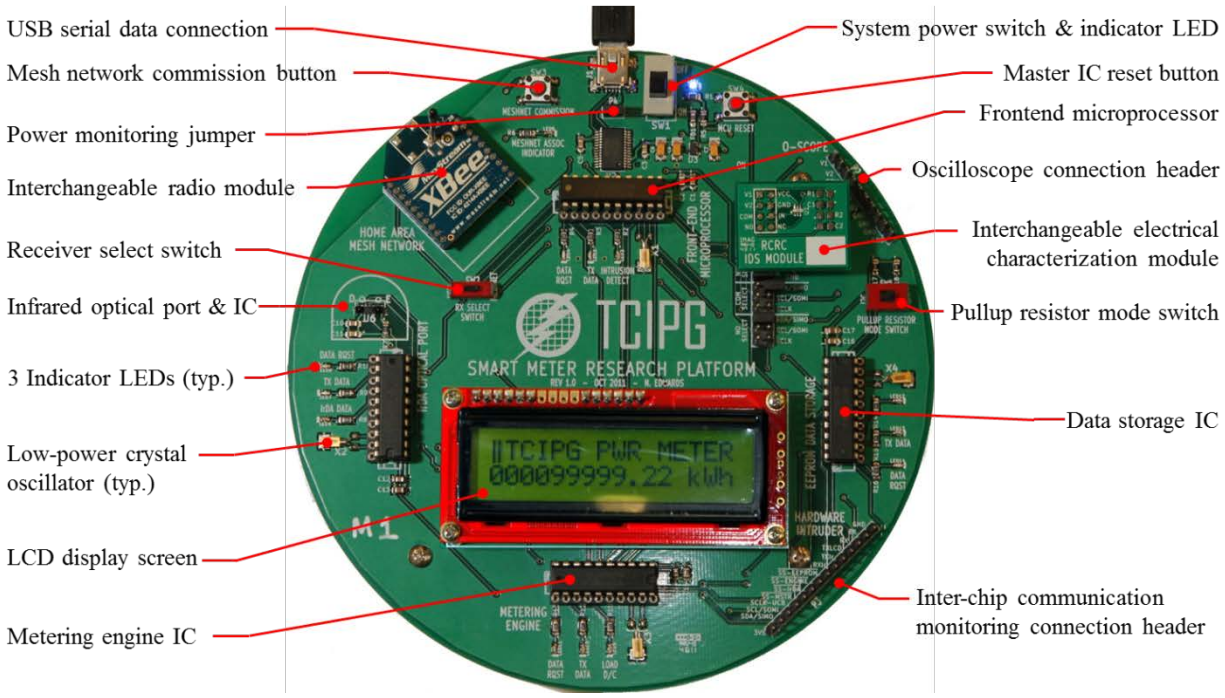


Figure 5.1: Smart meter research platform

5.2 External Communication Interfaces

The primary way to remotely interact with the research platform is through a USB connection or radio modules which serially transmit and receive data to the software/firmware residing on a front-end microprocessor. The front-end microprocessor transmits data simultaneously to the USB connection and the radio module to provide an alternate means of monitoring an experiment. In contrast, the front-end microprocessor can only receive data from one source at a time so that collisions are avoided on a shared hardware interface. The receive function (USB or radio) is selected by a mechanical switch.

The interchangeable radio module is connected with a standard 20-pin connection header used by several COTS manufacturers to provide flexibility for different radio technologies (the Digi brand is shown in Fig. 5.1). Examples of the COTS modules include WiFi (IEEE 802.11), Bluetooth (IEEE 802.15.1), ZigBee (IEEE 802.15.4), and WiMAX (IEEE 802.16) which are available in different transmission powers (i.e. Class 1 or Class 2 radios, etc.) and different frequency bands (i.e. 900 MHz, 2.4 GHz, etc.). Some radio modules even offer the ability to upload custom software for radio network configurations and routing algorithms. The digital interface of the radio module to the microprocessor allows for various baud rates and protocols (e.g. SPI, UART, etc.).

For the purpose of quickly configuring a radio network, this invention includes a network commission button which is used by some COTS radio module manufacturers to enroll a radio module and MAC address into a particular radio network without having to write customer software/firmware.

5.3 Microprocessor Hardware

All four microprocessors in this system have their own low-power crystal oscillator and three indicator LEDs that can be configured through custom software/firmware. The low-power crystal oscillator provides a stable clock source that can be used in conjunction with a microprocessor's internal oscillator. They are also connected to a master reset button which will simultaneously restart the software/firmware executions. The four microprocessors can communicate with each other through an inter-chip communication bus. The pin connections and wire traces support either I2C or SPI protocols depending on the current software/firmware configuration and the position of a pullup resistor switch which selects the idle state of the bus (i.e. V_{CC} or V_{SS}). Software/firmware on all microprocessors can easily be customized for experimentation.

The front-end microprocessor emulates that of a smart meter and has the software/digital interface for external communication as described by the ANSI C12 standards. The front-end microprocessor also runs the software drivers for both the radio communications module and the USB connection. It also typically serves as the master device on the inter-chip communication bus. The data storage IC is a microprocessor that emulates a smart meter's non-volatile memory. The software is typically configured to allow for reading and writing of data (which might be power usage data or other logged information as mentioned in the ANSI C12.19 standards). The infrared optical port IC uses an infrared (IrDA) transceiver connected to the protocol supporting hardware of a microprocessor. The software can emulate the ANSI C12.18 standards for optical port communication with a smart meter.

The metering engine IC is a microprocessor that emulates a smart meter's analog sampling of power usage. Although this microprocessor does not actually sample the power, it has the ability to sample low-voltage waveforms through its internal analog-to-digital hardware (with the appropriately configured software). This gives a researcher more realistic data when running experiments. This microprocessor also emulates a smart meter's remote disconnect

function, by acknowledging and serving the disconnect command message that propagates from the radio module over the inter-chip communication bus. The metering engine IC also runs the software drivers for the LCD display screen and communicates with it through a serial interface. Software/firmware can be easily customized for experimentation.

5.4 Experiment and Monitoring Interfaces

The smart meter research platform provides several interfaces to connect external measurement or test equipment. The first interface allows the external equipment to monitor all inter-chip communication, including data sent to the LCD display screen, the radio module, and the USB connection.

The research platform also provides a connection header for installing an interchangeable module that aids the electrical characterization of the inter-chip communication buses. This is the primary interface used for the IDS circuit described in Chapter 4, which was built onto a small circuit board module that can be easily connected or removed during experimentation. The connection header is wired to several jumpers that allow for selection of the communication protocol, source clocks, and two I/O ports of the front-end microprocessor. The module can be designed to temporarily shift the electrical characteristics so that the system response can be studied. Fault or error propagation studies might be conducted through this connection header.

In addition to the interchangeable electrical characterization module, a third connection header is provided for system response measurement by external equipment such an oscilloscope. A two-pin jumper on the main power supply is also installed so that in-line power analysis can be performed. This provides another research mechanism to study the electrical characteristics of a smart meter or embedded system.

5.5 Power Supply and Noise Suppression

The smart meter research platform is powered through the USB connection which is controlled through a single power switch with an associated LED indicator. Since low power microprocessors are used, the largest power requirement in the system is that of the radio module, in particularly when transmitting data. The second largest power requirement is that of the LCD screen. To isolate any noise in the system power, low-noise voltage regulators and bypass capacitors are utilized for the radio module and separately for the rest of the system power. This provides some flexibility for different COTS radio modules to be utilized, each with

a different power requirement, and still to provide the noise isolation while keeping the system current draw less than 500 mA supplied by the USB connection.

CHAPTER 6 DESIGN OF EXPERIMENTS

6.1 Considerations for Sampling Time

As mentioned in Chapter five, it is important to conduct realistic experiments with real devices so that manufacturing process variations (difficult to model and nearly impossible to replicate) are captured in the resulting data. The initial observations to help determine the preliminary relationship of possible measurements to that of an intruder were conducted on a 5 kHz clock speed I^2C communication bus. The slower bus speed allowed for easier configuration of test firmware and observations to be more easily distinguished without system noise becoming a large factor. From this the metrics discussed in Table 4.1 were determined as ones with higher potential to identify a hardware Trojan on the bus and are easy to measure. But ultimately the experiments needed to be conducted at a more realistic bus speed of 100 kHz, which is established as a reliable speed for I^2C communication.

With 100 kHz as the target bus speed, it is important to consider the sampling frequency of the IDS system. Since the I^2C protocol is based on eight bits before an acknowledgment is sent, the sampling window or charge cycle of the IDS circuit should be long enough to observe at least two bits on the bus so that the measurements can be collected in a multifactor sense while reducing false negatives. For this initial research, a window of 40 μs is selected which gives the possibility of measuring electrical characteristics of up to four bits at 100 kHz depending on the binary-stream on the communication bus. The front-end microprocessor uses an internal timer to trigger and drive and I/O pin for the activation of the sampling widow.

In addition the Nyquist sampling frequency must account for not only the bit-time, but also the other metrics such as interval slopes or any time-offset measurements so that the sampling occurs at least twice as fast as the desired time. Future work will also study the effects of the sampling frequencies on the accuracy of the IDS system, primarily because there are engineering cost factors involved: faster sampling rates typically increase the cost of the supporting ADC hardware.

6.2 Considerations for Areas Under the Curve

It is important to size the resistors and capacitors such that the RC time constant for the IDS circuit encompasses enough time for a potential intruder to perturb the measured voltage

response signals. Yet the time constant should not be so large that too much energy is stored in the capacitor, thus making a perturbation unnoticeable due to the filtering characteristics of the IDS circuit. The other concern is that the time constant should not too small such that any noise on the system affects the IDS measurements. In addition the resistor and capacitor values should be varied to understand the effects of them on the IDS sensitivity.

6.3 Design of Experiment

This experiment uses randomized design which explored the entire combinatorial space of all the hardware. The smart meter research platform described in Chapter five was used to capture the data on the I^2C communication bus between the system's four microprocessors (MSP430G2 microcontrollers). Three different hardware intruders were used as a COTS devices attached to the communication bus; they included an AVR328p, PIC24FJ, and another MSP430G2 microcontroller. Each intruder had custom firmware to bypass the I^2C hardware (which strictly adheres to the protocol) so that an alternating binary pattern could be sent over the inter-chip communication bus for identification of an active attack on the system. It is questionable if a real intruder will use such a binary pattern, but this does give us more insight on whether such and IDS system design will work to identify an intrusion and to distinguish between different intruders. Since each intruder has different clock oscillator hardware, the actual bit rate varies slightly between them: AVR328p measured at 99.0 kHz, PIC24FJ at 100.0 kHz, and MSP430G2 at 99.6 kHz

Three different voltage response modes for each capacitor value were created by selecting different resistor values. The three modes as shown in Fig. 6.1 (high, medium, low) offer different characteristics for the areas under the curve, difference between *V1* and *V2*, peak voltages, and capacitor charging rate.

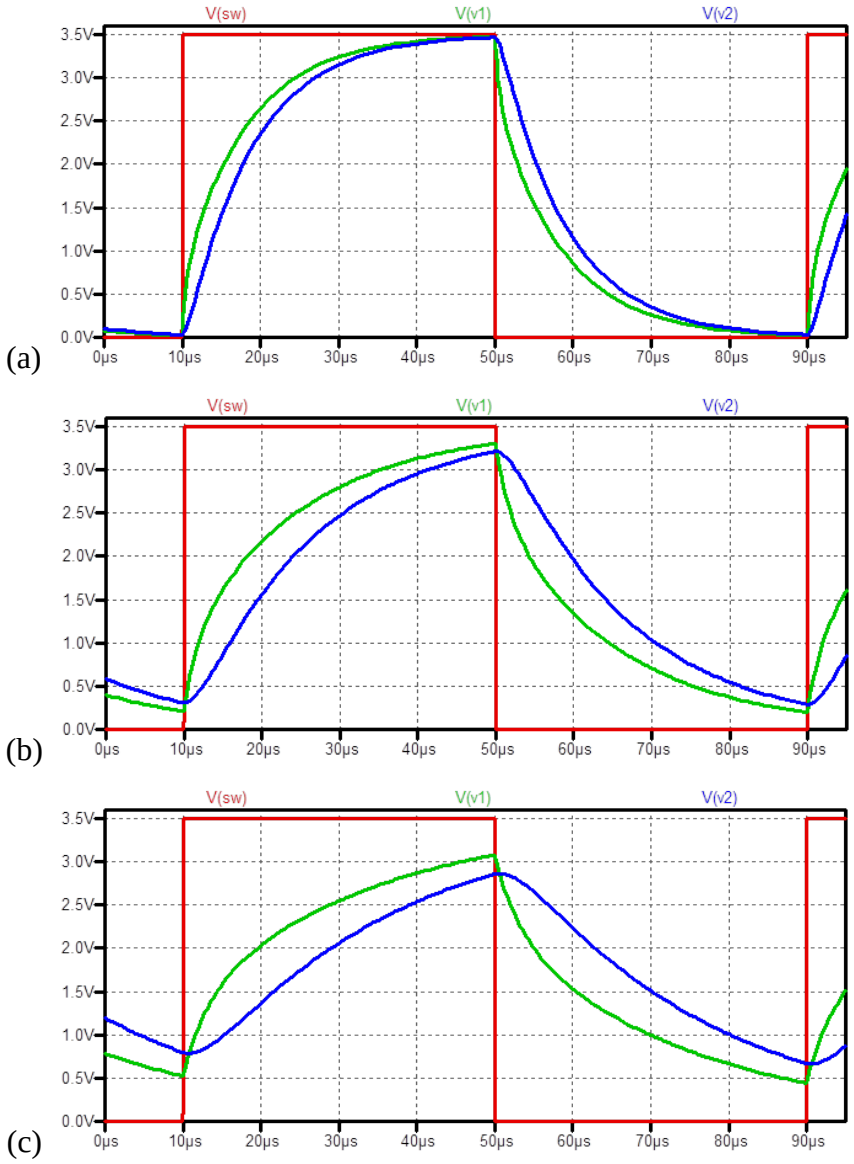


Figure 6.1: Three voltage response modes for each capacitor value

All RC components were in a 0805 surface mount component package. The random selection of components included those with different nominal tolerances which are determined by manufacturing process variation. For example one capacitor might have a 10% tolerance in its capacitance while a resistor might have a 5% tolerance. Table 6.1 shows the nominal values for the resistor-capacitor combinations. Three manufacturers were randomly selected for each ceramic chip capacitor value and for each resistor value; in total there were 45 different IDS modules. Detailed information on the random selection and placement of the components is located in Appendix C.

Table 6.1: Three resistor values for each capacitor value

Capacitor Nominal Value				
10 pF	**20 pF**	**39 pF**	**100 pF**	**200 pF**
84.5 kΩ	64.9 kΩ	49.9 kΩ	21.5 kΩ	12.4 kΩ
165 kΩ	143 kΩ	97.6 kΩ	49.9 kΩ	24.9 kΩ
249 kΩ	210 kΩ	165 kΩ	84.5 kΩ	45.3 kΩ
Resistor Values				

Randomization in the experiment was provided by 1) randomly selected manufacturers and RC components, 2) randomly selecting the IDS module for each hardware configuration, 3) randomly attaching the intruder microprocessor to the three different smart meter platforms, and 4) randomly selecting the system's core microprocessors from 45 preprogrammed MSP430G2 microcontrollers. During the experiment, the entire combinatorial space of experimental factors were explored without allowing repeats: zero to four system microprocessors, one to 45 IDS modules, and zero to three COTS intruder microprocessors, for a total of 675 combinations. The three smart meter research platforms were treated as blocking factors and simply rotated through the designed experiment to provide an equal amount of use for the other hardware combinations.

To ensure independence of measurements, the smart meter research platform was powered off to change the test configuration, then on before any measurement was taken. Immediately prior to activating the intruder hardware, two sequential measurements were taken of the voltage response without an intruder so that a sufficient number of measurements were taken to fully characterize the system noise. A total of 2036 experimental runs were completed, with 686 runs for a non-intruder state and 450 runs for each intruder in the random order previously described.

6.4 Experimental Test Setup

The smart meter research platform was powered over a USB connection to a computer. A 4-channel Hantek DSO3064A oscilloscope was used to capture data from the *V1*, *V2*, and *COM* nodes of the IDS circuit and the I/O wire trace that controls the analog switch. The DSO3064A has 8-bit voltage resolution and has a sampling bandwidth of 50 MHz (the Nyquist sampling rate is 100 MHz or 10 ns between each sample). The data is imported from the oscilloscope over its USB connection to a computer and then is saved as an ASCII text file with 10,245 samples per

channel times four channels. This resulted in a collection of 83,435,280 data points for the entire experiment. The oscilloscope capture trigger was set to the rising edge of the IDS charge cycle, although it captured additional voltage-time data prior to the trigger. The metrics listed in Table 4.1 were extracted from the raw voltage-time measurements using custom scripts in the statistical software R. Advanced data analysis was also performed in R.

The three intruder Trojans were configured with slight differences. The AVR328p was in a DIP package setup on a breadboard with a 3.5 V source. The PICF24FJ was configured on the USB powered Bus Pirate circuit board from Seed Studio, although the firmware was customized for the experiment. The MSP430G2 intruder was configured on one of Texas Instrument's LaunchPad development platforms and supplied power over a USB connection. Each intruder was attached to the SDA and SCL traces of the I^2C bus on the smart meter research platform.

CHAPTER 7 GRAPHICAL ANALYSIS

7.1 The Goals of Graphical Analysis

From graphical analysis we hope to answer the first research question, “Can we identify the presence of an intruder?” The second question is “How?” While graphical analysis or data visualization may not provide exact numerical results, it is highly beneficial to draw inferences about the relationships between experimental factors and the measured values. These graphical relationships can also help decide how to perform the numerical analysis and to refine the intrusion detection model. It also provides a mechanism for keen empirical observation, especially the real-time data displayed on an oscilloscope. Ultimately through graphical analysis we can visualize the signatures of normal versus anomalous activity.

7.2 Oscilloscope Trace Observations

Although it has not been previously discussed how the IDS metrics in Table 4.1 were decided upon, observing oscilloscope variations while trying to answer another research question led to the discovery of this particular IDS experiment, associated metrics and technical solutions. Fig. 7.1 shows one of the original oscilloscope traces using the two-stage RC circuit with analog switch. The green trace is the I/O signal driving the analog switch. The pink trace is the COM node of the analog switch with is immediately before *R1* of the IDS circuit. The yellow and blue traces indicate the voltage response of *V1* and *V2*, respectively.

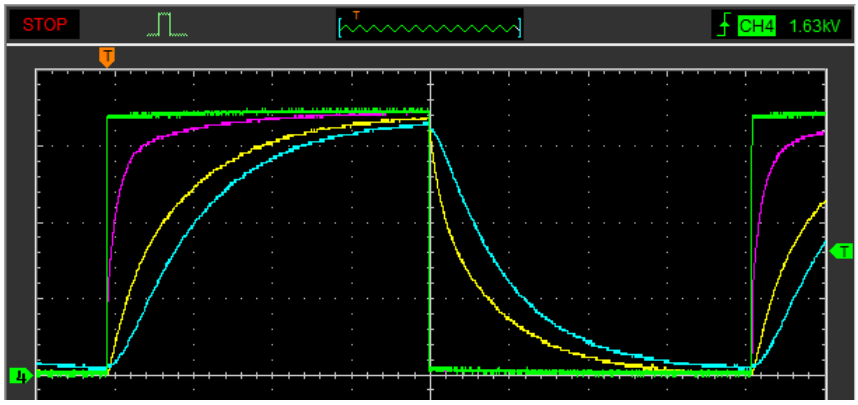


Figure 7.1: Oscilloscope trace of IDS circuit without an intruder

One observation is that the analog signal for COM is slightly rounded when compared to the switch control signal. This is primarily due to the IDS circuit and analog switch’s resistive-

capacitive characteristics. Another observation is that the charge and discharge cycles appear to have similar rates of decay and possibly are identical after a linear transformation is performed on the signal about the horizontal axis.

The oscilloscope trace of the IDS circuit with an intruder actively using the communication bus (Fig. 7.2) reveals dynamic response characteristics that help define which IDS metrics to use. We first notice that the COM signal is a direct representation of the intruder's activity, which happens to sink or source enough energy to overcome the capacitance of the analog switch noted in the non-intruder observation. In other words, the signal for COM is no longer slightly rounded on the rising edge transition. Other observations of the hardware Trojan's impact on the dynamic response of the RC circuit include decreased peak voltages, reduced areas under the curve, and a significant decrease in some of the interval slopes that follow a sinking activity. We also notice that the waveform shapes for *V1* and *V2* of the charge cycle (on the left) are much different than that of the discharge cycle (on the right). After observing several experimental runs, we also can observe that the intruder's activity is not synchronous to the IDS circuit, which indicates that it might be difficult to build an anomalous signature based on temporal elements.

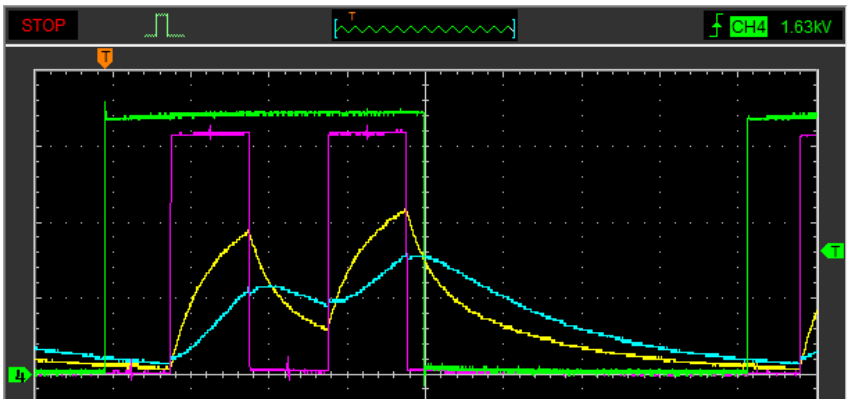


Figure 7.2: Oscilloscope trace of IDS circuit with an intruder's active attack

7.3 Graphical Analysis of Statistical Data

One of the most useful statistical plots is that of the probability density function (simply known as a density plot). This plot can overlay a histogram plot and helps identify distribution trends in the data. Based on the high number of experimental runs, not all of the several hundred graphics generated from the data can be shown or discussed in a thesis. For illustration purposes, the peak voltages of *V1* are selected. We first look at the density plot for *V1* peak voltages in

Fig. 7.3 that encompasses all the experimental data and is categorized by the four intruder types (one of which is non-intruder). The density plots reflect Gaussian approximation.

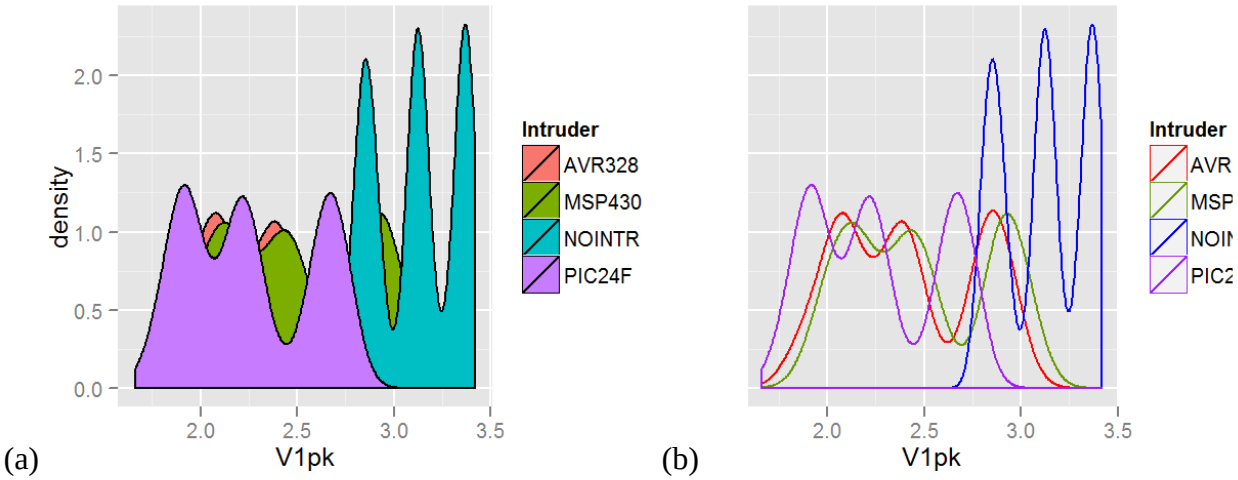


Figure 7.3: Density plots for *V1* peak voltages

The density signatures between intruders and the non-intruder subsets in Fig. 7.3 seem to be somewhat distinct. Three modes appear in the non-intruder data and are generally shifted to the higher peak voltages. If a threshold-based intrusion detection algorithm were used, it might include a narrow band on the peak density values, although this might not work well to analyze the first mode (on the left) of the non-intruder data since other intruders also have major density components at that voltage level. The signatures between intruders also appear slightly different than each other.

But if we look at the scatterplot of the same data for *V1* peak, a refined inference can be drawn. Fig. 7.4 shows the scatter plot of all the 2,036 experimental runs for *V1* peak, and like Fig. 7.3 it is also color coded by the intruder subset. It now appears that the three modes in the density plots are masking the 15 resistor-capacitor combinations. The three observed modes represent the distribution of resistor combinations per capacitor value as shown in Fig. 6.1 (high, medium, low) and are displayed as the blue horizontally aligned clusters in Fig. 7.4.

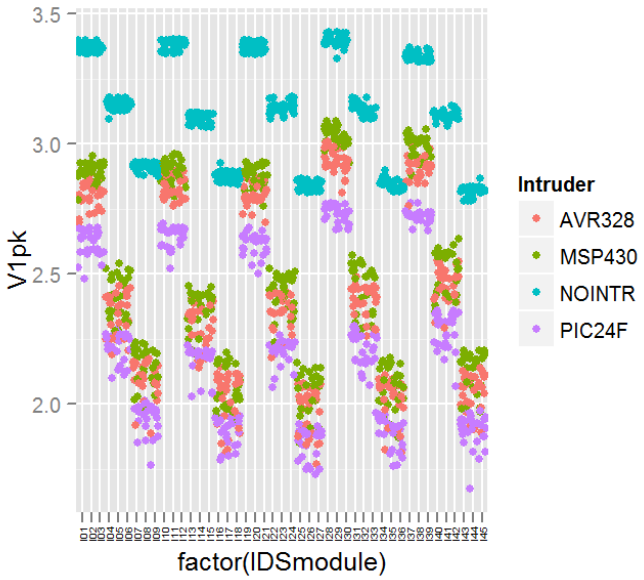


Figure 7.4: Scatterplot for *V1* peak voltages

The data subset for one particular resistor-capacitor mode reveals more information. Figure 7.5 shows isolation of the resistor-capacitor combination where R is nominally 49.9 kΩ and C is nominally 100 pF. The density plots of Fig. 7.6 also reflect this data subset and show that the Euclidean distances between each density signature are significant enough to easily identify the presence of a hardware intruder with a high probability of success. To summarize, the graphical analysis indicates that numerical analysis should look at the individual resistor-capacitor modes.

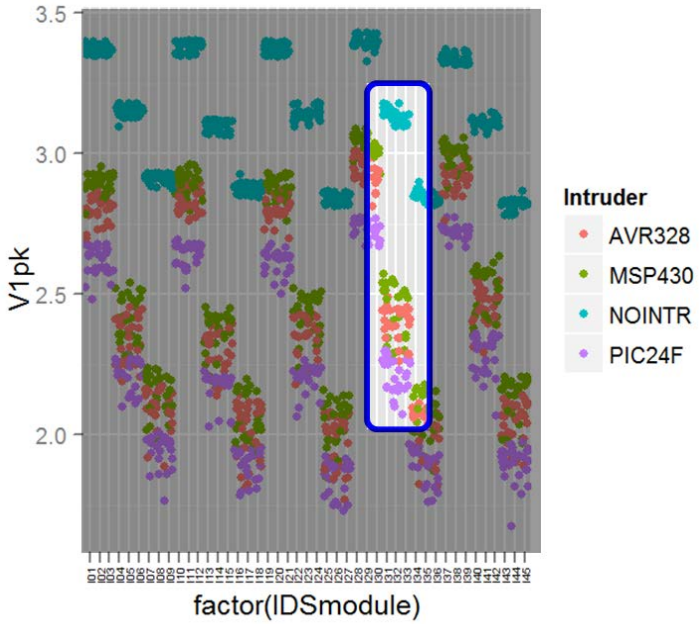


Figure 7.5: Scatterplot isolating one mode of *V1* peak voltages

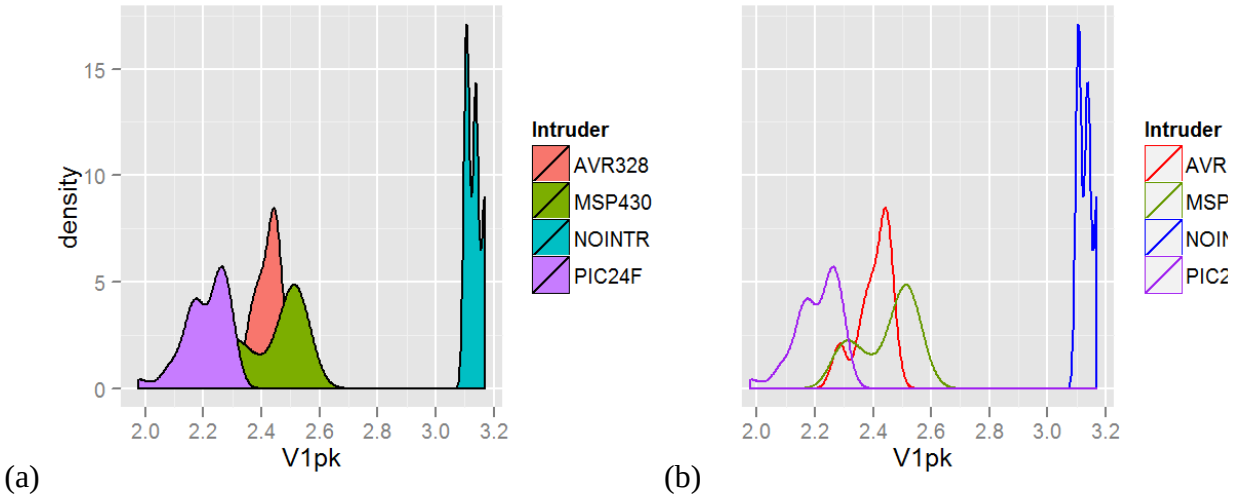


Figure 7.6: Density plots for *V1* peak voltages where R ≈ 49.9 kΩ and C ≈ 100 pF

So now the research question then becomes "can we distinguish between classes of intruders?" A preliminary observation is that the individual resistor-capacitor modes provide enough information for metrics like V1 peak voltage to answer this research question because the signatures of each intruder type appear unique. And while the technical details of how to answer this question are not as straightforward as using a threshold-based intrusion detection algorithm, the solution would provide some variety of cybersecurity attribution data and could possibly be extended into more advanced circuit-board-level hardware Trojan detection techniques.

Other metrics like *AreaV1V2_OnOff* and *SlopeV1qty* shown in Fig. 7.7 appear to strongly identify the presence of an intruder, but the density signatures of the intruder types appear nearly identical and therefore the metric cannot distinguish between types. Although other data

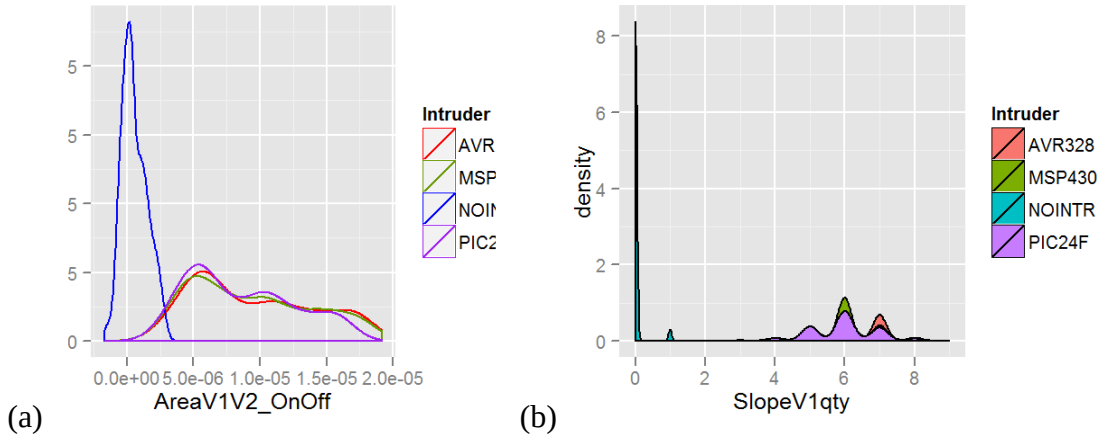
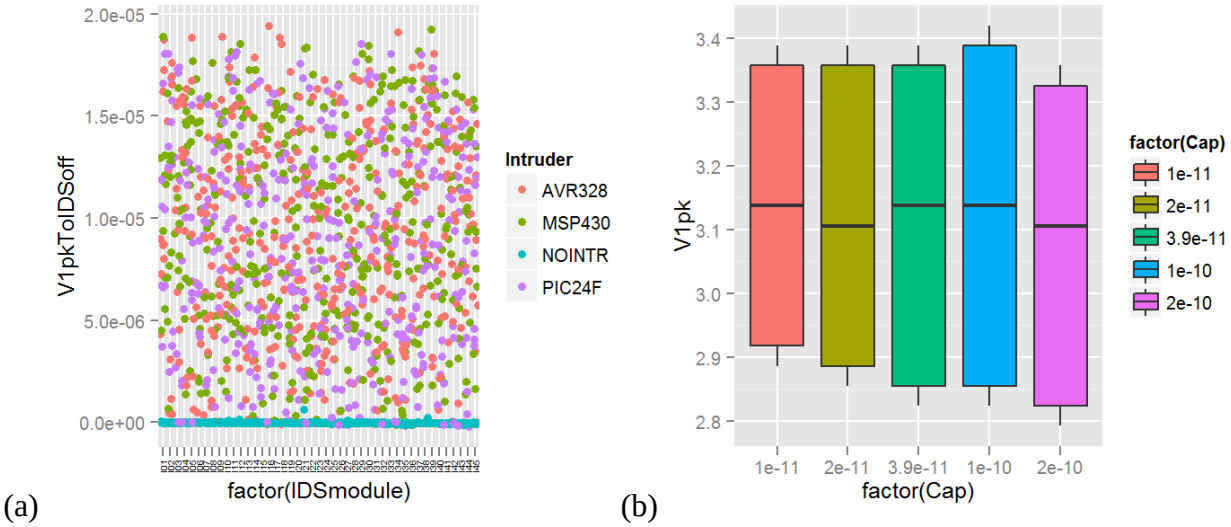


Figure 7.7: Density plots for *AreaV1V2_OnOff* (a) and *SlopeV1qty* (b)

visualization graphics like the scatterplot shown in Fig. 7.8 may indicate randomness if displayed in a time series sequence, they do not indicate any useful patterns or relationships. Ultimately from the graphical analysis we start to draw inferences about the relationships between the dynamic responses of the IDS circuit and the effects of the intruder types. These inferences will be confirmed through numerical analysis.



(a) (b)

Figure 7.8: Scatterplot of *V1pkToIDSoff* (a) and boxplot of *V1pk* (b)

A more complete set of characteristic graphs and plots for the hardware intrusion detection dataset are located in Appendix A.

CHAPTER 8 CHARACTERIZATION OF THE SYSTEM NOISE

8.1 Importance of Noise Characterization

For threshold-based IDS algorithms, it is critical to understand the noise levels of the system. A threshold-based algorithm will compare a measured value to the expected value in order to determine if there is an anomaly or intruder. The expected values for analog signal measurements in no way can be a statically defined value; some variance must be allowed in the comparison. Characterizing the noise will help determine which threshold or statistical variance is appropriate to distinguish between an intruder and system noise.

8.2 Collecting Noise Data

As mentioned in Chapter 6, there were 686 test runs with no intruder attached to the system. From graphical analysis shown in Chapter 7, we know that it would be inappropriate to collect summary statistics for the whole dataset because of the potential masking of modes. So of these runs, the metrics listed in Table 4.1 were tabulated with the statistics of total sum, total count, mean, variance, standard deviation, and standard deviation as a percentage of the mean for each of the 15 different combinations of resistor-capacitor values. In order to summarize the noise for a particular metric, we use the average variance of fitted values $\hat{y}$, where p is the number of parameters fitted [47]:

$$\bar{V}(\hat{y}) = \frac{1}{n}\sum_{i=1}^{n} V(\hat{y}_i) = \frac{p\sigma^2}{n} \tag{8.1}$$

The average variance also happens to be the mean square error (MSE) from the well-defined statistical analysis of variance (ANOVA) methodology. The standard deviations (both forms) of each metric were also calculated in a similar fashion. The summarization of noise for each metric is shown in Table 8.1 along with the units of measurement. It is important to note that all statistics and metrics are in terms of fundamental units, with the exception of the standard deviation of slopes *SDslopeV1_OnOff* and *SDslopeV2_OnOff*. These measurements were scaled to use volts per 10 µs so that the areas could more easily be correlated with the oscilloscope display.

Table 8.1: System noise summarization

Metric	Var	SD	SD Percent of Mean	Units
V1pk	0.000261383	0.016167359	0.51%	volts
V2pk	0.00027902	0.016703899	0.57%	volts
V1pkToIDSoff	1.15334E-15	3.39609E-08	769.56%	seconds
V2pkToIDSoff	6.38239E-14	2.52634E-07	146.59%	seconds
AreaV1on	9.25758E-13	9.62163E-07	0.99%	volt*seconds
AreaV2on	9.15446E-13	9.5679E-07	1.16%	volt*seconds
AreaV1V2on	4.99075E-14	2.234E-07	1.48%	volt*seconds
AreaV1off	1.6446E-13	4.05537E-07	1.32%	volt*seconds
AreaV2off	1.45365E-13	3.81268E-07	0.79%	volt*seconds
AreaV1V2off	7.26714E-14	2.69576E-07	1.78%	volt*seconds
AreaV1V2_OnOff	1.8434E-13	4.29348E-07	372.60%	volt*seconds
SDslopeV1_OnOff	0.003744857	0.061195238	16.74%	volt/10microseconds
SDslopeV2_OnOff	0.008931744	0.094507905	43.25%	volt/10microseconds
SlopeV1qty	0.034937338	0.186915324	417.87%	integer
SlopeV2qty	0.107346374	0.327637566	279.48%	integer
AreaV1_OnOff	9.89273E-13	9.94622E-07	1.63%	volt*seconds
AreaV2_OnOff	9.53049E-13	9.76242E-07	7.34%	volt*seconds

8.3 Analysis of System Noise

In general, the system noise (i.e. variance) is relatively small with the average standard deviation of most metrics equaling less than two percent of the means for each resistor-capacitor mode. Some of the metrics have a large average variance such as the time difference between peak voltage and the transition edge from charge cycle to discharge cycle, *V1pkToIDSoff*. One important observation in the noise characterization is that most of the system noise is orders of magnitude less than the expected voltage responses. For example *V1pkToIDSoff* shows an average standard deviation of 769.56 percent of the means, yet the variance is in terms of femtoseconds. Our expected responses for a 100 kHz bus speed might be in the order of nanoseconds; so even with the large variance when compared to arithmetic mean, it is still an acceptable measurement for the intrusion detection system.

The analysis of the noise in metrics *SlopeV1qty* and *SlopeV2qty* can also be justified. Of 686 test runs, the average variance is one-tenth of an integer or less. Since these metrics count

how many times the interval slopes are less than the threshold of zero, a fraction of an integer does not really present useful information. It is probably more important to look at the actual count per resistor-capacitor mode to gain a better understanding of the effects of the IDS circuit sizing. Table 8.2 shows the overall results for *SlopeV2qty*. It can be observed that the smaller capacitance values typically have more interval slopes less than zero, which indicates that the lower capacitance has less signal filtering capability and might be more sensitive to perturbations caused by an intruder.

Table 8.2: Noise characterization for *SlopeV2qty*

Resistor-Capacitor Mode	Sum	N	Mean	Var	SD	SD Percent of Mean
Cap = 1e-11: Res = 84500	18	45	0.4	0.2454	0.4954	1.238
Cap = 1e-11: Res = 165000	16	46	0.3478	0.2318	0.4815	1.384
Cap = 1e-11: Res = 249000	12	45	0.2666	0.2	0.4472	1.677
Cap = 2e-11: Res = 64900	15	45	0.3333	0.2272	0.4767	1.430
Cap = 2e-11: Res = 143000	10	45	0.2222	0.1767	0.4204	1.891
Cap = 2e-11: Res = 210000	4	46	0.0869	0.0811	0.2848	3.276
Cap = 3.9e-11: Res = 49900	8	47	0.1702	0.1443	0.3798	2.231
Cap = 3.9e-11: Res = 97600	5	45	0.1111	0.1010	0.3178	2.860
Cap = 3.9e-11: Res = 165000	3	47	0.0638	0.0610	0.2470	3.871
Cap = 1e-10: Res = 21500	3	46	0.0652	0.0623	0.2496	3.827
Cap = 1e-10: Res = 49900	1	45	0.0222	0.0222	0.1490	6.708
Cap = 1e-10: Res = 84500	0	46	0	0	0	0
Cap = 2e-10: Res = 12400	0	46	0	0	0	0
Cap = 2e-10: Res = 24900	2	46	0.0434	0.0425	0.2061	4.742
Cap = 2e-10: Res = 45300	1	46	0.0217	0.0217	0.1474	6.782
SUMS	**98**	**686**	**2.1548**	**1.6177**	**4.3033**	**41.922**
AVE VARIANCE				**0.1078**	**0.3284**	**2.794**

CHAPTER 9 IDS MODEL DEVELOPMENT & STATISTICAL ANALYSIS USING LOGISTIC REGRESSION

9.1 The Selection of Analysis Methodology for Intrusion Detection

There are several ways to approach intrusion detection for an embedded system. The first consideration is the methodology of detection as it will define the numerical analysis algorithms. As previously mentioned, NIST has summarized three main classes of intrusion detection: signature-based, anomaly-based, and stateful protocol analysis. For an embedded system, the type of threat becomes a major factor in deciding the methodology. For example a statically defined signature-based detection may work reasonably well for known intrusions on the network or within a host software process on the common general purpose computer running Windows operating system, but may suffer from low accuracy in the AMI domain as not all cybersecurity threats are known or cataloged. Anomaly-based offers more flexibility for the embedded system when addressing unknown threats, but sometimes can suffer from oversimplified detection models or ones that are too complex for an embedded system.

Stateful protocol analysis or other state-based analysis techniques such as hidden Markov models are very useful if the device under test can be described by discrete states. The problem with state-based analysis in an embedded system is that it cannot capture a side channel attack like the eavesdropping passive attack. It might be possible to describe a side channel attack by several discrete characteristics which can then be caste into system states, but this becomes very complex. Similarly, specification-based detection as described in [29–31] also requires the system to be described by discrete logical specifications and likely will not detect a side channel attack. Given the challenges of supply chain and unknown or uncharacterized cyber attacks on a deployed embedded system, the best option for intrusion detection ultimately is to combine the three methodologies into one unified intrusion detection system with its core focused on anomaly detection.

A simple and perhaps naïve technique to apply the previously mentioned methodologies is to use a threshold-based algorithm that compares a measured value or multiple values to a predetermined threshold or “golden value”. The threshold for each measured value is usually

determined by the system noise ceiling or standard deviation from experimental or randomly sampled manufacturing data. The comparison can be performed by regarding anything greater than the threshold as anomalous, using the hamming distance between measured value and threshold value, or using the probability of a measurement indicating an intruder (usually a probability density function with a Gaussian fit). To compile the results of multiple measurements and give a single IDS indicator, one could simply perform a logical exclusive-OR (XOR) on the results of all measurement comparisons.

While a threshold-based technique as previously described might be simple to implement and will not suffer much processor performance loss in a low-power embedded system, it has several significant disadvantages. First, the described technique assumes that the presence of an intruder will always be characterized by positive outputs on all IDS measurement comparisons. So if one metric fails to detect a deviation from the threshold, then the system may falsely indicate the lack of an intruder's presence. To reduce this risk, the number of metrics used for detection must be a minimal set, but that also consequently means that the IDS system loses the ability to fully characterize an intruder with additional metrics. Another major disadvantage to a simple threshold-based technique is that the system noise levels will change with external RF interference, temperature or other environmental effects. In other words, the "golden value" or thresholds that were determined during manufacturing test may no longer be valid. If the thresholds are set using a single value from aging or environmental testing, they may have too much variance due to the climate conditions between AMI systems and thus allow an intruder to hide beneath the margins of the thresholds. A final disadvantage is that a fixed threshold approach has worse prediction performance than other detection methods [48].

Although the simple threshold-based IDS algorithm may be improved using Bayesian methods to compensate for a system's operating environment or decay from aging, it may be more important to look at the general intrusion detection problem. Detecting an intruder on a cyber-physical system can be classified into two main categories: intruder vs. no-intruder. Each group can be further classified into subcategories based on one or many characteristics. For example an intruder may be one of several types based on the action or trigger event. Likewise, the non-intruder category (i.e. normal) may be categorized by several expected states. Along with each category or subcategory, an intrusion detection system must define the measurements and the algorithm to classify or describe each. And by defining the categories and subcategories,

one begins to create an intruder taxonomy described by metrics that are measurable by the embedded system.

By looking at intrusion detection as a problem of categorical data analysis, one can begin to see that the detection algorithm can be implemented with a large variety of statistical or machine learning techniques which result in a rich characterization of the intruder without the insufficiencies of a simple threshold-based detection technique. Some options include using artificial neural networks (ANN), fuzzy model extraction, Kth nearest neighbor (KNN), support vector machine (SVM), principal component analysis (PCA), generalized linear models (GLM), or logistic regression modeling (LRM). Each technique has a training phase where experimental data with known classes is used to build and optimize the classification model (i.e. intrusion detection) followed by a prediction phase. The prediction classifies sample data with unknown classes against a fitted model. While many of these techniques can have high classification accuracy and are lightweight enough for an embedded system, we will primarily use logistic regression because of a minimal number of assumptions, no error accumulation term, and familiarity of regression modeling.

9.2 Overview of Multinomial Logistic Regression

Logistic regression takes a form similar to other statistical regression methods, where a number of covariates and their coefficients determine the expected value or outcome of the model. This includes well recognized types of regression including linear regression modeling and generalized linear modeling (shown in Eq. 9.1) where the β terms are the linear coefficients and x terms are the covariates. The final term of the generalized linear model, ε, represents the error accumulation.

$$E(y) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_{n-1} x_{n-1} + \epsilon, \qquad \beta_0 = \alpha \tag{9.1}$$

A logistic regression model (Eq. 9.2) is similar except that there is no error term and the coefficients do not represent a linear relationship to a continuous value outcome.

$$logit(\pi_i) = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + \beta_{n-1} x_{n-1}, \qquad \beta_0 = \alpha \tag{9.2}$$

Instead, the discrete outcome is a likelihood or odds ratio of the classification: the probability of the outcome being of a certain class divided by the probability of the outcome not being the class [49]. Simply stated, the logistic regression model will give a likelihood that the covariate measurements indicate a particular class. The x covariates are thus called predictors.

Another difference between logistic regression and the generalized linear model are the coefficients. A basic interpretation of the β coefficients is that the odds of a classification increase by e^{β} for a 1-unit increase in x_n [49], [50]. So rather than a linear coefficient, the logit coefficient indicates the rate of how much the predictor will affect the outcome with each unit step increase. Eq. 9.3 shows the likelihood ratio relationship between the logit, or log odds, and the coefficients where α is the intercept:

$$logit\left[\pi(x)\right] = log\left(\frac{\pi(x)}{1-\pi(x)}\right) = \alpha + \beta x, \qquad \alpha = \beta_0 \tag{9.3}$$

In addition, classifier systems that use a multinomial logit model are commonly known as a maximum entropy classifier referring to its basis on the log of odds.

Within logistic regression there are several primary types, although this research will only discuss two cases: binomial and multinomial (also called polytomous). In terms of intrusion detection, binomial logistic regression would classify the presence of an intruder vs. no intruder – a binary response. Multinomial logistic regression extends the binomial concepts to a classification problem that requires multiple discrete-choice modeling where each outcome class can also be represented with a binomial choice. So if the research goal is to classify multiple types of intruders (a discrete set), multinomial logistic regression should be used. Logistic regression can also be combined with Bayesian methods to also account for system changes over time from operating environment or component decay, and thus makes it a very powerful analysis technique.

Unlike other statistical methods which require testing assumptions such as normality, linearity, or independence of observational errors, multinomial logistic regression has primarily one assumption – the assumption of independence from irrelevant alternatives. Under this assumption Hausman and McFadden state that the assumption facilitates estimation and forecasting based on the implication that the model can be estimated from data on binomial choices [51]. They also state that a specific multinomial logistic regression model has a necessary and sufficient characterization such that the ratio of the probabilities of choosing any two alternatives is independent of the attributes or availability of a third alternative. So if this assumption is true then the removal of an irrelevant, third choice, will not affect the estimated coefficients on the remaining predictors and they should not be statistically different from the

model before the removal of the irrelevant choice. Hausman and McFadden subsequently proposed such a test to verify the assumption for a multinomial logistic regression model.

To summarize the benefits of using multinomial logistic regression, it has fewer assumptions (of normal distribution, etc), it does not accumulate an error or residual over time, and there are well established methods to combine it with a Bayesian approach so that its model can be effectively refined with subsequent data. Yet the training dataset for multinomial logistic regression must be sizeable, the outcomes must be from a discrete set, and the assumption of independence from irrelevant alternatives must be met.

9.3 Intrusion Detection Model Development and Goodness-of-Fit

The overall goal in building a regression model is to have a minimal set of predictors but still maintain high prediction accuracy. There are several ways to build a regression model and eliminate the noncontributing variables such as backward elimination, forward selection, stepwise regression, and others. By eliminating the unnecessary predictors we reduce the noise in the estimation. With each modification of the model, several statistics can be analyzed to check the fit of the model and make sure that the change did not have statistical significance. Ultimately the optimal intrusion detect model should balance accuracy with computational performance.

The best approach to measure a model's goodness-of-fit is somewhat contextual; each possible goodness-of-fit statistic works differently for various data sets. For logistic regression we have selected two primary statistics: likelihood ratio statistic (a.k.a. LR statistic) and the Akaike Information Criterion (AIC). The LR statistic (Eq. 9.4) is a simple comparison of logistic regression deviance from one model to the next, where L_M represents the likelihood of the current model and L_S represents the likelihood of the saturated model:

$$G^2(M_0|M_1) = -2(L_M - L_S) \tag{9.4}$$

A simple definition of logistic regression deviance is shown in Eq. 9.5 or the discrete computational form in Eq. 9.6:

$$Deviance = D = -2ln\left(\frac{likelihood\ of\ the\ fitted\ model}{likelihood\ of\ the\ saturated\ model}\right) \tag{9.5}$$

$$Deviance = D = -2\sum_{i=1}^{n}\left[y_i ln\left(\frac{\hat{\pi}_i}{y_i}\right) + (1-y_i)ln\left(\frac{1-\hat{\pi}_i}{1-y_i}\right)\right] \tag{9.6}$$

Essentially the LR statistic is computed by performing an analysis of variance (ANOVA) of the deviances between the two models of comparison. The second statistic to check goodness-of-fit for our intrusion detection model is the Akaike Information Criterion, which was first published in 1974 by Hirotugu Akaike as a "versatile procedure for statistical model identification which is free from the ambiguities inherent in the application of conventional hypothesis testing procedure" [52]. AIC isdefined by Eq. 9.7 where *k* is the number of independently adjusted parameters within the model, and *L* is the maximum likelihood for the estimated model:

$$AIC = 2k - 2ln\,(L) \tag{9.7}$$

As shown, the AIC is two times the parameters plus the model deviance. The preferred model (i.e. one with a better fit) is the one with a minimum AIC value. Although there has been some debate on the performance of using AIC or LR statistics to check model fit in different applications, they are widely accepted as general purpose goodness-of-fit statistics and therefore are used in this research.

Since there is not much prior work characterizing analog signals on an embedded system for the purpose of intrusion detection, a relatively good starting approach for this research is to use a backward elimination procedure to build and refine the model. Backward elimination begins with a full regression model with all the predictors and then uses stepwise elimination to remove predictors that do not significantly affect the model [53]. A saturated model with all the predictors and interaction terms can be used as the starting model for backward elimination and has the possibility to generate a more accurate model, but due to the combinatorial effects of using many predictors it can make the model fit analysis too computationally intensive for an embedded system. In addition near-saturated models can be so specific that they might not detect an intruder.

To check the statistical significance of a predictor and decide which one to remove in the next step we will use the p-value from an analysis of variance comparing the modified model to that of the full model. The predictor with the highest p-value will be removed, after which the new model will then be refit and analyzed in a similar fashion. In any of the stepwise model building procedures it is important to remove one predictor at a time as the interaction between terms is not always clear and so that the assumption of independence from irrelevant alternatives can be checked. The backward elimination process generally repeats until all predictors have p-

values that are less than the desired level of significance; this research uses a significance level of $\alpha_{crit} < 0.001$ or a critical region greater than 99.9% for the predictors.

In order to analyze the experimental data from this research and build the intrusion detection model with backward elimination, a linear transformation is first applied so that the large-valued measurements (a multiplier of 10^3 or kilo) do not mask the small-valued measurements (a multiplier of 10^{-12} or pico). A full model with all 19 predictors and no interaction terms is fitted and analyzed. Table 9.1 shows the model fit analysis data for one of the iterations of backward elimination. The AIC for this fitted model is 1025.812 while the LR statistic comparing the deviances is 0.34837 with a reasonable chi-squared probability of 95.06%. As additional predictors are removed, we expect the AIC to increase because the model becomes less accurate. As mentioned before, the ideal and pragmatic intrusion detection model will balance goodness-of-fit with the prediction accuracy. Table 9.1 also shows that the IDS metric of *SDslopeV1_OnOff* should be the next predictor eliminated as its p-value is 0.2010404, which is much higher than others and indicates that this predictor does not significantly affect the logistic regression model.

Table 9.1: Analysis of model fit using backward elimination

```
AIC: 1025.812
  Resid. df Resid. Dev   Test    Df  LR stat.  Pr(Chi)
1      6054   917.8122
2      6051   917.4638 1 vs 2     3 0.3483742 0.950688
Analysis of Deviance Table (Type II tests)

Response: Intruder
               LR Chisq Df Pr(>Chisq)
IDSmodule         80.10  3  < 2.2e-16 ***
Cap                9.88  3  0.0195772 *
Res              139.90  3  < 2.2e-16 ***
V1pk             578.27  3  < 2.2e-16 ***
V2pk             196.09  3  < 2.2e-16 ***
V1pkToIDSoff      25.65  3  1.129e-05 ***
V2pkToIDSoff       9.37  3  0.0247645 *
AreaV1on         132.26  3  < 2.2e-16 ***
AreaV2on         145.05  3  < 2.2e-16 ***
AreaV1off         24.56  3  1.911e-05 ***
AreaV2off         48.25  3  1.888e-10 ***
AreaV1V2_OnOff    51.51  3  3.815e-11 ***
SDslopeV1_OnOff    4.63  3  0.2010404
SDslopeV2_OnOff   56.38  3  3.490e-12 ***
SlopeV1qty        29.36  3  1.883e-06 ***
SlopeV2qty        16.96  3  0.0007219 ***
AreaV2_OnOff      30.66  3  1.002e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
IDS MODEL ADJUSTER - remove next predictor with highest Pr
[1] "SDslopeV1_OnOff"
```

9.4 Intrusion Detection Model Performance

As the predictors approach smaller p-values, it may become more important to look at the prediction performance rather than the level of significance. Since good experimental design uses random sampling to reduce the residual effects, it would also be appropriate to check the prediction performance with similar rigor for a dataset. In machine learning and categorical data analysis it is well accepted to use a process called *n*-fold cross validation where the dataset is divided into *n*-folds, then *n*-1 folds are used to train the prediction model and one fold is used for the classification. The fold used for the prediction is incremented along with the folds used for training until all folds have been used for a prediction using the fitted (i.e. trained) model. The results are then compiled into an error matrix (also known as a confusion matrix or contingency table) such that the true values are cross-tabulated with the predicted classifications. For the hardware intrusion dataset of 2,036 experimental runs, 40-fold cross validation is chosen to test the concept of using analog signals for intrusion detection without requiring a high performance computer for the analysis. Table 9.2 shows the error matrix for the *fit10* model using Venables and Ripley's cross-validation algorithm from their classification example on forensic glass [54].

Table 9.2: Error matrix for *fit10*

```
40-fold CROSS VALIDATION
CONFUSION MATRIX:
    predicted
true   1   2   3   4
   1 686   0   0   0
   2   0 330  97  23
   3   0  77 372   1
   4   0  23   0 427
SUM TOTAL of MATRIX = 2036
Class[1:4] = NOINTR AVR328 MSP430 PIC24F
```

The numerical data contained within the error matrix provides several useful statistics. Among the common ones are accuracy, true positive rate (recall), false positive rate (type I error) and precision (positive predictive value). The overall accuracy is the sum of the error matrix times its identity matrix divided by the sum total of the error matrix. This is simply stated as the number of correct predictions divided by the total number of predictions. So the accuracy of the

model *fit10* shown in Table 9.2 is calculated by $\frac{686+330+372+427}{2036} = 0.8915$ or 89.15% accurate. The other statistical measures are defined in Eq. 9.8 through 9.10:

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{9.8}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN} \tag{9.9}$$

$$Precision = PPV = \frac{TP}{TP + FP} \tag{9.10}$$

While the true positive rate, false positive rate, and precision of an error matrix provide useful information, they cannot be used alone to assess the accuracy of a predictive model. One such technique that results in a single statistic is Kappa analysis, which was proposed by Cohen in 1960 for use in social sciences but has become a de facto standard among researchers for general discrete multivariate classification of remote sensor data [55], [56]. It is easy to make the case that the hardware intrusion detection data acquired from analog signals is also a use-case of remote sensor data well suited for the Kappa analysis with the goal of distinguishing between intruder types. The resulting Kappa or $\hat{K}$ statistic provides a single metric that indicates prediction performance and can be used to determine if one error matrix is significantly different from another; a larger Kappa value is more desirable when comparing two fitted models.

The primary assumption of the $\hat{K}$ statistic is the assumption of a multinomial sampling model. Eq. 9.11 shows a basic representation of the maximum likelihood estimate of $\hat{K}$ and is based on the actual agreement between the remotely sensed classification and the reference data as indicated by the major diagonal, and the chance agreement indicated by error matrix row and column totals [56]:

$$\hat{K} = \frac{p_O - p_C}{1 - p_C} = \frac{actual\ agreement - chance\ agreement}{1 - chance\ agreement} \tag{9.11}$$

The computational form of the $\hat{K}$ statistic used to analyze the hardware intrusion detection data is shown in Eq. 9.12 where *n* is the total number of observations in the error matrix, *k* is the number of rows, n_{ii} is the major diagonal (identity), while n_{i+} and n_{+i} are marginal totals of row *i* and column *i* respectively [55], [56]:

$$\hat{K} = \frac{n\sum_{i=1}^{k} n_{ii} - \sum_{i=1}^{k} n_{i+}n_{+i}}{n^2 - \sum_{i=1}^{k} n_{i+}n_{+i}} \tag{9.12}$$

By combining backward elimination with 40-fold cross validation and looking at goodness-of-fit metrics, true positive rates, false positive rates, precision, and $\hat{K}$ statistics we can select an appropriate IDS model that balances accuracy with computational complexity. The algorithms to analyze the hardware intrusion detection data were implemented in the R statistical software (version 2.15) with a summary of the results presented in Table 9.3. Detailed outputs of several model characterizations are located in Appendix B.

Table 9.3: Goodness-of-fit and prediction performance of hardware IDS logit models

Model	Dev G^2	df	Pr(Chi)	AIC	Overall Accuracy	Mean TPR	Mean FPR	Mean PPV	$\widehat{K}$ Statistic
fitF	4650.7	54	0	1031.46	0.8939	0.8800	0.0441	0.8794	0.8566
fit1	-5E-04	0	1	1031.46	0.8934	0.8794	0.0443	0.8789	0.8560
fit2	-0.006	0	1	1031.47	0.8939	0.8800	0.0441	0.8794	0.8566
fit3	0.3484	3	0.9507	1025.81	0.8939	0.8800	0.0441	0.8796	0.8566
fit4	2.0493	6	0.9151	1021.51	0.8949	0.8811	0.0437	0.8806	0.8579
fit5	8.1532	9	0.5188	1021.62	0.8924	0.8783	0.0447	0.8778	0.8546
fit6	14.623	12	0.2627	1022.09	0.8924	0.8783	0.0447	0.8778	0.8546
fit7	27.062	15	0.0282	1028.53	0.8939	0.8800	0.0441	0.8790	0.8566
fit8	34.669	15	0.0027	1036.13	0.8919	0.8778	0.0449	0.8772	0.8540
fit9	46.847	18	0.0002	1042.31	0.8919	0.8778	0.0449	0.8768	0.8540
fit10	66.195	21	1E-06	1055.66	0.8915	0.8772	0.0451	0.8769	0.8533
fit11	89.806	24	2E-09	1073.27	0.8861	0.8711	0.0471	0.8702	0.8460
fit12	115.04	27	8E-13	1092.51	0.8875	0.8728	0.0465	0.8721	0.8480
fit13	186.61	30	0	1158.07	0.8767	0.8606	0.0506	0.8601	0.8334
fit14	262.45	33	0	1227.92	0.8654	0.8478	0.0550	0.8469	0.8181

Model *fitF* represents the full model with all predictors and no interaction terms, while model *fit14* contains 14 fewer predictors due to the backward elimination process. With regard to goodness-of-fit, one observation of Table 9.3 is that the deviance G^2 and degrees of freedom increase as we reduce the number of predictors. The exception is *fitF* which is compared to an empty model and appropriately has a high deviance. Another observation is that the p-values

listed in column *Pr(Chi)* indicate that the removal of predictors after *fit7* significantly affects the deviance and goodness-of-fit. This relationship is also captured by the AIC value which begins to have a larger step-increase after *fit7*. To summarize, the goodness-of-fit decreases after fit7 and the prediction performance metrics subsequently become the model selection criteria.

The overall accuracy can be used to roughly select the minimal-set model, but it is important to use a metric like the $\hat{K}$ statistic and also the characterizations provided by true positive rates, false positive rates, and precision. A natural boundary exists in the Table 9.3 data after *fit10* where the overall accuracy decreases by more than 0.5% and the $\hat{K}$ statistic decreases by 0.007 when compared to the full model. In addition, the mean true positive rates, false positive rates, and precision for *fit10* are not significantly different from that of the full model. Conclusively, *fit10* makes a reasonable selection for the IDS model with 89.15% accuracy, a $\hat{K}$ statistic of 0.8533, true positive rate of 87.72%, false positive rate of 4.51%, and precision of 0.8769.

The coefficients and predictors of each intruder class for *fit10* are shown in Table 9.4. Although the intruder class names are shown, they should be interpreted as $logit(\pi(x))$ which is described by Eq. 9.3 where *x* is the intruder class:

Table 9.4: Estimated parameters (coefficients) of logit models for *fit10*

Logit	Intercept	IDS module	Res	V1pk	V2pk	V1pkToIDSoff
$log(\pi_{AVR328}/\pi_{NOINTR})$	34.09	-0.128	-13.45	43.38	-82.42	0.97
$log(\pi_{MSP430}/\pi_{NOINTR})$	6.79	-0.031	-5.48	93.15	-87.66	3.14
$log(\pi_{PIC24F}/\pi_{NOINTR})$	89.73	-0.442	-28.34	-181.5	102.66	2.13

Logit	AreaV1on	AreaV2on	AreaV2off	AreaV1V2_OnOff	SDslopeV2_OnOff	SlopeV1qty
$log(\pi_{AVR328}/\pi_{NOINTR})$	-85.7	9.04	-11.98	2.9	27.14	21.79
$log(\pi_{MSP430}/\pi_{NOINTR})$	-190.5	142.17	18.21	-16.14	7.01	23.89
$log(\pi_{PIC24F}/\pi_{NOINTR})$	144.34	-265.73	-61.75	34.52	38.27	10.61

9.5 Receiver Operating Characteristic Curves

Another important technique to analyze a classification model uses receiver operating characteristic curves (ROC). This curve has its origins in radar signal detection during World

War II and plots the probability of detecting true signals (true positive rate or sensitivity) versus false signals (false positive rate or one minus the specificity). Although there is some debate in statistics research on model performance characteristics that can be derived from a ROC curve, the area under the curve (AUC) is universally accepted as a measure of the model's ability to discriminate between outcomes [50]. The AUC ranges from zero to one, with 0.5 representing the probability of chance or maximum entropy [57] and is represented by the diagonal where sensitivity equals 1-specificity (dashed line in Fig. 9.1).

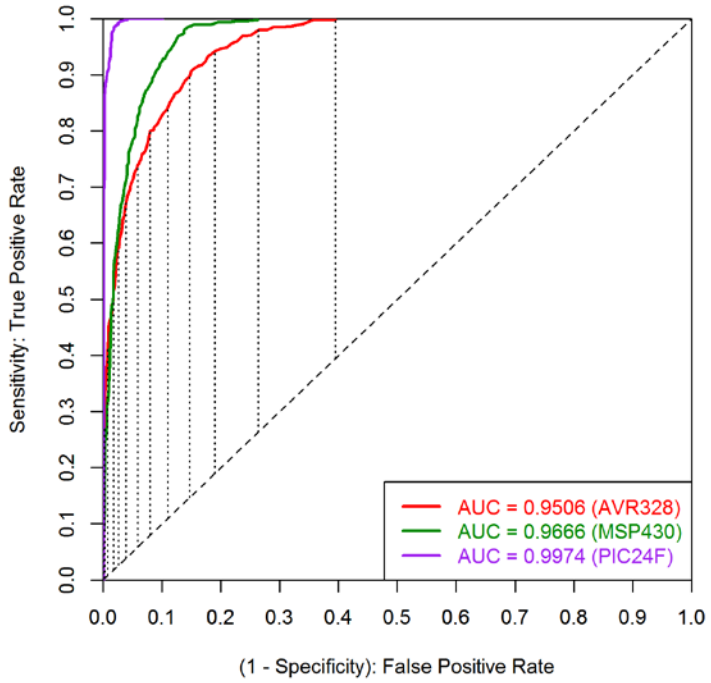


Figure 9.1: ROC curves for hardware intrusion detection model *fit10*

Despite other performance statistics or goodness-of-fit measurements for a model, the AUC validates a model against something of reality. As previously mentioned, the AUC measures a model's ability to discriminate between outcomes - simply stated a relative measure compared to the probability of chance. If the AUC is much greater than 0.5, then the prediction performance is much better than chance. Likewise, if the AUC is less than 0.5, random guessing may have better prediction performance than the model. So if a researcher presents a model that has higher prediction performance indicated by a particular statistic and yet the AUC is less than 0.5, then the model and statistical measurements need to be reconsidered. Alternately, a poorly fitting model may still have good discrimination (i.e. AUC) which further emphasizes the importance that model performance should be assessed using both goodness-of-fit and AUC

techniques [50]. For the hardware intrusion detection research, Fig. 9.1 shows that the predictive model has much higher probability of successful classification than that of chance for each of the intruder classes. In particular, the AUC for the PIC24F intruder is above the 99 percentile while that of the MSP430 and AVR328 intruders are 0.9666 and 0.9506 respectively.

9.6 Sensitivity, Specificity, and Precision Curves

The overall goal of intrusion detection in this statistical discussion is to predict the intruder class based on measured predictors and the fitted IDS model. This applies to both the binomial logit (i.e. presence of an intruder versus no intruder) and multinomial logit (distinguishing between several classes of intruders). In the multinomial case, measured predictors of a fitted model will provide a probability of classification for each intruder type with the total sum equaling one. Usually a class with the highest probability becomes the predicted outcome or identified intruder, but this is ultimately determined by a probability threshold (normally a default of 0.5).

In order to further discuss probability thresholds for prediction, the relationships observed from an error matrix such as Table 9.2 must be correlated to the common terminology of sensitivity and specificity. The true positive rate of a model (Eq. 9.8) is also called the sensitivity and is associated to type II errors (β or false negative rate) with the relationship:

$$\boldsymbol{\beta = 1 - sensitivity}. \tag{9.13}$$

The false positive rate (Eq. 9.9) is also known as the type I error (α) and is associated to specificity (i.e. true negative rate) with the relationship:

$$\boldsymbol{\alpha = 1 - specificity}. \tag{9.14}$$

There are several other relationships that can be derived from an error matrix [58], but the identification of an optimal probability threshold for model prediction is primarily derived from the sensitivity and specificity metrics. Table 9.5 shows the sensitivity, specificity, and precision calculations of the Table 9.2 error matrix for each of the four intruder classes. From the calculations it is clear that the non-intruder class performs at 100% in each metric. In addition, this IDS model distinguishes the PIC24F intruder class much better than distinguishing the AVR328 intruder class.

Table 9.5: Sensitivity, specificity, and precision calculations for *fit10* by intruder class

```
Class[1:4] = NOINTR AVR328 MSP430 PIC24F
METRICS:
  class      sens      spec precision
1     1 1.0000000 1.0000000 1.0000000
2     2 0.7333333 0.9203822 0.7674419
3     3 0.8266667 0.9200988 0.7931770
4     4 0.9488889 0.9792925 0.9467849
```

A classification design challenge is that the optimal threshold for an intruder class may not be the maximum probability in the predicted set (which includes probabilities for each intruder type). By iterating through different probability thresholds for the sensitivity and specificity of a fitted model and plotting the results (Figs. 9.2 through 9.4), an optimal threshold can be observed at the intersection of both curves. This intersection occurs at 0.27, 0.36 and 0.34 for the intruder classes AVR328, MSP430 and PIC24F respectively. Yet if a probability threshold of 0.27 is selected for a class and since the total sum of the set equals one, it is possible that the other classes might also have probabilities higher than the threshold. This means that the classification may not reach convergence. Subsequently the optimization of prediction probability thresholds requires further research.

Figures 9.2 through 9.4 also show plots of the precision versus probability thresholds, although it is clear that optimizing for precision rather than specificity will have a much reduced sensitivity. Additional ROC, sensitivity, specificity, and precision curves for *fitF* and *fit14* are located in Appendix B.

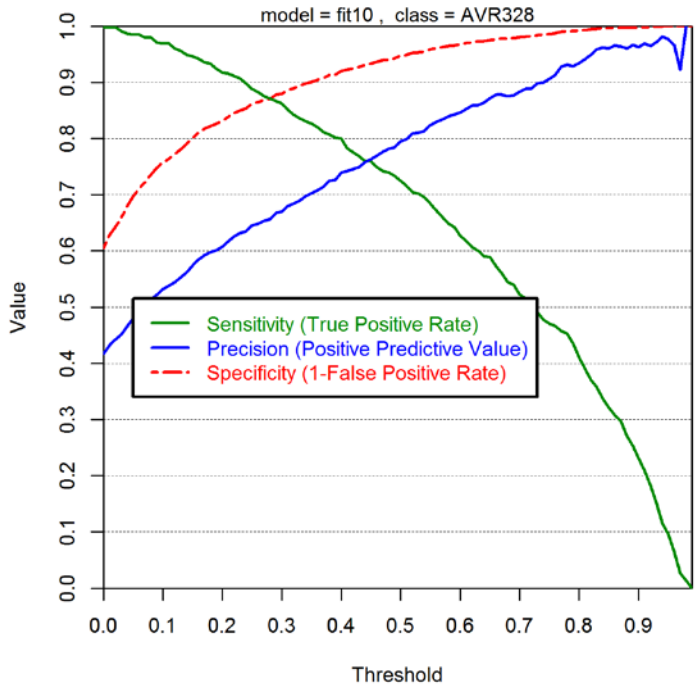


Figure 9.2: Intruder class AVR328 plot of sensitivity, specificity and precision versus all possible probability thresholds for hardware intrusion detection model *fit10*

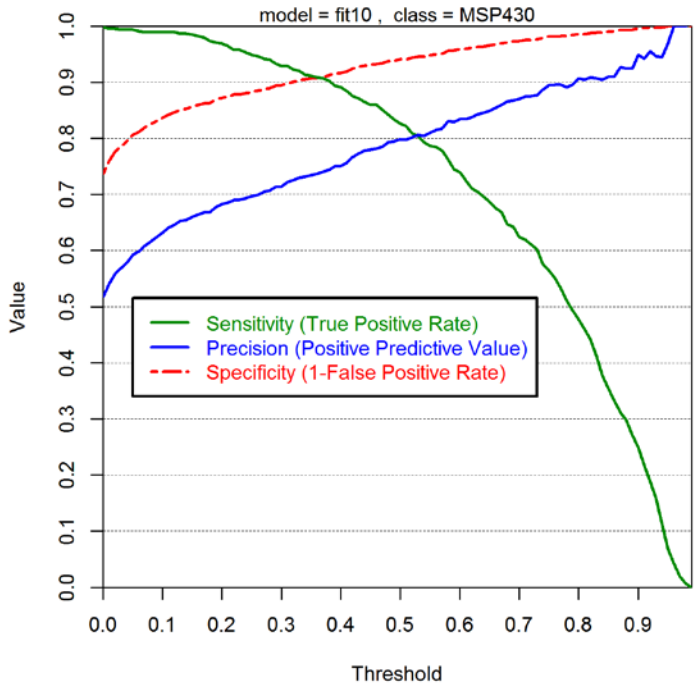


Figure 9.3: Intruder class MSP430 plot of sensitivity, specificity and precision versus all possible probability thresholds for model *fit10*

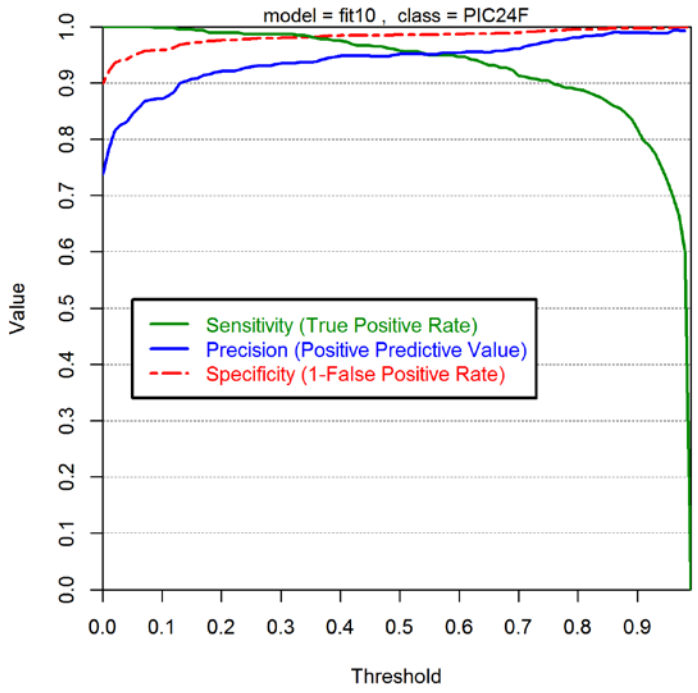


Figure 9.4: Intruder class PICF24F plot of sensitivity, specificity and precision versus all possible probability thresholds for model *fit10*

CHAPTER 10 FUTURE WORK AND TECHNOLOGY ROADMAP

10.1 Statistical Optimization of IDS Model and Detection Algorithm

Aside from the potential optimizations previously discussed on the prediction probability thresholds which balance sensitivity and specificity, a fundamental question must be answered: how good of performance do we need for the hardware intrusion detection problem? The selected IDS model from backward elimination, *fit10*, has an overall accuracy of 89.15% which is far better than current industry embedded systems which lack a hardware intrusion detection solution. But this accuracy would not suffice in a human authentication application.

Regardless, the IDS detection model and algorithms are likely to improve by applying several additional techniques. The first expected improvement would come from using a Bayesian approach to multinomial logistic regression modeling. A Bayesian approach accounts for previously known information during the training phase, rather than a blind approach to each measurement and prediction. With a Bayesian approach, the accuracy of the predictions or intrusion detection should improve with each iteration of model training.

A second improvement would result from the stratification of training data. Currently 686 data entries exist for the non-intruder class state and 450 runs for each intruder. Stratification would normalize the data set such that each class will have an equal number of data entries and prevent a bias in training the model. The IDS detection model is currently biased toward the non-intruder class and consequently provides 100% identification on the presence of an intruder, but only 89.15% accuracy in distinguishing between intruder classes. Stratification of the training data may slightly reduce performance in classifying the non-intruder, but should disproportionately increase the performance of identifying the other intruder classes.

Additional improvements to this intrusion detection system might be obtained by using a third approach that combines several of the previously discussed techniques into a tiered-classification algorithm. The first intrusion detection classifier would be optimized to generally detect the presence of an intruder at the cost higher false positive rates, but it would be fast and less computationally intensive. Depending on the outcome, a second classifier algorithm which

is more costly than the first could be optimized to distinguish between the intruder classes. In this approach the outcome of the first algorithm must be a measure of probability rather than discrete classification. In addition this approach also requires additional research to understand the conditions when certain class identification might do better. Likewise, a third orthogonal algorithm could also be applied to refine the classification performance and confidence of prediction.

10.2 Technology Roadmap

Since this cybersecurity research may be one of the first to address hardware Trojans installed within an embedded system (not specific to IC based inclusions), there is much work to do before it becomes a mature solution ready for integration into real critical infrastructure systems. The following research questions and concepts still need to be answered:

- Determine the effects of IDS sampling frequency in order to optimize detection vs. hardware resources
- Solution for passive hardware eavesdropping
- Solution for multiple authorized master devices on the communication bus
- Apply these concepts to different types of hardware intruders including voltage follower op amp (Fig. 3.5)
- Securely integrate this IDS technology onto an embedded system during manufacturing
- Compare the performance of IDS model(s) on embedded microprocessors and hardware described in Chapter 4
- Independent vulnerability analysis of this hardware intrusion detection technology

10.3 Future Work

There are several extensions to the work presented in this research which may prove to be fruitful and extend the application space of this research's approach:

- Distinguishing between individual ICs of the same model and lot
- Distinguishing counterfeit parts (different RC characteristics)
- Comparing the performance with other machine learning techniques
- Study the analog characteristics of circuit components to determine if normal manufacturing process variance can create unique hardware signatures that are very-difficult to replicate.

CHAPTER 11 CONCLUSIONS

Like many cybersecurity solutions, the technology and approach introduced in this research is not a stand-alone solution for all security issues. But it is a very cost-effective solution for a new capability and shows promise not only in its ability to identify the presence of a hardware Trojan on an inter-chip communication bus, but also to distinguish between types of intruders. The detection results of this technology can possibly be used for security attribution purposes and start to address the complexity of supply-chain hardware security issues. It can also be combined with network centric IDS systems and host-based systems for high-resolution and complete security view on critical infrastructure embedded systems.

To summarize the work of this research, it seeks to help solve the growing cybersecurity challenges for critical infrastructure systems. The first several chapters provide a background on the cybersecurity issues with embedded systems. Chapter 1 introduced the generalized problem, Chapter 2 thoroughly discusses the cybersecurity issues of supply chain and a device's engineering lifecycle, and Chapter 3 detailed an embedded system hardware threat model. And while this background information is not all inclusive it does provide more depth and additional dimensions to cybersecurity from a system's engineering perspective; ones that are not addressed in traditional security research focused on process or network traffic-based intrusion detection, nor are they addressed by typical security penetration testing.

The subsequent chapters introduce a new approach using resistor-capacitor circuits and their dynamic response to characterize a system and any attached hardware intruders. Chapter 4 describes the technical details of the IDS hardware, Chapter 5 describes the custom design smart meter research platform that enables data collection for this research and future research, and Chapter 6 outlines the design of the experiment used to collect empirical data of this IDS technology on a real embedded system.

The final chapters analyze the data so that inferences can be drawn about the usefulness and performance of this IDS technology. Chapter 7 helps quickly identify relationships between data and experimental factors that can be further analyzed using numerical analysis and Chapter 8 characterizes the minimal noise of the IDS system. Chapter 9 discusses numerical approaches

and detection algorithms using statistical and categorical data analysis, builds an IDS model using backward elimination while checking the goodness-of-fit and intruder classification performance, and also presents the importance of validating a model against the probability of chance. And while there are possibilities for improvement of the IDS model and classification performance through Bayesian approach, training data stratification or using tiered-response detection algorithms, it still presents a promising approach that begins to address a much larger cybersecurity problem for critical infrastructure.

Ultimately this technology combines the use of an analog signal response from a resistor-capacitor circuit and machine learning techniques to not only identify the presence of a hardware Trojan on an inter-chip communication bus at 100% accuracy for the dataset of over 2000 measurements, but which also correctly distinguishes between several types of implanted Trojans at 89% accuracy. While this research has focused on the security of inter-chip communication, it demonstrates the possibility of using low-power analog signals for device-level information assurance. And despite an extensive list of issues to solve before integrating into real systems, supply chain intricacies, and low-power embedded system design constraints, the new perspectives presented in this research have many implications for the growing complexities of securing cyber-physical systems.

REFERENCES

[1] M. Keogh and C. Cody, “Cybersecurity for State Regulators With Sample Questions for Regulators to Ask Utilities,” National Association of Regulatory Utility Commissioners (NARUC) Grants & Research, Jun. 2012.

[2] U.S. Department of Commerce, Bureau of Industry and Security Office of Technology Evaluation, “Defense Industrial Base Assessment: Counterfeit Electronics,” Jan. 2010.

[3] “NCER,” *Ecycling Process*. [Online]. Available: http://www.electronicsrecycling.org/. [Accessed: Jun. 23, 2012].

[4] “NIST SP 800-53 Rev. 4 - DRAFT Security and Privacy Controls for Federal Information Systems and Organizations,” Feb. 2012.

[5] J. Yoshida, “Disasters, shortages, counterfeits: For industry, 2011 was a year of wakeup calls,” *EDN*, vol. 57, no. 2, pp. S11–S14, Jan. 2012.

[6] R. A. Lebron, R. Rossi, and W. Foor, “Risk-Based COTS Systems Engineering Assessment Model: A Systems Engineering Management Tool and Assessment Methodology to Cope with the Risk of Commercial Off-the-Shelf (COTS) Technology Insertion During the System Life Cycle,” in *Strategies to Mitigate Obsolescence in Defense Systems Using Commercial Components*, Budapest, Hungary, 2000.

[7] J. T. Hanlon, “The Future of Components for High Reliability Military and Space Applications,” Sandia National Laboratories, Feb. 1996.

[8] J. Minihan, E. Schmidt, G. Enserro, and M. Thompson, “Commercial Off-the-Shelf (COTS) Components and Enterprise Component Information System (eCIS): Topical Report on Enhanced Surveillance Program 703001,” Kansas City Plant, U.S. Department of Energy, KCP-613-8421s, Jun. 2008.

[9] Developmental Test Command (DTC), “MIL-STD-810G Environmental Engineering Considerations and Laboratory Tests.” U.S. Department of Defense (DoD), 31-Oct-2008.

[10] G. Box and N. Draper, “A Basis for the Selection of a Response-Surface Design,” *J. Am. Stat. Assoc.*, vol. 54, no. 287, pp. 622–654, 1959.

[11] G. Taguchi, S. Chowdhury, and Y. Wu, *Taguchi’s Quality Engineering Handbook*. Hoboken, New Jersey: John Wiley & Sons, Inc, 2007.

[12] W. Shaw, F. Speyerer, and P. Sandborn, “Final Report - Diminishing Manufacturing Sources and Material Shortages (DMSMS) Non-Recurring Engineering (NRE) Cost Metric Update,” Defense Microelectronics Activity, McClellan, CA, Contract H94003-10-F-0109, Sep. 2010.

[13] Wikipedia contributors, “Six Sigma,” *Wikipedia, the free encyclopedia*, Jun. 16, 2012. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Six_Sigma&oldid=497896848. [Accessed: Jun. 17, 2012].

[14] T. Ōno, *Toyota Production System: Beyond Large-Scale Production*. Cambridge, Mass.: Productivity Press, 1988.

[15] Alabama Industrial Development Training (AIDT) and Alabama Department of Commerce, “Just-In-Time Manufacturing,” Sep. 2006.

[16] R. Schonberger, *Japanese Manufacturing Techniques: Nine Hidden Lessons in Simplicity*. New York: Free Press, 1982.

[17] U.S. Department of Justice, “Press Release: Administrator of Florida-based VisionTech Components, LLC Pleads Guilty in Connection With Sales of Counterfeit High Tech Devices Destined to the U.S. Military and Other Industries,” U.S. Department of Justice, Nov. 2010.

[18] U.S. Attorney’s Office - U.S. Department of Justice, “Press Release: Administrator of VisionTech Components, LLC Sentenced to 38 Months in Prison for Her Role in Sales of Counterfeit Integrated Circuits Destined to U.S. Military and Other Industries,” Oct. 2011.

[19] United States District Court for the District of Columbia, *United States of America v. Stephanie A. McCloskey: Government’s Memorandum in Aid of Sentencing*. 2011.

[20] M. Tehranipoor and F. Koushanfar, “A Survey of Hardware Trojan Taxonomy and Detection,” *IEEE Design & Test of Computers*, vol. 27, no. 1, pp. 10–25, Feb. 2010.

[21] X. Wang, M. Tehranipoor, and J. Plusquellic, “Detecting Malicious Inclusions in Secure Hardware: Challenges and Solutions,” in *Hardware-Oriented Security and Trust, 2008. HOST 2008. IEEE International Workshop on*, 2008, pp. 15 –19.

[22] R. Karri, J. Rajendran, K. Rosenfeld, and M. Tehranipoor, “Trustworthy Hardware: Identifying and Classifying Hardware Trojans,” *Computer*, vol. 43, no. 10, pp. 39 –46, Oct. 2010.

[23] Y. Jin, N. Kupp, and Y. Makris, “Experiences in Hardware Trojan design and implementation,” in *Hardware-Oriented Security and Trust, 2009. HOST ’09. IEEE International Workshop on*, 2009, pp. 50 –57.

[24] A. S. Sedra and K. C. Smith, *Microelectronic Circuits*, 5th ed. New York: Oxford University Press, 2004.

[25] P. R. Gray and R. G. Meyer, “MOS Operational Amplifier Design-A Tutorial Overview,” *Solid-State Circuits, IEEE Journal of*, vol. 17, no. 6, pp. 969 –982, Dec. 1982.

[26] J. E. Solomon, “The Monolithic Op Amp: A Tutorial Study,” *Solid-State Circuits, IEEE Journal of*, vol. 9, no. 6, pp. 314 –332, Dec. 1974.

[27] “LM110/LM210/LM310 Voltage Follower,” National Semiconductor Corporation, Nov. 1994.

[28] “NIST SP 800-94 Guide to Intrusion Detection and Prevention Systems (IDPS),” Feb. 2007.

[29] R. Berthier, W. H. Sanders, and H. Khurana, “Intrusion Detection for Advanced Metering Infrastructures: Requirements and Architectural Directions,” in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*, 2010, pp. 350–355.

[30] R. Berthier and W. H. Sanders, “Specification-Based Intrusion Detection for Advanced Metering Infrastructures,” in *Dependable Computing (PRDC), 2011 IEEE 17th Pacific Rim International Symposium on*, 2011, pp. 184 –193.

[31] P. Jokar, H. Nicanfar, and V. C. M. Leung, “Specification-based Intrusion Detection for home area networks in smart grids,” in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, 2011, pp. 208 –213.

[32] S. Manich, M. S. Wamser, and G. Sigl, “Detection of Probing Attempts in Secure ICs,” presented at the HOST 2012, San Francisco, California, USA, 2012.

[33] D. Agrawal, S. Baktir, D. Karakoyunlu, P. Rohatgi, and B. Sunar, “Trojan Detection using IC Fingerprinting,” in *IEEE Symposium on Security and Privacy, 2007. SP ’07*, 2007, pp. 296–310.

[34] W. E. Cobb, E. W. Garcia, M. A. Temple, R. O. Baldwin, and Y. C. Kim, “Physical Layer Identification of Embedded Devices Using RF-DNA Fingerprinting,” in *Military Communications Conference - MILCOM*, 2010, pp. 2168–2173.

[35] W. E. Cobb, E. D. Laspe, R. O. Baldwin, M. A. Temple, and Y. C. Kim, “Intrinsic Physical-Layer Authentication of Integrated Circuits,” *Information Forensics and Security, IEEE Transactions on*, vol. 7, no. 1, pp. 14 –24, Feb. 2012.

[36] R. Maes and P. Tuyls, “Process Variations for Security: PUFs,” in *Secure Integrated Circuits and Systems*, I. M. R. Verbauwhede, Ed. Boston, MA: Springer US, 2010, pp. 125–141.

[37] G. Carlock, “The Two-Stage RC Low-Pass Matched Filter,” *IEEE Transactions on Communications*, vol. 20, no. 1, pp. 73– 74, Feb. 1972.

[38] A. Siska and M. He, “Golden Gloves ADC Championship Match - SAR vs. Sigma - Delta - Cypress Semiconductor,” May 18, 2010. [Online]. Available: http://www.cypress.com/?rID=43323. [Accessed: Nov. 11, 2011].

[39] "Itron CENTRON GPRS Smart Meter (datasheet)." SmartSynch, 2011.

[40] R. K. Iyer, Z. Kalbarczyk, and A. Slagell, "From Measurements to Security Science: Data-Driven Approach | Information Trust Institute." [Online]. Available: http://www.iti.illinois.edu/research/data-science/measurements-security-science-data-driven-approach. [Accessed: Jun. 10, 2012].

[41] R. L. Helinski, "A Physical Unclonable Function Derived From the Power Distribution System of an Integrated Circuit," Dissertation, 2010.

[42] S. Butterworth, "On the Theory of Filter Amplifiers," *Experimental Wireless & The Wireless Engineer*, pp. 536–541, Oct. 1930.

[43] "TS5A3160 1-Ohm SPDT Analog Switch (datasheet rev. C)," Texas Instruments Incorporated, Mar. 2012.

[44] D. W. Hamer, "Ceramic Capacitors for Hybrid Integrated Circuits," *Spectrum, IEEE*, vol. 6, no. 1, pp. 79 –84, Jan. 1969.

[45] M. Kahn, "Multilayer Ceramic Capacitors - Materials and Manufacture," AVX Corporation, Jul. 2004.

[46] "X7R Dielectric General Specifications," AVX Corporation, 2010.

[47] G. E. P. Box, W. G. Hunter, and J. S. Hunter, *Statistics for Experimenters: An Introduction to Design, Data Analysis, and Model Building*. John Wiley & Sons, 1978.

[48] C. Liu, P. M. Berry, T. P. Dawson, and R. G. Pearson, "Selecting Thresholds of Occurrence in the Prediction of Species Distributions," *Ecography*, vol. 28, no. 3, pp. 385–393, 2005.

[49] A. Agresti, *An Introduction to Categorical Data Analysis*, 2nd ed. Wiley-Interscience, 2007.

[50] D. W. Hosmer and S. Lemeshow, *Applied logistic regression*, 2nd ed. Wiley-Interscience Publication, 2000.

[51] J. Hausman and D. McFadden, "Specification Tests for the Multinomial Logit Model," *Econometrica*, vol. 52, no. 5, pp. 1219–1240, 1984.

[52] H. Akaike, "A New Look at the Statistical Model Identification," *Automatic Control, IEEE Transactions on*, vol. 19, no. 6, pp. 716 – 723, Dec. 1974.

[53] J. J. Faraway, *Practical regression and ANOVA using R*. Available: http://cran.r-project.org/doc/contrib/Faraway-PRA.pdf, 2002.

[54] W. N. Venables and B. D. Ripley, *Modern Applied Statistics with S*, 4th ed. Springer, 2002.

[55] R. G. Congalton, “A Review of Assessing the Accuracy of Classifications of Remotely Sensed Data,” *Remote Sensing of Environment*, vol. 37, no. 1, pp. 35–46, Jul. 1991.

[56] R. G. Congalton and K. Green, *Assessing the Accuracy of Remotely Sensed Data: Principles and Practices*, 2nd ed. CRC Press, 2008.

[57] C. E. Shannon, “A Mathematical Theory of Communication,” *The Bell System Technical Journal*, vol. 27, pp. 379–423, 623–656, Oct. 1948.

[58] Wikipedia contributors, “Sensitivity and specificity,” Jul. 5, 2012. [Online]. Available: http://en.wikipedia.org/w/index.php?title=Sensitivity_and_specificity&oldid=499251670. [Accessed: Jul 7, 2012].

APPENDIX A. Characteristic Graphs and Figures of Data

A.1 Scatterplots – Complete Dataset

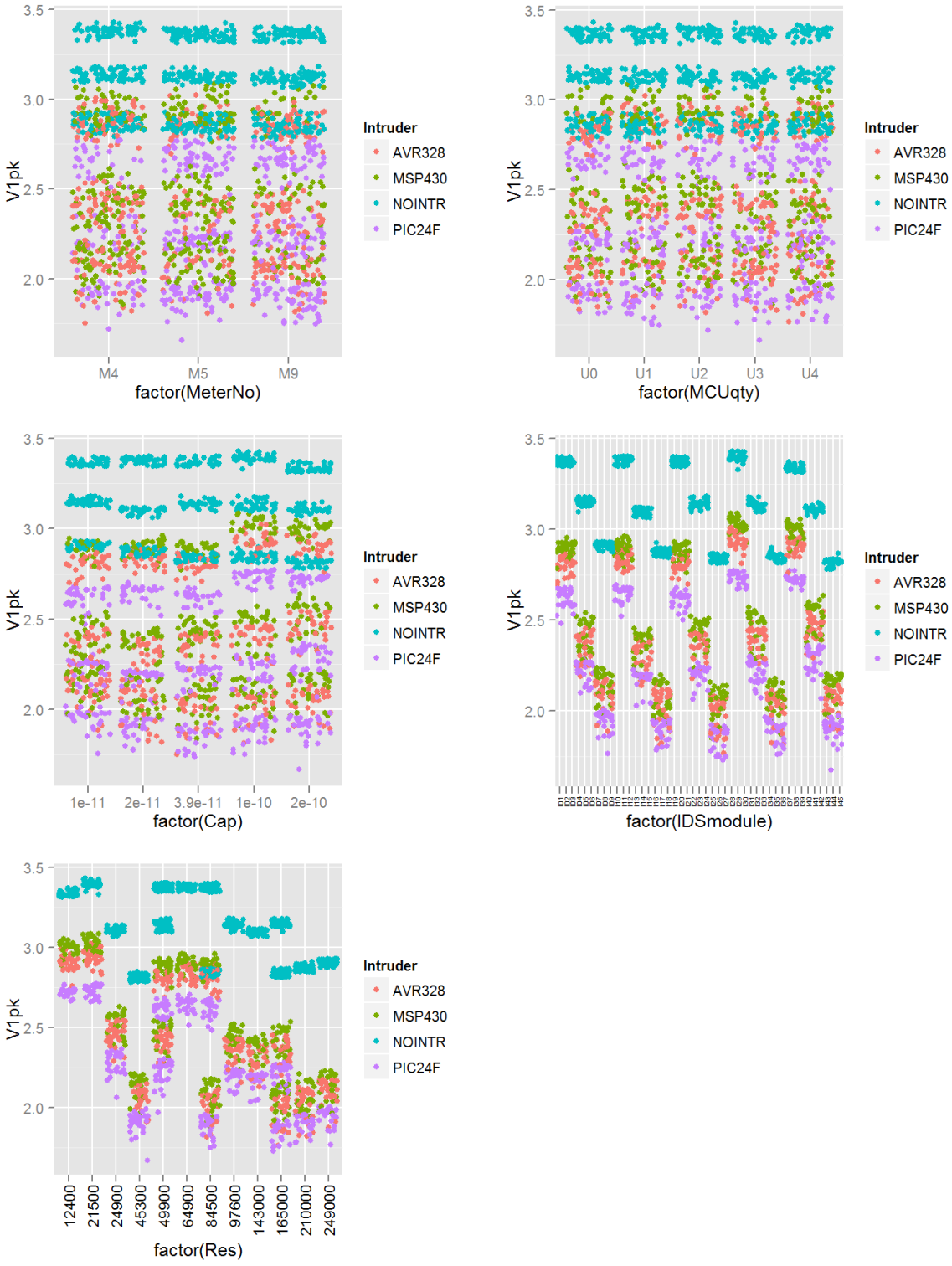


Figure A.1: Scatterplot of V1pk (complete dataset)

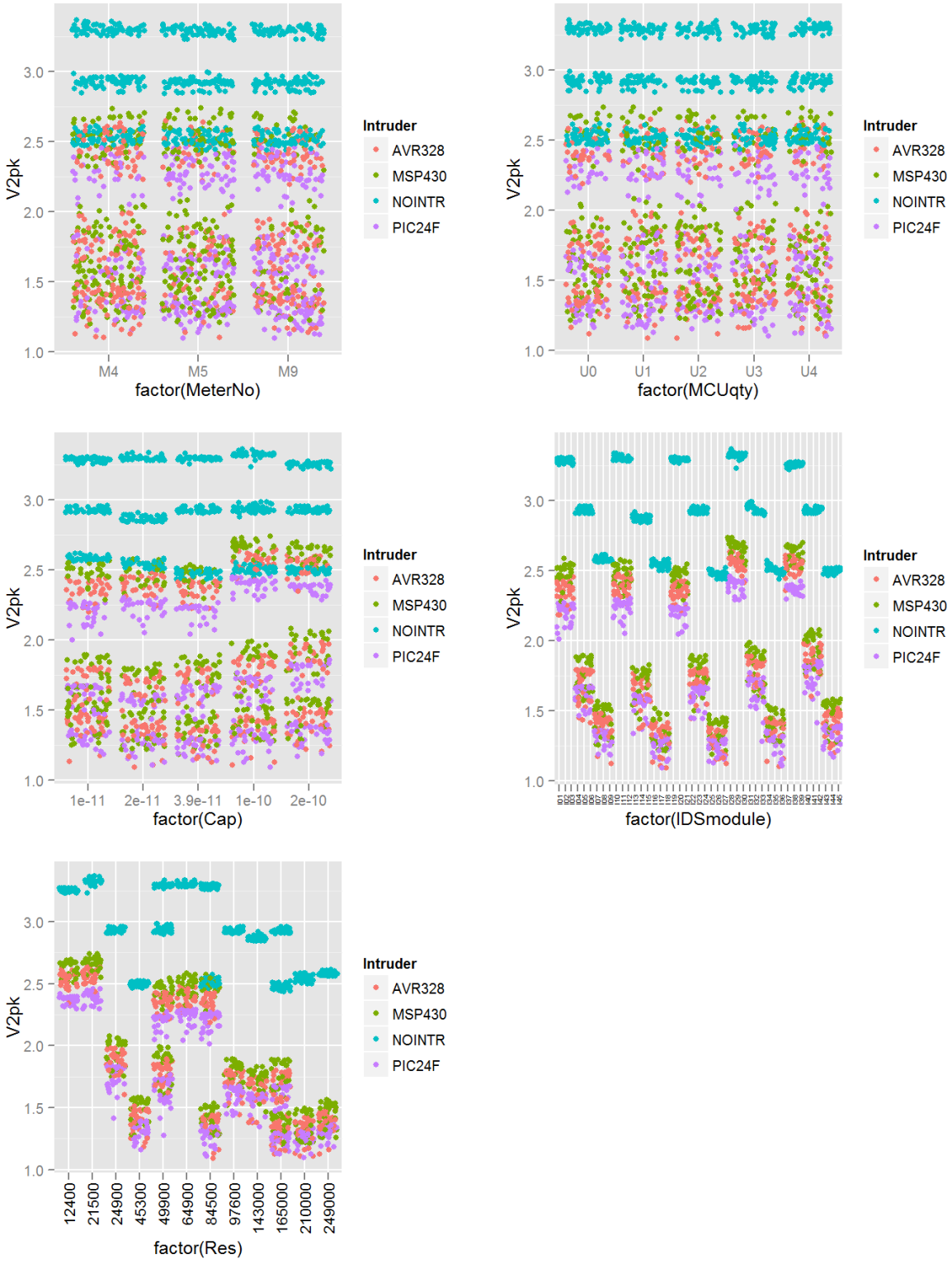


Figure A.2: Scatterplot of V2pk (complete dataset)

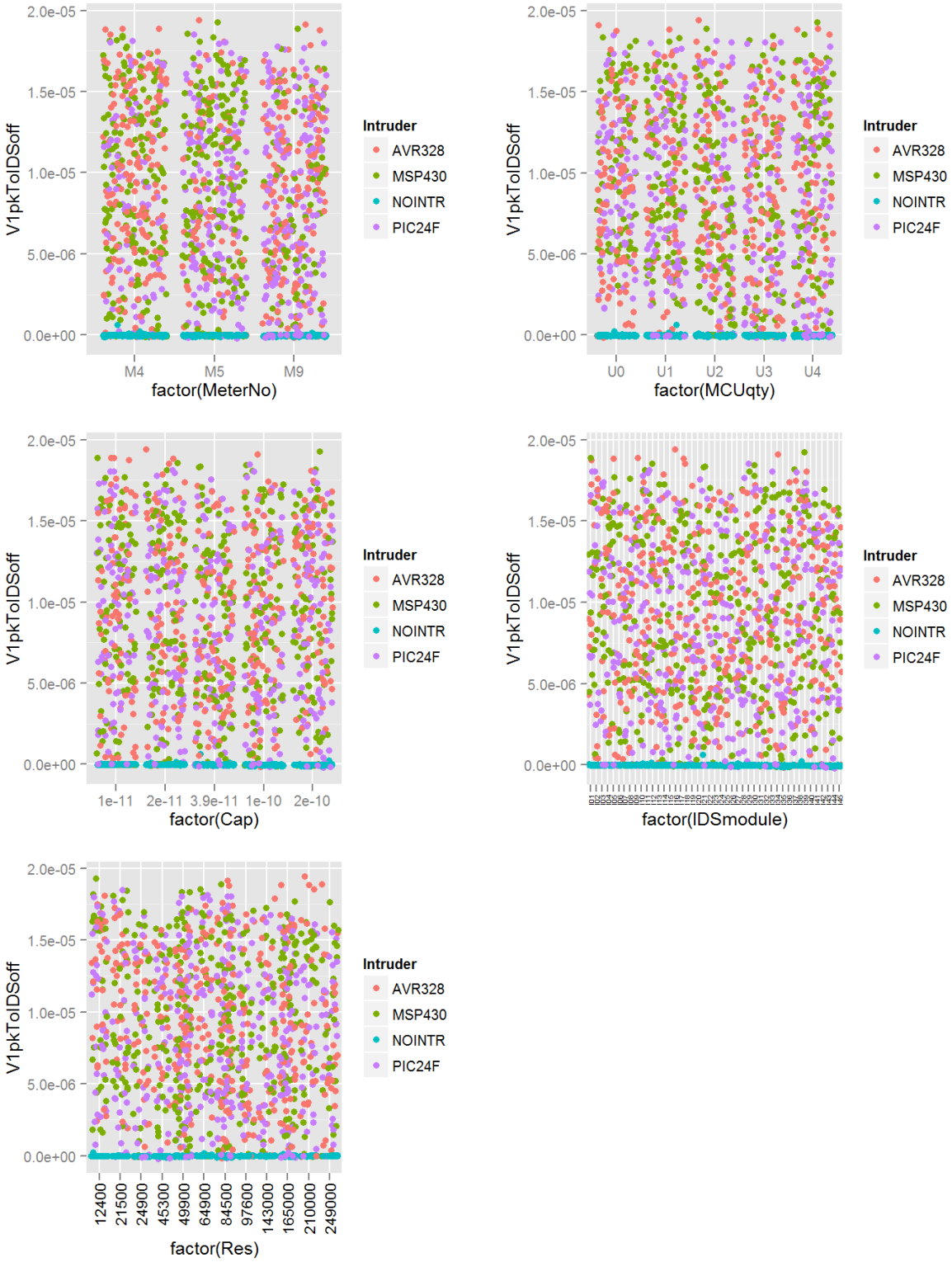


Figure A.3: Scatterplot of V1pkToIDSoff (complete dataset)

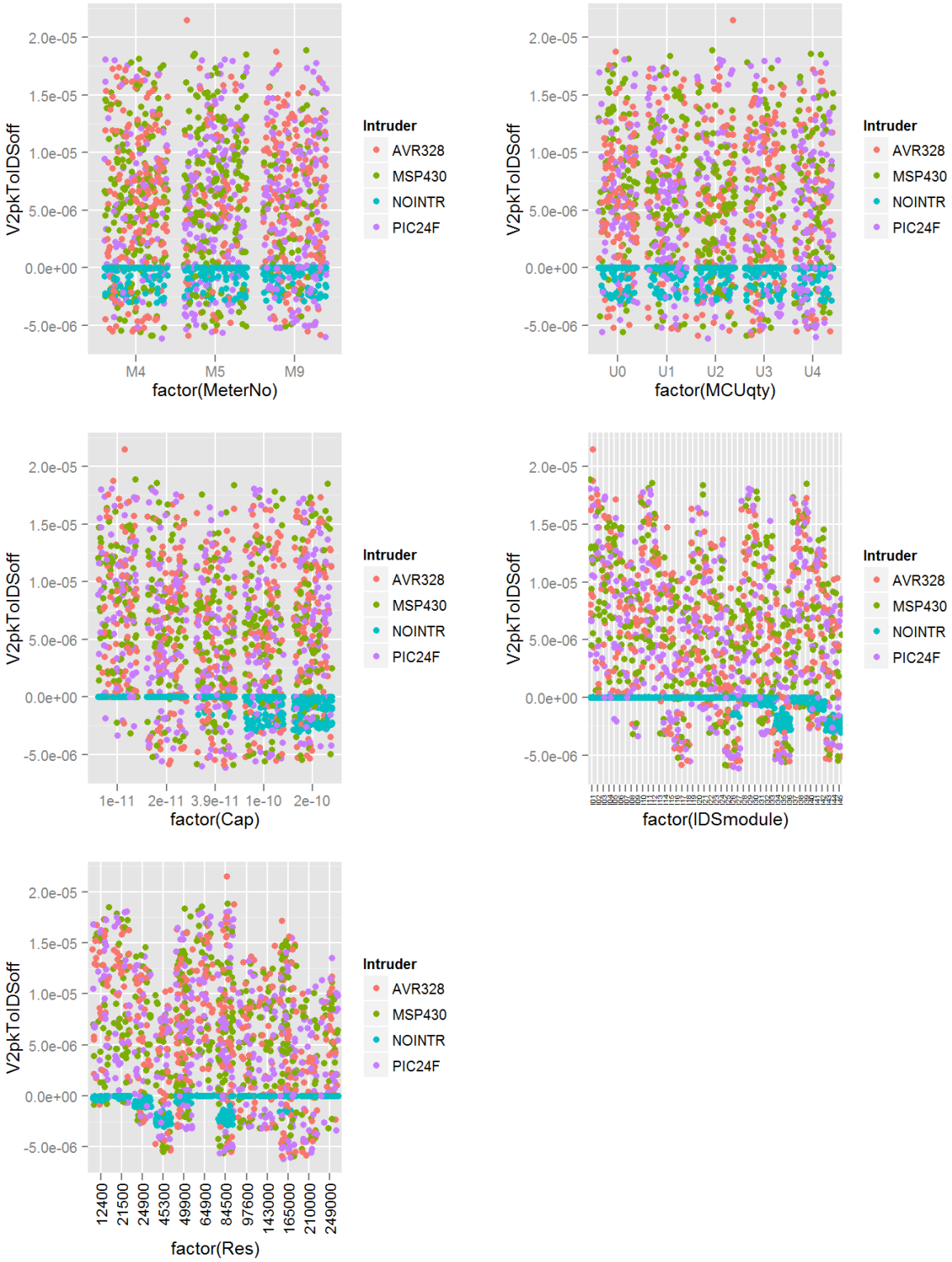


Figure A.4: Scatterplot of V2pkToIDSoff (complete dataset)

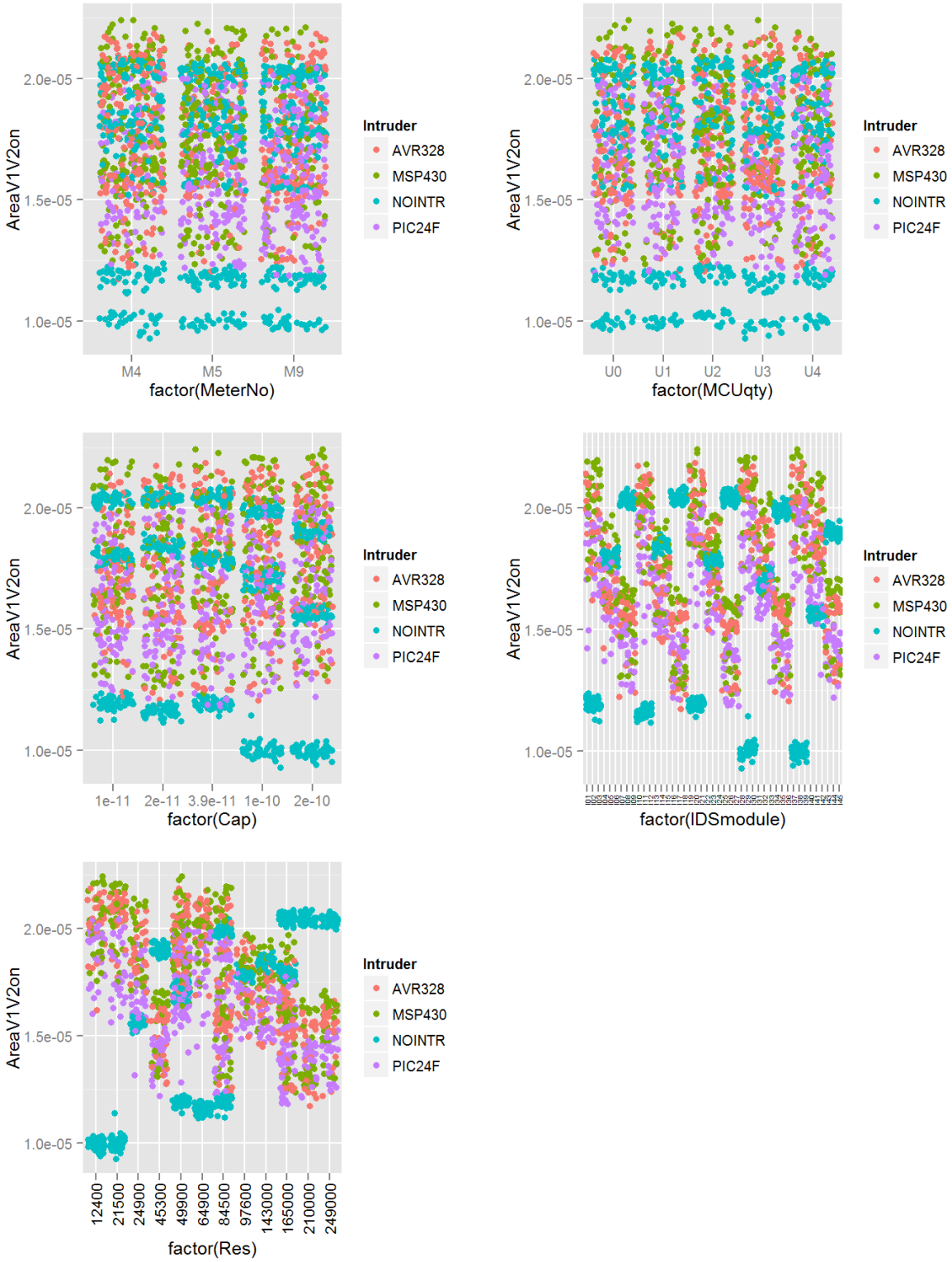


Figure A.5: Scatterplot of AreaV1V2on (complete dataset)

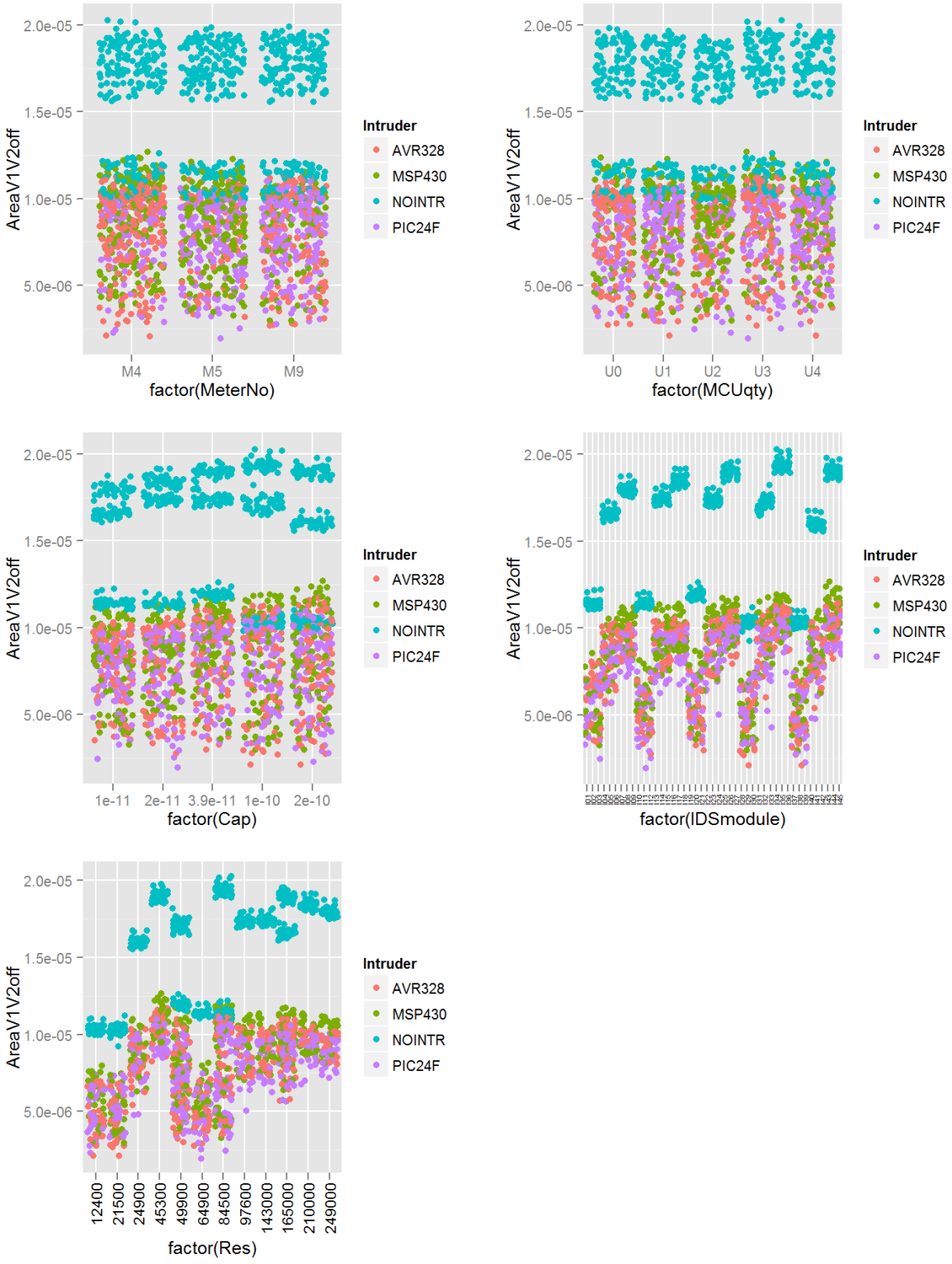


Figure A.6: Scatterplot of AreaV1V2off (complete dataset)

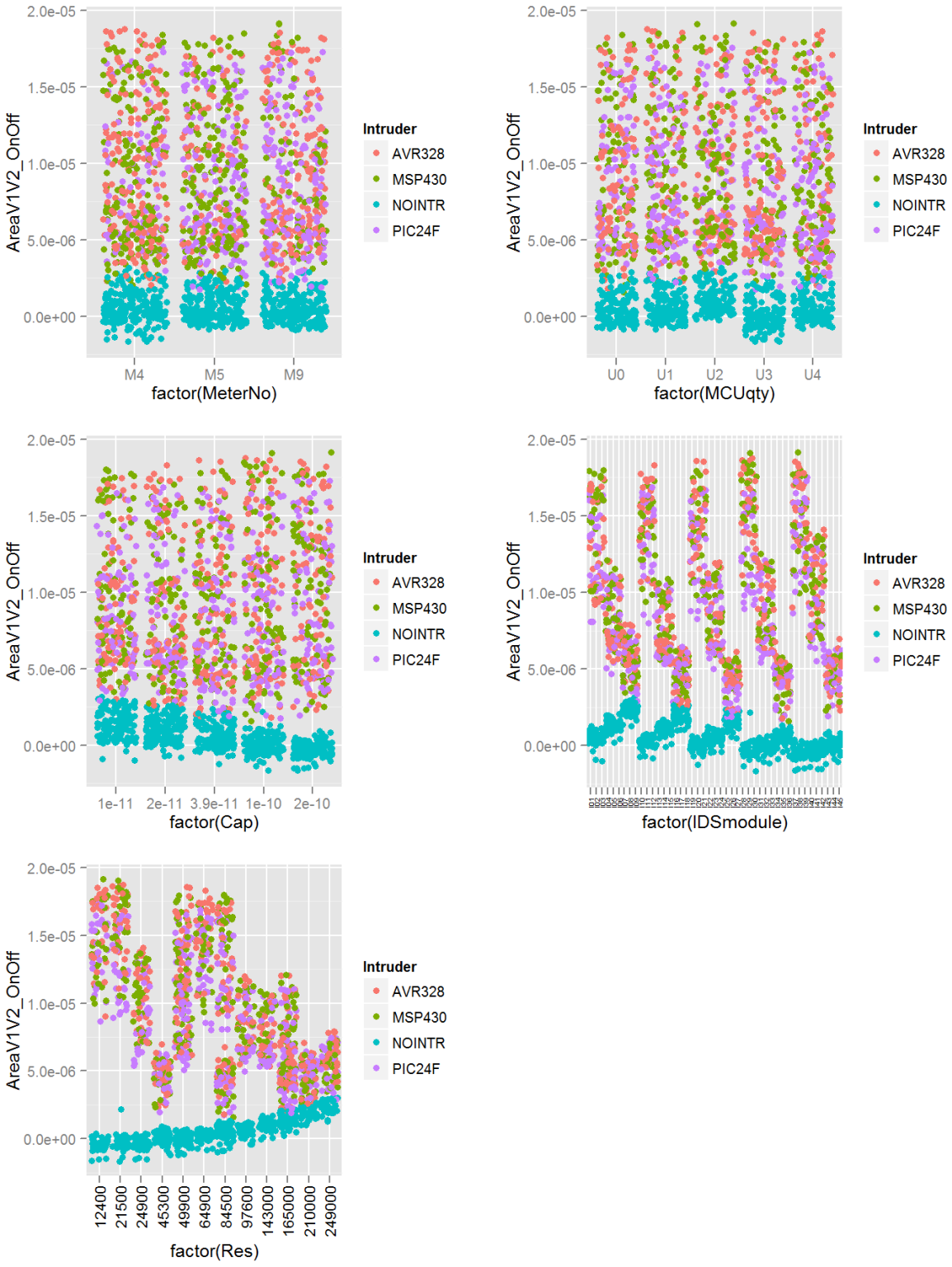


Figure A.7: Scatterplot of AreaV1V2_OnOff (complete dataset)

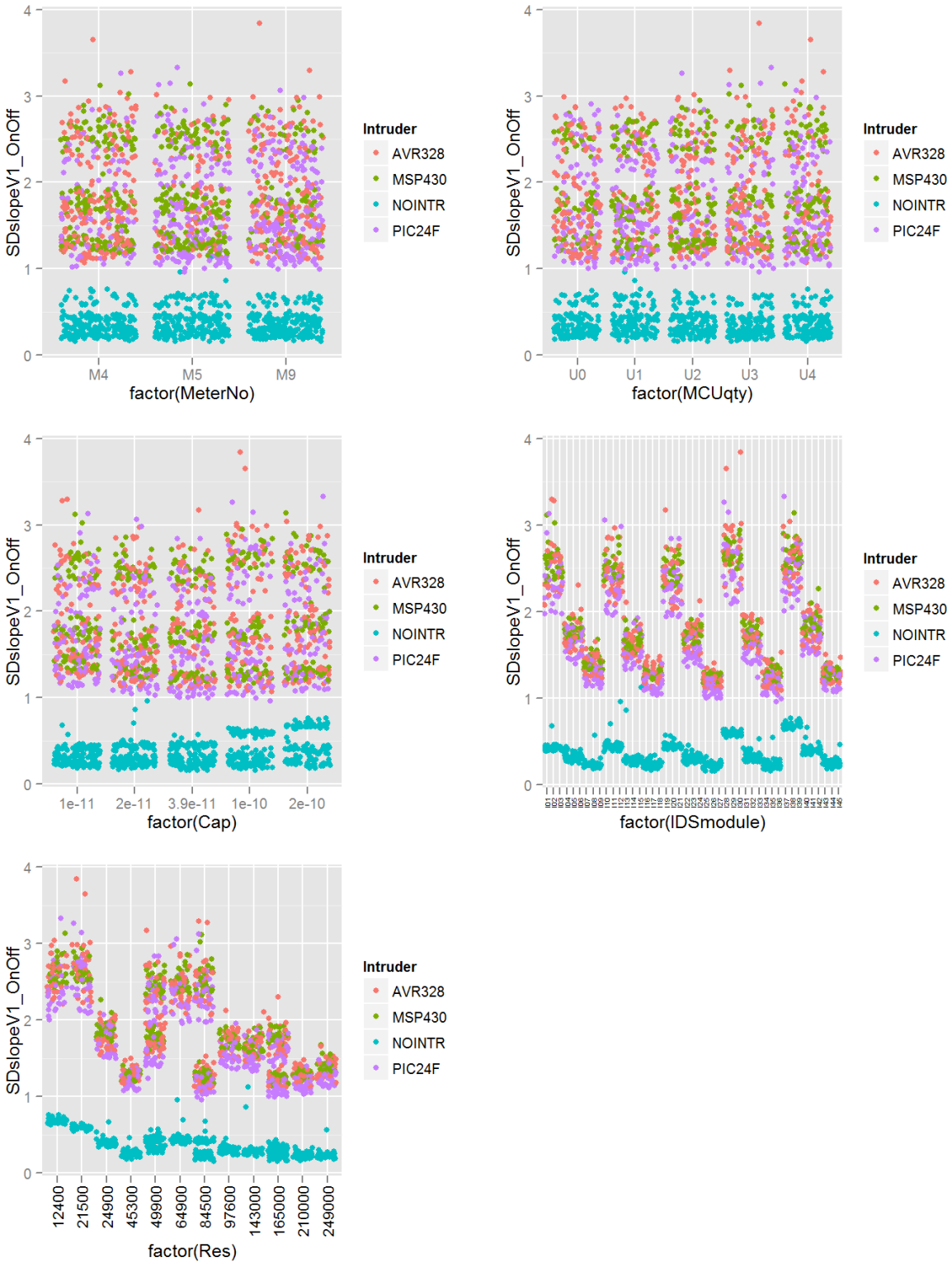


Figure A.8: Scatterplot of SDslopeV1_OnOff (complete dataset)

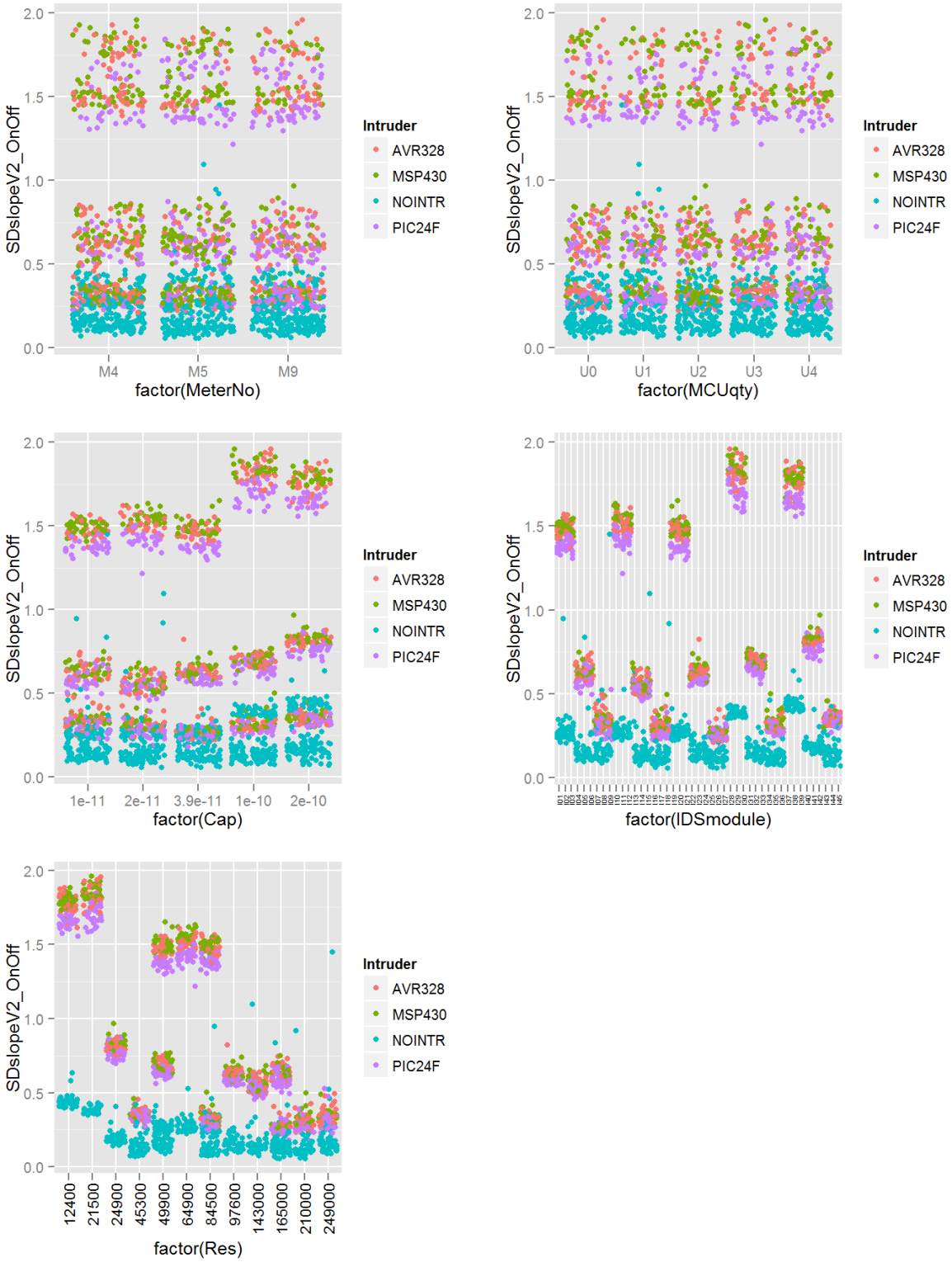


Figure A.9: Scatterplot of SDslopeV2_OnOff (complete dataset)

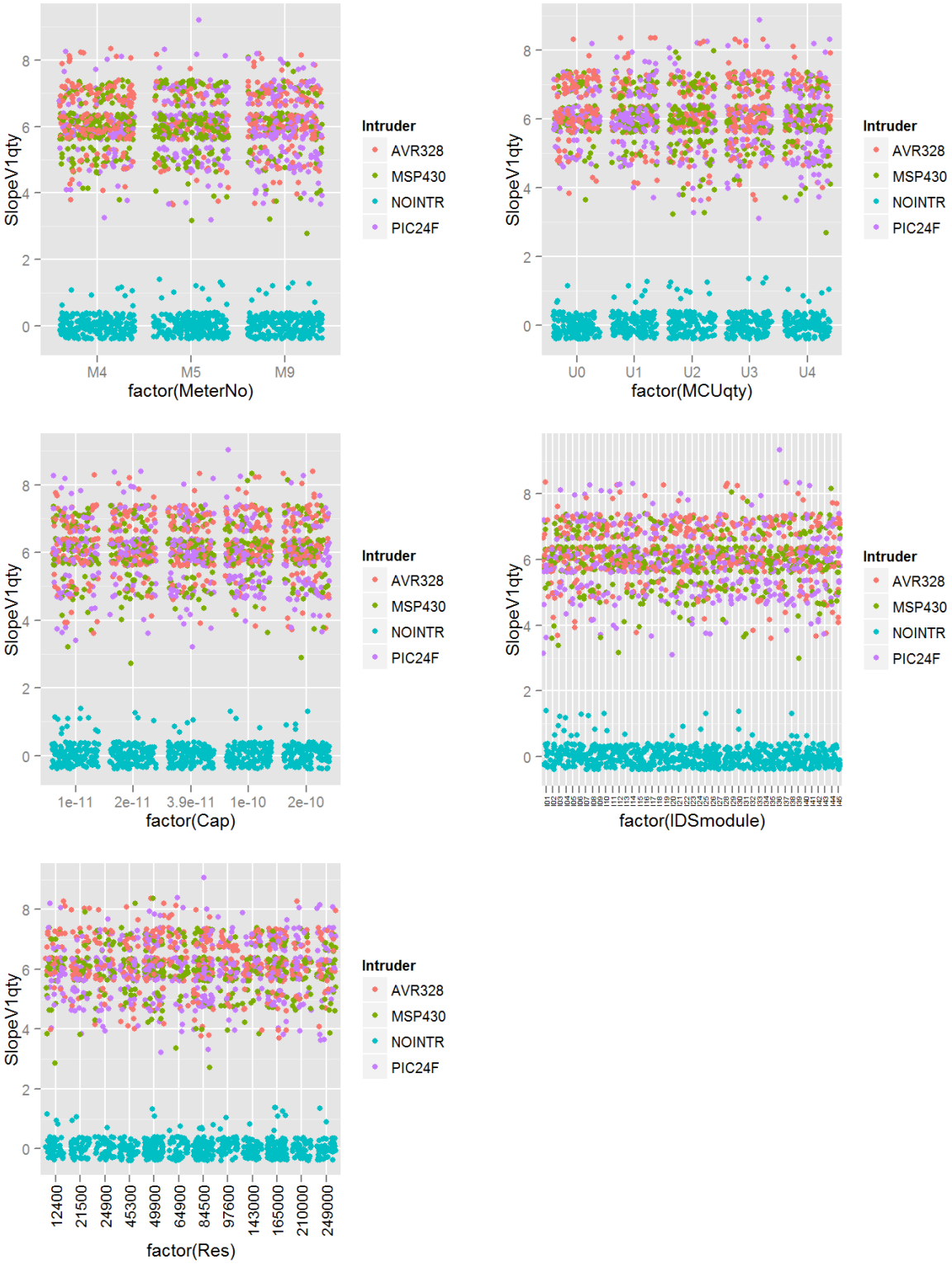


Figure A.10: Scatterplot of SlopeV1qty (complete dataset)

Figure A.11: Scatterplot of SlopeV2qty (complete dataset)

A.2 Density Plots of Complete Dataset

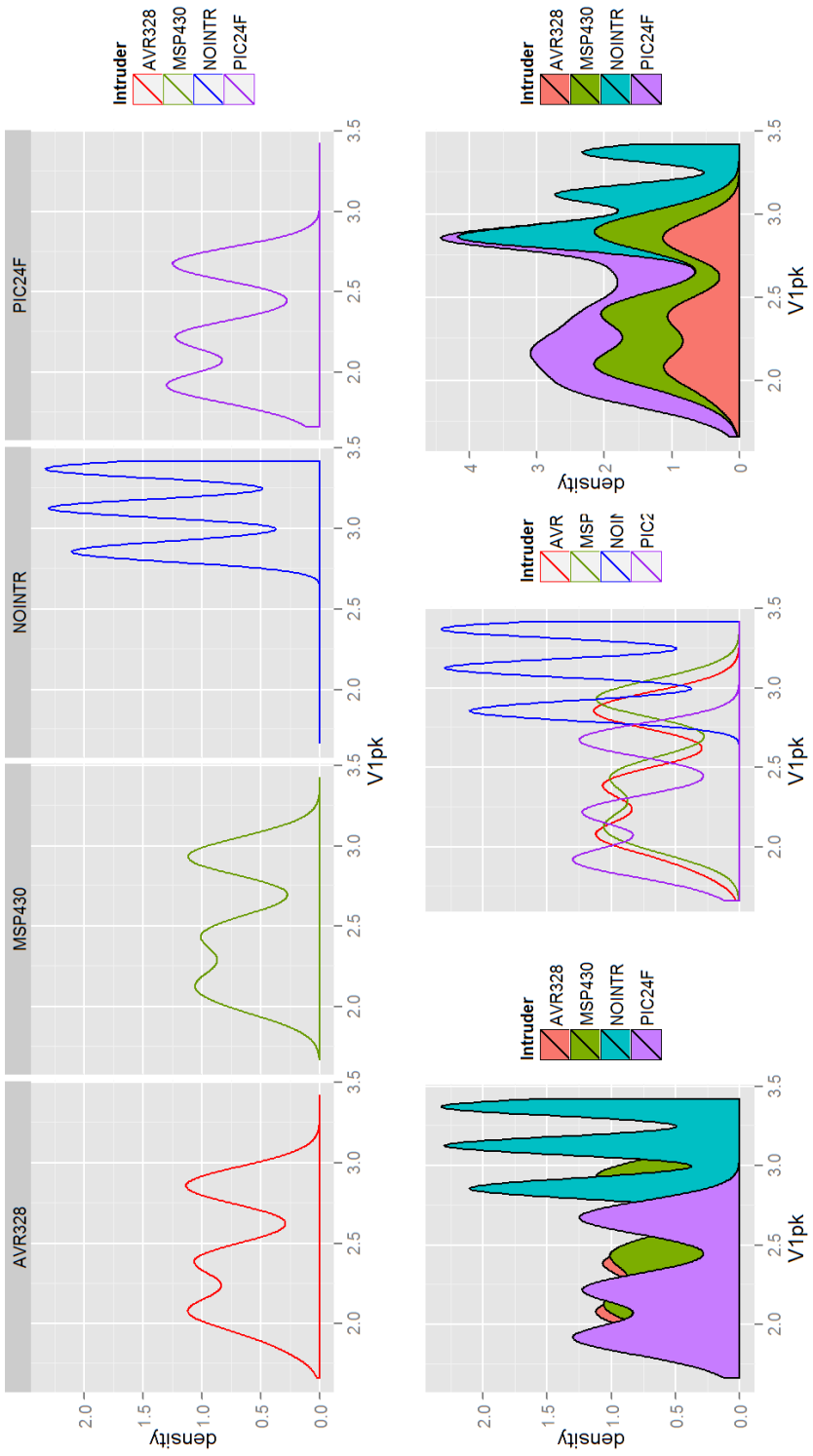


Figure A.12: Density plot of V1pk (complete dataset)

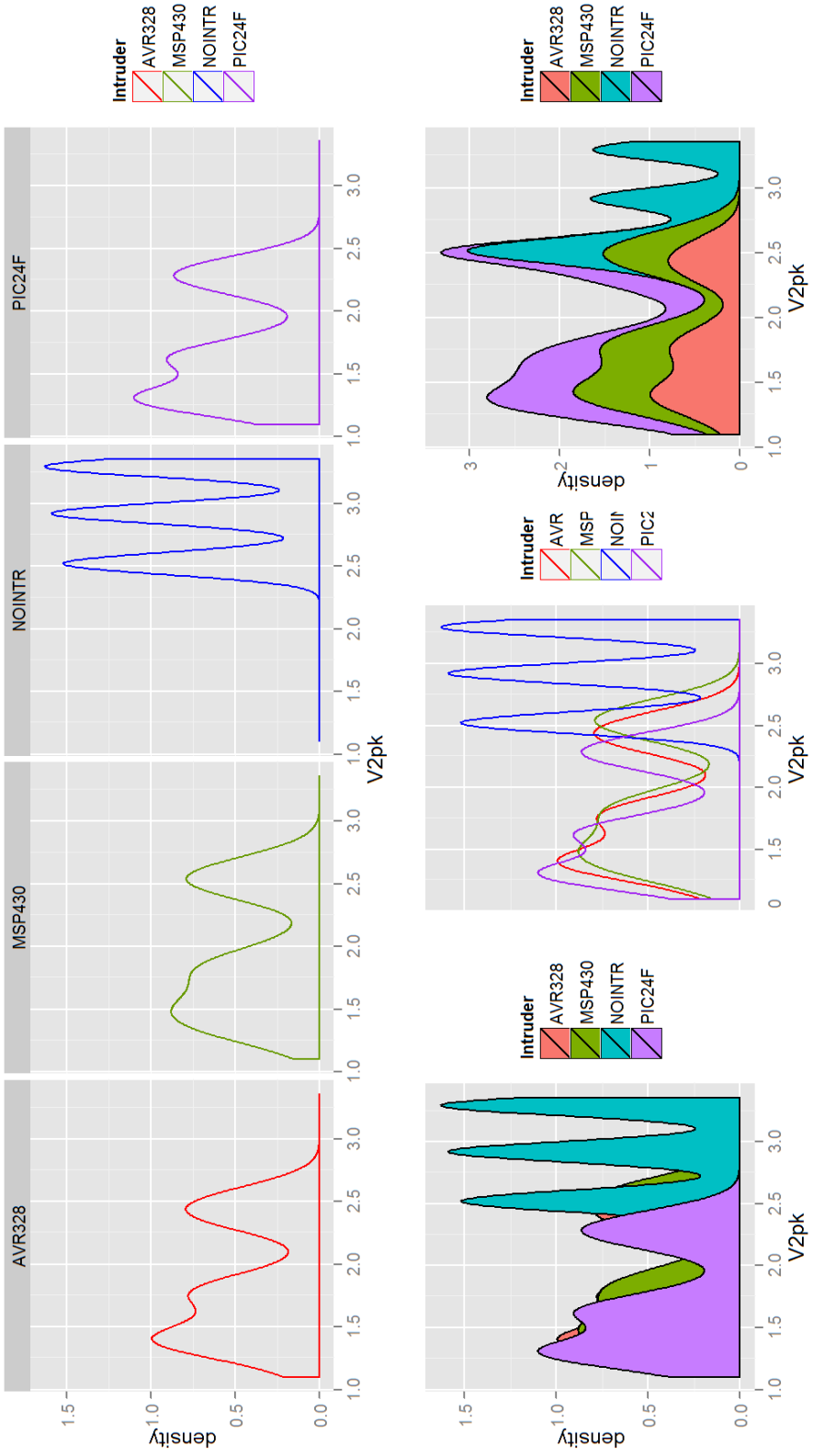


Figure A.13: Density plot of V2pk (complete dataset)

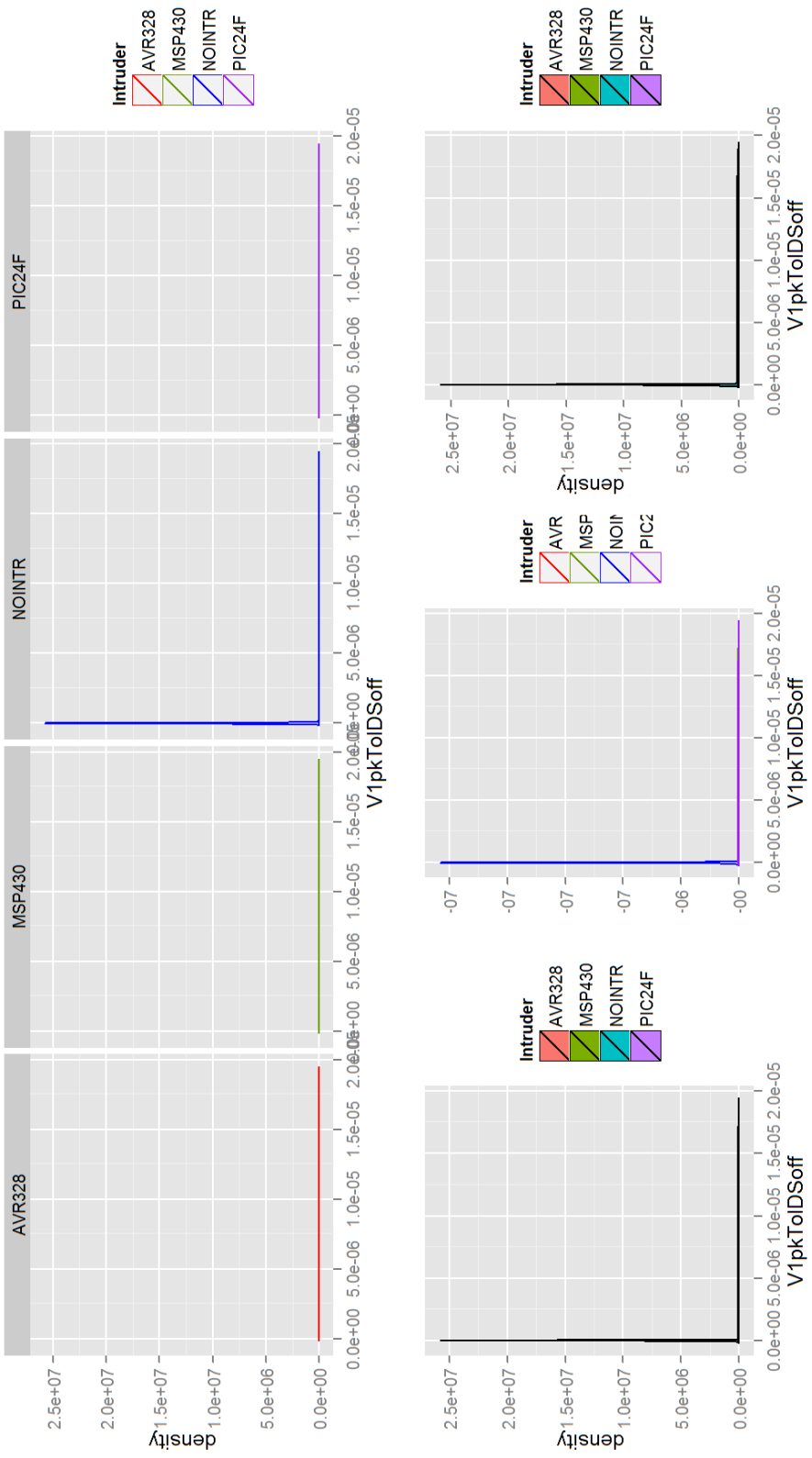


Figure A.14: Density plot of V1pkToIDSoff (complete dataset)

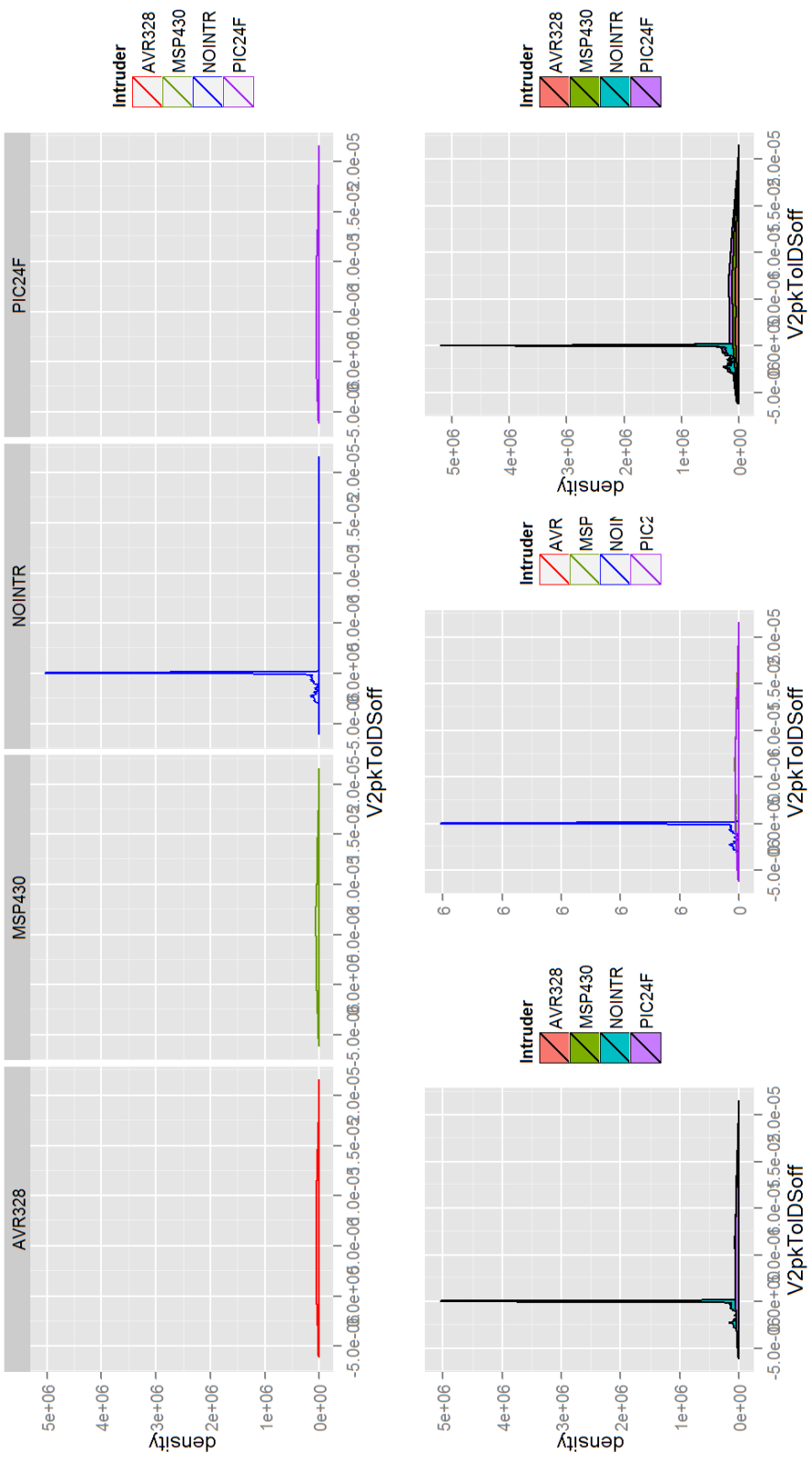


Figure A.15: Density plot of V2pkToIDSoff (complete dataset)

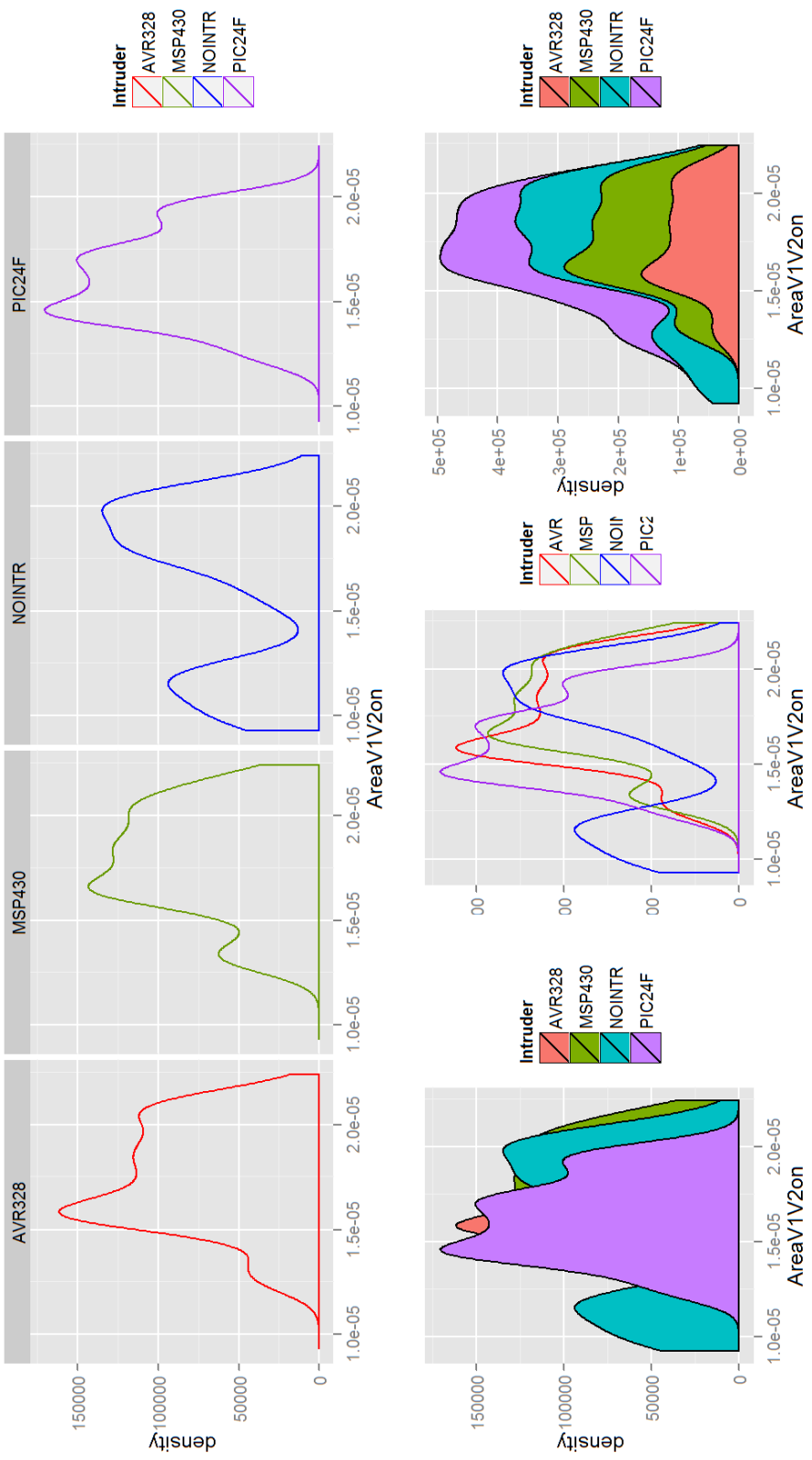


Figure A.16: Density plot of AreaV1V2on (complete dataset)

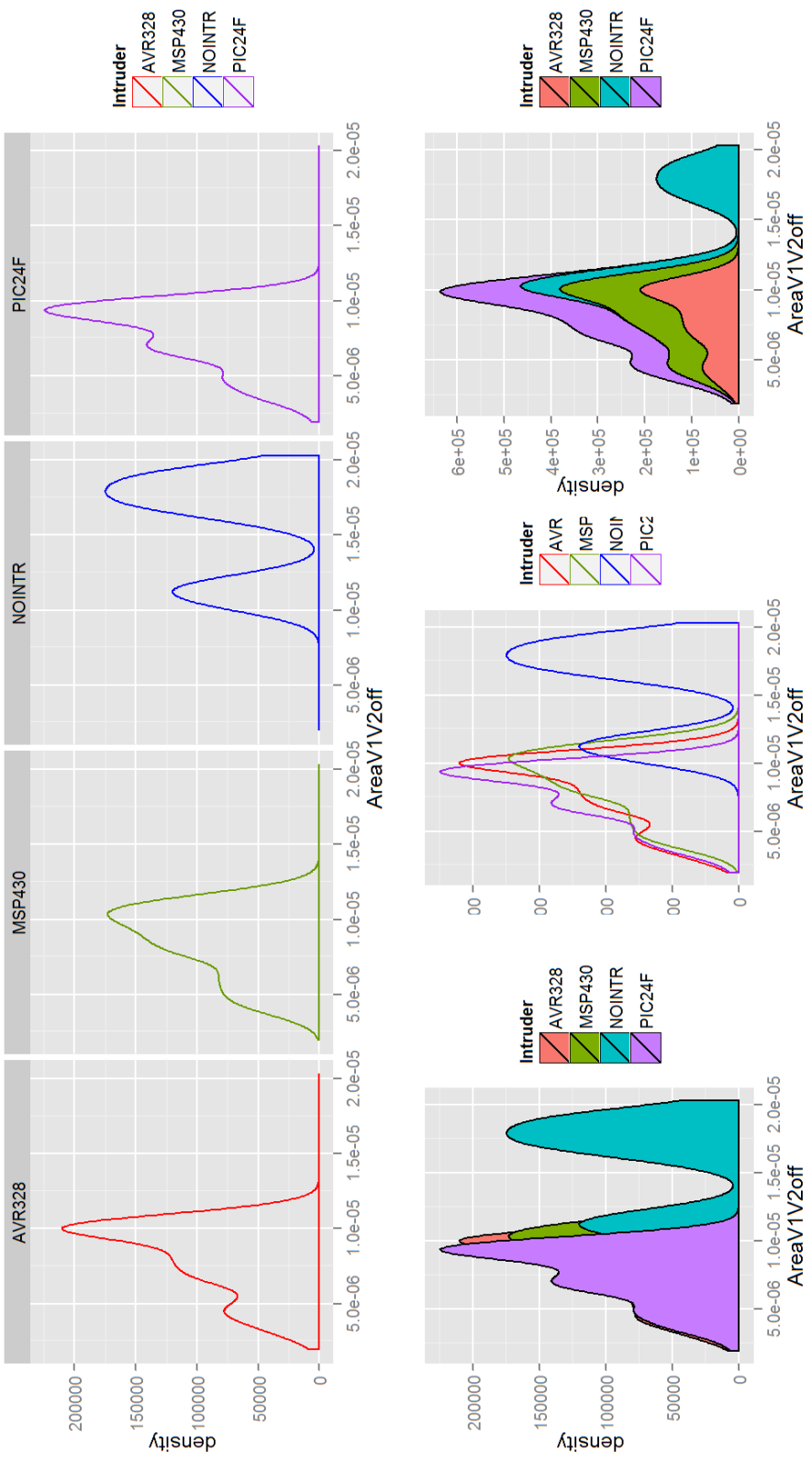


Figure A.17: Density plot of AreaV1V2off (complete dataset)

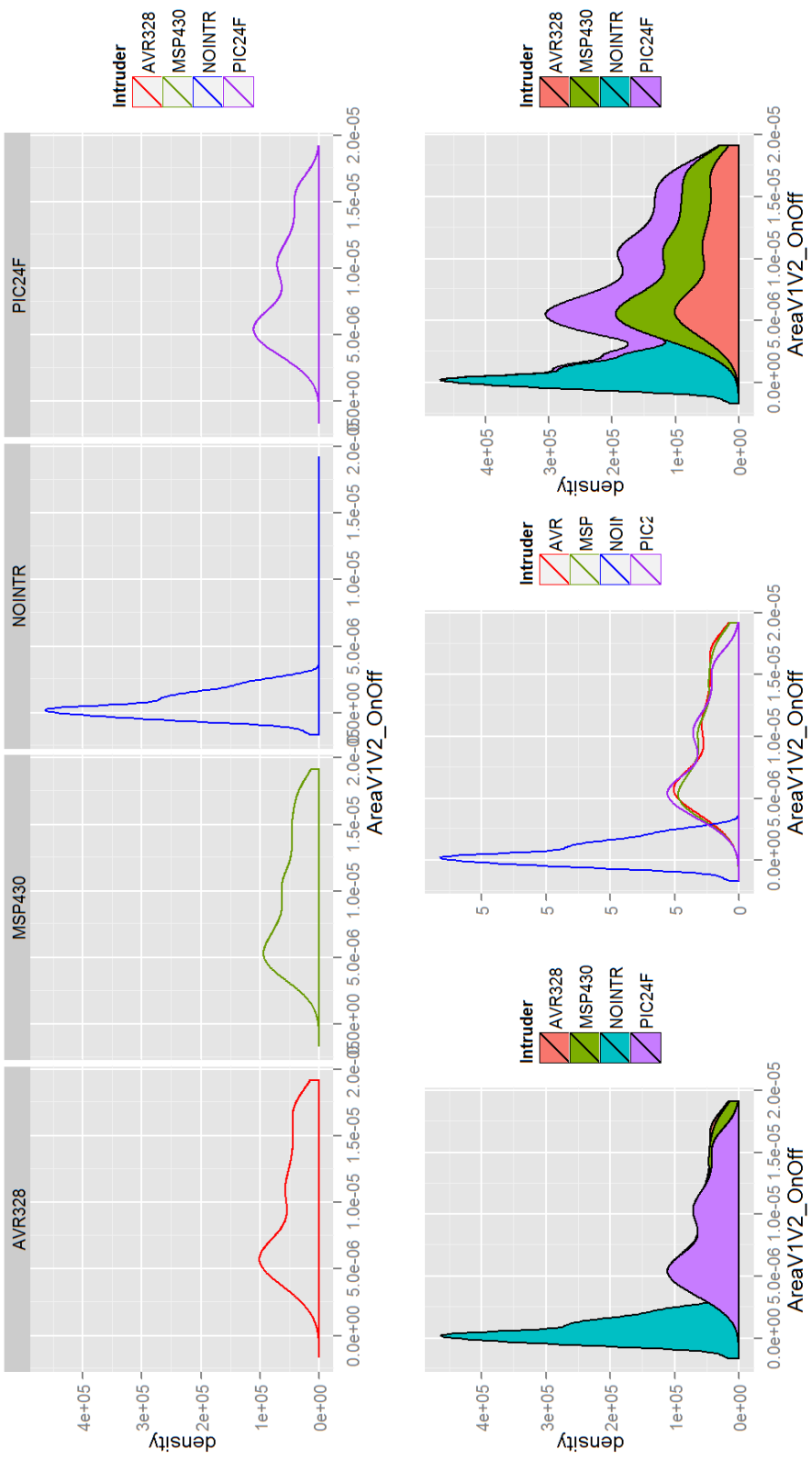


Figure A.18: Density plot of AreaV1V2_OnOff (complete dataset)

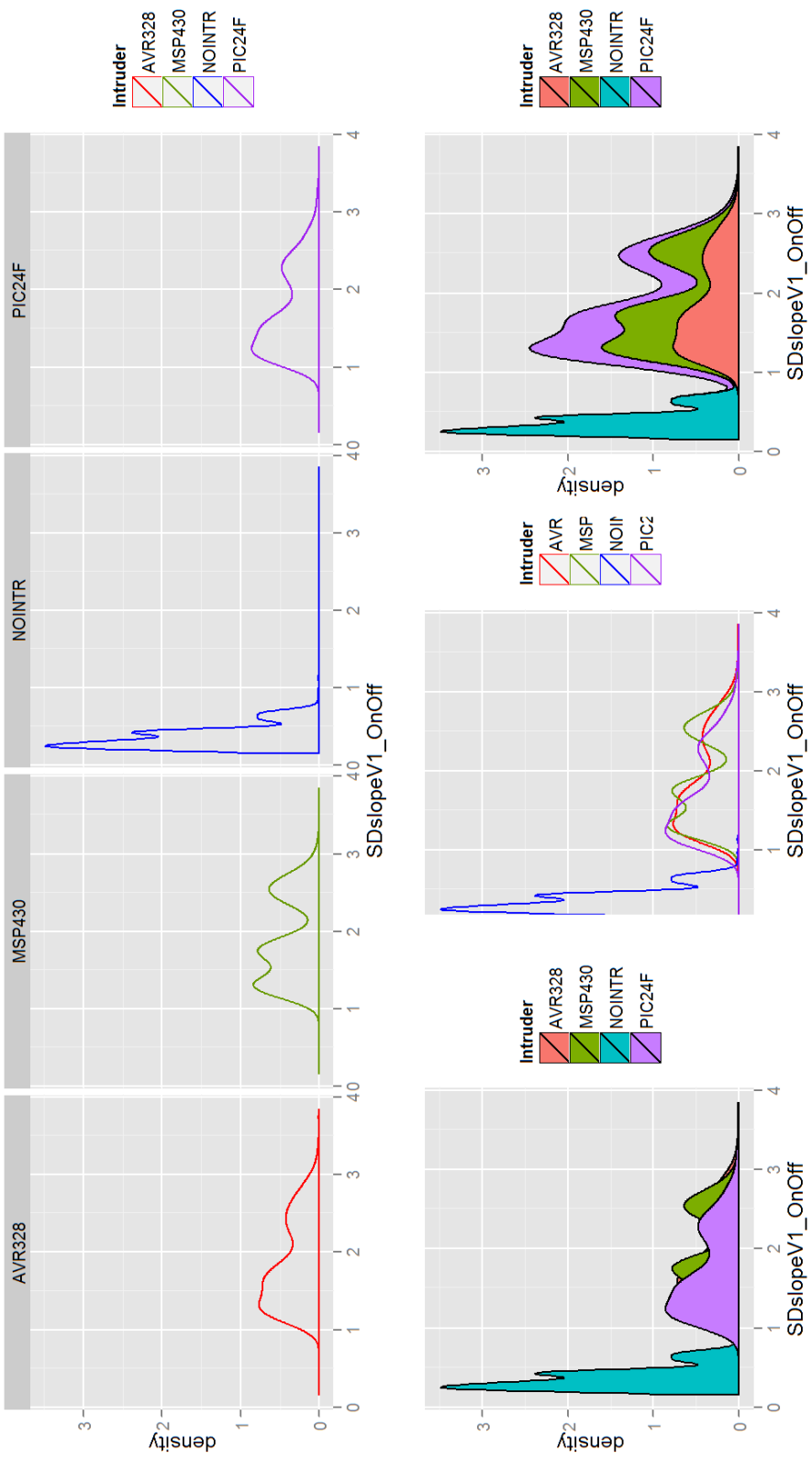


Figure A.19: Density plot of SDslopeV1_OnOff (complete dataset)

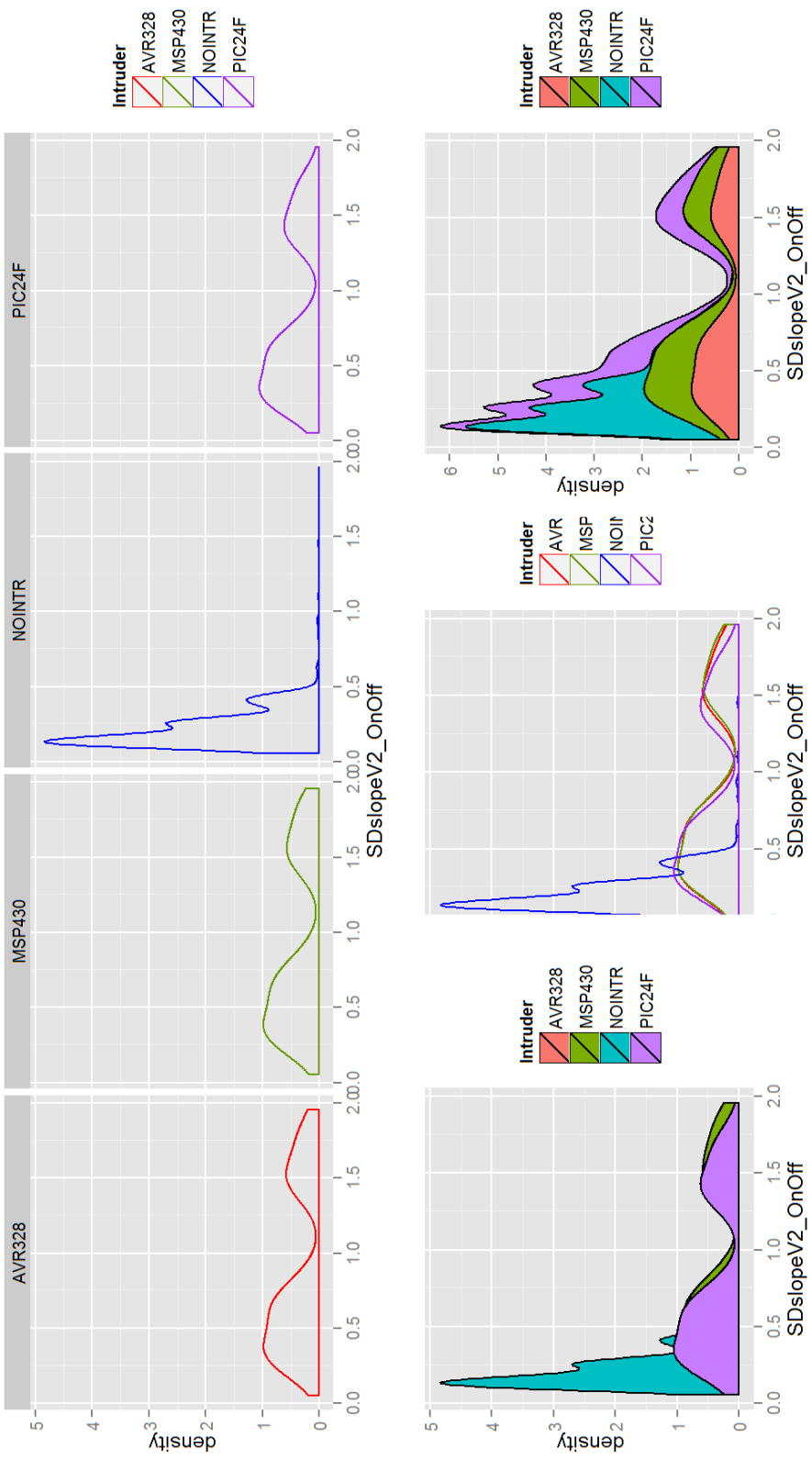


Figure A.20: Density plot of SDslopeV2_OnOff (complete dataset)

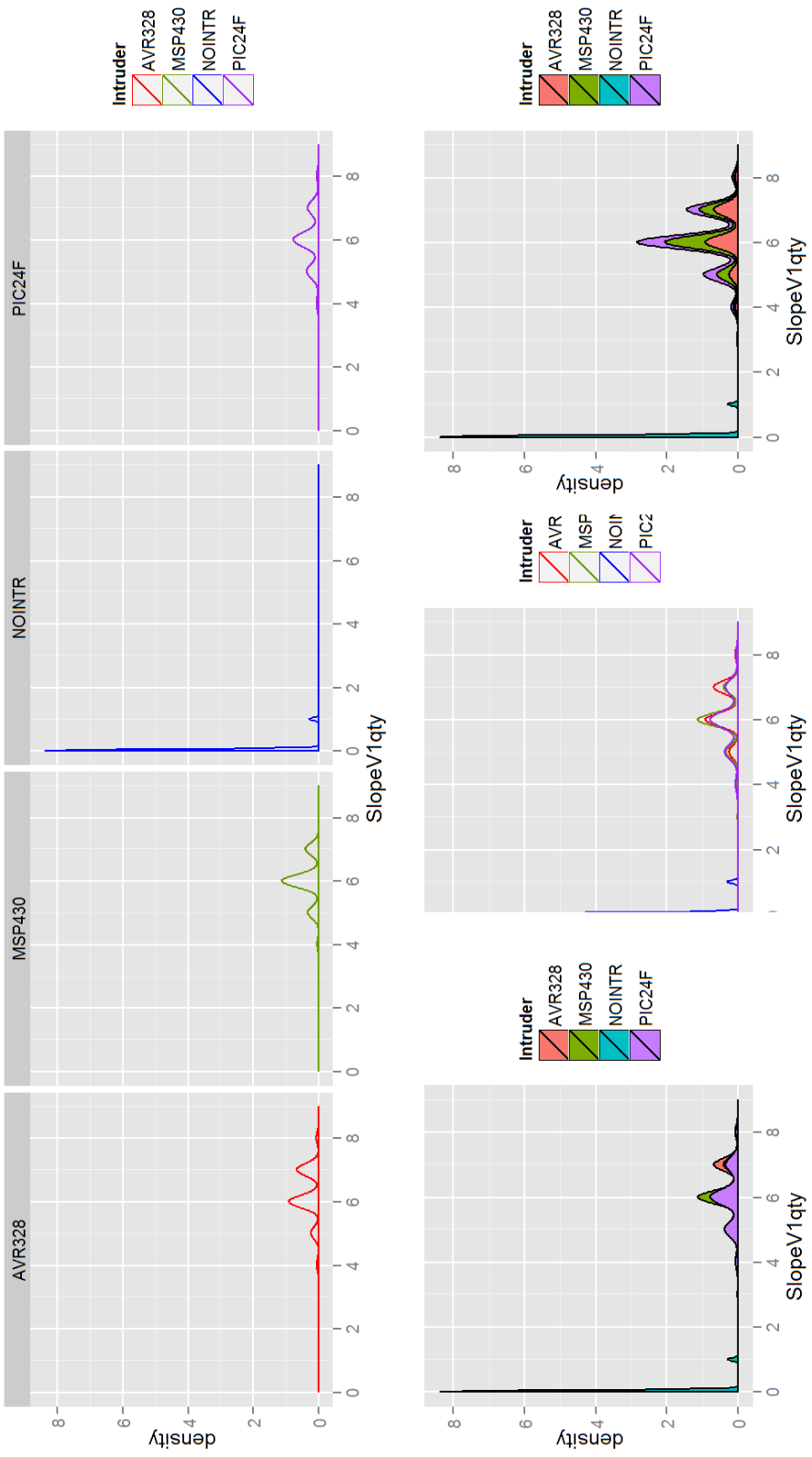


Figure A.21: Density plot of SlopeV1qty (complete dataset)

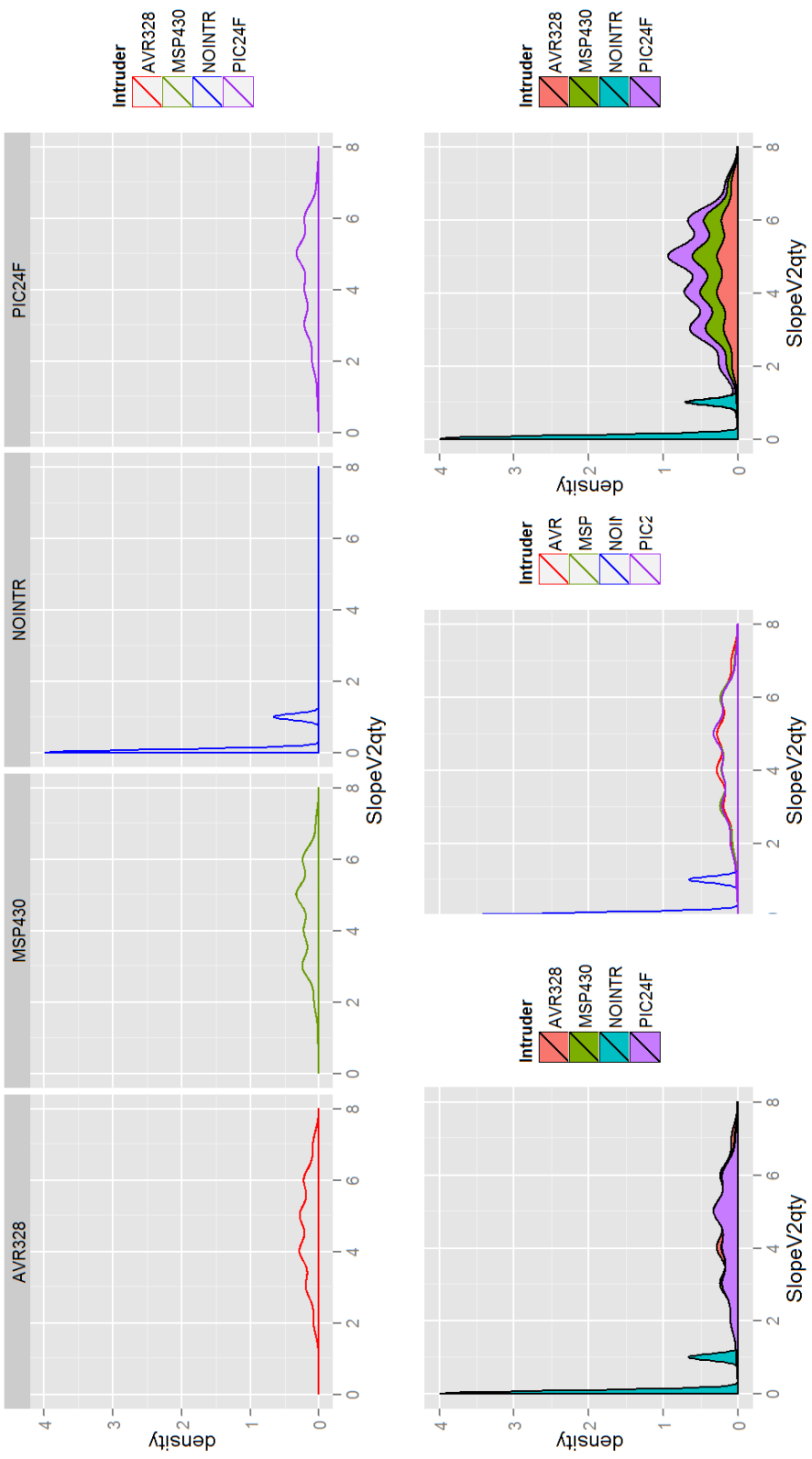


Figure A.22: Density plot of SlopeV2qty (complete dataset)

A.3 Density Plots – R≈49.9 KΩ and C≈100 pF

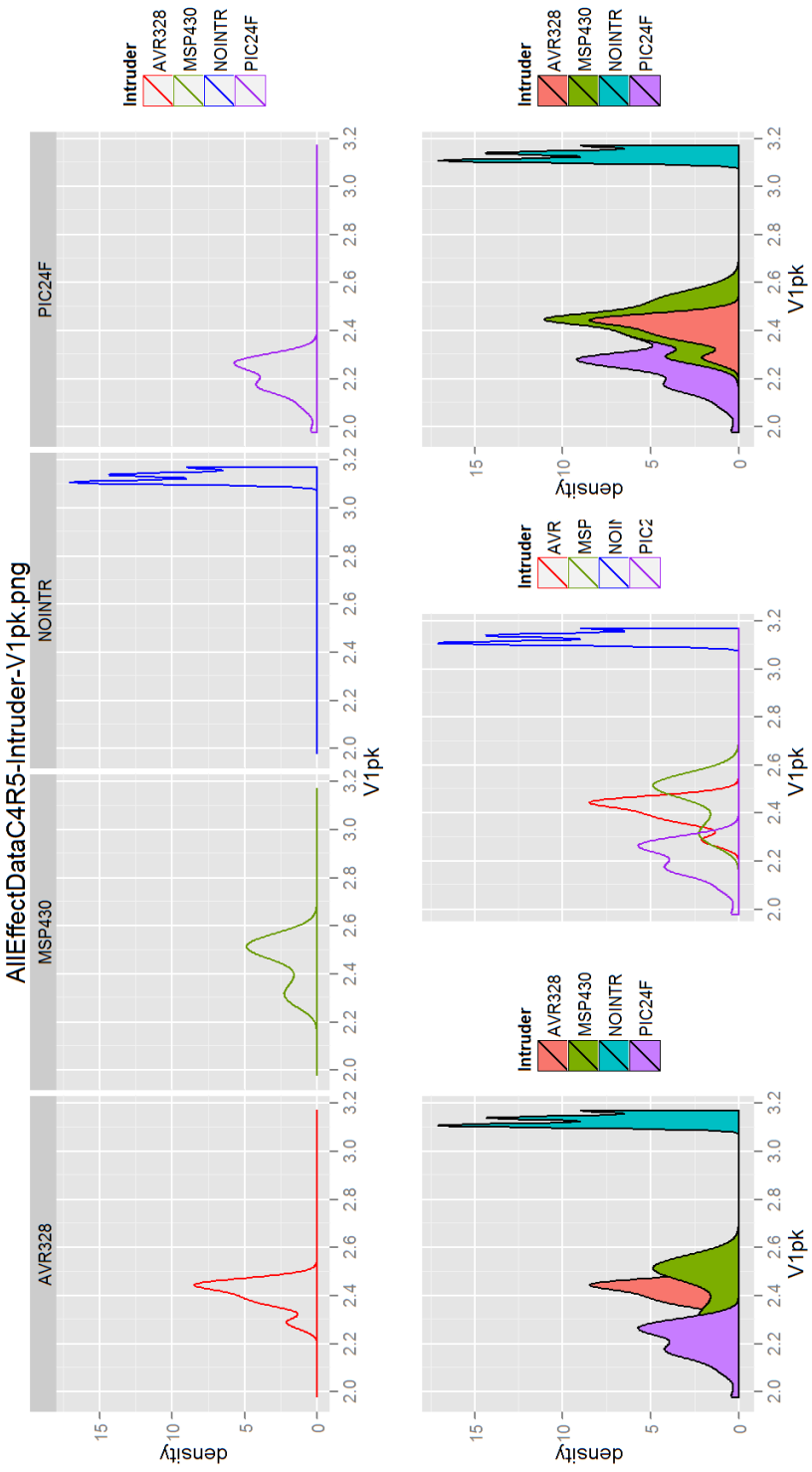


Figure A.23: Density plot of V1pk, R ≈ 49.9 kΩ and C ≈ 100 pF

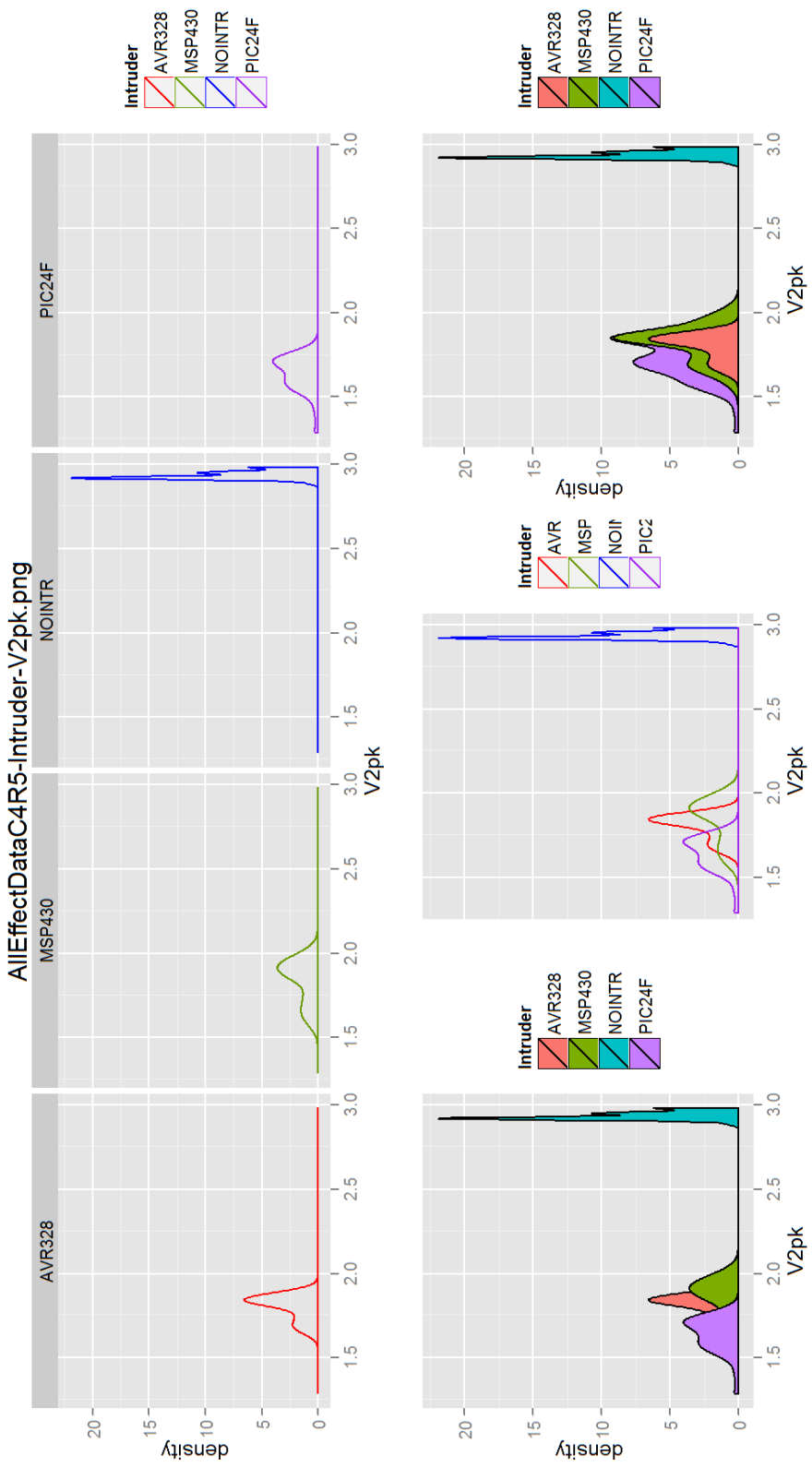


Figure A.24: Density plot of V2pk, R ≈ 49.9 kΩ and C ≈ 100 pF

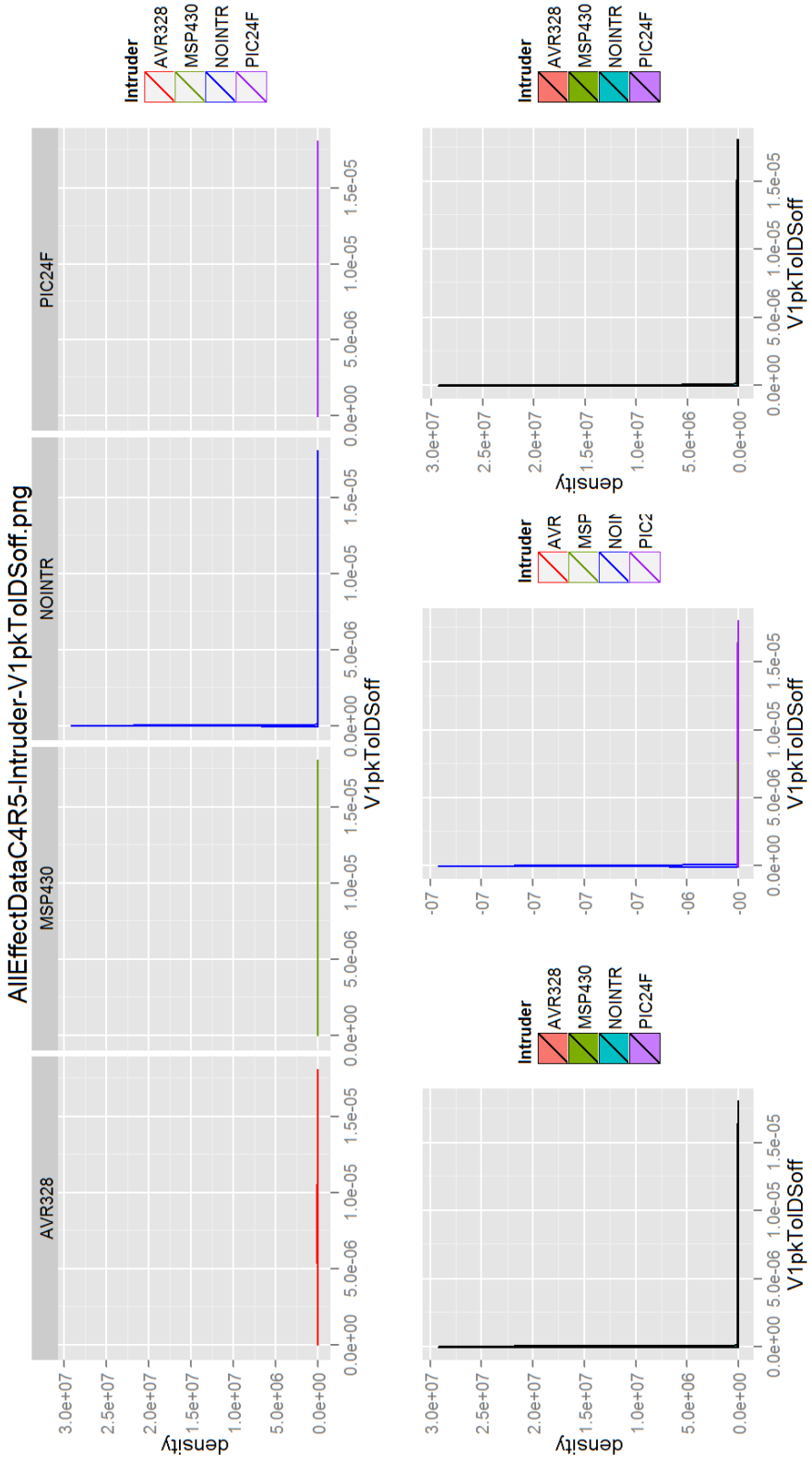


Figure A.25: Density plot of V1pkToIDSoff, R ≈ 49.9 kΩ and C ≈ 100 pF

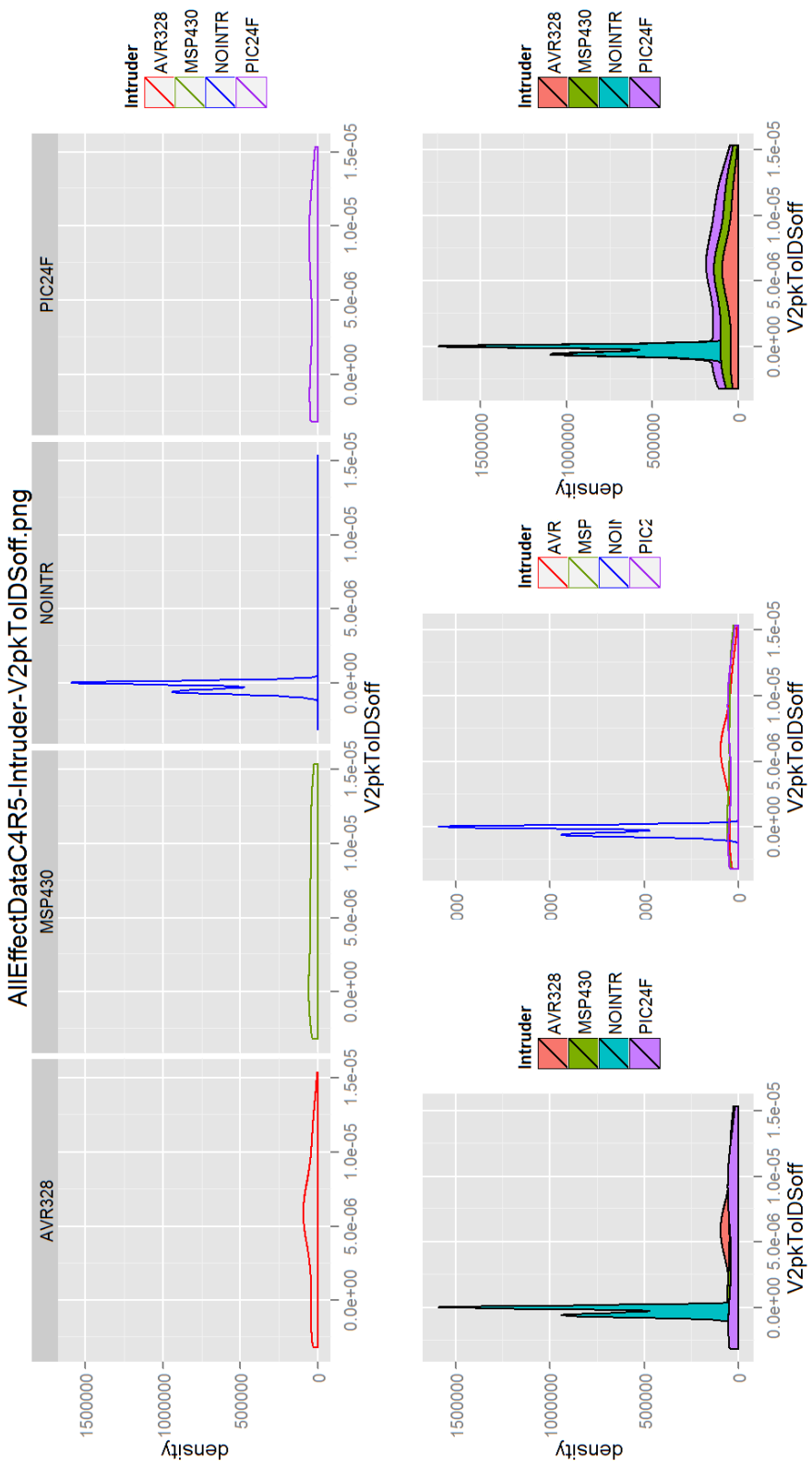


Figure A.26: Density plot of V2pkToIDSoff, R ≈ 49.9 kΩ and C ≈ 100 pF

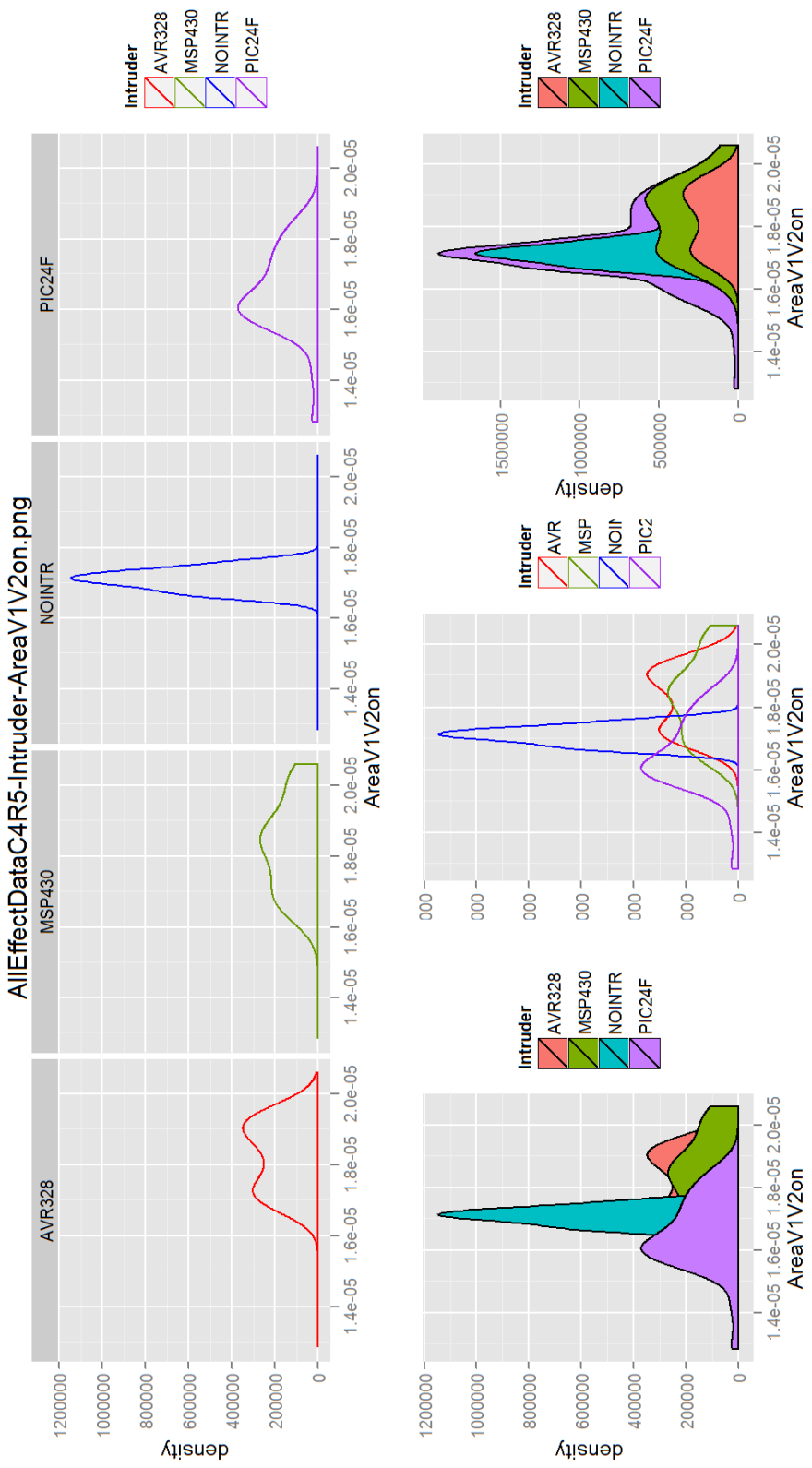


Figure A.27: Density plot of AreaV1V2on, R ≈ 49.9 kΩ and C ≈ 100 pF

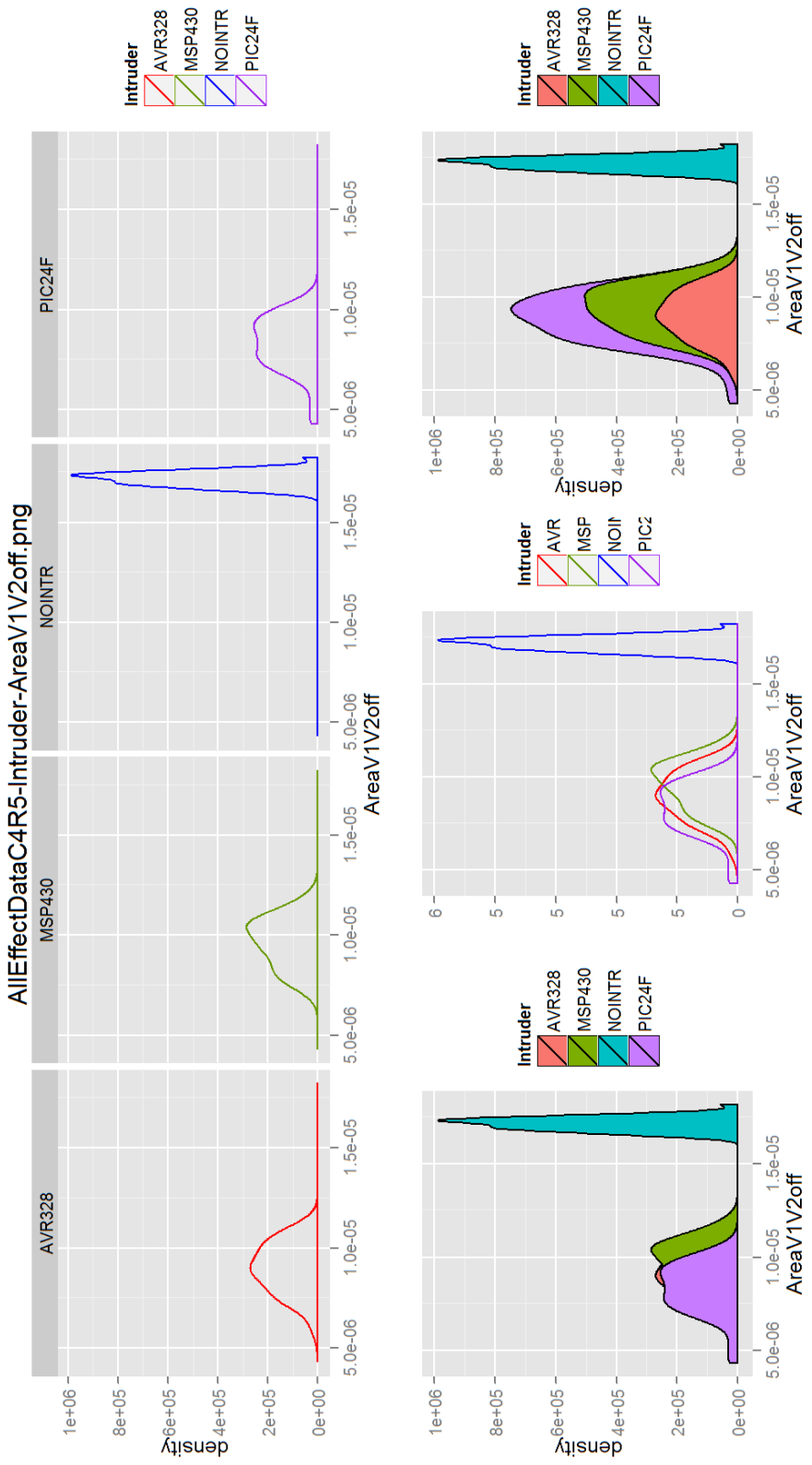


Figure A.28: Density plot of AreaV1V2off, R ≈ 49.9 kΩ and C ≈ 100 pF

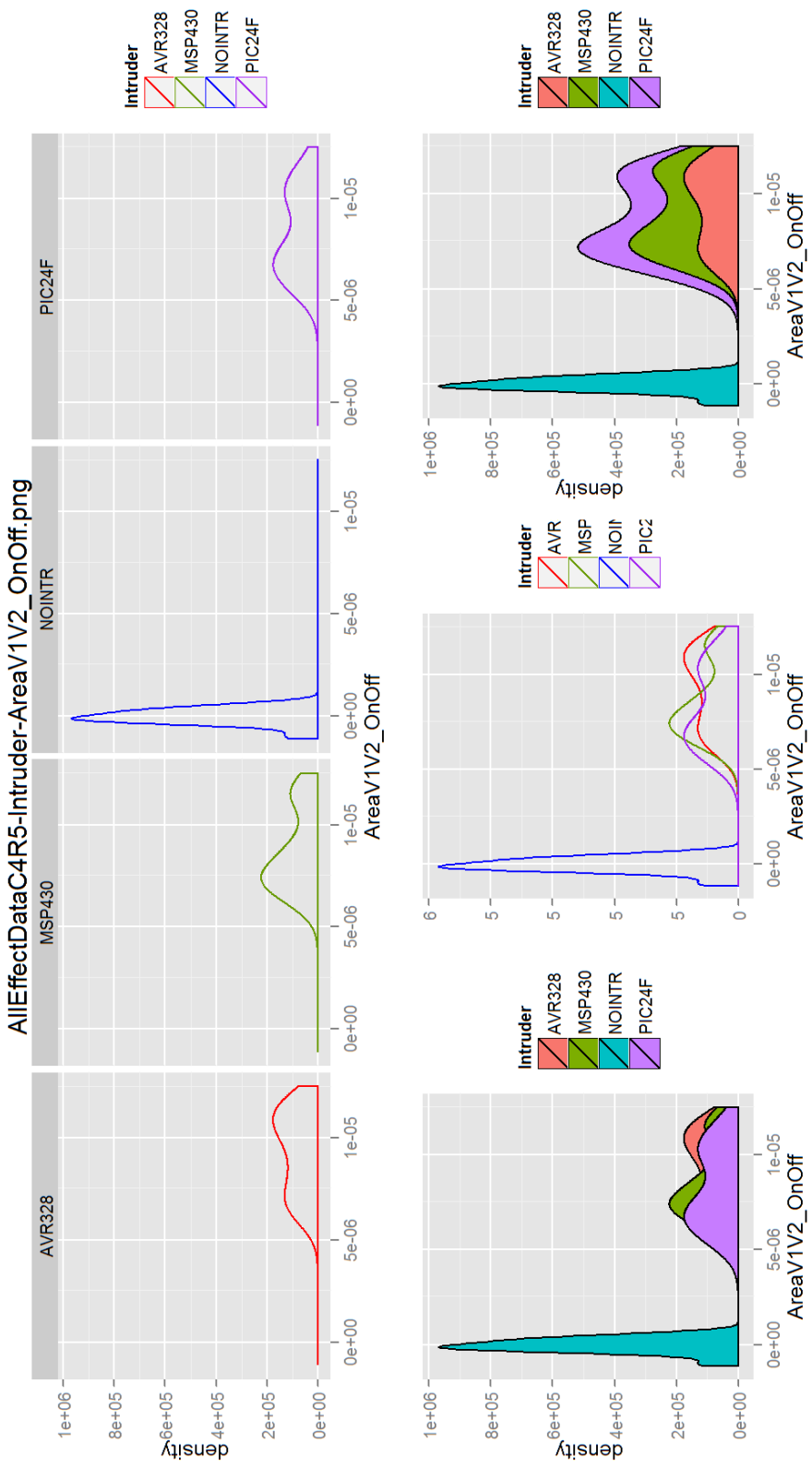


Figure A.29: Density plot of AreaV1V2_OnOff, R ≈ 49.9 kΩ and C ≈ 100 pF

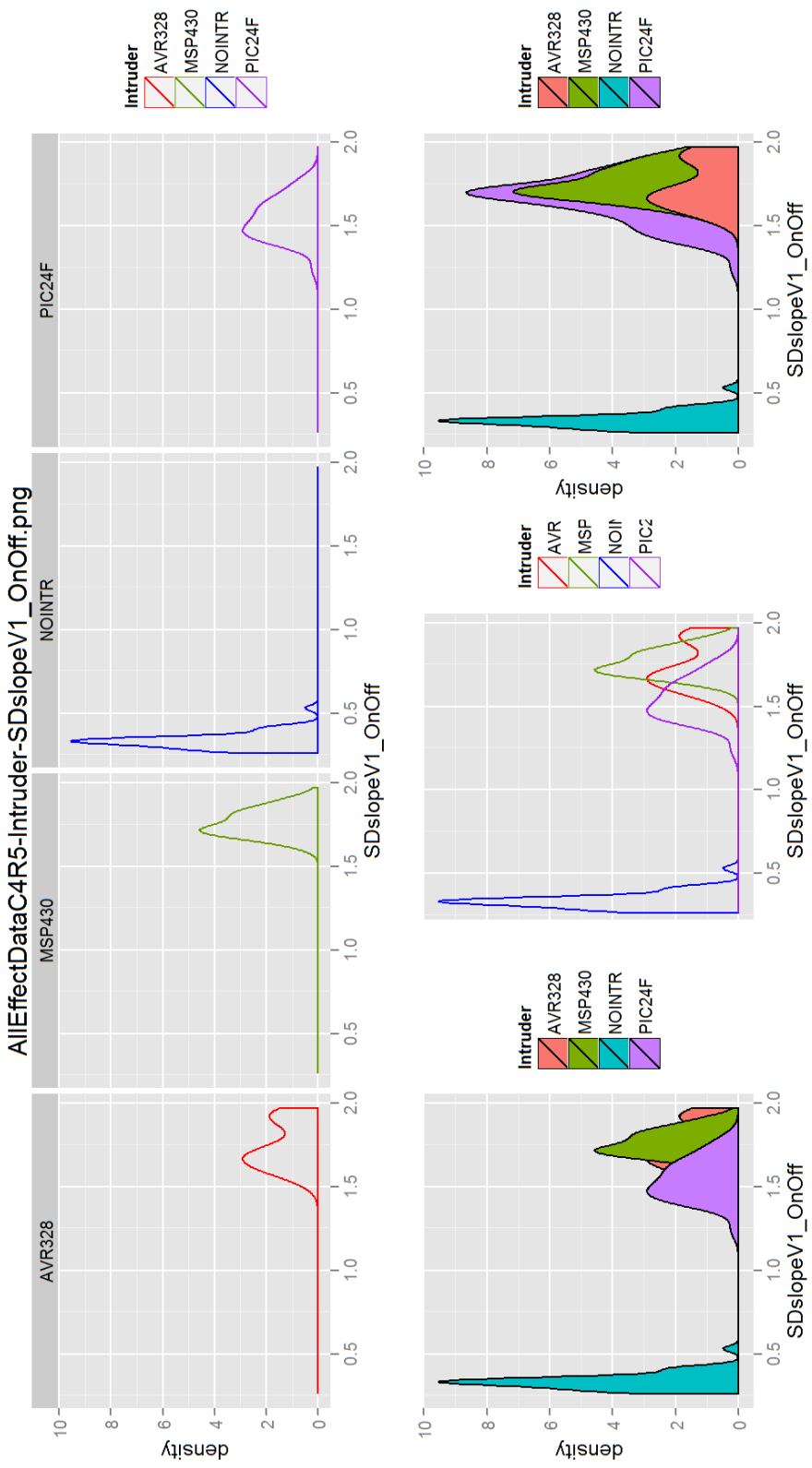


Figure A.30: Density plot of SDslopeV1_OnOff, R ≈ 49.9 kΩ and C ≈ 100 pF

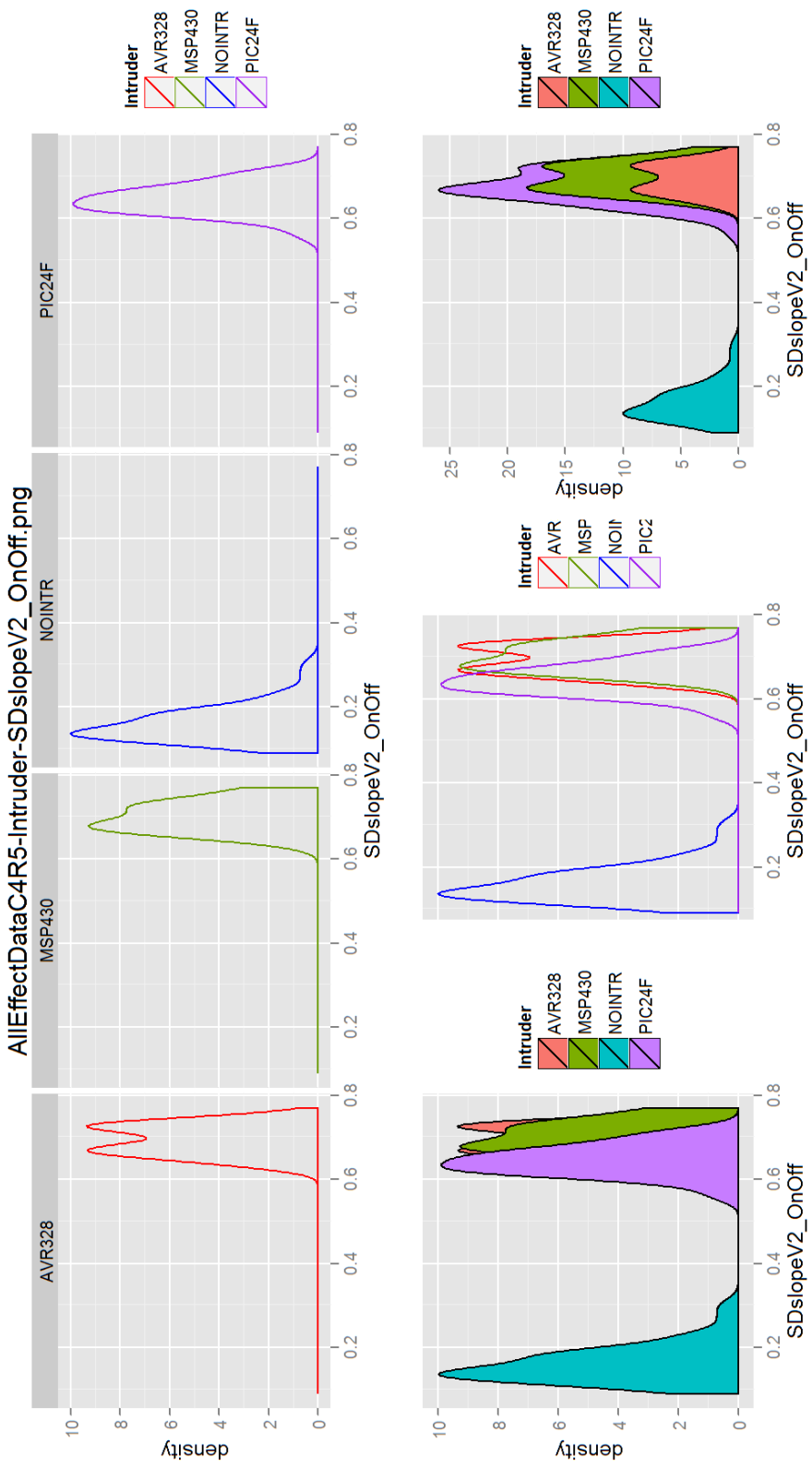


Figure A.31: Density plot of SDslopeV2_OnOff, R ≈ 49.9 kΩ and C ≈ 100 pF

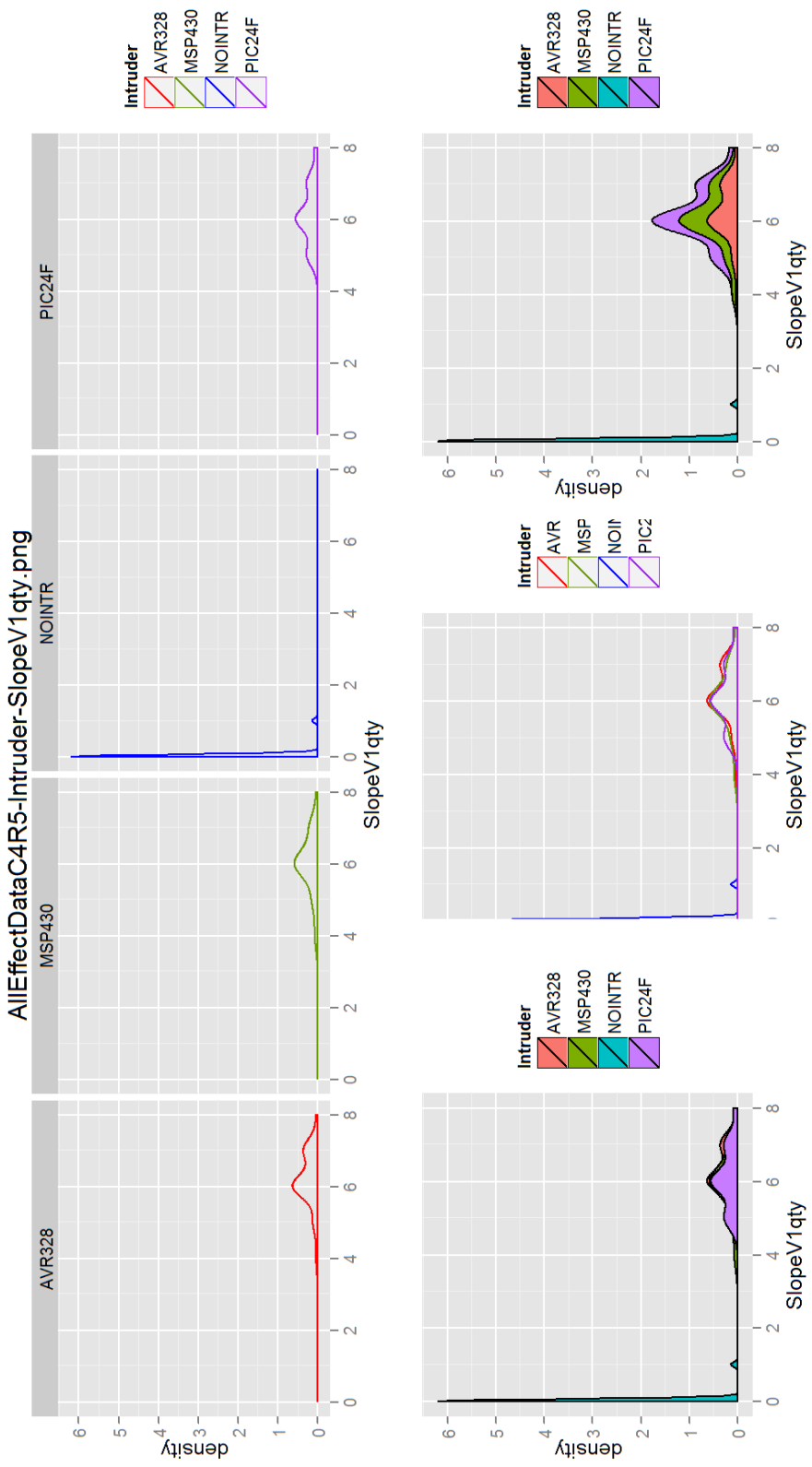


Figure A.32: Density plot of SlopeV1qty, R ≈ 49.9 kΩ and C ≈ 100 pF

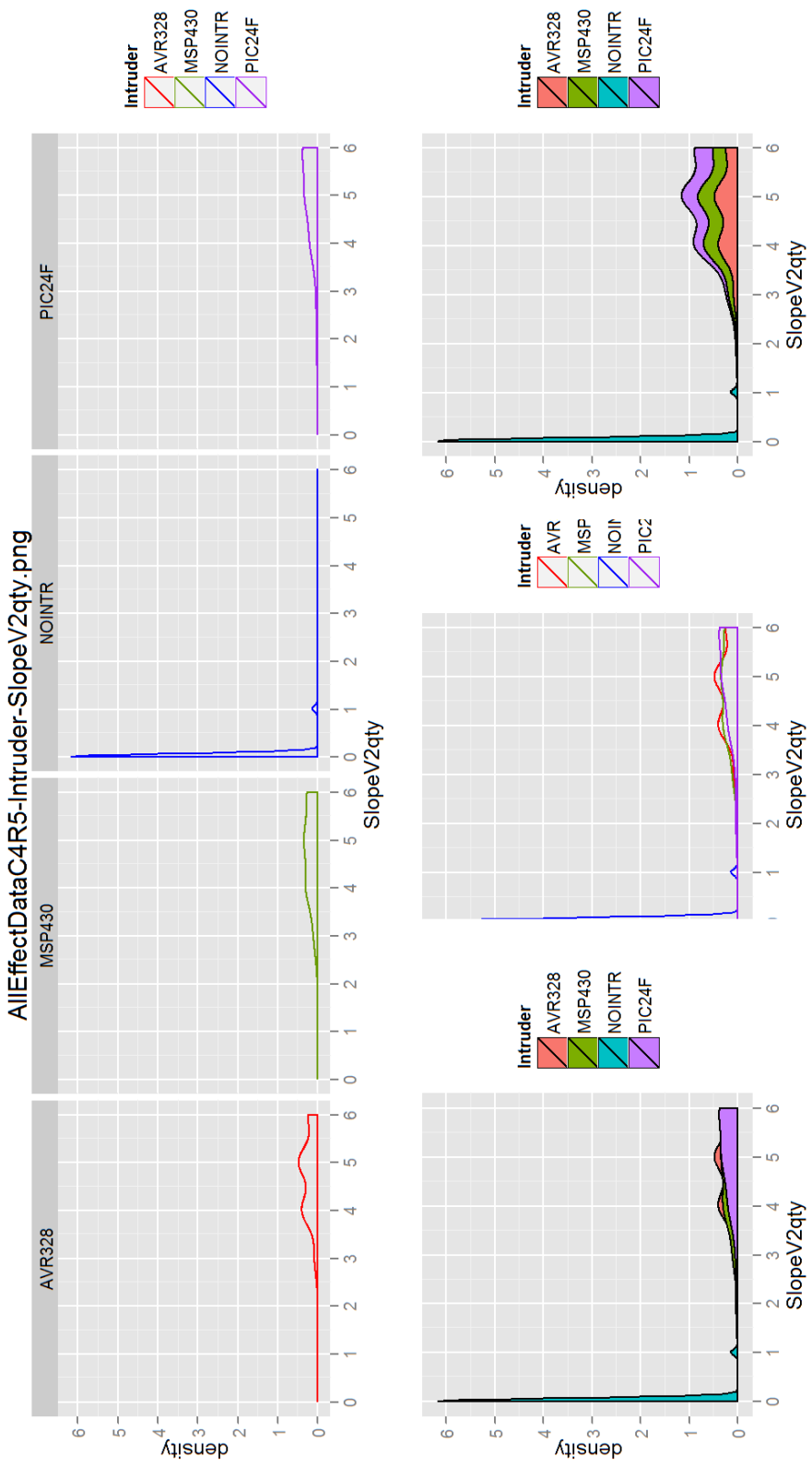


Figure A.33: Density plot of SlopeV2qty, R ≈ 49.9 kΩ and C ≈ 100 pF

APPENDIX B. Logistic Regression Model Analysis Data

B.1 Output of Model Fitting: fitF (full model)

```
## NOTE: Logistic Regression Model Fit Procedures:
 # Use the Pr(>Chisq) statistic as the primary decider for the model - record G^2,
AIC, df, Prediction Error (miss rate)
 # Start with full model using (Y ~ .) operator and all predictor columns.  Remove
the cols of unused predictors:
# Then iterate until all predictors Xn are significant at desired alpha level
(default is 0.05)

**************** NEW MODEL *****************
# weights:  88 (63 variable)
initial  value 2822.495319
iter  10 value 1482.482864
iter  20 value 893.597388
iter  30 value 617.695503
iter  40 value 524.688433
iter  50 value 495.273563
iter  60 value 476.281060
iter  70 value 467.766178
iter  80 value 463.867074
iter  90 value 461.544223
iter 100 value 460.753938
iter 110 value 460.426767
iter 120 value 459.835746
iter 130 value 459.356880
iter 140 value 459.304163
iter 150 value 459.231504
iter 160 value 459.151393
iter 170 value 459.008089
iter 180 value 458.959203
iter 190 value 458.902422
iter 200 value 458.838094
iter 210 value 458.753646
iter 220 value 458.747932
iter 230 value 458.733804
iter 240 value 458.733144
iter 250 value 458.731892
iter 250 value 458.731891
iter 250 value 458.731891
final  value 458.731891
converged
Call:
multinom(formula = Intruder ~ ., data = multidata, maxit = 1000)

Coefficients:
  (Intercept)  IDSmodule        Cap        Res       V1pk      V2pk
2   17.228683  0.2455088 -1.4102064  -6.204550   52.93219 -84.47125
3   -8.664804  0.3723091 -2.8621933   3.355777  108.74229 -97.09592
4   56.906778 -0.1046092 -0.1707526 -18.192985 -202.04147 128.07029
  V1pkToIDSoff V2pkToIDSoff   AreaV1on   AreaV2on AreaV1V2on AreaV1off
2    -2.082298    0.2256163  -21.50961  -12.28094 -4.3364467  13.14286
3     1.714197   -2.1324478 -127.64431  159.93367 -5.6640626 -78.33229
4    -1.156317   -1.7206410   74.44861 -220.28919  0.2205705  28.88744
  AreaV2off AreaV1V2off AreaV1V2_OnOff SDslopeV1_OnOff SDslopeV2_OnOff
2 -46.49984   0.7668306       4.033838        26.21070        7.806660
3  78.52817  -0.8420152      -6.585581        24.29444       -9.620816
```

```
4 -62.15008 -14.7087251      37.711567        21.81076       21.054773
  SlopeV1qty SlopeV2qty AreaV1_OnOff AreaV2_OnOff
2   26.41804   16.34087    -23.59511     -4.18498
3   27.03313   16.38136    -85.96294     11.90354
4   11.27217   23.58262     71.41970    -37.70302

Std. Errors:
  (Intercept) IDSmodule       Cap       Res     V1pk     V2pk V1pkToIDSoff
2    1.948528  8.492471 0.5605115 0.7492481 3.692008 4.088029    0.5040206
3    2.384443  8.492446 0.6831421 0.9146324 3.731552 4.390233    0.6736369
4    3.165210  8.492578 0.9951042 1.3279323 2.854521 3.423957    0.8508637
  V2pkToIDSoff AreaV1on AreaV2on AreaV1V2on AreaV1off AreaV2off AreaV1V2off
2    0.6239874 2.931345 6.649695   1.557534  1.691072  3.874027    2.353587
3    0.8473658 2.829076 6.318849   1.838505  2.068840  3.555370    2.584504
4    1.0684997 2.552702 2.556552   2.626746  2.876093  1.332372    3.264209
  AreaV1V2_OnOff SDslopeV1_OnOff SDslopeV2_OnOff SlopeV1qty SlopeV2qty
2       1.539388        1.497276        1.829499   1.110018  0.7826893
3       1.852843        1.993922        2.356258   1.354728  0.9808477
4       2.382634        2.622209        3.214743   2.019868  1.4016136
  AreaV1_OnOff AreaV2_OnOff
2     2.866894     2.965550
3     2.816503     3.814772
4     2.125523     4.226577

Value/SE (Wald statistics):
  (Intercept)   IDSmodule        Cap        Res      V1pk      V2pk
2    8.841897  0.02890899 -2.5159276  -8.281035  14.33696 -20.66307
3   -3.633891  0.04384003 -4.1897482   3.668990  29.14130 -22.11635
4   17.978831 -0.01231772 -0.1715927 -13.700236 -70.77947  37.40417
  V1pkToIDSoff V2pkToIDSoff   AreaV1on   AreaV2on  AreaV1V2on  AreaV1off
2    -4.131375    0.3615718  -7.337798  -1.846842 -2.78417379   7.771906
3     2.544690   -2.5165609 -45.118725  25.310572 -3.08079755 -37.862908
4    -1.358992   -1.6103336  29.164632 -86.166530  0.08397098  10.043985
  AreaV2off AreaV1V2off AreaV1V2_OnOff SDslopeV1_OnOff SDslopeV2_OnOff
2 -12.00297   0.3258135       2.620417       17.505589        4.267103
3  22.08720  -0.3257937      -3.554312       12.184246       -4.083092
4 -46.64618  -4.5060608      15.827677        8.317704        6.549441
  SlopeV1qty SlopeV2qty AreaV1_OnOff AreaV2_OnOff
2  23.799648   20.87785    -8.230198    -1.411198
3  19.954653   16.70123   -30.521161     3.120380
4   5.580648   16.82533    33.601003    -8.920464

Residual Deviance: 917.4638
AIC: 1031.464

### Confidence Intervals ###
, , 2
                      2.5 %      97.5 %
(Intercept)      13.4096384  21.0477274
IDSmodule       -16.3994293  16.8904468
Cap              -2.5087888  -0.3116240
Res              -7.6730492  -4.7360507
V1pk             45.6959831  60.1683900
V2pk            -92.4836388 -76.4588591
V1pkToIDSoff     -3.0701603  -1.0944360
V2pkToIDSoff     -0.9973765   1.4486090
AreaV1on        -27.2549442 -15.7642841
AreaV2on        -25.3141001   0.7522265
AreaV1V2on       -7.3891581  -1.2837352
AreaV1off         9.8284144  16.4572957
AreaV2off       -54.0927962 -38.9068880
AreaV1V2off      -3.8461159   5.3797771
```

```
AreaV1V2_OnOff    1.0166931   7.0509830
SDslopeV1_OnOff  23.2760921  29.1453062
SDslopeV2_OnOff   4.2209085  11.3924112
SlopeV1qty       24.2424480  28.5936397
SlopeV2qty       14.8068286  17.8749144
AreaV1_OnOff    -29.2141184 -17.9760988
AreaV2_OnOff     -9.9973519   1.6273920

, , 3
                       2.5 %       97.5 %
(Intercept)      -13.3382258   -3.9913826
IDSmodule        -16.2725789   17.0171971
Cap               -4.2011271   -1.5232594
Res                1.5631307    5.1484237
V1pk             101.4285779  116.0559925
V2pk            -105.7006196  -88.4912206
V1pkToIDSoff       0.3938928    3.0345009
V2pkToIDSoff      -3.7932543   -0.4716413
AreaV1on        -133.1892023 -122.0994269
AreaV2on         147.5489555  172.3183869
AreaV1V2on        -9.2674667   -2.0606584
AreaV1off        -82.3871413  -74.2774384
AreaV2off         71.5597776   85.4965722
AreaV1V2off       -5.9075501    4.2235196
AreaV1V2_OnOff   -10.2170866   -2.9540758
SDslopeV1_OnOff   20.3864221   28.2024529
SDslopeV2_OnOff  -14.2389967   -5.0026362
SlopeV1qty        24.3779150   29.6883524
SlopeV2qty        14.4589321   18.3037845
AreaV1_OnOff     -91.4831842  -80.4426955
AreaV2_OnOff       4.4267227   19.3803560

, , 4
                       2.5 %       97.5 %
(Intercept)       50.703080   63.1104755
IDSmodule        -16.749756   16.5405377
Cap               -2.121121    1.7796157
Res              -20.795684  -15.5902857
V1pk            -207.636232 -196.4467157
V2pk             121.359458  134.7811239
V1pkToIDSoff      -2.823979    0.5113456
V2pkToIDSoff      -3.814862    0.3735799
AreaV1on          69.445404   79.4518111
AreaV2on        -225.299937 -215.2784385
AreaV1V2on        -4.927758    5.3688986
AreaV1off         23.250401   34.5244804
AreaV2off        -64.761483  -59.5386793
AreaV1V2off      -21.106457   -8.3109927
AreaV1V2_OnOff    33.041690   42.3814449
SDslopeV1_OnOff   16.671321   26.9501896
SDslopeV2_OnOff   14.753992   27.3555547
SlopeV1qty         7.313303   15.2310409
SlopeV2qty        20.835504   26.3297284
AreaV1_OnOff      67.253751   75.5856471
AreaV2_OnOff     -45.986962  -29.4190869

************************
40-fold CROSS VALIDATION
CONFUSION MATRIX:
    predicted
true   1   2   3   4
   1 686   0   0   0
```

```
   2   0 331  97  22
   3   0  77 372   1
   4   0  19   0 431
SUM TOTAL of MATRIX = 2036
METRICS:
  class      sens      spec precision
1     1 1.0000000 1.0000000 1.0000000
2     2 0.7355556 0.9235060 0.7751756
3     3 0.8266667 0.9200988 0.7931770
4     4 0.9577778 0.9800866 0.9493392
Class[1:4] = NOINTR AVR328 MSP430 PIC24F

Overall Error Rate =  10.60904 %
Overall Accuracy =  89.39096 %
Mean Sensitivity (TPR) =  88 %
Mean False Positive Rate (1-SPC) =  4.407715 %
Mean Precision (PPV) =  87.9423 %
Khat Statistic =  0.8566197

*** anova(fit0, fitF, test=Chisq) ***
Likelihood ratio tests of Multinomial Models

Response: Intruder

Model
1
1
2 IDSmodule + Cap + Res + V1pk + V2pk + V1pkToIDSoff + V2pkToIDSoff + AreaV1on +
AreaV2on + AreaV1V2on + AreaV1off + AreaV2off + AreaV1V2off + AreaV1V2_OnOff +
SDslopeV1_OnOff + SDslopeV2_OnOff + SlopeV1qty + SlopeV2qty + AreaV1_OnOff +
AreaV2_OnOff
  Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
1      6105  5568.1864
2      6051   917.4638 1 vs 2    54 4650.723       0
Analysis of Deviance Table (Type II tests)

Response: Intruder
               LR Chisq Df Pr(>Chisq)
IDSmodule          83.43  3  < 2.2e-16 ***
Cap                 9.51  3  0.0231793 *
Res               138.70  3  < 2.2e-16 ***
V1pk              558.80  3  < 2.2e-16 ***
V2pk              192.86  3  < 2.2e-16 ***
V1pkToIDSoff       28.21  3  3.287e-06 ***
V2pkToIDSoff       11.58  3  0.0089829 **
AreaV1on            6.67  3  0.0832205 .
AreaV2on          137.87  3  < 2.2e-16 ***
AreaV1V2on          4.62  3  0.2017310
AreaV1off           6.69  3  0.0826288 .
AreaV2off          16.06  3  0.0011023 **
AreaV1V2off         3.54  3  0.3159060
AreaV1V2_OnOff      4.10  3  0.2505444
SDslopeV1_OnOff     6.08  3  0.1079836
SDslopeV2_OnOff    61.24  3  3.193e-13 ***
SlopeV1qty         34.67  3  1.432e-07 ***
SlopeV2qty         18.32  3  0.0003778 ***
AreaV1_OnOff        6.32  3  0.0969307 .
AreaV2_OnOff       24.28  3  2.188e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
IDS MODEL ADJUSTER - remove next predictor with highest Pr
[1] "AreaV1V2off"
```

B.2 Output of Model Fitting: fit10 (selected IDS model)

```
## NOTE: Logistic Regression Model Fit Procedures:
 # Use the Pr(>Chisq) statistic as the primary decider for the model - record G^2,
AIC, df, Prediction Error (miss rate)
 # Start with full model using (Y ~ .) operator and all predictor columns.  Remove
the cols of unused predictors:
 # Then iterate until all predictors Xn are significant at desired alpha level
(default is 0.05)

*************** NEW MODEL *****************
# weights:  52 (36 variable)
initial  value 2822.495319
iter  10 value 1576.496251
iter  20 value 904.969947
iter  30 value 642.723847
iter  40 value 557.285559
iter  50 value 514.797448
iter  60 value 497.292086
iter  70 value 494.270138
iter  80 value 493.027082
iter  90 value 492.461812
iter 100 value 492.313782
iter 110 value 492.151061
iter 120 value 492.049538
iter 130 value 491.910724
iter 140 value 491.842153
final  value 491.829383
converged
Call:
multinom(formula = Intruder ~ ., data = multidata[, -c(14, 20,
    11, 16, 8, 3, 12, 19, 21)], maxit = 1000)

Coefficients:
  (Intercept)   IDSmodule       Res       V1pk      V2pk V1pkToIDSoff
2   34.087537 -0.12732062 -13.44443   43.37896 -82.41595    0.9600343
3    6.783198 -0.03015854  -5.47104   93.14823 -87.65957    3.1304438
4   89.728319 -0.44152667 -28.33285 -181.48034 102.65795    2.1235833
    AreaV1on    AreaV2on AreaV2off AreaV1V2_OnOff SDslopeV2_OnOff SlopeV1qty
2  -85.69365    9.039451 -11.97166       2.893178        27.13420   21.78123
3 -190.57411  142.160762  18.20492     -16.135828         7.00139   23.88304
4  144.33058 -265.725035 -61.74259      34.519887        38.26685   10.60183

Std. Errors:
  (Intercept) IDSmodule       Res     V1pk     V2pk V1pkToIDSoff AreaV1on
2    1.968941  3.804507 0.6904555 4.790336 5.481514    0.3370906 5.449905
3    2.465970  3.804510 0.8514835 5.104165 6.346641    0.4231639 5.863612
4    3.269702  3.804587 1.2187797 7.572810 9.431636    0.5950709 5.200744
  AreaV2on AreaV2off AreaV1V2_OnOff SDslopeV2_OnOff SlopeV1qty
2 5.892581  2.457128       1.750669        1.607131   0.944608
3 6.137080  3.121915       2.164410        2.087313   1.192280
4 4.695777  4.114437       2.957325        2.785322   1.659504

Value/SE (Wald statistics):
  (Intercept)   IDSmodule        Res       V1pk      V2pk V1pkToIDSoff
2   17.312626 -0.03346573 -19.471820   9.055515 -15.03525     2.848001
3    2.750722 -0.00792705  -6.425304  18.249456 -13.81196     7.397710
4   27.442356 -0.11605114 -23.246902 -23.964728  10.88443     3.568622
   AreaV1on   AreaV2on  AreaV2off AreaV1V2_OnOff SDslopeV2_OnOff SlopeV1qty
2 -15.72388   1.534039  -4.872218       1.652613        16.88363  23.058491
3 -32.50115  23.164234   5.831331      -7.455069         3.35426  20.031409
4  27.75191 -56.588089 -15.006326      11.672672        13.73875   6.388553
```

```
Residual Deviance: 983.6588
AIC: 1055.659

### Confidence Intervals ###
, , 2
                      2.5 %     97.5 %
(Intercept)      30.2284835  37.946590
IDSmodule         -7.5840182   7.329377
Res              -14.7976940 -12.091158
V1pk              33.9900716  52.767843
V2pk             -93.1595242 -71.672386
V1pkToIDSoff       0.2993489   1.620720
AreaV1on         -96.3752629 -75.012027
AreaV2on          -2.5097962  20.588698
AreaV2off        -16.7875436  -7.155780
AreaV1V2_OnOff    -0.5380699   6.324427
SDslopeV2_OnOff   23.9842862  30.284123
SlopeV1qty        19.9298369  23.632632

, , 3
                      2.5 %      97.5 %
(Intercept)        1.949985   11.616412
IDSmodule         -7.486862    7.426545
Res               -7.139917   -3.802163
V1pk              83.144255  103.152214
V2pk            -100.098755  -75.220377
V1pkToIDSoff       2.301058    3.959830
AreaV1on        -202.066578 -179.081642
AreaV2on         130.132306  154.189218
AreaV2off         12.086079   24.323761
AreaV1V2_OnOff   -20.377994  -11.893662
SDslopeV2_OnOff    2.910332   11.092448
SlopeV1qty        21.546217   26.219868

, , 4
                       2.5 %      97.5 %
(Intercept)       83.3198212   96.136817
IDSmodule         -7.8983801    7.015327
Res              -30.7216162  -25.944088
V1pk            -196.3227728 -166.637902
V2pk              84.1722782  121.143612
V1pkToIDSoff       0.9572658    3.289901
AreaV1on         134.1373107  154.523851
AreaV2on        -274.9285885 -256.521482
AreaV2off        -69.8067348  -53.678438
AreaV1V2_OnOff    28.7236360   40.316138
SDslopeV2_OnOff   32.8077229   43.725985
SlopeV1qty         7.3492600   13.854395

************************
40-fold CROSS VALIDATION
CONFUSION MATRIX:
    predicted
true   1   2   3   4
   1 686   0   0   0
   2   0 330  97  23
   3   0  77 372   1
   4   0  23   0 427
SUM TOTAL of MATRIX = 2036
METRICS:
  class      sens      spec precision
```

```
1     1 1.0000000 1.0000000 1.0000000
2     2 0.7333333 0.9203822 0.7674419
3     3 0.8266667 0.9200988 0.7931770
4     4 0.9488889 0.9792925 0.9467849
Class[1:4] = NOINTR AVR328 MSP430 PIC24F

Overall Error Rate =  10.85462 %
Overall Accuracy =  89.14538 %
Mean Sensitivity (TPR) =  87.72222 %
Mean False Positive Rate (1-SPC) =  4.505662 %
Mean Precision (PPV) =  87.68509 %
Khat Statistic =  0.8533007

****** anova(fit0, fit10, test=Chisq) ******
Likelihood ratio tests of Multinomial Models

Response: Intruder

Model
1
1
2 IDSmodule + Res + V1pk + V2pk + V1pkToIDSoff + AreaV1on + AreaV2on + AreaV2off +
AreaV1V2_OnOff + SDslopeV2_OnOff + SlopeV1qty
  Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
1      6105  5568.1864
2      6072   983.6588 1 vs 2    33 4584.528       0

****** anova(fitF, fit10, test=Chisq) ******
Likelihood ratio tests of Multinomial Models

Response: Intruder

Model
1
IDSmodule + Res + V1pk + V2pk + V1pkToIDSoff + AreaV1on + AreaV2on + AreaV2off +
AreaV1V2_OnOff + SDslopeV2_OnOff + SlopeV1qty
2 IDSmodule + Cap + Res + V1pk + V2pk + V1pkToIDSoff + V2pkToIDSoff + AreaV1on +
AreaV2on + AreaV1V2on + AreaV1off + AreaV2off + AreaV1V2off + AreaV1V2_OnOff +
SDslopeV1_OnOff + SDslopeV2_OnOff + SlopeV1qty + SlopeV2qty + AreaV1_OnOff +
AreaV2_OnOff
  Resid. df Resid. Dev   Test    Df LR stat.      Pr(Chi)
1      6072   983.6588
2      6051   917.4638 1 vs 2    21 66.19498 1.412423e-06
Analysis of Deviance Table (Type II tests)

Response: Intruder
               LR Chisq Df Pr(>Chisq)
IDSmodule         163.51  3  < 2.2e-16 ***
Res               170.02  3  < 2.2e-16 ***
V1pk              590.80  3  < 2.2e-16 ***
V2pk              183.71  3  < 2.2e-16 ***
V1pkToIDSoff       24.11  3  2.368e-05 ***
AreaV1on          288.87  3  < 2.2e-16 ***
AreaV2on          410.74  3  < 2.2e-16 ***
AreaV2off         160.33  3  < 2.2e-16 ***
AreaV1V2_OnOff    106.18  3  < 2.2e-16 ***
SDslopeV2_OnOff    87.69  3  < 2.2e-16 ***
SlopeV1qty         24.75  3  1.746e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
IDS MODEL ADJUSTER - remove next predictor with highest Pr
[1] "V1pkToIDSoff"
```

B.3 Output of Model Fitting: fit14 (lower performance)

```
## NOTE: Logistic Regression Model Fit Procedures:
 # Use the Pr(>Chisq) statistic as the primary decider for the model - record G^2,
AIC, df, Prediction Error (miss rate)
 # Start with full model using (Y ~ .) operator and all predictor columns.  Remove
the cols of unused predictors:
 # Then iterate until all predictors Xn are significant at desired alpha level
(default is 0.05)

*************** NEW MODEL *****************
# weights:  36 (24 variable)
initial  value 2822.495319
iter  10 value 1566.930487
iter  20 value 769.792031
iter  30 value 651.190219
iter  40 value 599.665521
iter  50 value 593.281778
iter  60 value 591.292119
iter  70 value 590.371570
iter  80 value 590.083992
iter  90 value 589.998414
final  value 589.957975
converged
Call:
multinom(formula = Intruder ~ ., data = multidata[, -c(14, 20,
    11, 16, 8, 3, 12, 19, 21, 7, 18, 17, 15)], maxit = 1000)

Coefficients:
  (Intercept)    IDSmodule        Res       V1pk      V2pk   AreaV1on
2    50.35327 -0.064367497 -13.253101   51.24005 -51.97816  -63.46759
3    18.36721 -0.003318551  -7.414385   89.77363 -85.59076 -128.05640
4   103.10139 -0.329255481 -26.616632 -121.86835 106.10120   58.87664
    AreaV2on  AreaV2off
2  -15.79917 -36.315179
3   81.66899   4.573242
4 -167.79143 -87.411641

Std. Errors:
  (Intercept) IDSmodule       Res     V1pk     V2pk AreaV1on AreaV2on AreaV2off
2    1.244644 0.5783854 0.5939403 3.505296 3.298434 3.757072 4.281145  1.622692
3    1.521330 0.5783274 0.7067899 3.520704 3.320817 3.908536 4.578383  1.963548
4    1.651375 0.5787623 1.0070353 5.043488 4.704208 3.926632 3.885620  2.103449

Value/SE (Wald statistics):
  (Intercept)    IDSmodule       Res      V1pk      V2pk  AreaV1on   AreaV2on
2    40.45596 -0.111288253 -22.31386  14.61790 -15.75844 -16.89283  -3.690407
3    12.07312 -0.005738188 -10.49022  25.49877 -25.77401 -32.76326  17.837954
4    62.43365 -0.568895884 -26.43069 -24.16351  22.55453  14.99418 -43.182665
  AreaV2off
2 -22.37958
3   2.32907
4 -41.55634

Residual Deviance: 1179.916
AIC: 1227.916

### Confidence Intervals ###
, , 2
                 2.5 %     97.5 %
(Intercept)  47.913813  52.792728
IDSmodule    -1.197982   1.069247
```

```
Res         -14.417203 -12.089000
V1pk         44.369797  58.110303
V2pk        -58.442969 -45.513346
AreaV1on    -70.831311 -56.103860
AreaV2on    -24.190058  -7.408279
AreaV2off   -39.495598 -33.134760

, , 3
                   2.5 %      97.5 %
(Intercept)   15.3854585   21.348964
IDSmodule     -1.1368194    1.130182
Res           -8.7996678   -6.029102
V1pk          82.8731799   96.674086
V2pk         -92.0994377  -79.082076
AreaV1on    -135.7169850 -120.395805
AreaV2on      72.6955247   90.642458
AreaV2off      0.7247584    8.421726

, , 4
                   2.5 %       97.5 %
(Intercept)   99.864755  106.3380275
IDSmodule     -1.463609    0.8050978
Res          -28.590385  -24.6428792
V1pk        -131.753399 -111.9832907
V2pk          96.881125  115.3212813
AreaV1on      51.180586   66.5727011
AreaV2on    -175.407104 -160.1757536
AreaV2off    -91.534325  -83.2889572

************************
40-fold CROSS VALIDATION
CONFUSION MATRIX:
    predicted
true   1   2   3   4
   1 686   0   0   0
   2   0 300 120  30
   3   0  93 355   2
   4   0  29   0 421
SUM TOTAL of MATRIX = 2036
METRICS:
  class      sens      spec precision
1     1 1.0000000 1.0000000 1.0000000
2     2 0.6666667 0.9051322 0.7109005
3     3 0.7888889 0.9025183 0.7473684
4     4 0.9355556 0.9725322 0.9293598
Class[1:4] = NOINTR AVR328 MSP430 PIC24F

Overall Error Rate =  13.45776 %
Overall Accuracy =  86.54224 %
Mean Sensitivity (TPR) =  84.77778 %
Mean False Positive Rate (1-SPC) =  5.495434 %
Mean Precision (PPV) =  84.69072 %
Khat Statistic =  0.8181195

****** anova(fit0, fit14, test=Chisq) ******
Likelihood ratio tests of Multinomial Models

Response: Intruder
                                                                  Model Resid. df
1                                                                     1      6105
2 IDSmodule + Res + V1pk + V2pk + AreaV1on + AreaV2on + AreaV2off      6084
  Resid. Dev   Test    Df LR stat. Pr(Chi)
```

```
1   5568.186
2   1179.916 1 vs 2    21  4388.27       0

****** anova(fitF, fit14, test=Chisq) ******
Likelihood ratio tests of Multinomial Models

Response: Intruder

Model
1
IDSmodule + Res + V1pk + V2pk + AreaV1on + AreaV2on + AreaV2off
2 IDSmodule + Cap + Res + V1pk + V2pk + V1pkToIDSoff + V2pkToIDSoff + AreaV1on +
AreaV2on + AreaV1V2on + AreaV1off + AreaV2off + AreaV1V2off + AreaV1V2_OnOff +
SDslopeV1_OnOff + SDslopeV2_OnOff + SlopeV1qty + SlopeV2qty + AreaV1_OnOff +
AreaV2_OnOff
  Resid. df Resid. Dev   Test    Df LR stat. Pr(Chi)
1      6084  1179.9159
2      6051   917.4638 1 vs 2    33 262.4522       0
Analysis of Deviance Table (Type II tests)

Response: Intruder
          LR Chisq Df Pr(>Chisq)
IDSmodule   146.59  3  < 2.2e-16 ***
Res         156.68  3  < 2.2e-16 ***
V1pk        782.75  3  < 2.2e-16 ***
V2pk        612.47  3  < 2.2e-16 ***
AreaV1on    221.81  3  < 2.2e-16 ***
AreaV2on    357.89  3  < 2.2e-16 ***
AreaV2off   553.05  3  < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
IDS MODEL ADJUSTER - remove next predictor with highest Pr
[1] "IDSmodule"
```

B.4 ROC, Sensitivity, Precision and Specificity Curves

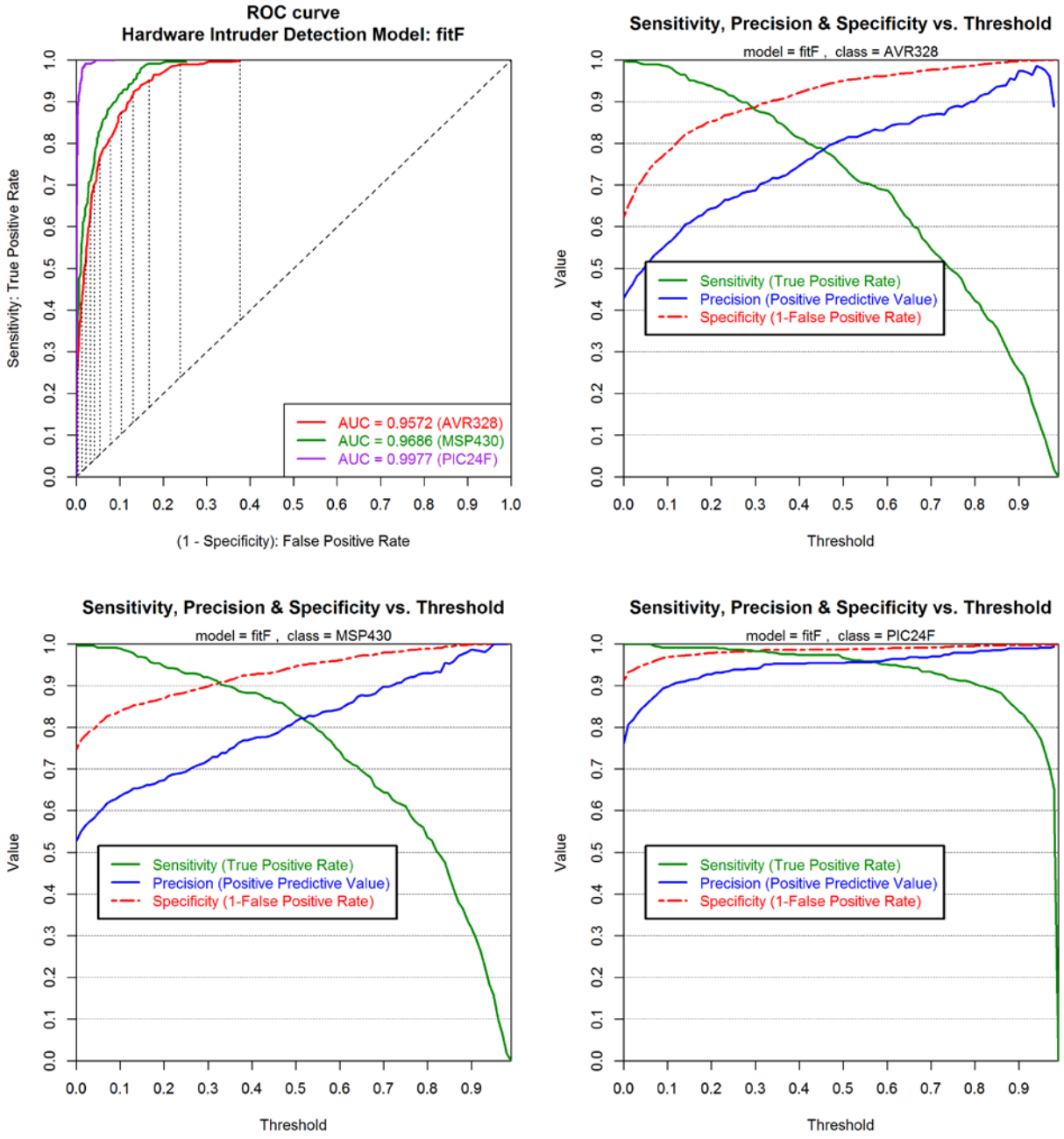


Figure B.1: ROC, Sensitivity, Precision and Specificity for model fitF (full model)

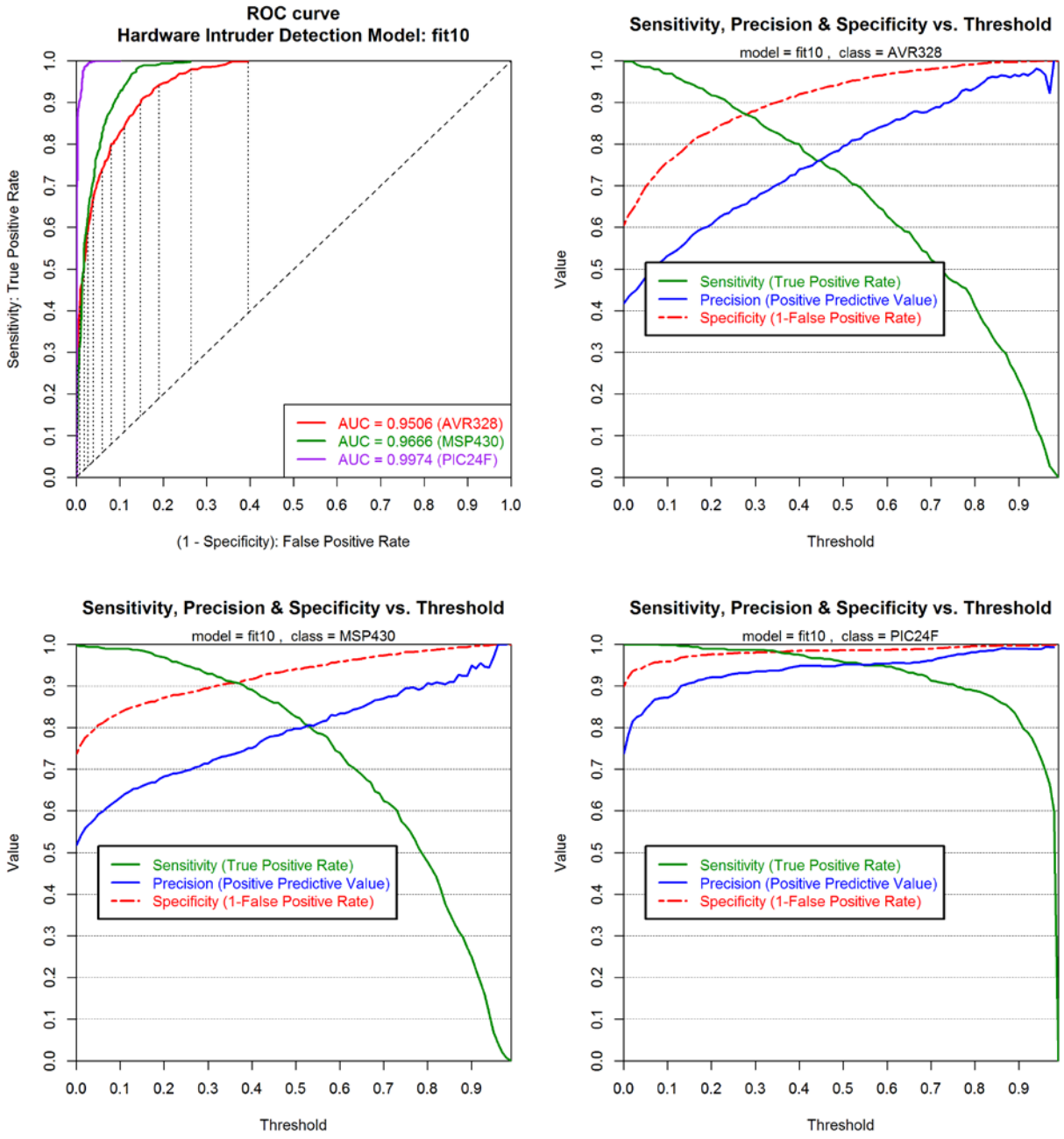


Figure B.2: ROC, Sensitivity, Precision and Specificity for model fit10 (selected IDS model)

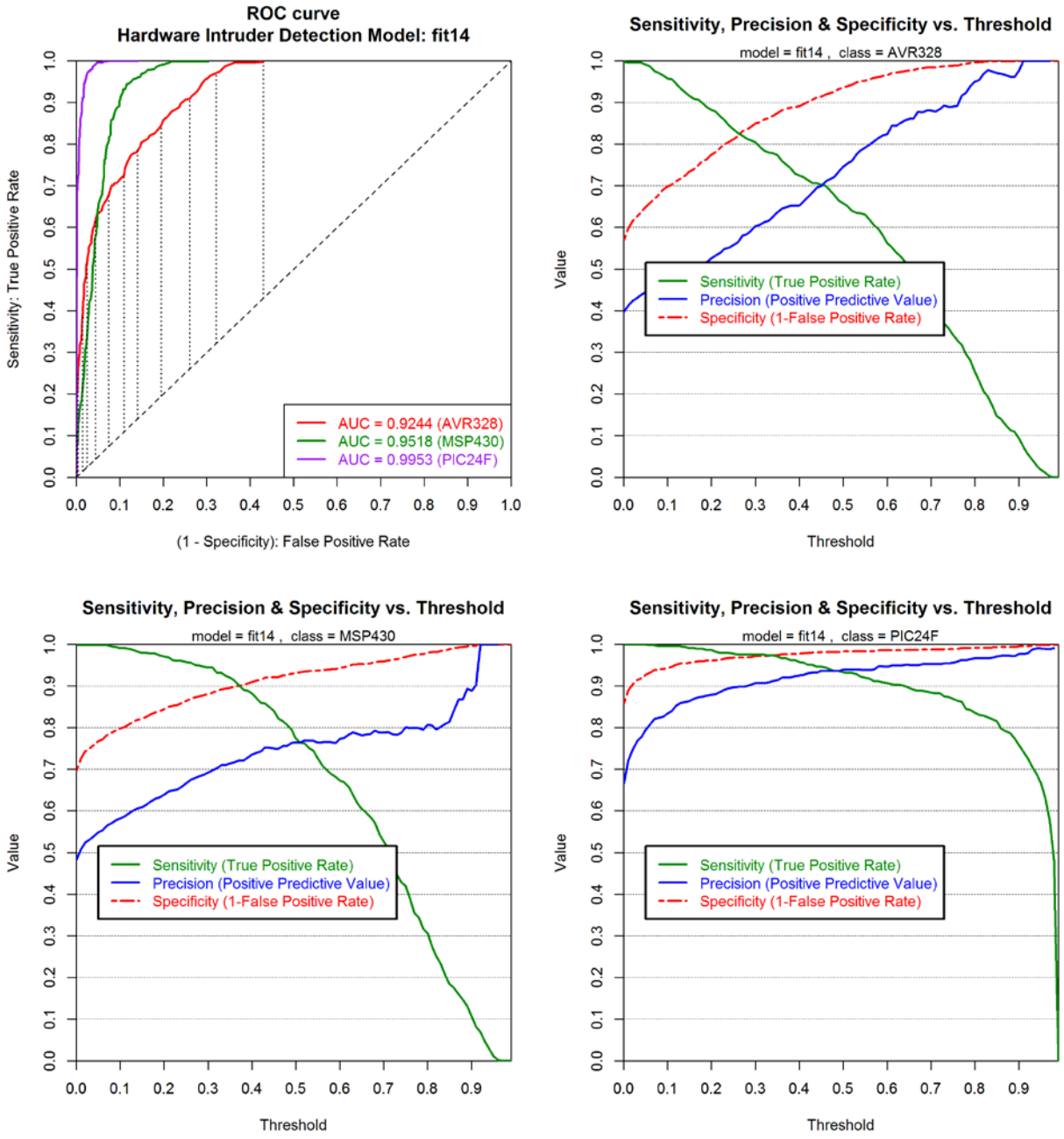


Figure B.3: ROC, Sensitivity, Precision and Specificity for model fit14 (lower performance)

APPENDIX C. Randomly Selected Components for IDS Circuit

C.1 Capacitors Randomly Selected for IDS Modules

Code	Value	Mfg Part No	Description
A	C10pF	VJ0805A100KXACW1BC	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 0805 10pF 50volts C0G 10%
B	C10pF	C0805C100M5GACTU	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 50volts 10pF C0G 20%
C	C10pF	08055C100KAT2A	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 50V 10pF 10%
D	C20pF	CC0805JRNPO9BN200	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 20pF 50V NPO 5%
E	C20pF	08051A200KAT2A	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 100V 2pF 10%
F	C20pF	GRM2195C2A200JZ01D	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 0805 20pF 100volts C0G 5%
G	C39pF	CC0805JRNPO9BN390	CAP CER 39PF 50V 5% NP0 0805
H	C39pF	08051A390JAT2A	CAP CER 39PF 250V 5% NP0 0805
I	C39pF	GRM21A5C2E390JW01D	CAP CER 39PF 100V 5% NP0 0805
J	C100pF	VJ0805A101KXBAC	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 100pF 100volts C0G 10%
K	C100pF	C0805C101K5GACTU	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 50volts 100pF C0G 10%
L	C100pF	08051A101MAT2A	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 100pF 100V 20%
M	C200pF	C0805C201J5GACTU	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 50volts 200pF C0G 5%
N	C200pF	GRM2165C2A201JA01D	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 0805 200pF 100volt C0G +/-5%
P	C200pF	08051A201JAT2A	Multilayer Ceramic Capacitors (MLCC) - SMD/SMT 0805 200pF 100volts C0G 5%

C.2 Resistors Randomly Selected for IDS Modules

Code	Value	Mfg Part No	Description
A	R84.5K	292-84.5K-RC	Thick Film Resistors - SMD 1/10WATT 84.5KOHMS
B	R84.5K	CRCW080584K5FKEA	Thick Film Resistors - SMD 1/8watt 84.5Kohms 1% 100ppm
C	R84.5K	RK73H2ATTD8452F	Thick Film Resistors - SMD 1/8watt 84.5Kohms 1%
D	R165K	292-165K-RC	Thick Film Resistors - SMD 1/10WATT 165KOHMS
E	R165K	CRCW0805165KFKEA	Thick Film Resistors - SMD 1/8watt 165Kohms 1% 100ppm
F	R165K	RK73H2ATTD1653F	Thick Film Resistors - SMD 1/8watts 165Kohms 1%
H	R249K	292-249K-RC	Thick Film Resistors - SMD 1/10WATT 249KOHMS
I	R249K	RC0805FR-07249KL	Thick Film Resistors - SMD 249K OHM 1%
J	R249K	ERJ-6ENF2493V	Thick Film Resistors - SMD 0805 249Kohms 1% Tol
L	R64.9K	292-64.9K-RC	Thick Film Resistors - SMD 1/10WATT 64.9KOHMS
M	R64.9K	CRCW080564K9FKEA	Thick Film Resistors - SMD 1/8watt 64.9Kohms 1% 100ppm
N	R64.9K	RK73H2ATTD6492F	Thick Film Resistors - SMD 1/8watt 64.9Kohms 1%
O	R143K	292-143K-RC	Thick Film Resistors - SMD 1/10WATT 143KOHMS
P	R143K	ERJ-6ENF1433V	Thick Film Resistors - SMD 0805 143Kohms 1% Tol
Q	R143K	CRCW0805143KFKEA	Thick Film Resistors - SMD 1/8watt 143Kohms 1% 100ppm
R	R210K	RC0805FR-07210KL	Thick Film Resistors - SMD 210K OHM 1%
S	R210K	292-210K-RC	Thick Film Resistors - SMD 1/10WATT 210KOHMS
T	R210K	CR0805-FX-2103ELF	Thick Film Resistors - SMD 210K 1%
U	R49.9K	292-49.9K-RC	Thick Film Resistors - SMD 1/10WATT 49.9KOHMS 1%
V	R49.9K	CR0805-FX-4992ELF	Thick Film Resistors - SMD 49.9K 1%
W	R49.9K	RC0805FR-0749K9L	Thick Film Resistors - SMD 49.9K OHM 1%
X	R97.6K	292-97.6K-RC	Thick Film Resistors - SMD 1/10WATT 97.6KOHMS
Y	R97.6K	CR0805-FX-9762ELF	Thick Film Resistors - SMD 97.6K 1%
Z	R97.6K	RC0805FR-0797K6L	Thick Film Resistors - SMD 97.6K OHM 1%
ZA	R21.5K	RK73H2ATTD2152F	Thick Film Resistors - SMD 1/8watt 21.5Kohms 1%
ZB	R21.5K	292-21.5K-RC	Thick Film Resistors - SMD 1/10WATT 21.5KOHMS
ZC	R21.5K	CRCW080521K5FKEA	Thick Film Resistors - SMD 1/8watt 21.5Kohms 1% 100ppm
ZD	R12.4K	RN732ATTD1242B25	Thin Film Resistors - SMD 1/10W 12.4Kohm 0.1% 25ppm
ZE	R12.4K	288-0805-12.4K-RC	Thin Film Resistors - SMD 12.4K OHM 0.1% 10PPM
ZF	R12.4K	TNPW080512K4BEEN	Thin Film Resistors - SMD 12.4Kohms .1% 25ppm
ZG	R24.9K	ERA-6AEB2492V	Thin Film Resistors - SMD 0805 24.9Kohm 0.1% 25ppm
ZH	R24.9K	RN732ATTD2492F25	Thin Film Resistors - SMD 1/10W 24.9Kohm 1% 25ppm
ZI	R24.9K	RG2012P-2492-B-T5	Thin Film Resistors - SMD 1/10W 24.9K ohm 0.1% 25ppm
ZJ	R45.3K	CRCW080545K3FKEA	Thick Film Resistors - SMD 1/8watt 45.3Kohms 1% 100ppm
ZK	R45.3K	RK73H2ATTD4532F	Thick Film Resistors - SMD 1/8watt 45.3Kohms 1%
ZL	R45.3K	RG2012P-4532-B-T5	Thin Film Resistors - SMD 1/10W 45.3K ohm 0.1% 25ppm

C.3 Placement of Components on IDS Modules

	CAP MFG	10pF A	10pF B	10pF C	20pF D	20pF E	20pF F	39pF G	39pF H	39pF I	100pF J	100pF K	100pF L	200pF M	200pF N	200pF P
	R MFG															
R84.5K	A	1														
	B		2													
	C			3												
R165K	D	4														
	E		5													
	F			6												
R249K	H	7														
	I		8													
	J			9												
R64.9K	L				10											
	M					11										
	N						12									
R143K	O				13											
	P					14										
	Q						15									
R210K	R				16											
	S					17										
	T						18									
R49.9K	U							19								
	V								20							
	W									21						
R97.6K	X							22								
	Y								23							
	Z									24						
R165K	D							25								
	E								26							
	F									27						
R21.5K	ZA										28					
	ZB											29				
	ZC												30			
R49.9K	U										31					
	V											32				
	W												33			
R84.5K	A										34					
	B											35				
	C												36			
R12.4K	ZD													37		
	ZE														38	
	ZF															39
R24.9K	ZG													40		
	ZH														41	
	ZI															42
R45.3K	ZJ													43		
	ZK														44	
	ZL															45