NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

USING A K-NEAREST NEIGHBORS MACHINE LEARNING APPROACH TO DETECT CYBERATTACKS ON THE NAVY SMART GRID

by

Vincent C. Chan

September 2020

Thesis Advisor: Preetha Thulasiraman
Second Reader: Monique P. Fargues

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704-0188*

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2020	**3. REPORT TYPE AND DATES COVERED** Master's thesis

4. TITLE AND SUBTITLE USING A K-NEAREST NEIGHBORS MACHINE LEARNING APPROACH TO DETECT CYBERATTACKS ON THE NAVY SMART GRID	**5. FUNDING NUMBERS** RMNPB
6. AUTHOR(S) Vincent C. Chan	

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) ONR/ESTEP, Arlington, VA 22203	**10. SPONSORING / MONITORING AGENCY REPORT NUMBER**

11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release. Distribution is unlimited.	**12b. DISTRIBUTION CODE** A

13. ABSTRACT (maximum 200 words)

In 2019, the Naval Facilities Engineering Command (NAVFAC) deployed the Navy smart grid across multiple bases in the United States. The smart grid can improve the reliability, availability, and efficiency of electricity supply. While this brings about immense benefit, placing the grid on a network connected to the internet increases the threat of cyberattacks aimed at intelligence collection, disruption, and destruction. In this thesis, we propose an Intrusion Detection System (IDS) for the NAVFAC smart grid. This IDS comprises a feature extractor, classifier, anomaly detector, and response manager. We use the K-Nearest Neighbors machine learning algorithm to show that various attacks (web attacks, FTP/SSH attacks, DOS, DDOS and port scanning) can be grouped into broader attack classes of Active, Denial, and Probe for appropriate response management. We also show that in order to reduce the load on the security operations center (SOC), the accuracy of the classifier can be maximized by optimizing the value of k, which is the number of data points nearest to the sample under consideration that decides the class assigned.

14. SUBJECT TERMS smart grid, cyber-security, machine learning, K-Nearest Neighbors, KNN	**15. NUMBER OF PAGES** 81
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	**18. SECURITY CLASSIFICATION OF THIS PAGE**	**19. SECURITY CLASSIFICATION OF ABSTRACT**	**20. LIMITATION OF ABSTRACT**
Unclassified	Unclassified	Unclassified	UU

NSN 7540-01-280-5500 Standard Form 298 (Rev. 2-89) Prescribed by ANSI Std. 239-18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release. Distribution is unlimited.

USING A K-NEAREST NEIGHBORS MACHINE LEARNING APPROACH TO DETECT CYBERATTACKS ON THE NAVY SMART GRID

Vincent C. Chan
Commander, Republic of Singapore Navy
MEng, Imperial College London, 2008

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING SCIENCE
(ELECTRICAL ENGINEERING)

from the

NAVAL POSTGRADUATE SCHOOL
September 2020

Approved by: Preetha Thulasiraman
Advisor

Monique P. Fargues
Second Reader

Douglas J. Fouts
Chair, Department of Electrical and Computer Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In 2019, the Naval Facilities Engineering Command (NAVFAC) deployed the Navy smart grid across multiple bases in the United States. The smart grid can improve the reliability, availability, and efficiency of electricity supply. While this brings about immense benefit, placing the grid on a network connected to the internet increases the threat of cyberattacks aimed at intelligence collection, disruption, and destruction. In this thesis, we propose an Intrusion Detection System (IDS) for the NAVFAC smart grid. This IDS comprises a feature extractor, classifier, anomaly detector, and response manager. We use the K-Nearest Neighbors machine learning algorithm to show that various attacks (web attacks, FTP/SSH attacks, DOS, DDOS and port scanning) can be grouped into broader attack classes of Active, Denial, and Probe for appropriate response management. We also show that in order to reduce the load on the security operations center (SOC), the accuracy of the classifier can be maximized by optimizing the value of k, which is the number of data points nearest to the sample under consideration that decides the class assigned.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

AMI	Advanced Metering Infrastructure
BAN	Base Area Network
CICIDS	Canadian Institute for Cybersecurity's Intrusion Detection Evaluation Dataset
DARPA	Defense Advanced Research Projects Agency
DOD	United States Department of Defense
DOE	United States Department of Energy
DOS	Denial of Service
DDOS	Distributed Denial of Service
FDIA	False Data Injection Attack
FTP	File Transfer Protocol
HAN	Home Area Network
ICS	Industrial Control Systems
IDS	Intrusion Detection System
IoT	Internet of Things
IP	Internet Protocol
KDD	International Conference on Knowledge Discovery and Data Mining
KNN	K-Nearest Neighbors
MITM	Man in the Middle
ML	Machine Learning
NAN	Neighborhood Area Network
NAVFAC	US Naval Facilities Engineering Command
PCA	Principal Component Analysis
SOC	Cybersecurity Operations Center
SSH	Secure Shell
TCP	Transport Control Protocol
WAN	Wide Area Network
WSN	Wireless Sensor Network

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

I would like to express my heartfelt thanks to the following people and organizations, without whom this Thesis would not be possible:

My wife, Grace, and daughters, Marie and Hannah. For literally following me across the world, and for allowing me the time to focus on this endeavor.

The Republic of Singapore Navy, for giving me this opportunity to enrich myself.

My advisor, Professor Thulasiraman, for her guidance and support.

And to all those at the front line during the Covid-19 pandemic, for your selfless effort in keeping us safe.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

The 4th Industrial Revolution, a term introduced in 2016 by Professor Klaus Schwab, founder and executive chairman of the World Economic Forum, describes the world's move into cyber-physical systems and promises a myriad of technological advancements [1]. Nine technologies that enable this cyber-physical revolution are described in Figure 1. A combination of these technologies gave rise to the idea of a Smart Grid.

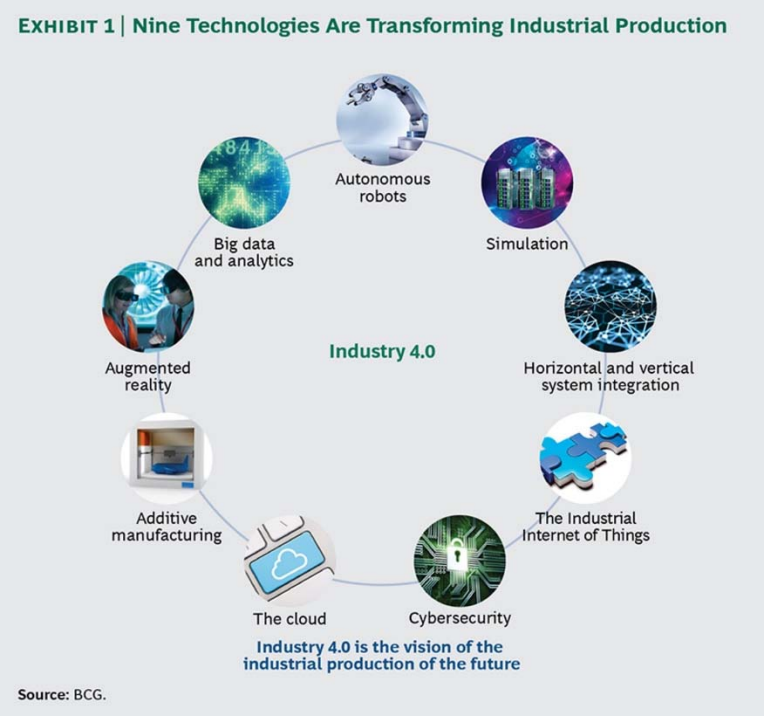

Figure 1. 4th Industrial Revolution Technologies. Source: [2].

A. WHAT IS A SMART GRID?

A smart grid is described by the U.S. Department of Energy (DOE) [3] as a rebuild of the existing power grid, and one that allows two-way communication between the user and the supplier. The smart grid is envisioned to improve reliability, availability, and

efficiency of the electricity supply. The benefits of the smart grid as given by the DOE are as follows:

- More efficient transmission of electricity
- Quicker restoration of electricity after power disturbances
- Reduced operations and management costs for utilities, and ultimately lower power costs for consumers
- Reduced peak demand, which will also help lower electricity rates
- Increased integration of large-scale renewable energy systems
- Better integration of customer-owner power generation systems, including renewable energy systems
- Improved security [3]

Figure 2 illustrates the evolution from the traditional power grid to the smart grid. In a traditional grid, the power generated flows in one direction from supplier to consumer. In a smart grid, there is a two-way flow of power and more importantly, information such as current and predicted power demand, current state of the grid, and consumer information is transmitted efficiently.

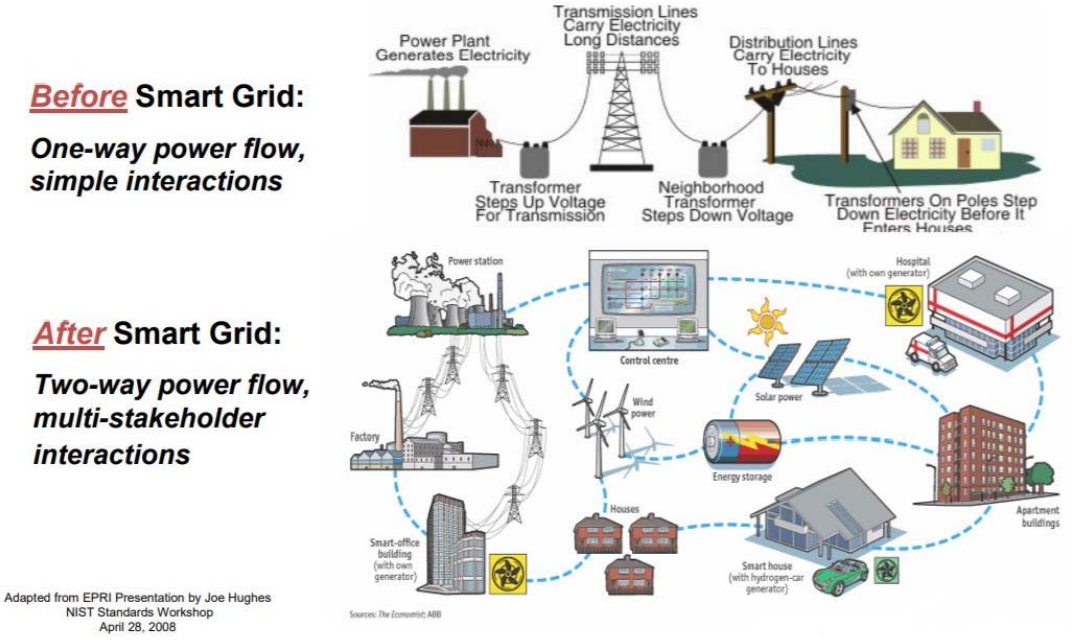

Figure 2. The Evolution of the Electric Utility System. Source: [4].

The concept of the smart grid introduces a whole new paradigm in power supply. New sources of energy can be added to the grid, more efficient power generation can be achieved by better demand estimation, and losses and anomalies can be detected and analyzed.

The smart grid is enabled by the addition of a sophisticated communications infrastructure that enhances the traditional grid. This communication infrastructure is shown in Figure 3. Internet of Things (IoT) devices such as sensors and smart meters record and transmit data (usually wirelessly) to an aggregator or a hub, which then forwards it via a wide area network (such as the Internet) to a control center. At its core, the smart grid is still an Industrial Control System (ICS)—a combination of components which provide control to an industrial process.

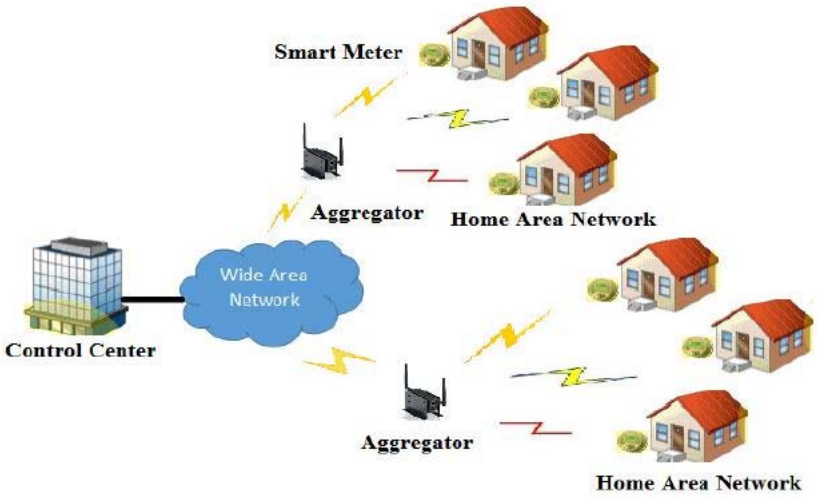

Figure 3. Simplified Smart Grid Communications Network. Source: [5].

B. WEAKNESS OF THE SMART GRID

Although the smart grid can bring immense benefit, it comes at a cost. The power grid was once protected by obscurity—legacy ICS systems tend to be built upon proprietary protocols and hardware, and are physically secure—most of the hardware can be put under lock and key, and the loss of a single node simply results in a power outage to a local area. However, as information technology components are introduced into the smart grid, many

proprietary systems are now being replaced by low-cost Internet Protocol (IP) systems. These additions combined with network connectivity (both wireless and through the internet), exposes the smart grid to a myriad of cyberattacks with numerous entry points [6].

According to the 2019 Symantec Internet Security Threat Report [7], an IoT device experiences an average of 5200 cyberattacks a month. IoT and ICS have also over time become targets of criminal and targeted attack groups. Placing the grid on a network, regardless of whether it is connected to the internet, exposes it to the same cybersecurity risks as any online system. Therefore, it is imperative that when a smart grid is fully deployed, it comes equipped with a comprehensive cybersecurity architecture that can detect, classify, and respond to cyberattacks on the grid.

C. RESEARCH MOTIVATIONS AND CONTRIBUTIONS

In 2013, the U.S. Naval Facilities Engineering Command (NAVFAC) foresaw the need for the Navy and Marine Corps to leverage the smart grid industry and formulated its plan for the Navy's own smart grid [8]. The NAVFAC has since invested heavily in infrastructure and research to deploy a smart grid that is tailored for Navy operations. In 2019, NAVFAC began full scale implementation of the smart grid at various installations in the mid-Atlantic region [9]. In placing the Navy's critical infrastructure on a network, both state and non-state actors will be particularly interested in what information they can learn, and what damage they can cause the Navy through these means. The research in this thesis contributes to a funded project by the Office of Naval Research's Energy Systems Technology Evaluation Program (ESTEP) which is studying the security of the Navy's smart grid.

Different cyberattacks warrant different responses to maintain smart grid availability; the grid cannot be taken offline every time there is an attack. While there is substantial research in the literature that uses machine learning and neural networks to detect a cyberattack against an ICS, there is little established work that specifically investigates machine learning cyber analytics for the smart grid.

The work in this thesis is foundational and our objective is to provide a starting point to using machine learning applications for the Navy smart grid. The contributions of this thesis are as follows:

- Propose an Intrusion Detection System (IDS) for the NAVFAC smart grid. This IDS is comprised of a feature extractor, classifier, anomaly detector, and response manager to allow the grid to react according to the cyber threat.

- Introduce the CICIDS2017 data set and the use of the open source CICFlowMeter for feature extraction.

- Prove that different attacks can be grouped together based on objectives, and that machine learning using the K-nearest neighbors algorithm can be used to segregate the various classes of attack traffic from benign traffic.

D. THESIS ORGANIZATION

The remainder of this thesis is organized as follows: In Chapter II, we discuss the smart grid architecture and protocol stack, and scope the area studied. We also describe identified cybersecurity vectors with critical impact on the grid. In Chapter III we provide a brief overview on machine learning, explain how to choose a technique, and take a more in-depth look at the K-Nearest Neighbors (KNN) classifier. Chapter IV describes our proposed cybersecurity architecture for the Navy's smart grid, how attacks are grouped for this study, why the KNN classifier was chosen, and introduces the data set used. Chapter V presents the results for the simulations conducted and applies it to the proposed architecture. Chapter VI concludes the study and recommends possibilities for future work.

THIS PAGE INTENTIONALLY LEFT BLANK

II. CYBERSECURITY AND THE SMART GRID

A. SMART GRID ARCHITECTURE

A large portion of the smart grid consists of the Advanced Metering Infrastructure (AMI). This AMI consists of smart devices, data management systems, and a communications network. Figure 4 gives a generic view of such a structure and breaks it down into a Home area network (HAN), Neighborhood area network (NAN) and Wide area network (WAN).

In this section, the various networks are discussed, and the main components and protocols are described.

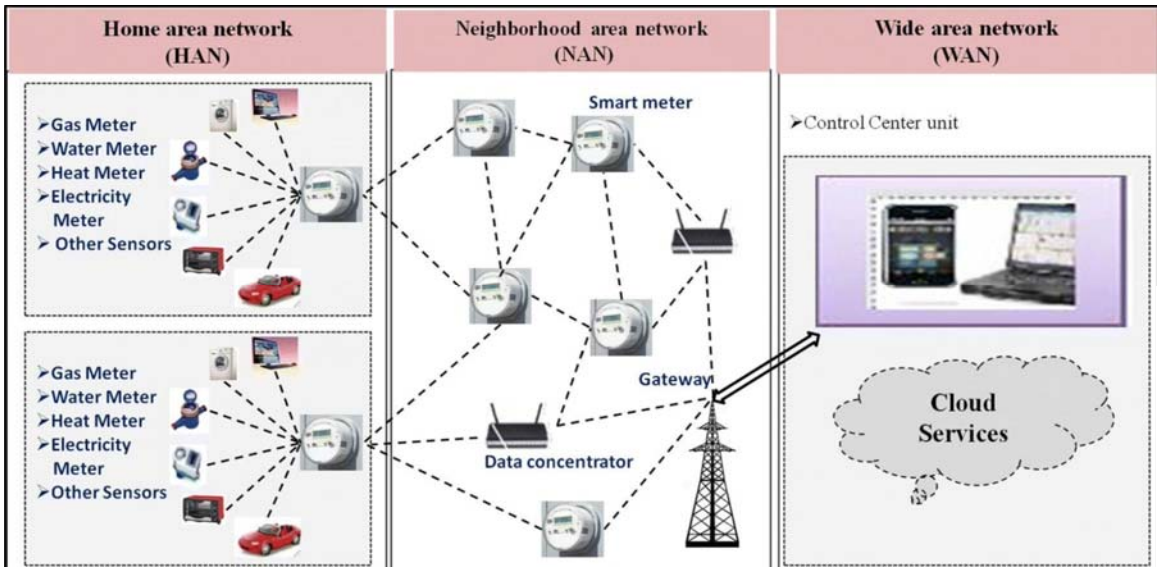

Figure 4. Multi-tier Smart Grid Communications Architecture. Source: [10].

1. Networks and Components

a. Home Area Network and the Smart Meter

The HAN, as its name suggests, is everything contained within a single home. The smart meter is the gateway from the HAN to the NAN. At the current level of commercial implementation, the smart meter is capable of reporting consumption as a function of time

and removes the need for manual meter readings. This allows both user and supplier to track usage trends and provide accurate pricing, allowing the supplier to reduce wastage by generating the correct amount of power, and the user to potentially time his activities to take advantage of windows with cheaper electricity to conduct activities like charging electric cars.

In the future, when more homes take advantage of IoT technology, the smart meter may be able to track electricity usage by appliance and allow the user to make more informed decisions on usage and therefore save cost. An IBM blog [11] also suggested in 2016 that the AMI will enable the seamless integration of microgrids, where renewables can efficiently supplement the grid.

In the context of the Navy, the HAN could represent a ship or a building. For buildings, the smart meter could help ensure efficient use of energy and prioritize critical systems when there is an outage. The deployment of smart meters on ships may present various advantages. Since each ship has its own generators, they can function themselves as microgrids when not drawing power from shore. The bases which house them can potentially save on utilities when they supply sufficient power to run various installations instead of drawing electricity from the public grid.

b. Neighborhood Area Network and the Data Concentrator

The NAN connects all the smart meters in a neighborhood to the utility's WAN infrastructure. To ensure the proper flow of data, the smart meters usually feed their data to a data concentrator and the concentrator is responsible for packaging the data to send to the utility. These devices also perform local control and manage the system health of the smart meters in its neighborhood. They are also capable of data storage and contingent modes of operation to mitigate the loss of WAN connectivity [12].

In the Navy context, the "neighborhood" will depend on the size of the installation and the ability to achieve seamless connectivity. There will likely be a data concentrator for each group of buildings, and perhaps each wharf of ships.

c. Wide Area Network

The WAN connects the NAN to the utility control center and to cloud services which host analytics and data storage. This connectivity could take various forms but would most likely use the internet to enable connectivity across a wide expanse.

For the Navy, the WAN would connect all the NANs within a facility using the Navy's own fiber optics or other secure communications methods and may even link various installations together using satellites. However, unless the Navy produces its own electricity, a gateway between the Navy grid and the internet or a utility's network will be necessary.

2. Means of Data Transmission

Kuzlu, Pipattanasomporn and Rahman [13] summarized potential communications technologies suitable for the various sections of the AMI based on range and data rate. This is adapted and updated as Table 1.

Table 1. Comparison of Communication Technologies for the Smart Grid. Adapted from [13].

Technology	Standard/Protocol	Max Theoretical Data Rate	Max Range	HAN	NAN	WAN
Wired Technologies						
Fiber Optic	PON	2.5 Gbps	60 km			X
	WDM	40 Gbps	100 km			
	SONET/SDH	10 Gbps	100 km			
DSL	ADSL	8 Mbps	5 km		X	
	HSDL	2 Mbps	3.6 km			
	VSDL	100 Mbps	1.5 km			
Coaxial Cable	DOCSIS	172 Mbps	28km		X	
PLC	HomePlug	200 Mbps	200m	X		
	Narrowband	500 kbps	3km		X	
Ethernet	802.3x	10 Gbps	100m	X	X	
Wireless Communications Technologies						
Z-Wave	Z-Wave	40 kbps	30m	X		
Bluetooth	802.15.1	721 kbps	100m	X		
ZigBee	ZigBee	250 kbps	100m	X	X	
	ZigBee Pro	250 kbps	1600m			
WiFi	802.11x	600 Mbps	100m	X	X	
WiMAX	802.16	75 Mbps	50km		X	X
Wireless Mesh	Various (e.g., RF mesh, 802.11, 802.15, 802.16)	Depending on Protocol	Depending on Deployment	X	X	
Cellular	4G	100 Mbps	Depending on Deployment		X	X
	5G	10 Gbps				
Satellite	Satellite Internet	1 Mbps	6000km			X

Many technologies exist and are ideal for different setups and environments. When so many different technologies are used, integration becomes a challenge. The most practical solution therefore is to design the smart grid around the widely used Transport Control Protocol (TCP) and Internet Protocol (IP). Although this will expose the network to cybersecurity risks, it is simply not cost-effective to secure the grid by designing a proprietary protocol due to the sheer number of components that will need to be programmed and networked.

B. ASSUMPTIONS MADE ON THE NAVY SMART GRID

Figure 5 shows the communication flows and a basic architecture for the Navy smart grid provided by NAVFAC. Based on this figure and given what we know about

traditional smart grids as described in the previous section, a few assumptions can be made about the Navy smart grid. These are:

1. The communications architecture will be based on TCP/IP for information exchange.
2. The Base Area Network (BAN) is a combination of the HAN and NAN and will use either wireless or power-line technologies due to the high cost of laying cables for this specific purpose.
3. The endpoint devices have limited cryptographic capability as they may be battery operated and be limited in processing power.
4. The smart meters are potentially located in physically unsecure locations.
5. The AMI Meter as depicted in Figure 5 does not connect directly to the backbone of the base. Several will channel information through a data concentrator as per the NAN in Figure 4. These data concentrators can be made physically secure.
6. The WAN between the data concentrator and the control center will use the Navy's private communications methods and will not be exposed to the public domain.
7. Data exchange across the WAN will be encrypted to DOD standards, and the basics of confidentiality, integrity and availability in cybersecurity will be implemented.
8. Not every base will house a cybersecurity operations center (SOC).

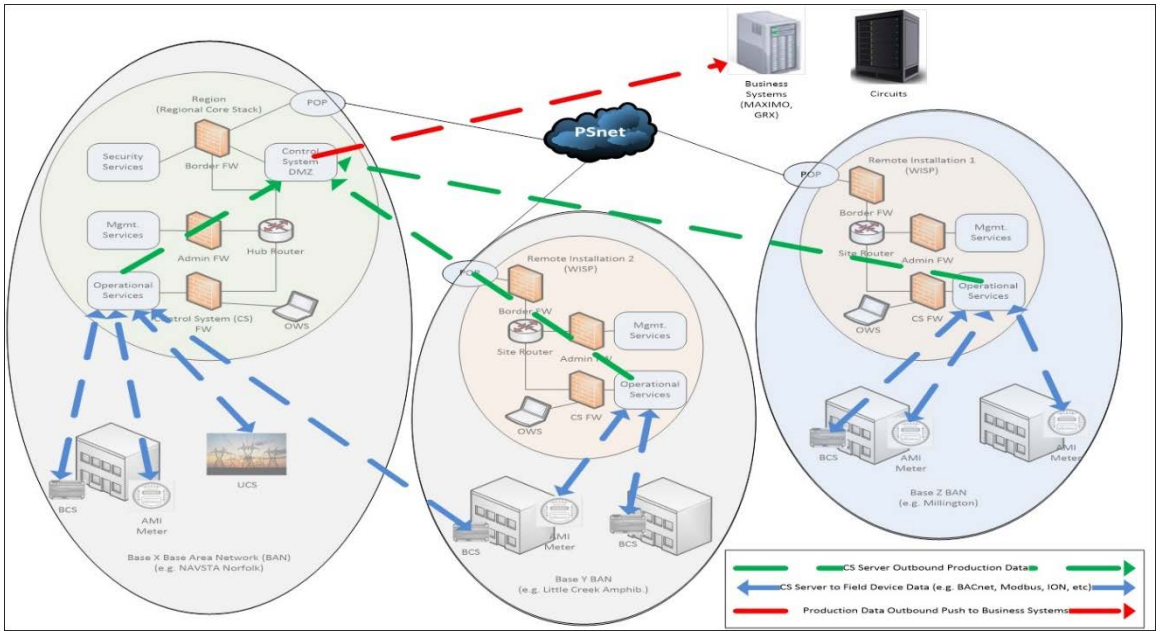

Figure 5. Communication Flows for the Navy Smart Grid. Source: NAVFAC.

Therefore, given the porosity within the BAN, any attempt to detect breaches in the network should be made at the data concentrator, before the data is transmitted to the control center. Also, given that not every base will contain a SOC, there will be limited resources to respond to a cyberattack.

C. CYBERSECURITY THREATS

Elmrabet et al. conducted a survey of cybersecurity threats to the smart grid, a large number of which are focused at the control center [14]. Of the attacks they describe, this thesis focuses on those which can take place prior to the data concentrator.

1. Port Scanning

The first step in Lockheed Martin's Cyber Kill Chain framework is reconnaissance [15], and this is where port scanning comes into play. Finding and identifying open ports allows an attacker to determine potential vulnerabilities and entry points for attacks.

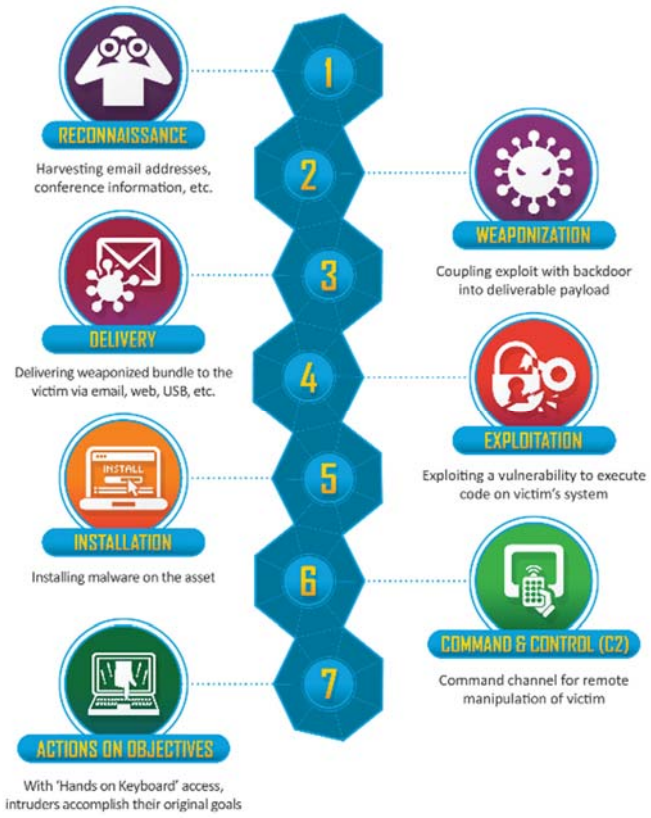

Figure 6. Cyber Kill Chain. Source: [15].

2. Denial of Service

Denial of service (DOS) takes several forms in the AMI and is extensively researched in the literature [16], [17], [18]. Diaz and Sanchez [19] describe many forms of attacks which aim to disrupt the communication flow between nodes. These can be summarized into three categories:

- Attacks which overwhelm the network's ability to process information via means such as fake packet injection.
- Attacks which introduce noise into the wireless network such as jamming the nodes.
- Attacks on the firmware of the sensors.

3. Active / Intrusion

Liang et al. [20] studied the possibility and potential effects of False Data Injection Attacks (FDIA). These have the potential for both physical and economic damage by injecting malicious measurement data to adversely affect the state estimation process of the smart grid.

4. Passive / Eavesdropping

As the HAN and NAN are largely wireless networks, they are particularly susceptible to eavesdropping. Without strong encryption, an adversary can easily intercept all communications between nodes and capture information to conduct traffic analysis or read data.

D. POTENTIAL IMPACT TO THE SMART GRID AND RESPONSES

It is useful to frame attacks on the smart grid in terms of the desired effect. Knapp and Samai [21] highlight three motivations—theft of information, denial of service, and manipulation of service. For the Navy, a similar set of effects for the attacker can be discerned, and each warrants a different response. The worst case for each is presented here.

1. Service Interruption

In this scenario, an attacker can shut down systems or infrastructure at critical phases of operations by conducting DOS attacks. For example, an attacker may take defenses offline during a missile raid, or create a distraction during a physical intrusion.

A possible countermeasure for DOS attack would be to take the affected nodes off the communications network, but automatically deliver a fixed amount of power to the systems under attack; i.e., revert to the traditional grid until the DOS attack is resolved.

2. Cascading Failure

Baldick et al. describe a cascading failure as "*a sequence of dependent failures of individual components that successively weakens the power system*" [22]. In their paper,

they discuss various causes and methods to model cascading failures. In general, the failure of one or more critical components causes a rerouting of power which in turn may overload other components, causing a series of failures and leading to a blackout. While these events are usually random, the bridging of cyber and physical systems make it possible that an attacker could plan and actively cause failures by methods such as FDIA. A well-planned attack on a naval facility could cause lasting damage on par with Stuxnet [23].

To effectively respond to such a scenario, the cybersecurity system must be able to detect the intrusion and if necessary, isolate the nodes under attack.

3. Gathering Intelligence

Through traffic analysis and studying the exchange of data that comes from eavesdropping, an attacker could determine locations of interest within a military installation. Data centers or servers could be characterized by an above average power draw, experimental labs correlated to power surges, power sources indicated by power provided to the grid, etc.

This set of attacks is passive by nature and usually undetectable. It is unlikely that any system can detect passive activity and the best mitigation and countermeasure is in the design of the system. Data from critical facilities may need to be encrypted or transmitted via wired means.

E. THE NEED FOR CYBERSECURITY

The previous section describes how an attacker could cause catastrophic damage to the Navy's interests and has numerous means to achieve that outcome. An attacker only has to succeed once, while we as the defender need to succeed all the time. It is therefore imperative to have a robust cybersecurity architecture to secure the Navy smart grid.

The need for a robust system gives rise to the idea of layered security and defense in depth. Many models exist, and one such model is the NIST cybersecurity framework shown in Figure 7. While the identification of threats (Identify) and the design of the system (Protect) is important, this thesis will focus on the detection (Detect) of cybersecurity events to enable the appropriate response (Respond).

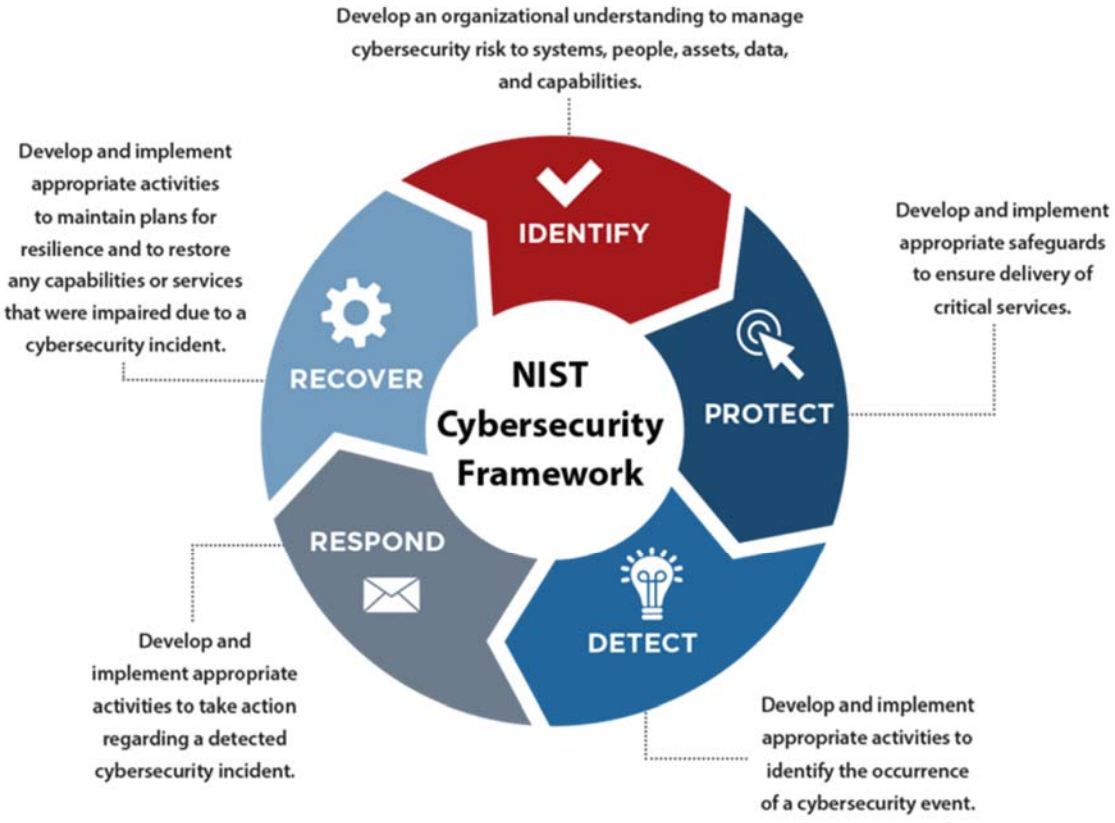

Figure 7. NIST Cybersecurity Framework. Source: [24].Equation Chapter 3 Section 1

III. MACHINE LEARNING

Machine learning (ML) has come a long way since its inception in the 1950s. Arthur Samuel, a pioneer in the field and the person credited with the phrase, stated that "Programming computers to learn from experience should eventually eliminate the need for much of this detailed programming effort" [25]. Today, there is extensive literature on ML techniques and their various applications, including in the field of cybersecurity.

A. OVERVIEW

As mentioned in Chapter II, this thesis focuses on the detection of cybersecurity events. This necessitates the development of what is commonly known as an Intrusion Detection System (IDS), which monitors a network and flags unusual events. There are two approaches to IDS: misuse detection which is based on rules and signatures, and anomaly detection which analyzes traffic to determine if it statistically deviates from what is known to be normal. ML is suitable to both types of detection depending on the technique chosen and the data available. In this section two general types of ML techniques are described—supervised and unsupervised.

It must be noted that data is core to the development of a ML algorithm and many data sets have been produced over the years for academic study. In [26], Amit et al. discuss some newer available data sets that add to the popular Defense Advanced Research Projects Agency (DARPA) data sets from the Massachusetts Institute of Technology, and the International Conference on Knowledge Discovery and Data Mining (KDD) series of data sets from the University of California. The generation of data sets is beyond the scope of this thesis. For our work in this thesis, we use the Canadian Institute for Cybersecurity's Intrusion Detection Evaluation Dataset (CICIDS) from the University of New Brunswick. The CICIDS will be discussed in detail in the next chapter.

1. Supervised Learning

In supervised learning, we use *labelled* data to train an algorithm, and when we input new data into the algorithm, it will output the label which best fits the given data.

There are two types of supervised learning: (1) classification and (2) regression methods. The former involves predicting a category whilst the latter predicts a quantity.

In cybersecurity, supervised techniques can be applied if we have data captured while an attack is taking place, and we are able to label the attack. Yousef et al. [27]. conducted a comparison on the use of K-means, Naïve Bayes, Support Vector Machine and Random Forest for wireless sensor networks using the KDD99 data set. Belavagi and Muniyal [28] compared Logistic Regression, Gaussian Naïve Bayes, Support Vector Machine and Random Forest using the NSL-KDD data set. Both determined that the Random Forest classifier was the best performing for the purpose of intrusion detection.

2. Unsupervised Learning

For unsupervised learning, we give *unlabeled* data to the algorithm in an attempt for the algorithm to find structure or relationships in the data, and group them accordingly. There are usually two tasks for unsupervised learning: (1) clustering and (2) dimensionality reduction. The former groups the data to allow the user to perform analysis and find patterns, while the latter reduces the number of features to those which are statistically more important in grouping.

In cybersecurity, unsupervised techniques are useful for anomaly detection and attacks for which there is no known signature or data, such as zero-day attacks. However, it is also well established that an unsupervised technique performs poorly compared to supervised techniques when labelled data is available due to the number of false alarms and the fact that not all anomalous events are cybersecurity events. Early work by Leung and Leckie studied various clustering methods to achieve anomaly detection [29]. More recently, the application of deep learning [30] and neural networks [31] have been studied in depth for the purpose of cyberattack detection.

B. SELECTING AN ALGORITHM

There are several requirements in choosing a ML algorithm. The first is to determine what kind of problem we want to solve, and therefore, what to do with the data.

This is followed by choosing a precise technique based on accuracy, training time, linearity, and number of parameters or features.

1. Objective

Mathworks summarized the ML techniques available in MATLAB (Figure 8) according to what it is best suited for.

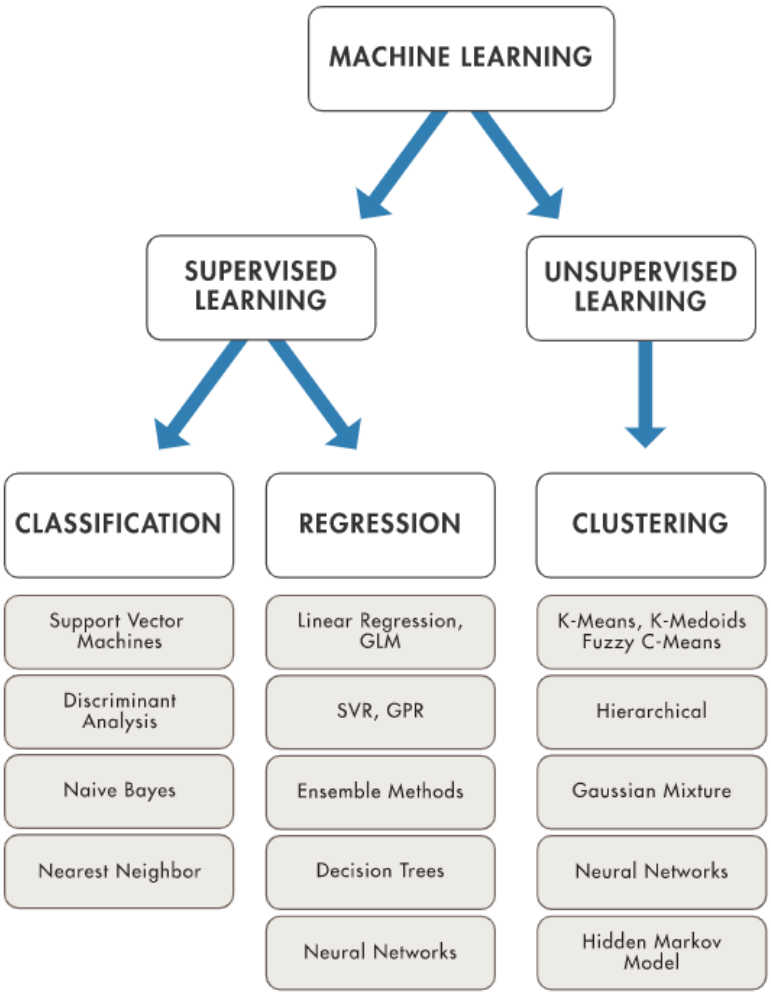

Figure 8. ML Techniques in MATLAB. Source: [32].

2. Training Time

Buczak and Guven conducted a survey of ML techniques and compiled the time complexity of training which can be found in Table 2 [33]. $O(...)$ represents the order of magnitude proportional to the variables in the brackets. Variables common to all entries of the table is that the data consists of n instances, each described by m attributes or features, and n is much greater than m. $O(n)$ and $O(n\log n)$ are linear time and appropriate for real-time applications. $O(n^3)$ is indicative of much slower algorithms and are used offline. However, it should be noted that once an algorithm is trained, the testing of incoming data is generally linear time and can be used in real-time.

Table 2. Complexity of ML Algorithms During Training. Adapted from [33].

Algorithm	**Typical Time Complexity**	**Streaming Capable**	**Comments**
Artificial Neural Networks	$O(emnk)$	Low	e: number of epochs k: number of neurons
Association Rules	$\gg O(n^3)$	Low	
Bayesian Network	$\gg O(mn)$	High	
Clustering, k-means	$O(kmni)$	High	i: number of iterations k: number of clusters
Clustering, hierarchical	$O(n^3)$	Low	
Clustering, DBSCAN	$O(n\log n)$	High	
Decision Trees	$O(mn^2)$	Medium	
Genetic Algorithms	$O(gkmn)$	Medium	g: number of generations k: population size
Naïve Bayes	$O(mn)$	High	
KNN	$O(n\log k)$	High	k: number of neighbors
Hidden Markov Models	$O(nc^2)$	Medium	c: number of states
Random Forest	$O(Mmn\log n)$	Medium	M: number of trees
Sequence Mining	$\gg O(n^3)$	Low	
Support Vector Machines	$O(n^2)$	Medium	

C. K-NEAREST NEIGHBORS

The rationale for technique selection will be covered in the next chapter, but at this point it is useful to introduce the ML algorithm used for this thesis. The KNN technique was first introduced by Thomas Cover in 1966 [34] and is one of the earliest methods of classification. He stated that there are two polar opposite cases, either the complete statistics of a distribution is known (and therefore the category for each new occurrence can be calculated), or there is no knowledge of the distribution other than making the inference from other samples (and therefore the category for each new occurrence is assigned based on the sample or samples most similar to it).

Consider an arbitrary distribution of three classes in two dimensions as shown in Figure 9. The red star is the new observation we wish to classify. The algorithm will calculate the distance (usually Euclidean) to every other point around it. From the diagram, one can see that if $k=3$, illustrated by the inner circle, it will be assigned to class 3. If $k=5$, it will be assigned to class 2 as per the outer circle.

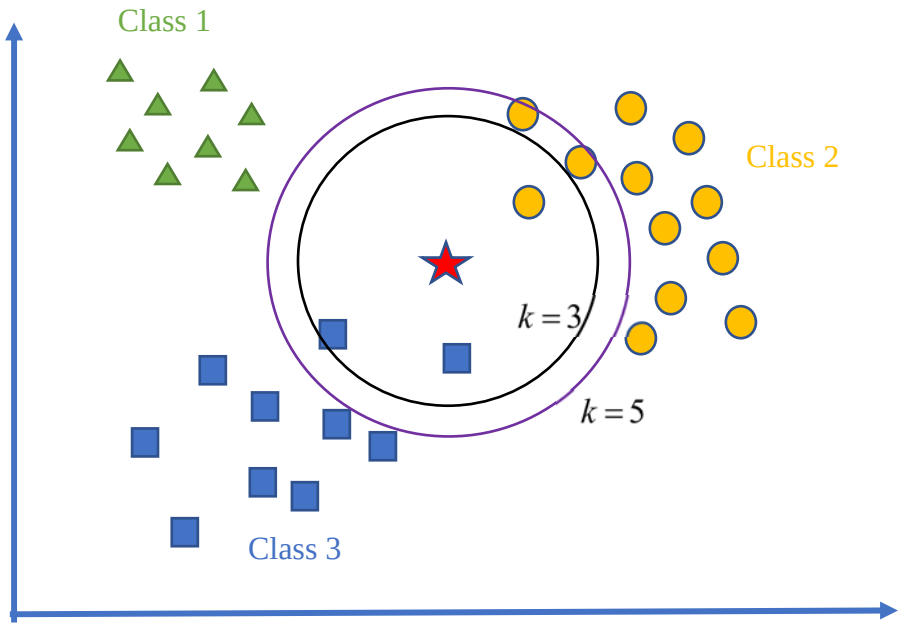

Figure 9. Illustration of the KNN Methodology.

This thesis makes use of the weighted KNN algorithm, where closer points are accorded higher value or "weights." This thesis uses the built-in MATLAB weighting function of *inverse of the distance*. However, any function that causes a value to decrease as the distance increases can be used in MATLAB to realize the weighted KNN.

Some parameters of note:

1. k

This is the main parameter for the algorithm as suggested by the fact that it features in the name. The k is the number of data points that are nearest to the sample under consideration. The class assigned will be that which occurs most frequently amongst these k data points. This algorithm is highly data dependent and is premised on the assumption that classes are well clustered and distinct. If that is the case, $k=1$ will suffice, i.e., the data point closest to it will be the most appropriate class. However, outliers do exist in the data and increasing k will help remove some of this bias.

2. Dimensionality

This is the number of features each data point has. The example on the previous page has two dimensions (x and y axis), but data used in machine learning can have any number of fields. While more fields may help segregate different classes better, they may also give rise to certain peculiarities. Beyer et al. [35]. showed that under certain conditions, as the number of dimensions increases, the difference between the distance to the nearest and farthest points decreases. This decrease will adversely affect the use of the KNN method. Their paper suggests that such phenomenon start to occur when there are between 10 to 15 dimensions and termed it the "curse of dimensionality." There are several techniques to reduce the number of features, one of which is using principal component analysis (PCA), a tool available in MATLAB.

3. Distance

In most cases the Euclidean distance is used as a metric to compute the nearest neighbors. The equation for the Euclidean distance between two n-dimensional points p and q is defined as:

$$d(p,q) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \quad . \tag{3.1}$$

The square of the distance between two points p and q is the sum of the square of the difference between each point for every feature i.

D. PRINCIPAL COMPONENT ANALYSIS

The reduction of the number of features in the data is extremely useful, both in order to avoid the "curse of dimensionality" mentioned earlier, as well as to reduce the time taken to train the model. While it is possible to write a program to assess and individually remove features that have little to no bearing on the resultant model, it is time consuming and inefficient. A popular method to achieve feature reduction, and one that is available in the MATLAB machine learning toolbox is PCA. PCA does not actually remove features by deleting them, but instead determines a new set of orthogonal features called "Principal Components" that forms a basis of the original set. MATLAB uses a Singular Value Decomposition algorithm [36] to produce a matrix which determines how each original feature contributes to the principal component, and conducts a matrix multiplication of the data with this matrix to reduce the number of variables for computation.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. PROPOSED ARCHITECTURE AND RESEARCH METHODOLOGY

A. PROPOSED SECURITY ARCHITECTURE

The IDS is a common feature in any cybersecurity system. However, the difficulty in implementing an IDS in a Wireless Sensor Network (WSN), such as in the AMI portion of the smart grid, is well established [18], [19], [37]. The WSN in the smart grid is characterized by the following:

1. Certain nodes may have limited resources such as computational power, bandwidth and memory which may constrain measures such as encryption throughout the WSN.
2. Nodes are in a potentially hostile environment as they are not physically secure and because they communicate wirelessly.
3. Network availability is a foremost consideration of a smart power grid and therefore any IDS requires high detection accuracy.

We propose a system where the IDS is placed at the data concentrators which receive the data from the wireless networks before pushing it to the local control center. These concentrators are assumed to be physically secure and computationally more capable than the nodes they serve. However, the IDS mechanism still needs to be of relatively low complexity to detect cyberattacks quickly. There should also be two-way communication between the IDS and the control center in order for the control center to update the detection models.

The proposed system is a combination of a classifier to quickly detect known attacks, and an anomaly detector to detect new threats. This system should be able to trigger specific responses to protect the integrity of the grid and isolate the areas under attack. Due to the current NAVFAC structure where not every base has a SOC, the classifier needs to be highly accurate in order to alleviate the workload of cybersecurity personnel and allow them to direct their efforts to investigate the anomalies. Figure 10 describes the proposed

security architecture. The components within the dotted area of Figure 10 compose the IDS.

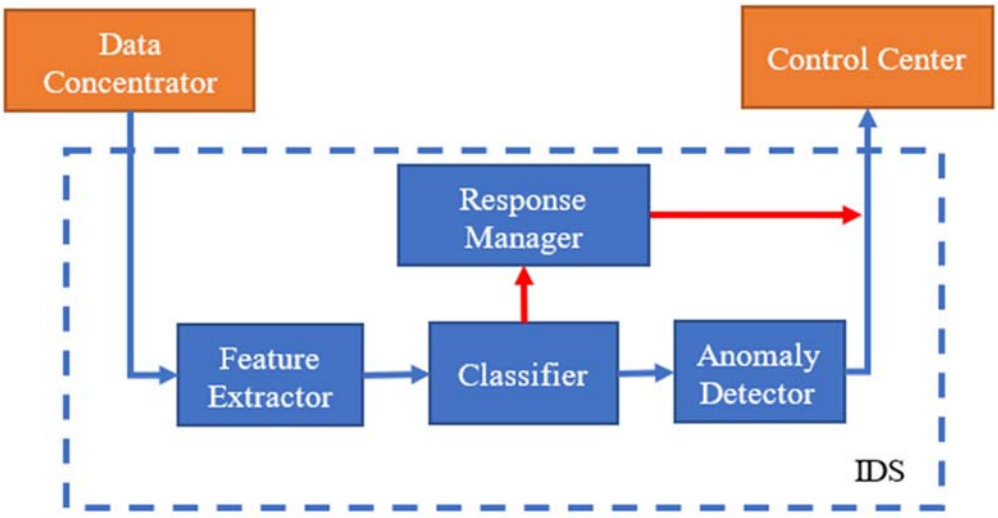

Figure 10. Block Diagram of Proposed Security System.

There are two possible models to realizing this IDS. If the total lag time is such that grid performance is unaffected, then only one data path is necessary (Figure 11).

Figure 11. Proposed Security System with Low-Lag IDS.

Otherwise, grid data needs to continue to be passed while the IDS is analyzing the traffic, and the IDS data is passed separately (Figure 12). This approach is less ideal as the delay may expose the control center to the attack data stream until the IDS can close the communications loop. In addition, in this scenario both data streams will compete for the WAN bandwidth which may cause further delays.

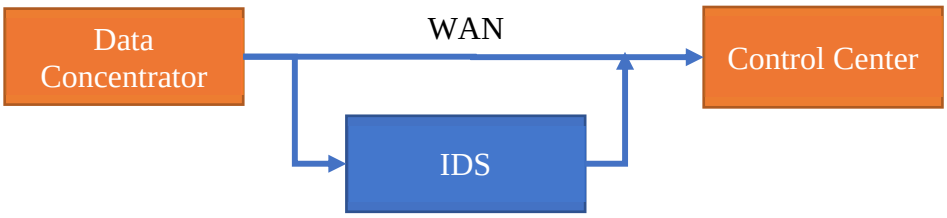

Figure 12. Proposed Security System with Higher Latency IDS.

As shown in Figure 10, the first part of the proposed security system consists of a data logger which extracts the necessary features for the ML algorithm to work on. The converted data stream is then passed through the classifier.

If an attack is detected by the classifier, the system will trigger the response manager. The response manager is responsible for a set of pre-planned responses. For example, if a DOS attack is detected from nodes 1 to 5, then the response may be to drop all communication packets from node 1 through 5 (to avoid negatively affecting the rest of the system), and provide a pre-determined amount of power to the services governed by nodes 1 to 5 (to prevent a blackout). This contingency action should be maintained until the threat has been dealt with and the SOC resets the response manager. Such an IDS will allow the system to quickly isolate the problem and ensure availability of the rest of the network.

It is unrealistic to expect that the classifier will be able to deal with all cyberattacks, especially if we consider zero-day attacks. The anomaly detector will deal with these, as well as attacks for which there is no pre-planned response. Upon detection of an anomaly, this component will trigger the SOC for further investigation and a man-in-the-loop response.

The rest of this thesis focuses on a ML technique to realize the classifier block and address the requirements for the feature extractor. As the anomaly detector will also likely be triggered by faults, and each SOC will have to deal with several bases, each alert will require significant man-hours to resolve. Therefore, the false alarm rate of the classifier must be minimized.

B. CLASSIFICATION OF ATTACKS

Elmrabet et al. pointed out that much of the literature focused on individual attacks and individual countermeasures [14]. Much of the literature on ML focuses on obtaining the classes defined in the data sets. This thesis notes that attacks of a similar nature share largely similar characteristics, and since the focus of the classifier is to trigger appropriate responses, the attacks should be grouped according to the desired response. Four general attack classes are proposed for analysis:

1. Active

The active attack seeks to gain access to the data stream to modify the exchanges between the AMI and the control center. The effects of this type of attack range from monetary loss due to inaccurate usage and possible siphoning of electricity, to destructive damage due to overloads and a possible cascading failure.

2. Probe

Referring to the Lockheed cyber kill chain that was shown in Figure 6, the first step is reconnaissance. There are some forms of reconnaissance which are completely passive and undetectable by an IDS such as “listening” to traffic. Such threats should be addressed through the security design of the system. Other forms include “probing” measures to discover weaknesses in a system which are not passive, such as port scanning. Detecting probes will indicate the presence of a potential adversary, and security personnel should endeavor to locate the source before a full attack is carried out.

3. Denial

Denial is the act of disrupting communications between the AMI and the rest of the system. This could be done to cause a localized blackout or to cause chaos to the system. Possible attacks include DOS, Distributed DOS (DDOS) or replay attacks.

4. Benign

Any traffic which is not classified as one of the previous classes would be classified as benign. Ideally this would mean that it consists of purely normal traffic. However, there

can be other attacks which cannot be grouped within the first three classes Therefore, all benign traffic should pass through the anomaly detector as an added layer of security.

C. CHOICE OF MACHINE LEARNING TECHNIQUE

As was stated previously, this thesis uses the KNN as the ML technique for this application. KNN is one of the earliest ML techniques. In recent times research has shown that non-linear techniques such as Support Vector Machines significantly outperform KNN in terms of false positive rates [38]. However, this result applies when there is only labelled data present and no unknown data. Liao and Vemuri [39] showed that a modified version of the KNN algorithm can produce high detection rates with a low false positive rate. Also, all these studies are based on either the DARPA or KDD data sets.

There is little work done in grouping data by attack type, and the data set described in the next section has not yet become mainstream. Therefore, from an experimental standpoint, it is prudent to return to a linear and not computationally intensive algorithm such as the KNN as a foundation for such work.

D. CICIDS2017 DATA SET

The basis of any ML algorithm is the suitability and availability of the training data. Much has been written about the challenges in obtaining such data, especially when organizations tend not to release real data logs freely due to privacy issues [26], [40]. This is the reason many studies resort to using the KDD and DARPA data sets, though they may be dated.

Sharafaldin, Lashkari and Ghorbani generated the CICIDS2017 data set [41] by setting up a comprehensive testbed consisting of different hardware and operating systems common to a network and running a variety of attack types to satisfy a framework of 11 criteria set by the Canadian Institute of Cybersecurity used to benchmark datasets [42]. By comparison, the KDD99 data set fulfilled six criteria and partially fulfilled two (not all common protocols were addressed, and the variety of attacks were limited). Traffic from this test bed was put through an open source program called CICFlowMeter to extract up

to 80 features from each “flow” which can be defined based on when a connection is terminated in a TCP exchange, or by time.

The data set contains eight files. The attack type (data labels) and number of data points for each file are listed in Table 3.

Table 3. Summary of Data from CICIDS2017.

File Description	**Data Labels**	**Number of Data Points**
Monday	-	-
Tuesday	FTP-Patator	7938
	SSH-Patator	5897
Wednesday	DoS Goldeneye	10293
	DoS Hulk	231072
	DoS Slowhttptest	5499
	DoS Slowloris	5796
	Heartbleed	11
Thursday Morning	Web Attack – Brute Force	1507
	Web Attack – SQL Injection	21
	Web Attack – XSS	652
Thursday Afternoon	Infiltration	36
Friday Morning	Botnet	1966
Friday Afternoon 1	PortScan	158930
Friday Afternoon 2	DDoS	128027
Total of All Files	Benign	2359087

E. DATA PREPARATION AND FEATURE SELECTION

In the initial stages of this research, while doing exploratory simulations on the CICIDS2017 dataset, some issues were apparent—a number of entries in each file had erroneous fields when loaded into MATLAB, and there was a disproportionate amount of data across the attack methods including the benign data. This finding agrees with those of Panigrahi and Borah [43] who studied the dataset, highlighted its shortcomings, and recommended some modifications. While this class imbalance might be reflective of actual system traffic, that is, a lot more traffic generated from normal usage vis-à-vis attacks, this imbalance is not useful for training an algorithm as a minority class may be obscured by the majority class and written off as a false positive. From Table 3 an example of this

imbalance is Heartbleed (11 occurrences) versus the Benign class (2.4 x10^6 occurrences). Therefore, although the data comes in an immediately usable form, additional preparation needs to be made.

The following steps were taken to prepare the data for simulation:

1. **Delete errors**. All data points which contain "NaN" or "Inf" in the data fields were removed. "NaN" means "not a number" indicating the value is undefined. "Inf" means "infinity." As KNN depends on the data being plotted and distances measured, the algorithm cannot tolerate data which cannot be represented on a scale.

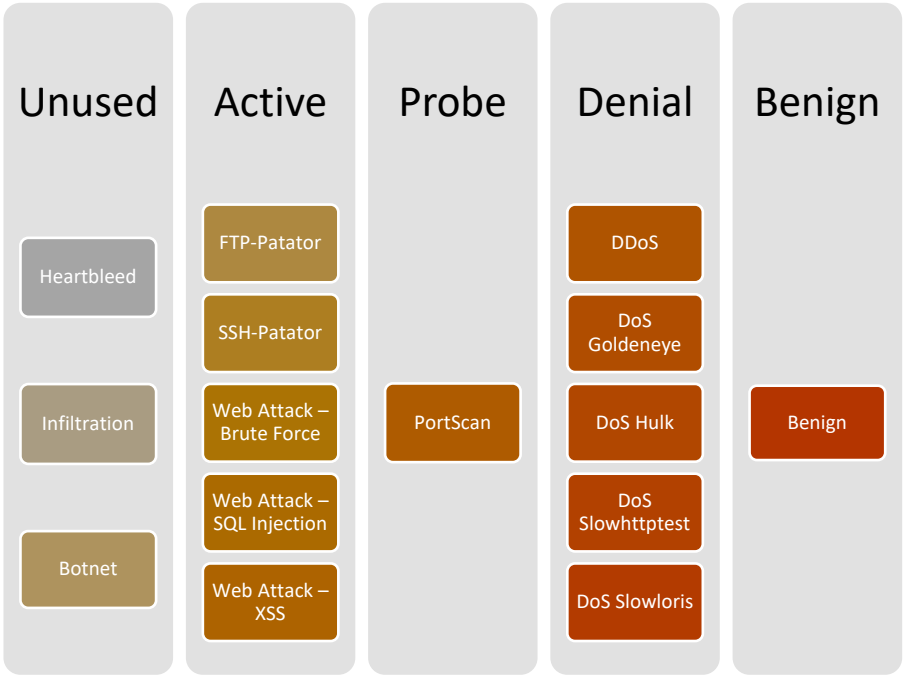

Figure 13. Reclassification of Data Labels.

2. **Relabel Data**. The relevant attack types from the dataset were selected and relabeled to the recommended classes in Section B, as presented in Figure 13. The Heartbleed (Wednesday file) and Infiltration (Thursday Afternoon file) data labels shown in Table 3 were not used due to the lack of data points. The Botnet label (Friday morning file) of Table 3 was also

not used in the data set because a Botnet can perform many attacks across the different classifications and the task for each entry is not clear in CICIDS2017.

Patator is an open source penetration tester, using brute force techniques to discover passwords. These passwords will allow an attacker to gain control of both hardware and software to conduct malicious activity. For the data set, the creators used the Secure Shell (SSH) and File Transfer Protocol (FTP) modules of the Patator program (see Table 3 Tuesday file).

The web attacks exploit weaknesses in web-based data storage and man-machine interfaces. SQL injection attempts to trick a SQL database into revealing stored information, Cross Site Scripting (XSS) attempts to find weaknesses in the developer's coding to get the website to execute the attacker's scripts, and the Web Attack—Brute Force uses brute force techniques over HTTP to guess administrator passwords for web-based accounts (see Table 3 Thursday Morning file).

3. **Remove "Port Number" feature**. Without studying the features in depth to reduce dimensionality, the "Port Number" feature was removed as the Navy smart grid architecture is not publicly available and port numbers may be reassigned as part of a proprietary system. It is undesirable that any result should be based on port numbers of a system other than the actual grid.

4. **Sort and Randomize**. The data was then sorted according to class and randomized within the class to ensure that the data that is subsequently extracted to form the training and testing set is not biased in time.

5. **Select Sample Size for Training and Test Sets**. The data is then scaled down to training and test sets. There should be no overlap of data between the two sets, and an appropriate set size was chosen to maximize accuracy and minimize training time. In general, for classes with more than 100,000 entries (Denial, Probe), 25,000 entries were used for the training set, and

25,000 used for the test set. Otherwise, 90% of the data was used for the training set and 10% of the data for the test set. Also, the proportion of benign data to attack data was set as 3:1. That is, three entries of benign data for every one attack entry. The data used and any deviations are detailed in the results section in the next chapter.

6. **Compile Data Files**. The data was then merged into training or test set files to fulfill the requirements of each simulation listed in the next section.

F. SIMULATION

1. General MATLAB Settings

The Appendix lays out how to set up and run MATLAB to allow the reader to use a similar simulation environment. The general parameters are listed in Table 4.

Table 4. General MATLAB Parameters Used.

Model Type Preset	Weighted KNN
Validation	Cross-Validation, 5 folds
Number of Neighbors	Varied based on simulation run
Distance Metric	Euclidean
Distance Weight	Inverse
Standardized Data	True
PCA	Enabled based on simulation run
PCA Variance	99%

2. Simulation Runs

For each simulation run, an appropriate training and test set was compiled. A different k was used in each run to determine the model's sensitivity to the k value. Also, each simulation was run with the MATLAB PCA feature toggled to determine its effect on accuracy and training time.

The statistics of significance are

1. Probability of Detection: Throughout the results section this will be the lowest percentage value for correctly classified attacks.
2. False Positive Rate: This will be the percentage of benign data points wrongly classified as attacks. This would wrongly trigger the response manager.

The simulations used in this thesis are as follows:

a. Pairwise

This set of simulations is to ensure that each type of attack data can be discerned from the benign traffic.

- Benign – Active
- Benign – Probe
- Benign – Denial

b. Triplets

This set of simulations is to ensure that different types of attacks can be discerned from each other.

- Benign – Active – Denial
- Benign – Active – Probe
- Benign – Denial – Probe

c. Combined

The final combined simulation is to show that a multi-attack classifier is possible with sufficient accuracy to function as part of the IDS.

- Benign – Active – Denial – Probe

V. RESULTS AND APPLICATION

A. PAIRWISE

This set of simulations investigates whether each type of attack data can be discerned from the benign traffic. A confusion matrix is usually used to visualize results in a classification problem and will be used extensively in this section. A sample matrix with a short explanation of the content in each box is presented in Figure 14. The color intensity is used in MATLAB to illustrate the difference in the numerical magnitude (for example: dark blue for 200,000 and light blue for 5,000) and should be ignored for the purpose of this section.

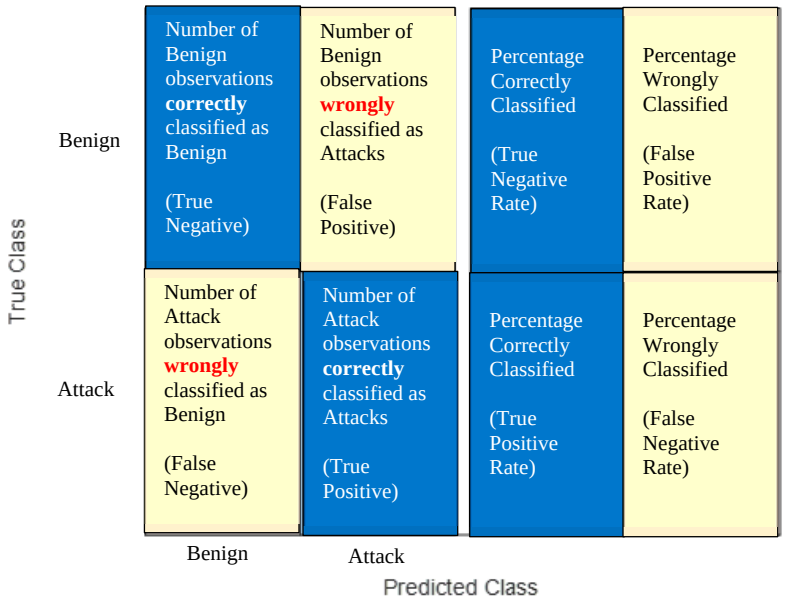

Figure 14. A Sample Confusion Matrix.

1. Benign – Active

In addition to being able to separate benign from attack data, this pairwise test also needs to prove that the web attack data can be combined with the SSH/FTP Patator data to form an overall "active" attack class. The data used is presented in Table 5.

Table 5. Data for Benign-Active Test Pair.

	Training Set	Test Set
Web Attack	~2000	201
Benign	~6000	600
FTP/SSH Brute Force	~10000	3835
Benign	~30000	~12000
Combined		
Active	~12000	NA
Benign	~36000	NA

Figure 15 and Figure 16 show that the brute force and web attack flow can be discerned from benign data with a greater than 94.5% probability of detection and a false positive rate of 1.4% and 0.3%, respectively. The two training sets were then combined, and the resultant model was applied to the web attack and FTP/SSH brute force test sets separately. This is necessary as the web attack test set is only 5% of the size of the other.

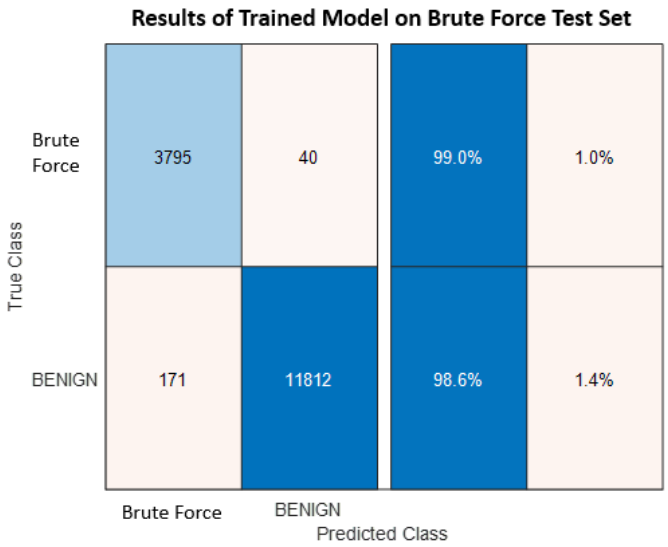

Figure 15. Results from Brute Force (Patator)-Benign Model.

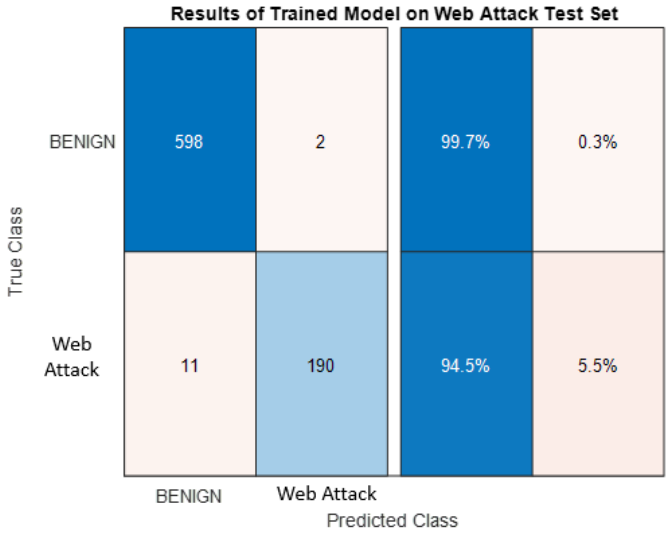

Figure 16. Results from Web Attack-Benign Model.

Figure 17 shows that both brute force and web attack flows are correctly classified by the combined model with an accuracy of greater than 92.0%. From the brute force data, the false positive rate is a maximum of 0.2%. The old data labels were retained intentionally under the “true class” to show that the combined model classified the data as an “active” attack.

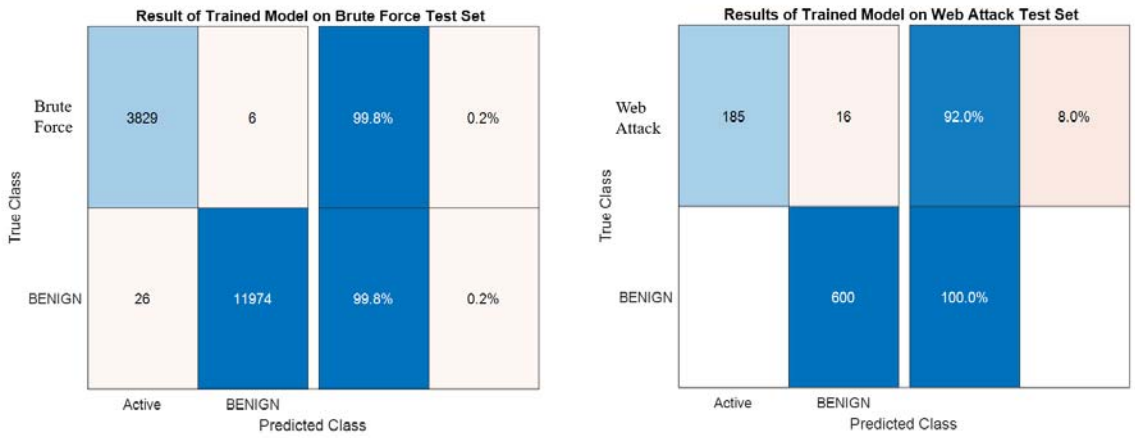

Figure 17. Results from Benign-Active Attack Model.

2. Benign – Probe

This pairwise test needs to prove that probe data can be discerned from benign data. The data used is reflected in Table 6.

Table 6. Data for Benign-Probe Test Pair.

	Training Set	Test Set
Probe	~50000	~50000
Benign	~150000	~150000

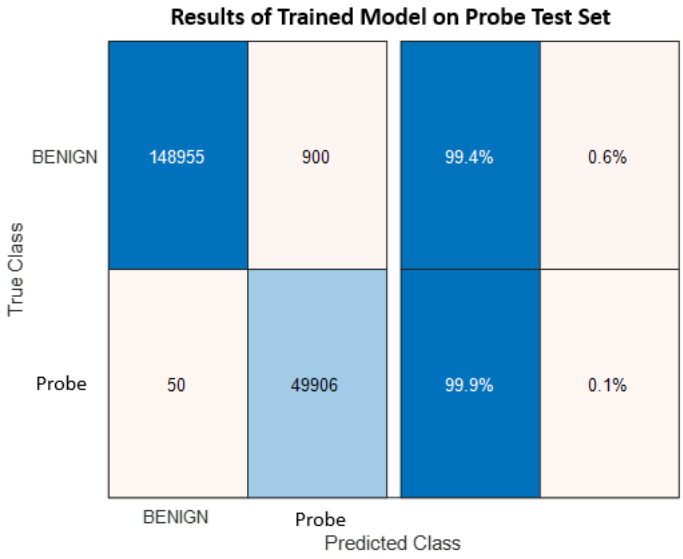

Figure 18. Results from Benign-Probe Attack Model.

Figure 18 shows that the benign-probe attack model has a probability of detection of 99.9% and a false positive rate of 0.6%

3. Benign – Denial

In addition to being able to separate benign from denial data, this pairwise test also needs to prove that the DOS data can be combined with the DDOS data to form an overall "denial" class. The data used is presented in Table 7.

Table 7. Data for Benign-Active Test Pair.

	Training Set	Test Set
DOS	~25000	~25000
Benign	~75000	~75000
DDOS	~25000	~25000
Benign	~75000	~75000
Combined		
Denial	~50000	~50000
Benign	~150000	~150000

Figure 19 shows that the benign-denial attack model has a probability of detection of 95.0% and a false positive rate of 0.1%. There were other observations made through this experimentation:

- If the model was trained with DOS data only, it shows poor results when a DDOS test set is used. The training set must include all types of data.
- The previous point also applies to the various types of DOS programs used. However, with PCA switched off, this requirement seems to be less strict. Figure 19 was produced with random DOS data and PCA off. However, with the same data but PCA on, the accuracy was only about 65%.
- When a properly curated denial training set was used (containing data from all the different DOS labels), the accuracy was greater than 90%.

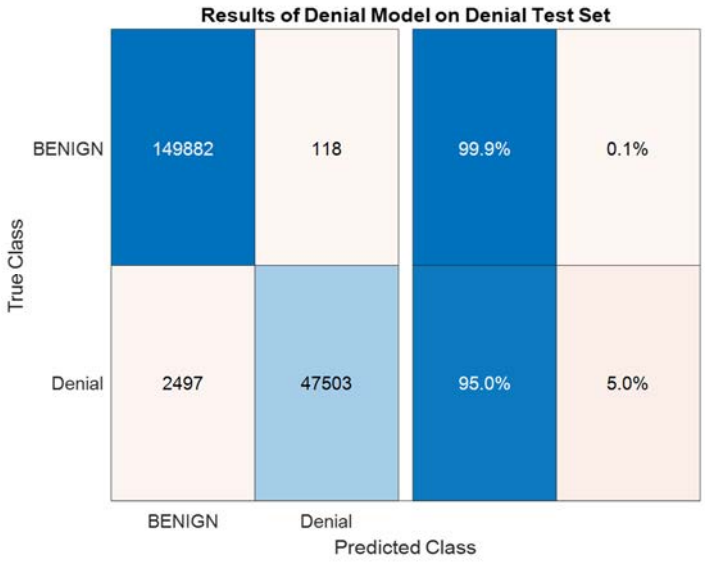

Figure 19. Results from Benign-Denial Attack Model.

B. TRIPLETS

This set of simulations is to ensure that different types of attacks can be discerned from each other by moving to sets of three before simulating the full four attack classes.

1. Benign – Active – Denial

As per the previous section, Table 8 shows the data used and Figure 20 shows the results of the Benign-Active-Denial triplet. The detection rate is at least 93.2% and the false positive rate is 0.4%.

Table 8. Data for Benign-Active-Denial Test Triplet.

	Training Set	Test Set
Active	~14500	~4000
Denial	~50000	~50000
Benign	~193500	~162000

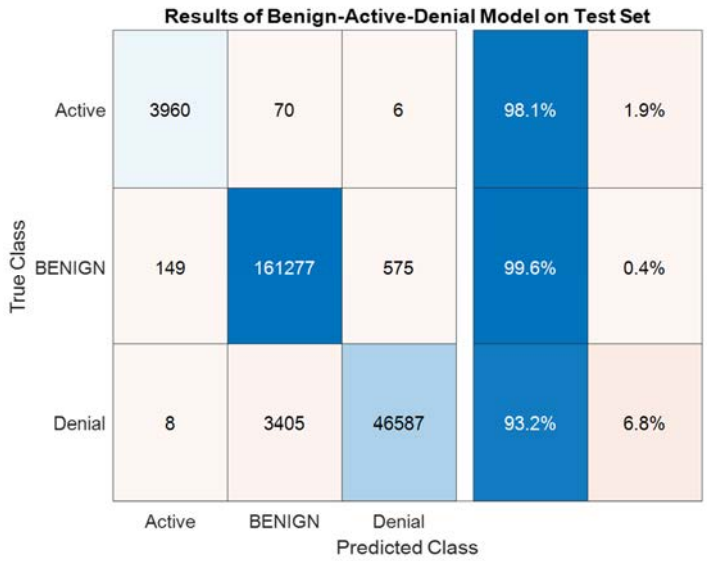

Figure 20. Results from Benign-Active-Denial Attack Model.

2. Benign – Active – Probe

Table 9 shows the data used and Figure 21 shows the results of the Benign-Active-Probe triplet. The detection rate is at least 98.2% and the false positive rate is 0.3%.

Table 9. Data for Benign-Active-Probe Test Triplet.

	Training Set	Test Set
Active	~14500	~4000
Probe	~25000	~25000
Benign	~118500	~87000

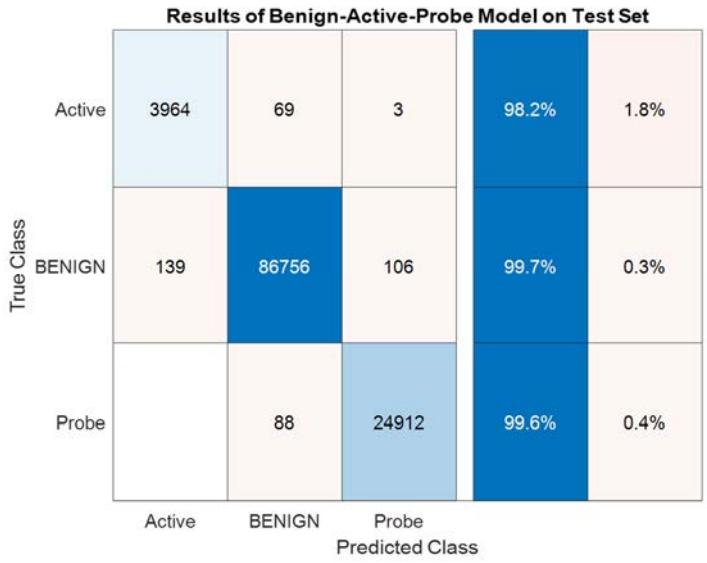

Figure 21. Results from Benign-Active-Probe Attack Model.

3. Benign – Denial – Probe

Table 10 shows the data used and Figure 22 shows the results of the Benign-Denial-Probe triplet. The detection rate is at least 92.9% and the false positive rate is 0.4%.

From this series of simulations, the "Denial" class seems to have the lowest detection rate of the four classes. When the flow is classified wrongly, it is usually classified as "Benign." This could be a function of the method of attack such as replay attacks, which repeat benign packets to achieve the denial effect. While this result warrants further study, a detection rate of above 90% is considered sufficient for this classifier's purpose, particularly when the need to minimize false positives is considered.

Table 10. Data for Benign-Denial-Probe Test Triplet.

	Training Set	Test Set
Denial	~50000	~50000
Probe	~25000	~25000
Benign	~225000	~225000

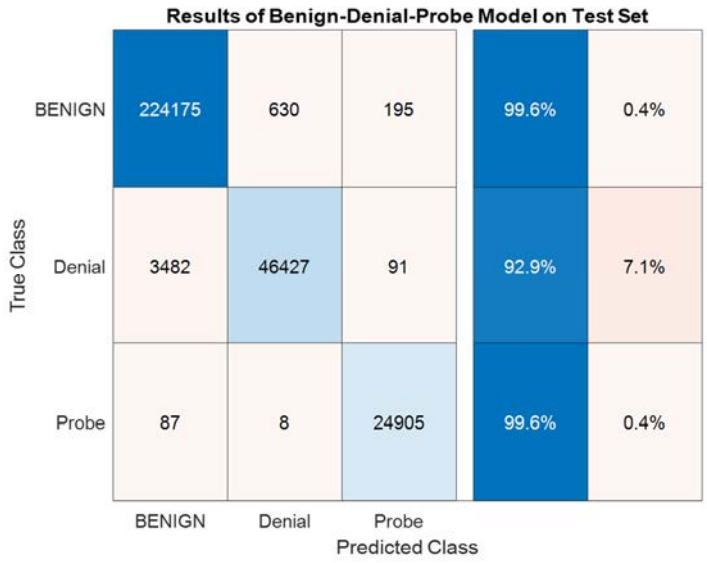

Figure 22. Results from Benign-Denial-Probe Attack Model.

C. COMBINED CLASSIFIER

Table 11 reflects the data used and Figure 23 presents the results for the final simulation where all four classes are combined. Low k values were used throughout sections A and B as the simulations returned sufficiently accurate results. In this section the model was trained for k values of 3, 4 and 5 and $k = 4$ was chosen as the model for 5 did not show any increase in accuracy. The detection rate of the combined classifier is 98.3% (based on the "active" class) and the false positive rate is 1.6%.

Table 11. Data for the Combined Classifier.

	Training Set	Test Set
Active	~14500	~1600
Denial	~50000	~50000
Probe	~25000	~25000
Benign	~268500	~230000

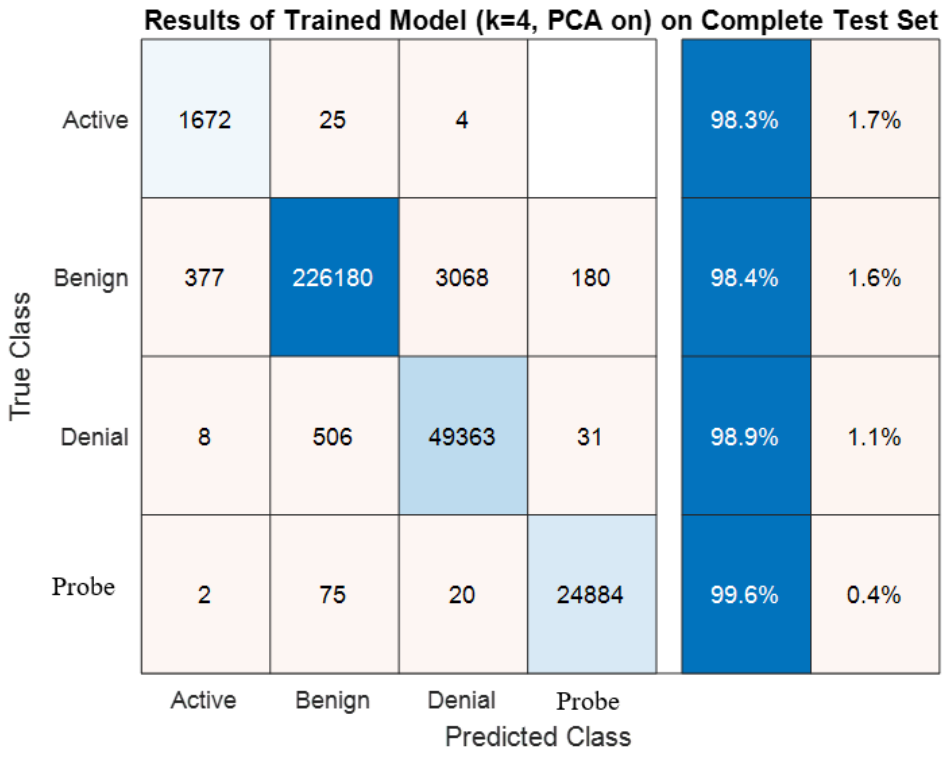

Figure 23. Results from the Combined Attack Model with PCA.

D. WITHOUT PCA

The same data used in section C was used to train another model where PCA was disabled and the k value was maintained at 4. The results are shown in Figure 24. The detection rate rose to 99.3% and the false positive rate dropped significantly to 0.6%. However, as mentioned in the PCA section of Chapter III, the trade-off is time. Using a laptop equipped with an Intel i7 1.8GHz Quad Core processor and 8GB RAM running Windows 10, the *training time* for the model increased from 48 seconds to 13707 seconds.

While that may be acceptable given training could be done offline and infrequently, the *processing time* for the test set also increased from 15.9 seconds to 3838 seconds.

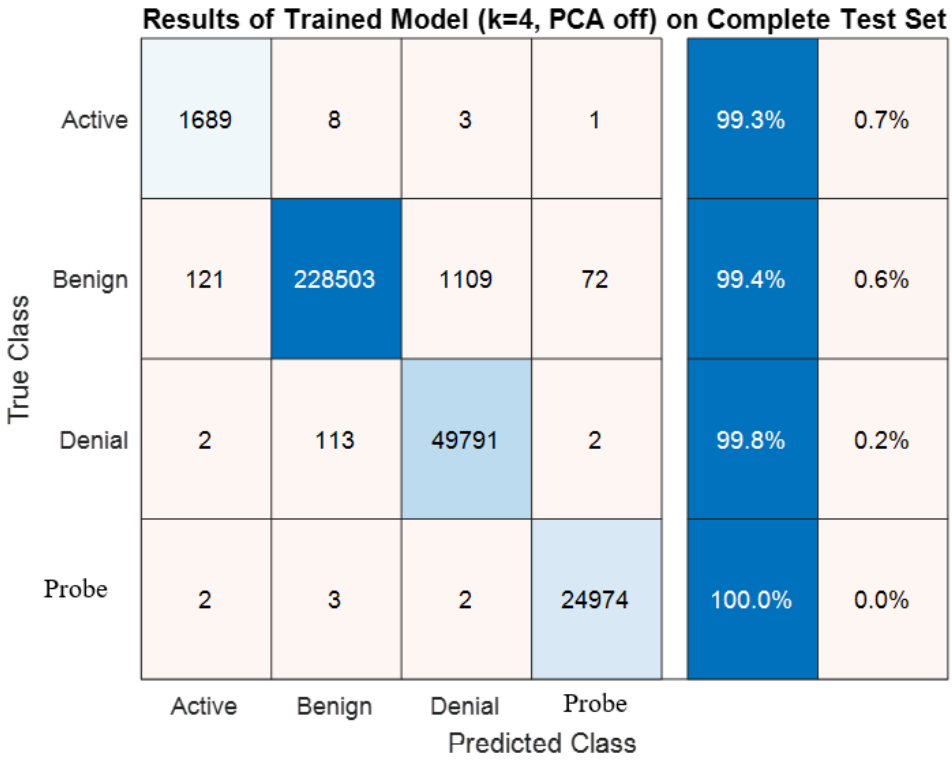

Figure 24. Results from the Combined Attack Model without PCA.

E. INCREASE k

For completeness, the final simulation was re-run with $k=10$ to assess if a larger increase in k value significantly impacts the results. The results are presented in Figure 25. The detection rate increased to 99.8% with a 0% false positive rate. Also, no attack was misclassified (which would cause an erroneous response from the response manager). Increasing k has a significant positive effect on the accuracy of the model, with only a linear training time increase (1.66 times in this case based on Table 2).

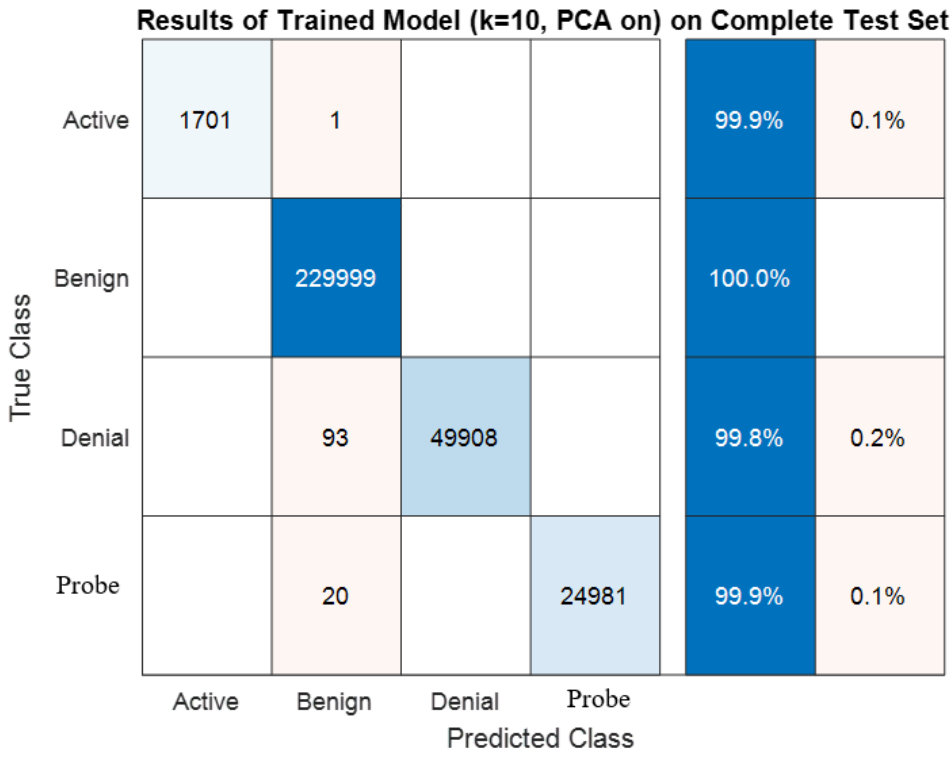

Figure 25. Results from the Combined Attack Model with PCA for $k = 10$.

F. ANALYSIS

As mentioned in the previous chapter, minimizing the false positive rate is the main aim of this system because the response is to be automated. If this is not minimized, then the grid runs the risk of shutting itself down frequently due to false alarms.

The results show that there are two ways to reduce the false positive rate. These methods and the required tradeoffs are summarized in Table 12.

Table 12. Methods to Reduce False Positive Rate.

Method	Effect on Time Taken
Increase k	Linear.
Disable PCA	Non-linear.

1. Increase k

Figure 25 shows significant improvement in false positive rate over Figure 23. Table 2 shows that if k is increased by a factor of n, the processing time is also increased but by a factor smaller than n. In this case, $k=4$ was increased to $k=10$ and therefore the time taken will be 1.66 times that of the original. Also, we can see that if we increase k any further, we will get no improvement of the false positive rate because it is already zero. While the actual smart grid data may not return such ideal numbers, there will be an upper limit to k beyond which the results will show no improvement. Therefore, the value of k should first and foremost be optimized to minimize the false positive rate as it yields the highest return for the extra time spent.

2. Disable PCA

The time taken when PCA was disabled was approximately 240 times that of when it was enabled based on measurements in MATLAB (Section C). MATLAB's PCA algorithm used 6 features to represent the 77 used. Therefore, the additional processing time could be said to be greater than $O\left(n^2\right)$ where the number of features was increased by a factor of n. However, it is harder to quantify the time-accuracy tradeoff for PCA as the processing time for a single data point is insignificant (0.0125s in this case). To recap, Chapter IV mentions that CICFlowMeter packages each TCP exchange or a user defined period as a "flow," and the features for each data point in the data set are produced from this. The number of AMI the concentrator manages, the duration of each exchange and the network access scheme must all be considered. If the additional processing time does not cause other data to be delayed or lost, then PCA should also not be applied to preserve the classifier accuracy.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSIONS

A. SUMMARY AND CONCLUSIONS

As the Navy maintains its technological edge and implements its smart grid to improve the reliability and efficiency of its electricity supply, its cybersecurity defenses must keep pace and be robust. In placing the Navy's critical infrastructure on a network, both state and non-state actors will be interested in what information they gain, and what damage they can cause through cyberattacks.

We have proposed a security system for the Navy smart grid in which an IDS is placed at the data concentrators to enable an automated response, offering the ability to quickly isolate the threat and maintain power availability to critical systems. It is assumed that all hardware before the concentrators may be easily compromised due to the difficulty in physically securing them, as well as the high likelihood that it will communicate via a wireless protocol. The concentrator is therefore the best place to situate a first layer of protection as it can be physically secured, has more computing power, and can maintain a secure communications link with the control center. The IDS that was developed in this thesis consists of a feature extractor, a classifier, an anomaly detector, and a response manager.

We introduced the CICIDS2017 dataset and grouped different attacks based on the perceived objectives. Using MATLAB's machine learning toolbox, we demonstrated that a classifier using the KNN technique is able to segregate the various classes of attack traffic from benign traffic. This implementation will enable the automated response.

Finally, we recommended two methods by which to increase the accuracy of the classifier in order to minimize false positives, and therefore, an unnecessary interruption to the smart grid. These methods are to increase k (specific to KNN) and use the full suite of features provided by the feature extractor. The potential trade off in terms of time was also discussed.

B. THESIS CONTRIBUTIONS

Our objective was to provide a starting point to using machine learning applications for the Navy smart grid. In this thesis we have contributed the following:

- Proposed the implementation of an IDS for the NAVFAC smart grid that consists of a feature extractor, classifier, anomaly detector, and response manager. We showed that is system implementation allowed the grid to react according to the cyber threat.

- Introduced the CICIDS2017 data set and the use of the open source CICFlowMeter for feature extraction.

- Proven that different attacks can be grouped together based on objectives, and that KNN is able to segregate the various classes of attack traffic from benign traffic.

C. RECOMMENDATIONS FOR FUTURE WORK

We conducted our simulations based on a publicly available dataset, generated by an experimental setup. The next step will be to show that similar techniques can be applied specifically to the Navy smart grid. To that end, the following steps are recommended:

1. Build a smart grid test bed. A scaled down version of the actual network architecture and communications protocols used will go a long way to generating actual data that can be used for further simulations and evaluating the effectiveness of PCA. A test bed will also allow researchers to conduct attacks on the grid without causing actual damage and produce valuable attack data for a classifier.

2. Manual feature selection. Once actual data is produced, the IDS can be optimized. MATLAB's PCA algorithm derives 6 features to represent 77 with a lowered accuracy. It is likely that a compromise can be found such that a sufficiently high accuracy can be obtained with less features, but without the use of PCA. This will involve determining which features

have no bearing on the outcome of the algorithm and manually removing them from the computations.

3. Investigate methods to reduce the false positive rate. Reducing the false positive rate is critical to the success of a fully automated response manager. With the actual data and communications architecture, the efficacy of various techniques to reduce the false positive rate such as requiring consecutive true positives to trigger a response, or running the same data through multiple ML classifiers can be studied.

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX. USING THE MATLAB MACHINE LEARNING TOOLBOX

This appendix aims to provide a step by step guide to using the MATLAB machine learning toolbox. The pictures are provided using version 2019a.

Step 1. After the MATLAB program is loaded, click on the “Apps” tab and locate “Classification Learner”

Figure A1. The MATLAB Apps Bar.

Step 2. After the Classification Learner is loaded, click the “New Session” dropdown and select “From File”

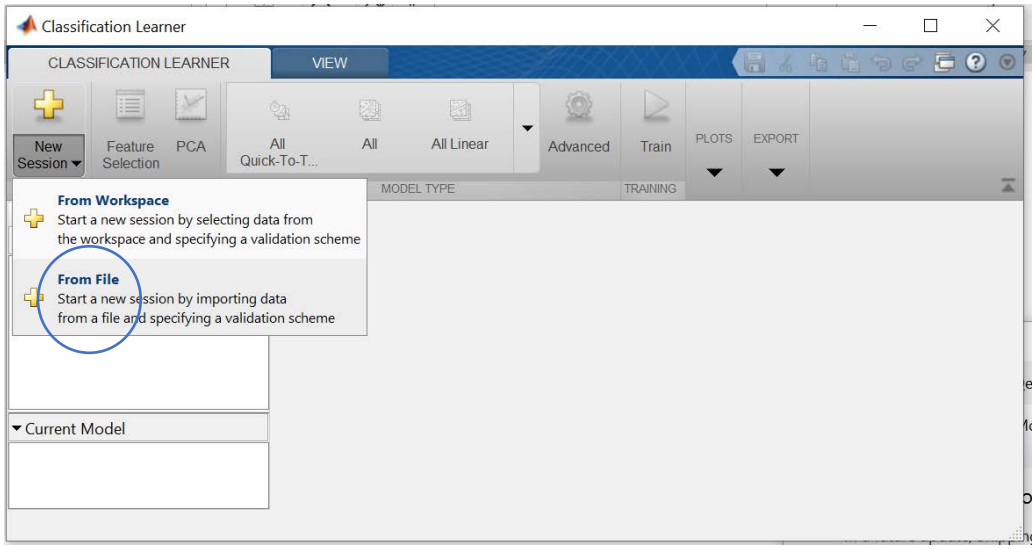

Figure A2. Starting a New Session.

Step 3. Choose your training set, and the file import window will open. It is recommended that you select “Exclude rows with” under “unimportable cells..” Then click “Import Selection”

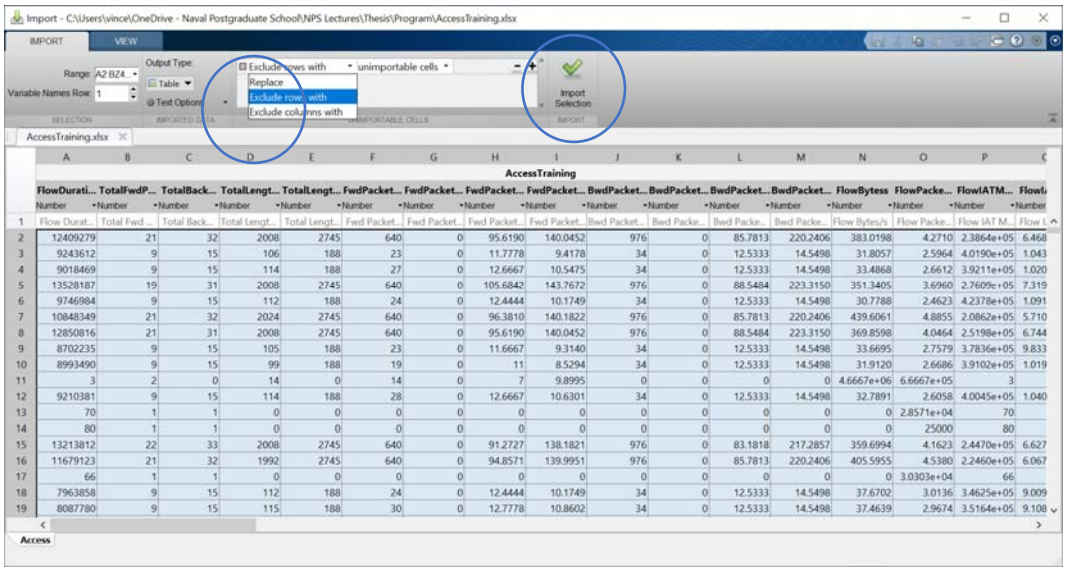

Figure A3. Importing Data in MATLAB.

Step 4. The data set window will open. Set your validation settings then click “Start Session.”

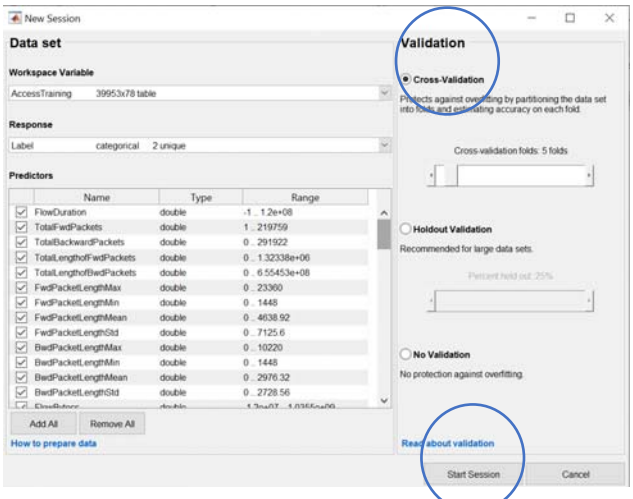

Figure A4. Validation Method Selection.

Step 5. In the main drop-down window. Select your ML technique.

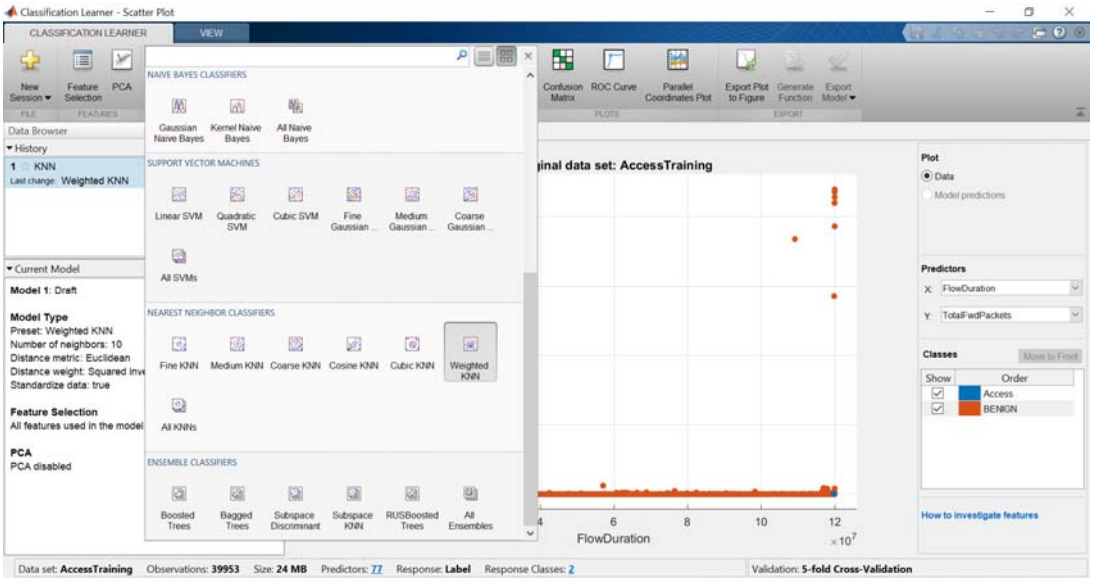

Figure A5. Selecting the Machine Learning Technique.

Step 6. Click “Advanced” and “PCA” to input the respective settings, then click “Train.”

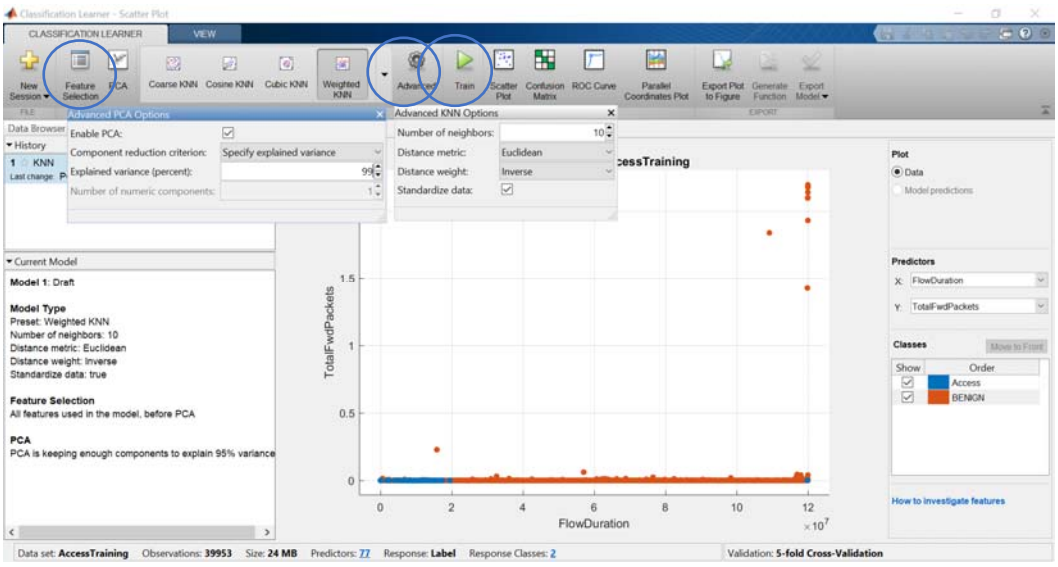

Figure A6. Training the ML Algorithm.

Step 7. The left panels display information about settings and timing. The training set confusion matrix can be called to see the training accuracy. Once satisfied, click on the "Export Model" drop down and click "Export Model," you will be prompted to give the model a name.

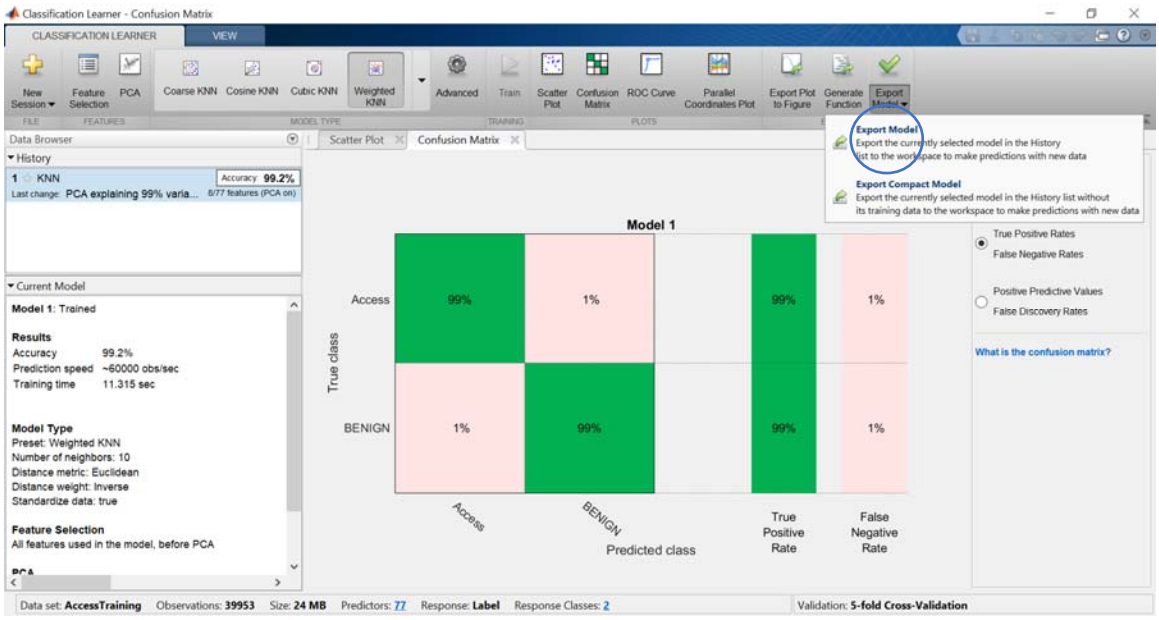

Figure A7. Exporting the Model.

Step 8. A workspace object will be generated with that name. On the "Home" tab select "Import Data" and repeat step 3 with your test set.

Figure A8. Importing Data into the Workspace.

Step 9. Run the following code. Replace the text highlighted in red with the appropriate workspace object name for the test set, and the orange with the model workspace object name. Edit the title as appropriate.

```
yfit1 = Activek3Poff.predictFcn(AccessTest);
[m1,order1] = confusionmat(AccessTest.Label,yfit1);
figure(1)
confusionchart(m1,order1, 'Title','Result of Trained Model
on Brute Force Test Set', 'RowSummary','row-normalized')
```

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

[1] K. Schwab, *The Fourth Industrial Revolution*. Switzerland: World Economic Forum, 2016.

[2] M. Lorenz et al., "Industry 4.0: The future of productivity and growth in manufacturing industries," Boston Consulting Group, April 9, 2015. [Online]. Available: https://www.bcg.com/publications/2015/engineered_products_project_business_industry_4_future_productivity_growth_manufacturing_industries.aspx

[3] Department of Energy, "The smart grid." Accessed 23 June, 2020. [Online]. Available: https://www.smartgrid.gov/the_smart_grid/smart_grid.html.

[4] G. Horst and R. Dugan, "Virtual Power Plant Modelling and Simulation," Electric Power Research Institute, 2010. [Online]. Available: https://smartgrid.epri.com/doc/07_AEP%20Smart%20Grid%20Project%20Modeling%20and%20Simulations.pdf

[5] N. Saxena and B. J. Choi, "A hybrid framework for evaluating intrusion detection in the AMI network," presented at the International Smart Grid Conference (ISGC), Gwangju, Korea, Oct. 20–23, 2015.

[6] Guide to Industrial Control Systems (ICS) Security, NIST Special Publication 800-82 rev. 2., Secretary of Commerce, Gaithersburg, MD, USA, 2015. [Online]. Available: http://dx.doi.org/10.6028/NIST.SP.800-82r2

[7] Symantec, "Internet security threat report," vol. 24, February 2019. [Online]. Available: https://docs.broadcom.com/doc/istr-24-2019-en

[8] Navy and Marine Corps Smart Grid CDD Industry Version. Commander, NAVFAC, Washington, DC, USA, 2014.

[9] US Navy Public Affairs, "NAVFAC Reaches Key Milestone in Deploying New Smart Energy Monitoring and Control Solution," Apr 30, 2019. [Online]. Available: https://www.navy.mil/submit/display.asp?story_id=109430#

[10] L. Wan, Z. Zhang, and J. Wang, "Demonstrability of Narrowband Internet of Things technology in advanced metering infrastructure," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 1–12, 2019.

[11] M. Bellias, "3 ways IoT will change smart meters for utilities," IBM, Dec. 1, 2016. [Online]. Available: https://www.ibm.com/blogs/internet-of-things/smart-meter-grid/

[12] P. Prakash, “Data concentrators: The core of energy and data management,” Texas Instruments, February 2018. [Online]. Available: https://www.ti.com/lit/wp/spry248a/spry248a.pdf

[13] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, “Communication network requirements for major smart grid applications in HAN, NAN and WAN,” *Computer Networks*, vol. 67, pp. 74–88, 2014.

[14] Z. E. Mrabet, H. E. Ghazi, N. Kaabouch, and H. E. Ghazi, “Cyber-security in smart grid: Survey and challenges,” *Computers & Electrical Engineering*, vol. 67, pp. 469–482, 2018. [Online]. doi: https://doi.org/10.1016/j.compeleceng.2018.01.015

[15] Lockheed Martin. “The Cyber Kill Chain,” 2011. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[16] Z. Heng, C. Peng, S. Ling, and C. Jiming, “Optimal DoS attack policy against remote state estimation,” *52nd IEEE Conf. on Decision and Control*, pp. 5444–5449, 2013.

[17] Y. Yuan, Z. Quanyan, S. Fuchun, W. Qinyi, and T. Basar, “Resilient control of cyber-physical systems against Denial-of-Service attacks,” *6th International Symposium on Resilient Control Systems*, pp. 54–59, 2013.

[18] N. Beigi, J. Mišić, V. Misic, and H. Khazaei, “A framework for intrusion detection system in advanced metering infrastructure,” *Security and Communication Networks*, vol. 7, pp. 195–205, 2014.

[19] A. Diaz and P. Sanchez, “Simulation of attacks for security in wireless sensor network,” *Sensors*, vol. 16, no. 11, p. 1932, 2016.

[20] L. Gaoqi, Z. Junhua, L. Fengji, S. R. Weller, and D. Zhao Yang, “A Review of False Data Injection Attacks Against Modern Power Systems,” *IEEE Transactions on Smart Grid*, vol. 8, no. 4, pp. 1630–1638, 2017.

[21] E. D. Knapp and R. Samani, Applied Cyber Security and the Smart Grid : Implementing Security Controls into the Modern Power Infrastructure. Waltham, MA, USA: Elsevier Inc, 2013.

[22] R. Baldick *et al.*, “Initial review of methods for cascading failure analysis in electric power transmission systems IEEE PES CAMS task force on understanding, prediction, mitigation and restoration of cascading failures,” *2008 IEEE Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century*, pp. 1–8, 2008.

[23] D. Kushner, “The real story of stuxnet,” *IEEE Spectrum*, vol. 50, no. 3, pp. 48–53, 2013.

[24] Framework for Improving Critical Infrastructure Cybersecurity, Secretary of Commerce, Gaithersburg, MD, USA, 2014. [Online]. Available: https://doi.org/10.6028/NIST.CSWP.04162018

[25] A. L. Samuel, "Some Studies in Machine Learning Using the Game of Checkers," *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959.

[26] I. Amit, J. Matherly, W. Hewlett, Z. Xu, Y. Meshi, and Y. Weinberger, "Machine learning in cyber-security - problems, challenges and data Sets," *The AAAI-19 Workshop on Engineering Dependable and Secure Machine Learning Systems*, 2019.

[27] Y. El, A. Toumanari, A. Bouirden, and N. El, "Intrusion detection techniques in wireless sensor network using data mining algorithms: comparative evaluation based on attacks detection," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 9, 2015.

[28] M. C. Belavagi and B. Muniyal, "Performance evaluation of supervised machine learning algorithms for intrusion detection," *Procedia Computer Science*, vol. 89, pp. 117–123, 2016.

[29] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters," *Twenty-eighth Australasian conference on Computer Science*, Newcastle, Australia, vol. 38, 2005.

[30] M. Z. Alom and T. M. Taha, "Network intrusion detection for cyber security using unsupervised deep learning approaches," *2017 IEEE National Aerospace and Electronics Conference (NAECON)*, pp. 63–69, 2017.

[31] B. Radford, L. Apolonio, A. Trias, and J. Simpson, "Network traffic anomaly detection using recurrent neural networks," presented at 2017 National Symposium on Sensor and Data Fusion, 2017.

[32] Mathworks. "Machine Learning." Accessed Jul. 14, 2020. [Online]. Available: https://www.mathworks.com/discovery/machine-learning.html

[33] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153–1176, 2016.

[34] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[35] K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, "When is "nearest neighbor" meaningful?," *Database Theory – Icdt'99*, vol. 1540, pp. 217–235, 1999.

[36] Mathworks. “PCA.” Accessed Jul. 28, 2020. [Online]. Available: https://www.mathworks.com/help/stats/pca.html

[37] P. Sharma, G. Bhagwatikar, and A. Kumar, “Wireless sensor network based smart grid communications: Cyber attacks, intrusion detection system and topology control,” *Electronics*, vol. 6, no. 1, p. 5, 2017.

[38] P. Laskov, P. Düssel, C. Schäfer, and K. Rieck, “Learning intrusion detection: supervised or unsupervised?” in *Lecture Notes in Computer Science*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 50–57.

[39] Y. Liao and V. R. Vemuri, “Use of K-Nearest Neighbor classifier for intrusion detection,” *Computers & Security*, vol. 21, no. 5, pp. 439–448, 2002.

[40] R. Kumar, A. Wicker, and M. Swann, “Practical machine learning for cloud intrusion detection: Challenges and the way forward,” 2017.

[41] I. Sharafaldin, A. Habibi Lashkari, and A. Ghorbani, “Toward generating a new intrusion detection dataset and intrusion traffic characterization.” *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, pp. 108–116, 2018.

[42] I. Sharafaldin, A. Gharib, A. Habibi Lashkari, and A. Ghorbani, “Towards a reliable intrusion detection benchmark dataset,” *Software Networking*, vol. 2017, pp. 177–200, 2017.

[43] R. Panigrahi and S. Borah, “A detailed analysis of CICIDS2017 dataset for designing intrusion detection systems,” *International Journal of Engineering & Technology*, vol. 7, pp. 479–482, 2018.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California