SECURITY SOLUTIONS FOR SUPERVISORY CONTROL AND DATA ACQUISITION (SCADA) NETWORKS IN INDUSTRIAL CONTROL SYSTEMS

by

Darshana Upadhyay

Submitted in partial fulfillment of the requirements
For the degree of Doctor of Philosophy

at

Dalhousie University
Halifax, Nova Scotia
April 2023

I dedicate this research affectionately to
my daughter, my mom & my mother-in-law

Contents

List of Tables . . . . . **vi**

List of Figures . . . . . **vii**

Abstract . . . . . **x**

List of Abbreviations . . . . . **xi**

Acknowledgements . . . . . **xii**

Chapter 1 Introduction **1**

1.1 Overview of SCADA Networks for Industrial Control Systems 1

1.2 Security of SCADA Systems 3

1.3 Motivation 5

1.4 Objective 6

1.5 Major Contributions 6

1.6 Report Organization 9

Chapter 2 Background and Related Work **10**

2.1 Literature review on Vulnerability assessment of SCADA Components (Module 1) 10

2.2 Literature review on Intrusion Detection Techniques (Module 2) . . . 11

2.3 Literature review on SCADA Communication Algorithms (Module 3) 13

Chapter 3 Vulnerability & Risk assessment of SCADA Components - Methodology, Experiments & Result discussion . . . **17**

3.1 Summary of the chapter 17

3.2 Vulnerability Assessments of Onion Omega2 18

3.2.1 Vendor Level Fixes 18

3.2.2 Network/Administrator Fixes 19

3.3 Generic Vulnerability Assessment 19

Chapter 4 Intrusion Detection Systems for SCADA based power grid - Methodology, Experiments & Result discussion 23

4.1 Summary of the chapter 24

4.2 Framework Design of proposed IDSs 25
4.2.1 Plant floor IDS: Proposed framework of GBFS-based WFI Scoring model for Tree-based classifiers 25
4.2.2 Control center IDS: Proposed Framework based on RFE-XGBoost filtering model along with majority vote ensemble method . . 26
4.2.3 Intermediate SCADA Center IDS: Proposed Framework based on semi-supervised ML approach using the concept of autoencoders for zero-day attacks 28

4.3 Dataset Description 29
4.3.1 Power system Dataset 29
4.3.2 Gas pipeline Dataset 32
4.3.3 Water Storage Dataset 33

4.4 Experiments and Results 33
4.4.1 IDS for Plant Floor 33
4.4.2 IDS For Control Center 39
4.4.3 IDS for Intermediate SCADA Centers (sub-MTUs) 44

4.5 Proposed IDS framework for Power Grid SCADA System 54

Chapter 5 Robust & Secure Solution for SCADA communication - Methodology, Experiments & Result discussion 56

5.1 Summary of the chapter 56

5.2 Multi-layered Framework for Secure SCADA Communication 57

5.3 Secure Key Exchange 58
5.3.1 Key Generation 58
5.3.2 Key Distribution 60
5.3.3 Key Extraction 60

5.4 Secure Information Exchange 62
5.4.1 Hybrid Multi-layered Architecture 63
5.4.2 Prime Counter 64
5.4.3 Hash Chaining 65

5.5 Experiments & Result Discussion 65
5.5.1 Algorithm selection of cipher suite for proposed framework . . 65
5.5.2 Computational speed of proposed framework 68
5.5.3 Calculation of Key storage cost & randomness evaluation . . . 69

5.6 Performance Analysis 71
5.6.1 Security Analysis 71
5.6.2 Storage cost 72
5.6.3 Execution speed 73

5.7 Deployment of SCADA Security solution on a Hardware Test bench . 73
5.7.1 Case Study 1: Controlling Servo Motor remotely using VTSCADA and MQTT Protocol 76
5.7.2 Case Study 2: Water Distribution System 77
5.7.3 Case Study 3: Voltage Level Indicators for Power system . . . 78
5.7.4 Performance evaluation of proposed security methods on Raspberry Pi 80

Chapter 6 Concluding Remarks 84

Bibliography 86

Appendix A Selected Publications from the thesis 93

List of Tables

4.1	Description of the output labels of the various categories of the datasets	32
4.2	Description of features	33
4.3	Main Features of Gas pipeline and Water storage dataset	34
4.4	Performance evaluation metrics of Proposed GBFS Based Classifier	40
4.5	Comparison of accuracy of majority vote ensemble algorithm with and without recursive feature elimination based feature selection	46
4.6	Hyper-tuning parameters of CTGAN model	47
4.7	Experimentation results summary computed using Variational Autoencoders	55
4.8	Comparative analysis of various methods	57
5.1	Comparative analysis of various Hash Functions	68
5.2	Comparative analysis of computational speed of various symmetric key algorithms	69
5.3	Comparative analysis of the computational speed of various public key cryptography	69
5.4	Considerable parameters of different cryptographic components	70
5.5	Total execution time calculation	71
5.6	Total Execution Time in Seconds	71
5.7	Comparative analysis of storage cost of keys	74
5.8	Comparative Analysis of various cipher suites	75
5.9	Total execution time of proposed security algorithms on Raspberry Pi	84
5.10	Computational speed of various phases of proposed security algorithms on Raspberry Pi (in milli seconds)	84
5.11	Comparison of memory utilization of proposed security algorithms on Raspberry Pi (in bytes)	84

List of Figures

1.1 Block diagram of a SCADA system, Legend: MTU: Master Terminal Unit, PLCs: Programmable Logic Controllers, RTUs: Remote Terminal Units, IEDs: Intelligent Electronic Devices . 4

1.2 Major Contributions of the Ph.D. work 9

2.1 Major Contributions of the proposal to ensure security of SCADA systems 14

3.1 Onion Omega2 vulnerability and mitigation (vendor fixes) . . 23

3.2 Onion Omega2 vulnerability and mitigation (vendor fixes) . . 24

3.3 Generic framework of vulnerability assessment process 24

4.1 Framework for a GBFS Based Intrusion Detection System at Plant Floor 28

4.2 proposed framework for intrusion detection of power grids at control center 29

4.3 proposed framework for intrusion detection of power grids at intermediate SCADA center 30

4.4 Represents the relative importance of each attribute of the dataset with 5000 records; computed by considering four estimators Num_trees = 100,500,700,1000 37

4.5 Best 15 Features of 15 Datasets for all the four categories - Binary, Three classes, Seven classes, and Multi-class 37

4.6 Comparative view of Different Machine Learning Classifiers for - four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets 40

4.7 Comparative view of Execution speed of Three GBFS-based Random Forest variances to classify normal and attack events for four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets 41

4.8 Comparative analysis of different features to evaluate the accuracy using RFE-XGBoost WFI scoring model. 43

4.9	Comparative view of different Machine Learning classifiers for four categories for each of the fifteen datasets	44
4.10	Precision-Recall Curves of RFE-based Majority vote ensemble method for four categories	45
4.11	Representation of power grid's data consist of the real and synthetic dataset (Fine-tuned post processed synthetic dataset)	47
4.12	Models train with normal data and inference is made on normal and attack data	49
4.13	Vanilla Autoencoder - Using Real Power-grids Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space	50
4.14	Vanilla Autoencoder - Using Synthetic Power-grids Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space	50
4.15	Vanilla Autoencoder - Using Real Gas Pipeline Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space	51
4.16	Vanilla Autoencoder - Using Synthetic Gas Pipeline Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space	51
4.17	Vanilla Autoencoder: Using Real Dataset of Water Storage System: Error distribution of error reconstruction of normal and Attack events with 2 and 16 cells in the latent space . . .	53
4.18	Vanilla Autoencoder: Using Synthetic Dataset of Water Storage System: Error distribution of error reconstruction of normal and Attack events with 2 and 16 cells in the latent space . . .	53
4.19	IDS framework for real-time SCADA systems for power grids .	56
5.1	Multi-layered framework for secure SCADA communication . .	59
5.2	Secure Key exchange mechanism for SCADA systems	61
5.3	Complete process diagram of secure communication between MTU and RTU	65
5.4	Randomness assessment of symmetric key	72
5.5	Test bench design for secure SCADA communications	77

5.6	Flow Diagram of Proposed Framework for SCADA Test bench	77
5.7	Layout of testbed	78
5.8	Experimental Setup	79
5.9	Sensors and Microcontrollers	80
5.10	Human Machine Interface (VTSCADA)	80
5.11	Modbuspal Simulator - Register values	81
5.12	Automation tool for input Voltage generator	81
5.13	HMI view of Voltage Indicator on VTSCADA	82
5.14	Modbus traffic captured using loopback address	82
5.15	Encyption applied on PLC data	83
5.16	Decryption of received data at MTU	83
5.17	Average execution time of proposed security module on Raspberry Pi	85
A.1	Publication history based on SCADA Security modules along with impact factor	95

Abstract

Supervisory Control and Data Acquisition (SCADA) networks play a vital role in Industrial Control Systems (ICSs). Industrial organizations perform operations remotely through SCADA systems to accelerate their processes. However, these network capabilities come at the cost of exposing the systems to cyber-attacks. Consequently, effective solutions are required to detect intrusions and secure SCADA systems as cyber-attacks on industrial infrastructure can have catastrophic consequences. Furthermore, SCADA field devices are equipped with micro-controllers for processing information and have limited computational power and resources. As a result, lightweight cryptography solutions are needed to strengthen the security of industrial plants against cyber threats. The Ph.D. work focuses on three major elements to secure the SCADA-based ICSs, namely, vulnerability assessment of field-site components, Intrusion Detection Systems (IDSs) for plant floor and control center, and robust cryptographic-based secure solution for SCADA communications. The overall goal of this thesis is to cover the landscape of SCADA weaknesses by providing efficient, lightweight, and robust solutions to strengthen the security of industrial applications. The thesis has the following major contributions:

- Module 1 : A thorough vulnerability analysis of industrial infrastructure has been made and recommendations have been provided by considering real incidents reported in vulnerability databases. Penetration testing has been carried out on one of the SCADA components, namely, Onion Omega2 (System-on-a-Chip).

- Module 2 : An integrated model of IDS framework for real-time SCADA systems has been proposed using defense-in-depth architecture by considering power grids as a candidate ICS application. In this approach, the displacement of three IDSs has been proposed, one at the plant floor using Gradient Boosting Feature Selection (GBFS) based filtering model to detect the intrusions in real-time, another at the control center using the majority vote-based ensemble method for accurate prediction. We propose an IDS at intermediate SCADA (placed at sub-MTUs) using Variational Autoencoder (VAE) based on semi-supervised learning to detect zero-day attacks. To improve the efficiency of the model, high-quality synthetic datasets were generated for SCADA-based power grids.

- Module 3 : A robust and low-cost security framework for SCADA has been proposed to mitigate cyber-attacks. The security model is based on a multi-layer framework that integrates both symmetric and asymmetric key cryptosystems. A SCADA hardware test bench is developed to experimentally evaluate the proposed framework.

The contributions of this thesis fulfill the objective of providing a full life-cycle strategy to innovate, design, and implement a security framework to protect SCADA networks against cyber-attacks in industrial control systems.

List of Abbreviations

ASKMA	Advanced SCADA Key Management Architecture
FPR	False Positive Rate
GBFS	Gradient Boosting Feature Selection
HKMA	Hybrid Key Management Architecture
HMI	Human Machine Interface
ICSs	Industrial Control Systems
IDSs	Intrusion Detection Systems
IEDs	Intelligent Electronic Devices
KDC	Key Distribution Center
MTU	Master Terminal Unit
PLCs	Programmable Logic Controllers
RFE	Recursive feature elimination
RSA	Rivest–Shamir–Adleman
RTUs	Remote Terminal Units
SCADA	Supervisory Control and Data Acquisition
SKE	SCADA Key Establishment
SKMA	SCADA Key Management Architecture
WFI	Weighted Feature Importance

Acknowledgements

Words can't adequately express my respect and gratitude to my supervisor, Dr. Srinivas Sampalli! I can simply connect him with the well-known Sanskrit prayer " *GururBrahma, GururVishnu, GururDevo Maheshwaraha, GururSaakshaat Parah Brahma, Tasmai Sri Gurave Namaha* ": like lord Brahma (the creator), he nurtures values and morality in my life, like Vishnu (the lord of perseverance), he implants determination staying focused on my goals, and as Shiva (the lord of destruction), he dispels the darkness of ignorance and leads me on the path to enlightenment. I bow to my mentor, the Supreme Being right before my eyes. I am extremely grateful to him for his invaluable advice, continuous support, and patience during my PhD study. His immense knowledge and plentiful experience have encouraged me in all the time of my research and daily life. He has made a profound impact on my professional growth, and I am genuinely thankful for all his support and trust in me.

I greatly acknowledge the support of NSERC for the CRD grant and industrial partners Cistel technology and technology Sanstream for the collaborative research opportunity. My gratitude extends to all the researchers of Cistel technology for their technical expertise and advice. I would particularly like to single out Dr. Marzia Zaman, Cistel technology, for her invaluable suggestions in formulating the research questions and methodology. Her insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

Additionally, this endeavor would not have been possible without the generous support of Dr. Jaume Manero, from the Technical University of Catalonia, Spain, and Dr. Hiroyuki Ohno, from Kanazawa University, Japan. Their motivation always models the strength and positivity of the mind. It has been always a pleasure working with Dr. Manero, especially his technical depth in the domain of Machine Learning and Deep Learning helped me a lot. I am grateful to Dr. Ohno, for all his support in helping me understand the concepts of embedded systems and for his valuable guidance in implementing a testbed for my research work.

I also could not have undertaken this journey without my defense committee,

who generously provided knowledge and expertise. I would like to offer my deepest gratitude to my committee members, Dr. Zincir-Heywood, Dr. Keselj, and Dr. Manero. I would also like to thank my master thesis mentors, Dr. Priyanka Sharma and Dr. Sharada Valliveti, from Nirma University, India, who planted the seeds of research in me.

Special thanks also go to the anonymous reviewers of my papers that were published during this journey. Their suggestions and comments contributed to improving my work. My sincere gratitude to Dr.Mike McAllister, Graduate Coordinator, for his valuable guidance. I would also like to thank the Faculty Graduate Administrator Ms.Vidhya Ramamoorthy for making the whole process easier. Special thanks to Ms. Megan Baker for her kind cooperation during my teaching and research assistantship. Also to Ms. Helena Martel from the Faculty of Graduate Studies, who worked tirelessly to coordinate and help with my thesis defense and submission.

I want to immensely thank my husband, Pratik, for his moral support and wise counsel. My deep love to both my children, Darpi & Darpan. They have made me stronger, better, and more fulfilled than I could have ever imagined. I love them to the moon and back. Their belief in me has kept my spirits and motivation high during this process.

Well, if you ever want to feel unconditional love, look at your parents' eyes! My heartfelt thanks to my parents, Urmila Upadhyay, and Pramod Upadhyay, who have made huge sacrifices for my growth. Their inspiration, kindness, and vision have encouraged me to work hard. I also want to thank my father-in-law, Harekrushna Upadhyay, and my sister-in-law, Kiran Jitendra Vyas, for their everlasting support and motivation.

Last but not least, thank you to my lovely friend, Ms. Rinki Parikh, who stood beside me during my hard times and always motivates me. I would also like to thank my cheerful friends, Sagarika, Deborah, Nupur, Qiaodan, Sweta, and Meera for sharing happy moments to rest my mind outside of my research. I would like to thank all the MYTech research group members for bringing joy by sharing and celebrating every happy event with me.

Thank you, God, for the life you have blessed me with such beautiful surroundings!

Chapter 1

Introduction

1.1 Overview of SCADA Networks for Industrial Control Systems

There has been a surge in the deployment of Supervisory Control and Data Acquisition (SCADA) systems to control and monitor industrial infrastructure over the Internet [1]. Organizations such as oil and natural gas, power stations, water & sewage systems, chemical plants, manufacturing units, railway, and other transportation use SCADA systems to monitor and control their infrastructure such as oil pipelines, solar panels, water pipelines, boilers, railway tracks, and plant floor components across open access networks [2, 3].

A SCADA system typically includes a control server (also known as Master Terminal Unit (MTU)), SUB-MTUs, communication links (e.g. satellite, radio or microwave links, cellular network, switched or lease lines and powerlines), and geographically dispersed field control devices, namely, Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs) [2, 4]. The block diagram of a typical SCADA system is depicted in Figure 1.1.

For continuous monitoring and control of plant floor devices, sensors, and actuators are used to measure different attributes of machinery and transmit that information to field devices [5]. Further, the field control devices, namely, PLCs, RTUs, and IEDs supply digital status information to the MTU (typically placed at the remote location) to determine the acceptable ranges according to parameters set in the server. This information will then be transmitted back to the field control device(s) where actions may be taken to optimize the performance of the system. Moreover, the status information is stored in a database and is displayed on a Human Machine Interface (HMI) at the control center, where operators can interact with the plant floor machinery for centralized monitoring and system control [6]. Large SCADA networks such as those on a power plant require hundreds of field devices and dedicated subsystems to reduce the load on the centralized server [2].

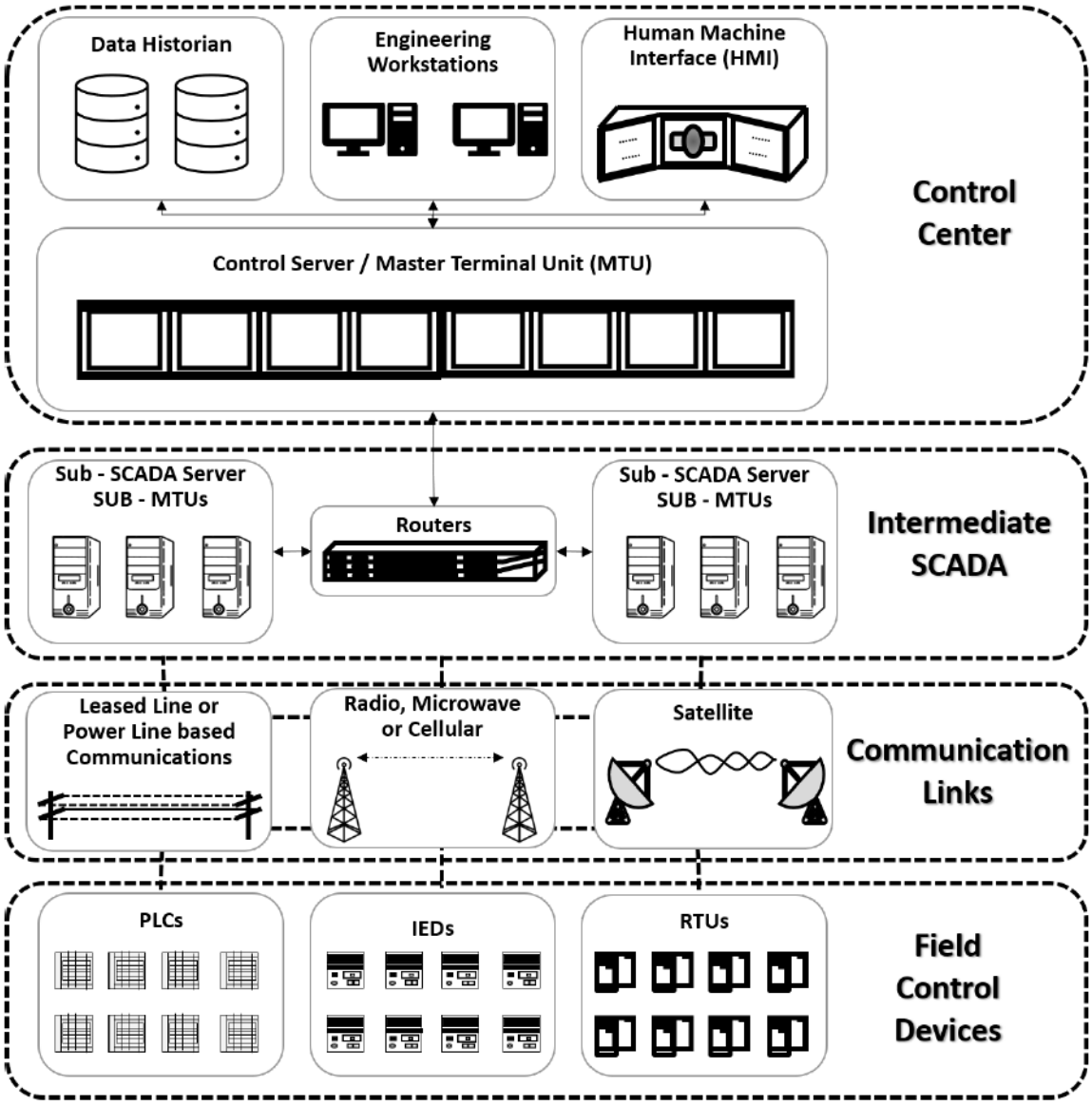


Figure 1.1: Block diagram of a SCADA system, Legend: MTU: Master Terminal Unit, PLCs: Programmable Logic Controllers, RTUs: Remote Terminal Units, IEDs: Intelligent Electronic Devices

1.2 Security of SCADA Systems

SCADA communication messages have sensitive information as they are used to monitor and control the plant floor devices. For example, in water and sewage systems, communication messages are used to raise and lower water tank levels or open and close the safety valves. Since these control devices are operated and monitored remotely, they can make them high-value targets for attackers to launch various cyber-attacks that can compromise the control systems, communication, and emergency services. Consequently, one of the critical aspects of the SCADA systems is the secure transmission of messages so that they cannot be tampered during the communication. Moreover, the SCADA devices must be authenticated and maintain confidentiality of the information during the transmission so that no interceptor can misuse the system.

As mentioned above, there are various SCADA-based ICS applications, such as gas refineries, water & sewage systems, power plants, railway monitoring systems, etc. We have considered power grids application for the development of intrusion detection systems for SCADA systems. Originally, power grids were designed to generate and distribute electricity in an efficient and timely manner, rather than focusing on security aspects of the critical infrastructure of the system. However, the increase of interconnectivity and remote accessibility places power grids under the risk of internal and external attacks.

Real-time cyber attacks can disrupt entire power grids. For example, in 2003 the Davis-Besse nuclear power plant near Oak Harbor, Ohio was infected by a Slammer worm that traveled from a consultant's network to the process control network and generated unwanted traffic [7]. As a result, the plant personnel could not access the safety parameter display system for around five hours which showed sensitive data about the reactor core, temperature, and radiation sensors of the power plant. In 2006 the Browns Ferry nuclear plant in Athens, Alabama was shut down after the failure of critical reactor components and controllers due to a cyber attack on their internal network [8]. In 2008, the second unit of the Hatch nuclear power plant in Baxley, Georgia experienced an automatic shutdown due to routine software updates to a single computer on the plant floor. The update was performed to synchronize data between the plant and business networks [8]. Another incident in an Iranian

nuclear plant was reported in 2011 where the plant process was interrupted due to the Stuxnet worm. This attack was initiated by connecting an infected USB drive to the Programmable Logic Controller (PLC) at the plant floor [9]. The Ukraine power plant cyber attack was reported in 2015 [10]. This was the first known successful attack on power grids where attackers were able to disrupt the electricity supply to the end users. Thus, power grid attacks are one of the most critical issues in industrial control systems and it is important to protect them by applying adequate safety measures [11].

General safeguards include defense-in-depth architecture which separates the control and corporate network traffic, strong access control and authentication mechanisms, restricted perimeters using DMZ (demilitarized zone), vulnerability assessment, and risk management systems [2]. However, these safeguards are difficult to deploy and maintain owing to legacy-inherited security loopholes and restrictions [12]. Therefore, these relevant preemptive measures are not sufficient to protect the SCADA system from cyber attacks. An additional protection layer is also required which detects and prevents the system from malicious events and threats.

Generally, packet filtering and identification of threats are key to securing these systems. However, traditional firewalls do not always fulfill all the security requirements of critical infrastructures. For example, in 2019, the western US power grid infrastructure was hacked. The intruders created periodic blind spots for grid operators for about 10 hours, by identifying a vulnerability in the firewall configuration [13].

In the last few years, many key management techniques have been published to secure SCADA communication, namely, SCADA Key Establishment (SKE), SCADA Key Management Architecture (SKMA), Advanced SCADA Key Management Architecture (ASKMA), and Hybrid Key Management Architecture (HKMA) [14], [15], [16], [17], [18], [19]. These techniques fall under two main categories, namely, centralized key management and decentralized key management schemes. Moreover, each of these categories uses three approaches to generate and extract the session key, namely, symmetric, asymmetric, and hybrid. The drawback of the centralized scheme is that if the Key Distribution Center (KDC) is down, the communication is cut off, which is not acceptable in SCADA systems. In a decentralized approach, the keys are created

using keying material and may only affect the single communication link in case of a breakdown.

The symmetric key based approach is efficient in terms of confidentiality and high availability but does not provide authentication and integrity. On the other end, an asymmetric key provides message integrity, authentication, and privacy, but may compromise availability. Hence, hybrid techniques are more suitable for SCADA systems. A few key management techniques have been proposed using hybrid methods. For example, Rezai et al. [17] propose an advanced Hybrid key management architecture (HSKMA), which improves the key management architecture proposed by Choi et al. [18]. However, it uses a centralized KDC to distribute the keys. Moreover, the communication between the MTU and the sub-MTU is established using Elliptic-Curve Cryptography (ECC) based asymmetric key cryptography while the sub-MTU and the RTU communicate using Rivest–Shamir–Adleman (RSA) public key cryptography. The same approach has been used to enhance the scheme proposed by Rezai et al. [20] using a decentralized system in [16]. In this scheme, the master keys are refreshed using ECC and symmetric cryptography is used for encryption, decryption, and session key updates. However, this scheme does not validate the message integrity and authentication. Moreover, none of the previous methods has practical implementation proof that it provides immunity against quantum attacks [21]. Furthermore, it has been known that RSA does not guarantee perfect forward secrecy [18]. In summary, none of the techniques covers all the security aspects.

1.3 Motivation

SCADA systems play a vital role in ICSs and rely on real-time request-response mechanisms to operate the substation components accurately by consuming minimal CPU and battery resources. For such time-critical applications, the deployed intrusion detection system should act quickly to capture malicious activities using minimal resources in a given time period for large-scale deployments. Moreover, accuracy plays a vital role to predict the nature of incoming traffic. Therefore, for proper validation & training, the control center should be equipped with a highly accurate IDS. Furthermore, effective IDS should be able to capture both known and zero-day attacks. To prevent intrusions and for secure communications, we also need to

consider the deployment of lightweight cryptographic solutions. The foregoing discussion brings in the need for an effective security framework that can not only detect but also protect SCADA networks from potential intrusions.

1.4 Objective

The primary goal of our work is to propose a security framework for intrusion detection & prevention in SCADA networks. The work focuses on three major aspects, namely, vulnerability assessment of SCADA components, efficient and accurate intrusion detection systems for known and zero-day attacks, and a robust security framework for SCADA communication to protect the network from potential intrusions.

An IDS is proposed & implemented on SCADA based smart grid. Furthermore, to prevent the system from various attacks, a robust & low-cost security framework is proposed. The framework is based on a multilayered architecture that combines both symmetric and asymmetric key cryptography techniques.

The proposed techniques were implemented & evaluated on a SCADA test bench.

1.5 Major Contributions

Figure 1.2 illustrates the major contributions of this Ph.D. work. It has three main modules.

Module 1: Vulnerability and risk assessment of SCADA components

1. To cover the landscape of risk assessment, a comprehensive review of various types of potential weaknesses of the SCADA system has been made by taking real incidents reported in standard vulnerability databases and recommendations have been provided for the improvement of the security of Industrial Control Systems.

2. For vulnerability assessment, penetration testing has been carried out on one of the SCADA components called Onion Omega2 (System-on-a-Chip). Various product-level weaknesses have been made either at the network level or by vendor patching.

Module 2: Intrusion Detection System (IDS) for SCADA Networks

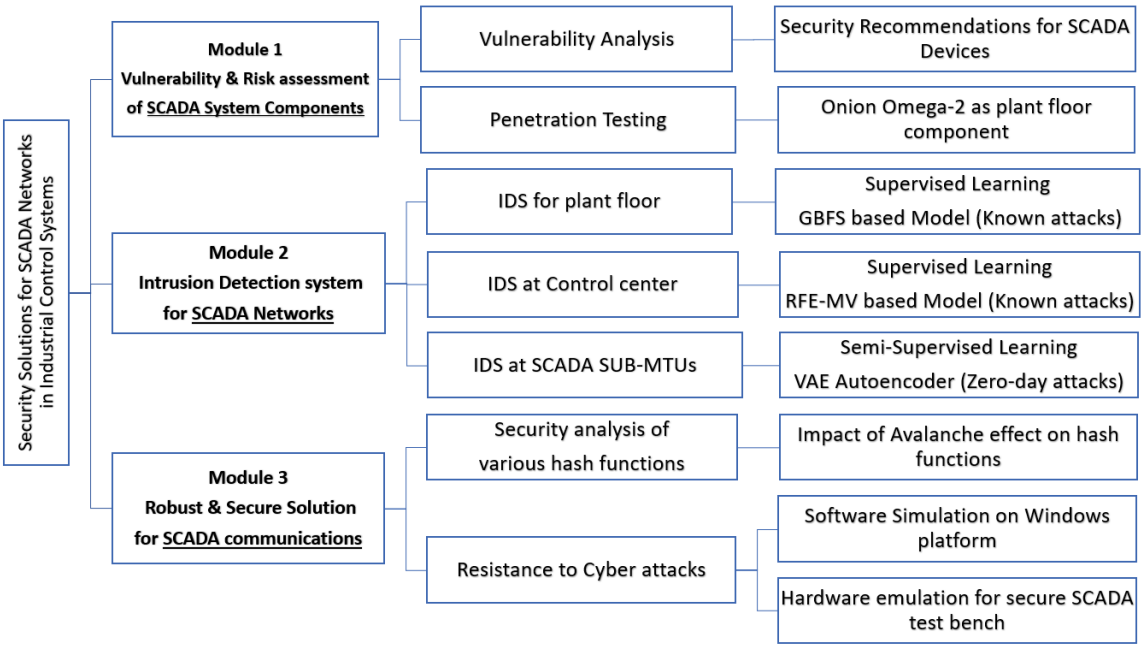


Figure 1.2: Major Contributions of the Ph.D. work

1. A Gradient Boosting Feature Selection (GBFS) based filtering model based on the Weighted Feature Importance (WFI) scoring technique has been proposed to reduce the complexity of classifiers at the plant floor of the power grids. Moreover, the implementation and evaluation of various decision tree-based machine learning techniques after feature selection have been carried out. This approach optimizes the False Positive Rate (FPR) and the execution time as it is more compatible with detecting intrusions in real-time communication.

2. For more accurate results and for verification, Recursive feature elimination (RFE) based filtering model has been proposed to find out the most stable features of the dataset at the control center. Further, these features are used to classify normal and attack vectors using majority vote-based IDS with multiple classifiers. Through this approach, the proposed framework achieves high processing speed and accurate prediction.

3. IDS at intermediate SCADA (Sub-MTUs), proposed in this work is intended to detect zero-day attacks. For this, we have proposed a novel approach using autoencoders to identify anomalous traffic. These semi-supervised models learn the various features of real-time traffic using normal events only. Furthermore,

for minimizing the overfitting problem and better validation, we have generated synthetic datasets using the CTGAN model. The experimental results reveal that the autoencoder models are able to detect unknown or zero-day attacks even with a slight modification in the normal events. The results are promising in terms of accuracy, precision, recall, and detection rate.

4. An integrated model of IDS framework for real-time SCADA systems has been proposed using defense-in-depth architecture by considering power grids as a candidate ICS application. In this approach, the displacement of three IDSs has been proposed, one at the plant floor using Gradient Boosting Feature Selection (GBFS) based filtering model to detect the intrusions in real-time, another at the control center using the majority vote-based ensemble method for accurate prediction. While IDS at intermediated SCADA (placed at sub-MTUs) has been proposed using Variational Autoencoders (VAE) based on semi-supervised learning to detect zero-day attacks. To improve the efficiency of the model, high-quality synthetic datasets were generated for SCADA-based power grids.

Module 3: Robust and Secure Framework for SCADA Networks

1. A robust and low-cost security framework for SCADA has been proposed to mitigate cyber-attacks. The security model is based on a multi-layer framework that integrates both symmetric and asymmetric key cryptosystems. Further, an efficient session key management mechanism has been proposed by merging random number generation with a hashed message authentication code. Moreover, for each session, three cryptographic techniques have been proposed based on the concept of Vernam cipher and a pre-shared session key, namely, random prime number generator, prime counter, & hash chaining. The proposed scheme is intended to design for a real-time request-response mechanism in the SCADA networks by supporting broadcast, multicast, and point-to-point communication. A SCADA hardware test bench has been developed to experimentally evaluate the proposed framework.

1.6 Report Organization

The rest of the report is organized as follows. Chapter 2 describes related research in the area of SCADA security by considering various attacks and protection schemes for all the three modules, namely, vulnerability assessment of SCADA Components, IDS for SCADA systems, and Robust framework for SCADA communication. Chapter 3 describes the methodology to assess vulnerabilities of SCADA system components by considering a case study of Onion Omega-2. The complete experimental setup, framework design, methodology for assessment, and result discussion of the proposed intrusion detection system are covered in Chapter 4. Chapter 5 presents the framework, experimental setup, and evaluations (performance assessment & security analysis) of the proposed model for secure SCADA communications. Chapter 6 covers the concluding remarks. Publication history is provided in the Appendix section.

Chapter 2

Background and Related Work

2.1 Literature review on Vulnerability assessment of SCADA Components (Module 1)

At present, various models and categories of embedded products are launched by different manufacturers most often, however, the decoding solution is not necessarily applied to each and every version of these products [22]. These bring product-level vulnerabilities in such devices, tools such as Binwalk, Firmware Reverse Analysis Console (FRAK), Interactive Disassembler (IDA), and Binary Analysis Toolkit (BAT) are used to decompress the file system of firmware [23], [24] The Binwalk tool is used to decompose the binary file and extract the metadata from it. FRAK evaluates the data provided by the equipment service provider and decompresses it. However, Binwalk has more capability to decompress the file compared to FRAK. BAT uses GPLtool, which recursively extracts the files from the binary firmware and also provides support to segment the document [22], [25].

Well-known, open-source information-gathering tools such as Nmap, Nikto and Sparta are used to identify open ports and services [26]. Furthermore, Nessus, OpenVAS, FoundScan and Internet Security Scanner have been used as popular scanning tools. These tools allow us to scan network devices and check them against their databases containing thousands of records for known vulnerabilities. OpenVAS vulnerability scanner developed by Greenbone Security is used to test various protocols and networks. Many vulnerability scanners and penetration testing tools are available in the Kali-Linux operating system [27]. Moreover, various test tools have been proposed and implemented in academic research to determine network flaws based on grammar and fuzzy logic methodology [28]. The PROTOS project group has developed tools using syntax-based generation according to the protocol type [29]. A popular search engine, Shodan, contains information of more than 600 million publicly available IoT devices such as various ports information and banner data information,

etc. This information is used to assess the weaknesses of SCADA/IoT devices in an attempt to mitigate the attacks [30].

Various frameworks have been proposed by researchers for life cycle assessments of infrastructure. Creery and Byres [31] propose a complete security assessment process model for the evaluation of control systems. The control system cyber security self-assessment tool (CS2SAT) was developed by the Idaho National Laboratory to evaluate control system security [23]. This tool has the capability to systematically evaluate the product by collecting all the necessary information from various resources to identify flaws in the system. Sandia National Laboratories has developed the information design assurance red team (IDART) to evaluate the security strength of SCADA systems [32]. Most of these assessment techniques have been proposed for identifying security flaws in control and monitoring systems in general, with limited application to SCADA systems. Furthermore, the previous approaches are mainly system-level assessment tools rather than at the device-level. In order to identify all security issues in SCADA systems, vulnerability assessment must be done for each component. n at the device-level. In order to identify all security issues in SCADA systems, vulnerability assessment must be done for each component. The approach should begin with risk assessment followed by vulnerability evaluation to validate the security.

2.2 Literature review on Intrusion Detection Techniques (Module 2)

Researchers have proposed several solutions for intrusion detection techniques to secure SCADA based power grids [33],[34]. Hink et al. provide a comparative analysis of various machine learning techniques using a power grids dataset and identify Adaboost-JRIP is one of the best classifiers [35]. However, the authors do not filter and reduce the dimension of the dataset. Hence, they are unable to achieve good accuracy and execution speed. Pan et al. have focused on hybrid IDS using data mining, where they have used common path mining to identify the location of attacks [36], [37].

Further, in [38], the authors apply Pearson Correlation Coefficient (PCC) for feature selection and extract 75% of features. They use an Expectation Maximization Clustering Technique (EMCT) to classify the events. Using this approach, they

Attributes	Machine Learning for Power System Disturbance and Cyber-attack Discrimination [9]	Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems [10], [11]	Machine Learning for Power System Disturbance and Cyber-attack Discrimination [12]	A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems [13]	An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems [14]	Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids [15]
Feature Section Method	Not Applied	Not Applied	PCC (Pearson's correlation coefficient)	ICA (Independent component analysis)	PCC	GBFS
Features (%)	100%	100%	75%	compared various subset of features	25%	12%
Classification Technique	Adaboost - JRIP	Common Path Mining	Expectation Maximization Clustering Algo	Beta mixture-hidden Markov models (MHMMs)	Gaussian Mixture-Kalman Filter Model (GMM-KF)	Tree-Based (XGBoost)
Algorithm used	Machine Learning	Data mining and pattern recognition	Maximum likelihood estimation	Statistical common estimation method	Bayesian filtering algorithm	Machine Learning
Measure used	Accuracy	Accuracy	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure, Training time
Dataset used	Oak Ridge National Laboratories (ORNL) - power grid dataset [16]					
Pros & Cons	Data are not pre-processed properly, Low accuracy and execution speed	Improvement in identification of attacks, provide location of attack.Moderate accuracy and execution speed	Improved execution speed but compromising in accuracy with multi-class datasets	Improved accuracy, no observations regarding execution speed	Improvement in accuracy, Tested for binary classification, No multi attack vectors classification	Significantly improved accuracy and execution speed

Figure 2.1: Major Contributions of the proposal to ensure security of SCADA systems

improve the execution speed but do not achieve better accuracy for a multi-class dataset. Moreover, this technique is enhanced by combining PCC with the Gaussian Mixture – Kalman Filter Model (GMM-KF) in [39]. The authors are able to reduce the percentage of the features to 25 and achieved good accuracy and execution speed. However, this experiment is limited to a binary dataset. Moustafa et.al.[40] have used ICA – Independent Component Analysis feature selection and Beta Mixture Hidden Markov (BMHM) classification model. The authors have obtained promising results in regards to accuracy. However, they have worked on a subset of the features, and hence we could not identify the exact number of features used in this section. We have recently proposed WFI based GBFS model for feature selection and extracted 12% of the most promising features in [6]. Our target was to achieve high execution speed and a better predictive model for real-time SCADA communication. The proposed GBFS model has further verified with different machine learning algorithms. We have identified that the proposed solution is suitable for tree-based classifiers. Note that all these experimental studies use the power grid dataset created by Oak Ridge National Laboratories (ORNL). Figure 2.1 summarizes the literature on IDSs for power grids.

The research work in [41] focuses on developing an IDS for network administrators

by combining supervised and unsupervised learning techniques using ensemble method. This approach has been tested on various datasets like KDD Cup 99, NSK-KDD, and Kyoto 2006+ and is able to classify around 95% of the incoming traffic correctly [41]. In [42], the authors propose sustainable ensemble learning to improve the detection rate by aggregating multiclass regression models such that ensemble learning adapts to different attacks. Cloud-based solutions for distributed anomaly detection systems can be found in [43]. In [44], the authors propose a Gaussian mixture based anomaly detection technique that relies on ensemble one-class statistical learning model that is designed to effectively recognize zero day attacks in real-time using the concept of edge networks.

One of the IDSs [45] is developed for unbalance data samples (KDDcup99), where it is seen that J48 and Random Forest work best for big sample classes while others such as Bayesian network and Random tree seem to be a good fit for small samples. Therefore, the authors [45] propose a solution based on ensemble learning by applying a majority vote classifier to improve the performance of classification. Further, this work is improved by combining the prediction of Bagging and Boosting using ensemble techniques with tree base algorithms as the base classifier in [46]. In [47], the authors propose a novel approach that combines permission and intents supplements with an ensemble method for accurate malware detection for cellular phone communication. Moreover, in [48], authors execute anomaly detection over the communication networks by combining the prediction of three different types of classifiers, namely, neural networks, decision trees, and logistic regression using a weighted majority voting scheme.

IDSs for real-time systems require low computational cost with high accuracy and execution speed. Such an IDS can be developed using a hybrid approach that combines the feature selection model along with an efficient classification scheme [49] which is the motivation behind our proposed framework.

2.3 Literature review on SCADA Communication Algorithms (Module 3)

SCADA networks are typically configured using proprietary protocols such as Modbus, IEC 61850, IEC 60870, DNP3, and Profinet, which do not support secure data

communication. Moreover, the remote procedure call (RPC) follows open communication and one of the real-time examples of the consequent vulnerability was the Blaster worm [2]. Furthermore, many network sniffing tools are freely available to view and gather the network traffic [50]. Therefore, secure data transmission is one of the important requirements for SCADA systems. Key management and encryption play a vital role in securing SCADA communication. Typically, in a SCADA communication, the MTU sends control signals to the RTUs to control the plant floor devices, which require three types of communication, namely, broadcast, multicast, and point-to-point. However, controller RTUs may need to operate other field RTUs. In case of an emergency shutdown, to acquire the clock information or synchronization, MTUs broadcast the signal to all the control devices such as RTUs, IEDs, and PLCs. To operate a specific substation device, the MTU requires multicast communication, whereas plant for machinery typically requires point-to-point communication. Hence, while designing a secure framework for SCADA networks, it is crucial to cover all three types of communication.

During the last two decades, many key management schemes have been proposed, which typically fall into two categories, namely, centralized key distribution such as [4], [14], [51], [52], and decentralized key distribution scheme such as [16], [53], [54], [55] . In the centralized scheme, the Key Distribution Center (KDC) plays a vital role in generating and distributing secret keys to establish secure communication between the communication parties. In contrast, the decentralized scheme requires pre-shared keying material that is used to create the session key. Once the session key is derived using keying essence, further communication takes place using that key. Furthermore, some key management schemes use the public key-based technique to establish secure transmission. Although this method is time-consuming, various research studies suggest that ECC is a suitable public-key cryptosystem [4], [16], [18].

Sandia Labs proposed a SCADA key establishment (SKE) method for managing cryptographic keys in the network [14]. This scheme is proposed for point-to-point communication amongst MTU, sub-MTU, and RTU and uses the symmetric key technique to establish secure communications between sub-MTUs and RTUs, while sub-MTUs and MTUs communicate using public key cryptography. For the symmetric key, the session key is generated using three types of keys, namely, long term key

(LTK), general seed key (GSK), and general key (GK) [14]. KDC assigns public and private key pair to each sub-MTU and MTU. However, this method does not support broadcast, multicast, and RTU to RTU communication. Moreover, it increases the overall key storage overhead and complexity as the long-term keys are managed manually. In [52], the authors propose a SCADA Key Management Architecture (SKMA) for secure session key management, which enhances the capability of SKE. While the SKE uses both a public key algorithm and a symmetric key algorithm, the SKMA uses only symmetric encryption algorithm. SKMA generates a session key using a pseudorandom function, keyed by the node-node key, and a timestamp that is based on the duration of the session. SKMA uses key establishment protocol based on ISO 11770-2 mechanism [15]. However, the scheme does not provide secure message broadcasting but supports RTU-RTU communication. Moreover, it does not provide any confidentiality and integrity.

Advanced SCADA Key Management Architecture (ASKMA) supports both message broadcasting and secure communications. Furthermore, evenly spreading the total amount of computation across the high power nodes (MTU or SUB-MTU) significantly avoids the performance bottleneck and keeps minimal burden on the low power nodes (RTU). It uses the LKH (Logical Key Hierarchy protocol) to construct a logical tree of symmetric keys. Each member knows all the symmetric keys from its leaf to the root, and if any new node joins the group, LKH updates the entire set of symmetric keys from its leaf to the root. Although the overall performance of ASKMA has many advantages, it can be less efficient during the multicast communication process. To solve this issue, ASKMA+ was proposed [14]. ASKMA+ divides the key structure into two classes, by applying the IoLus framework to construct each class as a logical key hierarchy (LKH) structure. Through this key structure, the authors proposed a more efficient key-management scheme supporting efficient multicast communication by considering the number of keys stored in a remote terminal unit (RTU). However, ASKMA+ does not address the availability issue in SCADA.

To satisfy the availability requirement, Hybrid Key Management Architecture (HKMA) and Advanced Hybrid Scada Key Management Architecture (AHSKMA) is proposed [17], but there were a chance that field devices will stop working during the replacement of field control devices. To solve this issue, Choi et al. propose a

hybrid key management scheme [18]. A centralized key distribution (CKD) protocol is applied between the sub-MTU and MTU, and LKH protocol is applied between sub-MTU and RTU. However, if the centralized key distribution server breaks down, the entire approach fails to execute the protocol. Rezai et al. [16] also use a hybrid key management method using ECC. Jiang et al. [19] propose Limited Self-Healing key distribution (LiSH), which offers revocation capabilities along with collusion-resistance for group communication in SCADA systems. The LiSH+ is used to address the dynamic revocation mechanism, which enhances the base method of LiSH. Kang et al. [54] propose a scheme for radial SCADA systems based on a pre-shared session key that relies on symmetric key cryptography. This solution enhances the performance of the radial SCADA system by using the master key concept.

AGA-12, Part 2, provides security features offering a new security protocol standard [56]. It uses cipher suites to secure communication amongst SCADA field devices, which covers authentication, confidentiality, and integrity. However, it fails to provide faster execution. Furthermore, it does not offer prevention against quantum and Denial of Service (DoS) attacks. In addition, AGA-12 uses the RSA algorithm for encryption, which was recently cracked and also does not provide key management [21]. The other security standards, such as IEC 62210, IEC 62351, fail to offer security against man-in-the-middle (MiM) attacks and also lack of strong key management. A novel key distribution method was proposed for smart grids in [57] which uses identity-based cryptography. This method adopts a hybrid approach to counteract man-in-the-middle and replay attacks. However, this method does not cover the authentication of the SCADA components. The authors in [58] introduce an authority roles for SCADA devices using attribute-based access control. The hybrid Diffie-Key exchange, along with the authentication scheme, was proposed in [59]. This scheme uses RSA and AES for session key generation and encryption. However, it does not provide high availability.

Chapter 3

Vulnerability & Risk assessment of SCADA Components - Methodology, Experiments & Result discussion

The research work reported in this chapter has resulted in the following publications:

- **D. Upadhyay** and S. Sampalli, "SCADA (Supervisory Control and Data Acquisition) systems: vulnerability assessment and security recommendations,"**Computers & Security, Elsevier,** vol. 89, p. 101666, 2020. *(Impact Factor: 5.105)*

- **D. Upadhyay**, S. Sampalli, and B. Plourde, "Vulnerabilities' assessment and mitigation strategies for the small linux server, Onion Omega2," **Electronics, MDPI**, vol. 9, no. 6, p. 967, 2020. *(Impact Factor: 2.39)*

3.1 Summary of the chapter

The Onion Omega2 is a small embedded Linux server for building SCADA/IoT communication systems. While it provides efficient functionality, it is important to be aware of its vulnerabilities and built-in security features. We have identified product-level vulnerabilities of Onion Omega2 using scanners and penetration tools. This helped us to identify the threats and vulnerabilities of Onion Omega2 and measure the level of risk. The vulnerabilities include missing patches, insecure system configurations, and other security-related updates. The identified vulnerabilities can either be fixed by the vendor and/or network administrator/engineer. Furthermore, this section illustrates effective countermeasures for identified vulnerabilities to harden the security of Onion Omega2. This study empowers vendors, software developers, and network engineers with the knowledge necessary to take proactive measures to ensure the security of the overall system built using Onion Omega2.

3.2 Vulnerability Assessments of Onion Omega2

Vulnerability assessment is typically a highly subjective process; it requires powerful analytical strategy and computational methodology [2], [60] For a thorough review of the vulnerability assessment of the Onion Omega2, we have followed standard vulnerability assessment tools and techniques. We started the security assessment process of Onion Omega2 by conducting firmware analysis using the Binwalk tool. Further, we have used standard tools such as Nikto, Sparta, OpenVAS, and Nessus to analyze the scan results of Onion Omega2 starting from basic port scans to advance level testing. We describe the scan results and mitigation techniques of Onion Omega2 in the following section. The section is mainly divided into two parts according to remediation strategies:

1. Vendor level fixes
2. Network/Administrator level fixes

In this study, we demonstrate a step-by-step process to evaluate product-level vulnerabilities of Onion Omega2, which focuses on both vendor and network-level fixes. Moreover, this assessment strategy not only adheres to the best practices, but also provides the roadmap to build and assess other secure embedded devices which include micro-controllers, platforms, and customized operating systems. We started our assessment process with basic port scans followed by the web server and DNS server assessment. Through this, we have evaluated common loopholes in SCADA products, which include buffer overflow, lack of bounds checking, command injections, cross-site scripting, and directory path traversal. We then assessed the security strength of remote login and Wi-Fi communication protocols. Furthermore, SSL/TLS (secure socket layer/transport layer security) security configuration is evaluated to analyze the strength of security protocols. By using a similar systematic vulnerability assessment process, security practitioners can determine the security weaknesses of any embedded devices from the primary to the advanced level.

3.2.1 Vendor Level Fixes

The following section highlights the vulnerability assessment of the Onion Omega2 which focuses on vendor-level fixes where we provide a comprehensive summary of

vendor-level fixes in the following Figure 3.1. According to the latest details, Onion Omega2—v0.3.2 b217 and up, has patched vulnerabilities 4 and 5 as mentioned in Figure 3.1. However, the first three vulnerabilities are not patched yet and have been left to Onion Omega2 end users to implement the security hardening. One reason these vulnerabilities have not been patched is that the Linux build system is open-source and the firmware of Onion Omega2 is meant to be a "jack of all trades", and compatible with the end users to build their own customized firmware that can be security hardened according to their needs.

3.2.2 Network/Administrator Fixes

This section presents the major weaknesses of Onion Omega2 for configuring SCADA networks. We have found three major vulnerabilities of Onion Omega2 that need to be fixed at the network-administrator level. The first vulnerability relates to the IP forwarding service that is enabled in Onion Omega2, which could act as an unreliable path to bypass the firewall. The second weakness was found in the configuration of the message queue telemetry transport (MQTT) broker. An MQTT broker is a lightweight protocol and is used to establish the communication with low-battery IoT devices. This protocol does not protect with an authentication password, consequently, allowing intruders to extract information from the internal network. The third vulnerability includes a medium-strength cipher configured for HTTPS protocol suites of the secure socket layer. In the following, we discuss each of these vulnerabilities in detail. The comprehensive summary of network-level fixes is provided in Figure 3.2.

3.3 Generic Vulnerability Assessment

This study mainly focused on the vulnerability assessment of Onion Omega2. However, the procedure that we have adopted can be useful for the assessment and analysis of other platforms. Vulnerability assessment is a continuous process due to constant technological changes and hence becomes the backbone for a successful defense of any industrial control system. This process is heavily dependent on asset management and risk assignment to prioritize security issues. Figure 3.3 depicts a generic flowchart for vulnerability analysis, in which we generalize the assessment process for other

devices, platforms, and networks in industrial control systems. The proposed strategy can be used to maintain the security and compliance standards of the system with open-source tools and technologies. Using this generic framework, vendors, security analysts, and network engineers can build their own configurations and run different scans to detect security flaws in the system. Our approach gives the ability to assess the infrastructure thoroughly, which covers different levels such as host, network, wireless, and application-level vulnerabilities.

Vulnerabilities	Description	Targeted Ports/Services	Scanning Tools Used	Mitigation
X-XSS-Protection header is not defined.	Cross-site scripting attack on index page is possible. This could allow the user agent to render the content of the site in a different fashion to the MIME type.	4200, 80 Http	Nmap, Nikto, Sparta	1. Vendor should configure the webserver including an X-Frame-Options/ X-XSS protection/X-Content-Type-Options header. 2. Fix the header protection by enabling MOD Header in httpd.conf file. 3. Enable XSS filter to sanitize the page if attack detected. 4. Use No Script extension for browser (e.g., Clear Click for Firefox)
Buffer Overflow attack possible.	Long string from remote host generates buffer overflow in memory.	4200, Http	Nmap, OpenVAS	Vendor should upgrade the web server to the latest version by adopting proper validation techniques for each segment of the program code.
DNS Server Cache Snooping Remote Information Disclosure	This may allow a remote attacker to determine domains that have recently been resolved via this name server.	53-DNS (UDP/TCP) dnsmasq 2.75	Nmap, Nessus	Vendor should update the DNS server with proper patches to fix the problem.
SSH Weak MAC Algorithms Enabled	SSH server is configured to allow either MD5 or 96-bit MAC algorithms, both of which are considered weak.	22-SSH	Nessus, OpenVAS	Vendor should patch it, or administrator should consult product documentation to disable MD5 and 96-bit MAC algorithms.
Key-Reinstallation Attack (KRACK) Vulnerability [36]	This breaks security of WPA2 protocol and able to access the critical information communicating between client and Access point.	WPA2	Krackattacks-scripts [37]	Vendor should fix the issue by properly configured wpa-supplicant in Linux kernel.

Figure 3.1: Onion Omega2 vulnerability and mitigation (vendor fixes)

Vulnerabilities	Description	Scanning Tools	Mitigation
IP forwarding enabled	An attacker can exploit this path to route packets through the specific host to bypass targeted firewalls, routers, or NAC filters.	OpenVAS, Nessus	1. Unless the remote host is a router/gateway, it is recommended to disable IP forwarding. 2. On Linux based firmware, disable the IP forwarding by doing [38]: echo 0 > /proc/sys/net/ipv4/ip_forward 3. On Windows based system, set the key 'IPEnableRouter' to 0 under [38]: HKEY_LOCAL_MACHINE\System \CurrentControlSet\Services\Tcpip\Parameters 4. On Mac OS X, to disable IP forwarding execute following command [38]: sysctl -w net.inet.ip.forwarding = 0
SSL medium strength cipher suites supported	The default configuration supports encryption that uses 3DES-CBC for symmetric key cryptography which is vulnerable to SWEET32.	Nessus, OpenVAS, Nikto	Reconfigure the affected application to avoid use of medium strength ciphers [39].
MQTT Broker does not have authentication	This may allow a remote attacker to gain access of the network without any authentication. Intruder can extract the information of the control/IoT networks.	OpenVAS	Enable authentication mode in Onion Omega2: By configuring Mosquitto Broker setting password in etc/mosquito/Passwd;

Figure 3.2: Onion Omega2 vulnerability and mitigation (vendor fixes)

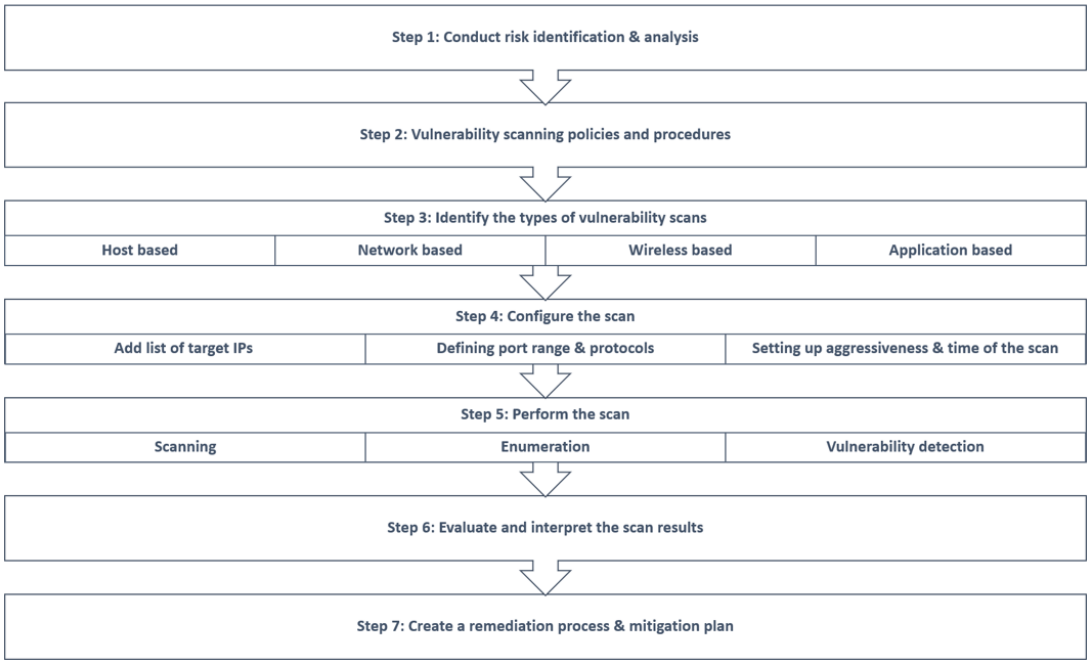


Figure 3.3: Generic framework of vulnerability assessment process

Chapter 4

Intrusion Detection Systems for SCADA based power grid - Methodology, Experiments & Result discussion

The research work reported in this chapter has resulted in the following publications:

- **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids," in **IEEE Transactions on Network and Service Management,** vol. 18, no. 1, pp. 1104-1116, March 2021, doi: 10.1109/TNSM.2020.3032618. *(Impact Factor: 4.19)*

- **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model With Majority Vote Ensemble Algorithm," in **IEEE Transactions on Network Science and Engineering,** vol. 8, no. 3, pp. 2559-2574, 1 July-Sept. 2021, doi: 10.1109/TNSE.2021.3099371. *(Impact Factor: 5.21)*

- **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "A Defense-in-Depth IDS Framework For Known and Zero-day Attack Detection in SCADA Systems," in **IEEE Transactions on Network Science and Engineering,** manuscript under review. *(Impact Factor: 5.21)*

- **D. Upadhyay**, Q. Lui, J. Manero, M. Zaman and S. Sampalli, "Comparative analysis of Tabular GAN models by validating synthetic data of Power Grids," in 20^{th} **IEEE International Conference on Smart Technologies, EUROCON 2023**, manuscript under review.

4.1 Summary of the chapter

This chapter presents three types of IDSs, namely, one for the plant floor, one for the control center, and another for intermediate SCADA (SUB-MTUs). The plant floor and control center IDSs have been proposed for the detection of known cyber attacks, while IDS at sub-MTUs has been proposed for unknown attacks. The plant floor IDS is developed using a GBFS-based feature selection approach to identify the most promising features for anomaly detection in power grids. To accelerate the execution speed and learning efficiency, a GBFS-based feature selection approach is applied on filtered data to compute the most promising features (15 features out of 128 features) from the entire dataset dynamically according to network/SCADA traffic. The dynamic approach of selecting the features from the entire dataset hides largely all the sensitive information of the power grid system. Finally, these reconstructed datasets are used by decision tree-based algorithms that classify the various attacks and normal events.

Whereas, the RFE-XGBoost-based feature selection approach along with the majority vote ensemble method is used to detect intrusions at the control center. The proposed framework comprises three key elements, namely, data preprocessing, feature selection, and anomaly detection. Initially, during data preprocessing, the features are mapped and scaled to a specific range. The RFE-XGBoost-based feature selection approach is subsequently applied to filtered data to compute the most stable features from the entire dataset (30 features out of 128 features). This approach enhances learning efficiency. Furthermore, the selection of the features is carried out dynamically according to network traffic. In the subsequent stage, these reconstructed datasets are used by nine heterogeneous classifiers to predict the various attacks and normal events. Finally, the majority vote-based ensemble algorithm is applied to predict the output based on the majority of the class labels predicted by each of the nine classifiers.

The experimental results reveal that the proposed framework fares well in terms of accuracy, detection rate, precision, and recall. Moreover, the proposed model outperforms some of the state-of-the-art published techniques. The model offers a blend of effectiveness with precision, as it uses a limited number of stable features, and the classification is carried out based on combined predictions of the nine most

promising classifiers. Moreover, this combination requires limited computational cost, which is one of the crucial factors for mission-critical applications. Thus the proposed model has the potential to leverage the competencies of real-time SCADA systems for power grids.

4.2 Framework Design of proposed IDSs

The following sub sections focus on the framework design of each of the three proposed IDSs that is intended to place in the various location of SCADA infrastructure by considering defense-in-depth architecture.

4.2.1 Plant floor IDS: Proposed framework of GBFS-based WFI Scoring model for Tree-based classifiers

This section presents the proposed framework for an intrusion detection system that distinguishes normal and malicious events by analyzing SCADA traffic on power grids. The proposed framework operates in three phases, namely, pre-processing the data, feature selection, and anomaly detection using a classification approach. The elements for each phase are illustrated in Figure 4.1.

During the data preprocessing phase, data cleansing, feature mapping, and feature normalization are applied to the raw dataset to obtain filtered data. Then the Gradient Boosting Feature Selection approach is applied on filtered data to select the most promising features from the entire dataset dynamically. Since power grids use a complex mix of SCADA systems to control field-site components, network monitoring devices such as SNORT and Syslog are used to capture the different types of features [35]. Usually, real-time data obtained from sensors or real-time systems always presents some consistency issues, the signal is lost, or the measuring devices get off the scale readings at some point. For this reason, we need to do a data cleansing operation to remove incorrect data. We remove infinities and NaN values, looking for empty sequence points that will be avoided by the algorithms. Furthermore, in order to extract the relevant features, we apply a Gradient Boosting Feature Selection which uses the Weighted Importance Feature extraction method to select the most promising features. This approach helps to improve the computational speed and also assists in providing a precise outcome for anomaly detection. Moreover, reduction in

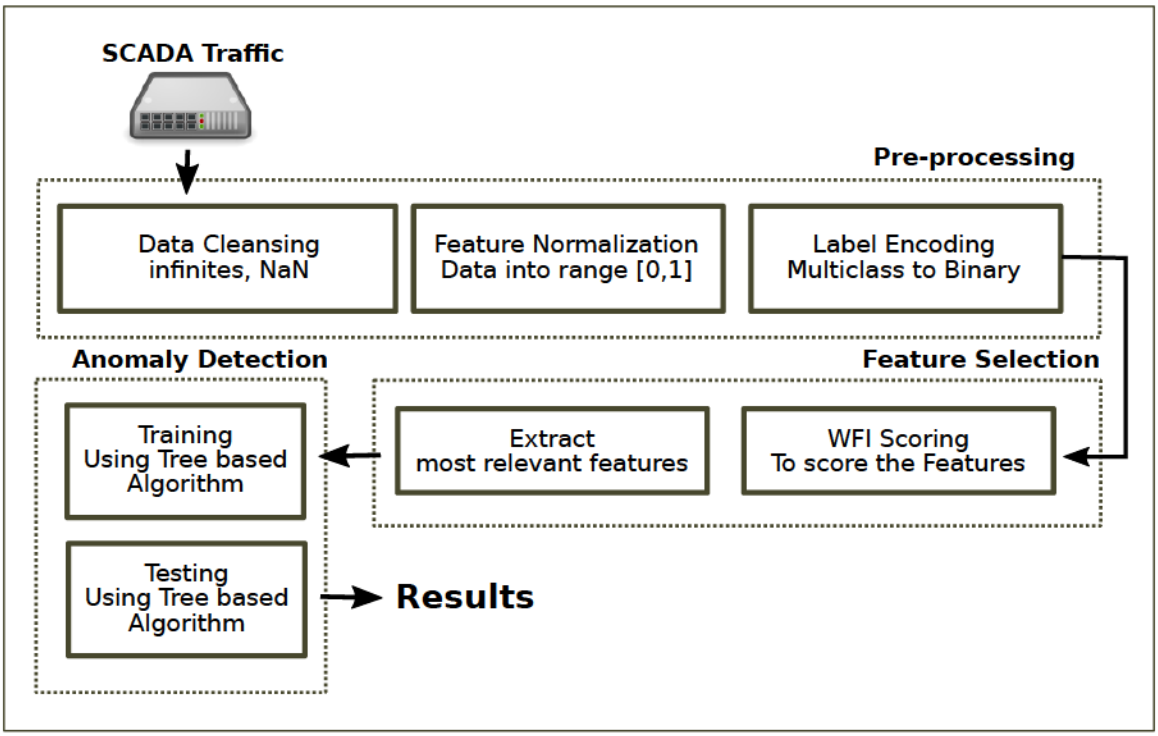


Figure 4.1: Framework for a GBFS Based Intrusion Detection System at Plant Floor

features helps in consuming less memory while training and testing the dataset during classification to classify normal and attack events.

4.2.2 Control center IDS: Proposed Framework based on RFE-XGBoost filtering model along with majority vote ensemble method

This section presents the proposed scheme for an intrusion detection system for power grids to classify traffic into attacks and normal events by analyzing SCADA traffic. This novel approach uses the RFE-XGBoost-based feature selection method to determine the most consistent features from the dataset based on feature importance scores. Furthermore, the majority vote ensemble method identifies accurate outcomes during classification. This combined approach accomplishes two significant aspects of real-time traffic monitoring namely, accuracy and computational speed. The entire framework is divided into three phases – data preprocessing, feature selection, and anomaly detection, as illustrated in Figure 4.2.

The data cleaning, feature mapping, and feature normalization are done in the preprocessing phase to obtain streamed and sanitized data. Since the power grid

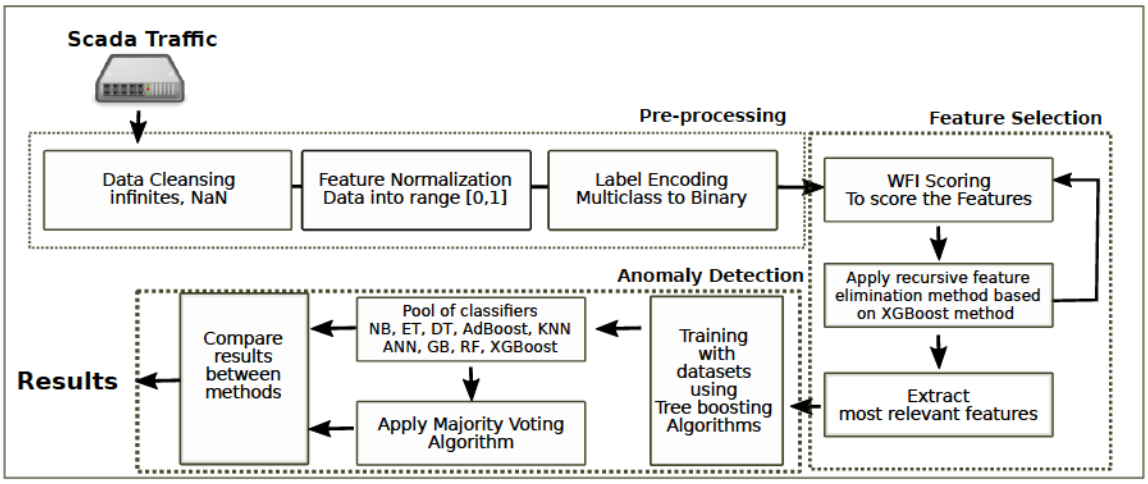


Figure 4.2: proposed framework for intrusion detection of power grids at control center

is part of large industrial control systems that use complex SCADA infrastructure to control the substation equipment, network monitoring devices such as SNORT, Wireshark, and Syslog are used to obtain the different types of features from the communication data [16]. Usually, streaming data that is obtained from sensors or actuators in real-time systems has reliability issues, such as lost signal or wrong observations due to failures in measuring devices which result in their inability to interpret the scale readings. For this reason, the data cleansing operation is a critical process to remove incorrect data (like infinities or NaN data). In this phase, we remove empty sequences that otherwise will generate issues such as inaccurate and faulty inferences with the algorithms. Moreover, the power grid records are collected at four PMUs (Phasor Measurement Units) which are situated at different locations in the power substation. Various internal attacks were launched by the ORNL to generate the IDS dataset for power grids reflecting the diverse nature of records. Another transformation that we performed in the data in this phase is the data normalization, to improve the training stability in the classifiers, especially for Artificial Neural Networks. For the normalization, a standard scaler method is used to normalize the records by considering zero mean and unit variance, was used.

In the feature selection phase, the importance of each feature is identified using the WFI scoring model. The recursive feature elimination approach is then applied to the binary dataset to eliminate irrelevant features recursively. Once the model determines the most consistent features, in the anomaly detection phase, the nine

classifiers, namely NB, ET, DT, RF, GB, XGBoost, ADBoost, KNN, and ANN are used to predict the output labels. Finally, the majority vote-based ensemble method predicts the class label for input samples based on the majority of the class labels predicted by each of these nine classifiers. The voting classifier uses "hard voting" to classify the input sample based on the majority class label.

4.2.3 Intermediate SCADA Center IDS: Proposed Framework based on semi-supervised ML approach using the concept of autoencoders for zero-day attacks

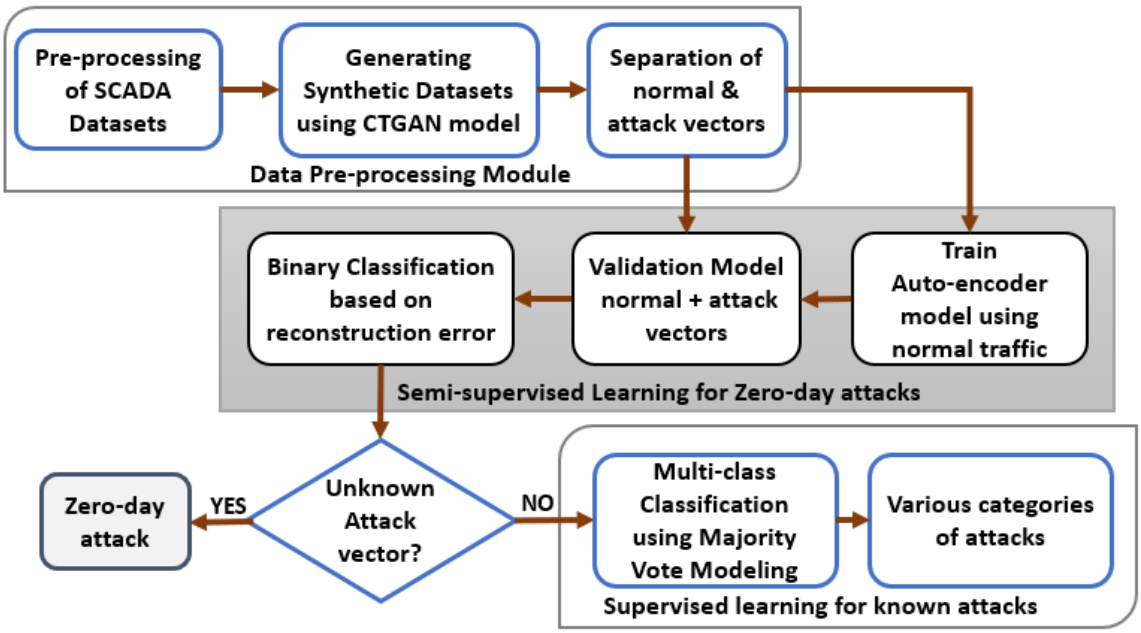


Figure 4.3: proposed framework for intrusion detection of power grids at intermediate SCADA center

This section presents the proposed scheme for an intrusion detection system for industrial control systems to detect zero-day attacks. This novel approach uses concepts of auto-encoders to determine the unknown attacks by training the model using normal traffic only. Furthermore, to validate the efficiency of the proposed model, the synthetic datasets have been generated using the CTGAN model. Once the model determines the attack vectors, in the next phase, the model detects the types of attacks whether it is known or unknown to the model. In case of unknown attacks, the record has been appended to the training dataset at the control center

to train the supervised learning model. In the case of known attacks, the supervised model will classify the type of known attacks. The approach of supervised learning has been proposed in one of our previous research where the accurate classification is done using the majority vote ensemble method. The researcher will find more details in [6, 61] for the detection of known attacks. This combined approach accomplishes two significant aspects of security, namely, the detection of unknown vulnerabilities in SCADA systems, and accurate real-time traffic monitoring for known attacks.

The entire framework is divided into three phases, namely, data preprocessing, semi-supervised modeling for the zero-day attacks, and anomaly detection for known attacks, as illustrated in Figure 4.3. We have followed the same approach for data preprocessing that was used in the development of the other two IDSs. For proper training of the deep learning models, we have generated new data samples from the existing samples using the CTGAN model. Further, this dataset has been separated into two subsets, namely, the normal dataset and the attack dataset. During the phase of semi-supervised learning, the autoencoders (vanilla and variational autoencoders) are trained using normal traffic. The training is accomplished using real datasets as well as synthetic datasets for the precise measurement of the model. Once the model has been trained, the validation is taken place using normal and attack events for the binary classification based on reconstruction error. In the case of the attack vector, the input event has been further verified at the control center to differentiate known and unknown attacks using a supervised learning model. For that, the majority vote-based ensemble method is used to predict the class label for input samples based on the majority of the class labels predicted by each of the nine classifiers as mentioned in the previous section.

4.3 Dataset Description

4.3.1 Power system Dataset

To determine the performance of the proposed approach, we have used three public benchmark datasets [62]. These datasets were created at Oak Ridge National Laboratories (ORNL) by setting up a power grid testbed [36]. This testbed was configured using various power grid components, namely, power generators – G1 and G2, IEDs – R1 to

Table 4.1: Description of the output labels of the various categories of the datasets

Categories	**Output Labels**
Binary (2)	Normal, Attack
Three States (3)	Normal, Attack, No Event
Seven States (7)	1-Natural SLG Fault, 1-Data injection attack 2- Remote Tripping Command injection attack 3- relay setting change attacks
Multi States (37)	1 – No event 8 – Natural events - 6 SLG faults - 2 Line maintenance events 28 – Attack events - 6 data injection SLG fault replay attacks - 4 command injection attacks against single IED - 2 command injection attacks against two IEDs - 10 relay setting change attacks on single IED - 4 relay setting change attacks on two IEDs - 2 relay disable and line maintenance attacks

R4, breakers – BR1 to BR4, and a three-bus two-line transmission system. In the case of fault detection, the IED trips the corresponding breaker depending on the nature of the fault. However, these IEDs are not smart enough to differentiate between original and fake failures. Moreover, operators can also manually trip the breakers and other system components during system maintenance [35]. The datasets derived from this power grid testbed contain measurements related to normal, disturbance, control, and cyber-attack behaviors captured during electrical transmission [37]. These datasets are randomly sampled and classified into three main categories, namely, binary, three-state, and multi-state. Initially, the multi-state dataset is constructed during the experiment and consists of a total of 37 scenarios.

These scenarios are mainly divided into three categories, namely, 8 natural events, one no event, and 28 attack events. The eight natural events are further divided into 6 SLG faults events and 2 line maintenance events, as listed in Table 4.1. Moreover, the 28 attack events are subcategorized into three major attack events, namely, Data Injection, Remote Tripping Command Injection, and Attack on Relay Settings. These

include 6 SLG fault replay attacks, 4 command injection attacks against a single IED, 2 command injection attacks against 2 IEDs, 10 relay setting change attacks on a single IED, 4 relay setting change attacks on 2 IEDs, and 2 relays disable and line maintenance attacks as listed in Table 4.1. These attack scenarios are simulated using the concept of an internal intruder, who can launch different attacks by issuing malicious injections from the substation [36]. Moreover, we have derived a seven-state dataset from the multi-states dataset.

Each power grid dataset consists of 128 features. To derive these features, 4 phasor measurement units (PMUs) are used to measure the electrical signals on an electrical power grid using a common time source to maintain time synchronization. Each PMU measures 29 features, hence in total 116 PMU measurements were carried out using 4 PMUs. These features are referred as R# - signal_Reference which indicates the index of PMU and type of measurement. For example, R1-PA1:VH represents the Phase A voltage phase angle measured by PMU R1 [63]. Also, 16 more columns are additionally inserted by control panel logs, snort alerts, and relay logs where relay and PMU are integrated together [64]. The last column represents the marker to label different events. The description of all the features is shown in Table 4.2. Also, each set of 15 sets consists average of 294 "no event" instances, 1221 natural events instances and 3711 attack vectors across the classification schemes [35].

Table 4.2: Description of features

Feature	Description
PA1:VH-PA3:VH	Phase A-C Voltage Phase Angle
PM1:V-PM3:V	Phase A-C Voltage Magnitude
PA4:IH-PA6:IH	Phase A-C Current Phase Angle
PM4:I-PM6:I	Phase A-C Current Magnitude
PA7:VH-PA9:VH	Pos.-Neg.-Zero Voltage Phase Angle
PM7:V-PM12:V	Pos.-Neg.-Zero Voltage Magnitude
PA10:VH-PA12:VH	Pos.-Neg.-Zero Current Phase Angle
PM10:V-PM12:V	Pos.-Neg.-Zero Current Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA:Z	Apparent impedance seen by relays
PA:ZH	Apparent impedance Angle seen by relays
S	Status Flag for relays

For the experimental study of zero-day attacks, we have targeted three SCADA-based

Table 4.3: Main Features of Gas pipeline and Water storage dataset

Feature	Description
PLC addresses	Station addresses of the Modbus devices for request and response
PLC read write functions	Modbus codes for reading and writing coil and register values
Length	length of the Modbus packet
Contol_mode	modes of the system such as automatic, manual, off
Control_scheme	Represents either the scheme is pump or solenoid
Pump	Pump control is on or off
crc_rate	pressure rates and measurements
Dead_band, Rest_rate, Set_points (* only for Gas pipeline dataset)	PID (Process ID) parameters for various measurements
HH, LL (* only for water storage dataset)	Level of the water in storage tank

industrial applications, namely, power grids, gas pipelines, and water storage systems. For power grids, we have created one single dataset by combining 15 datasets into one. The readers can find a detailed description of the dataset in [62]. The power system dataset consists of a total of 128 features with 22,714 normal events and 55,663 attack events.

4.3.2 Gas pipeline Dataset

The gas pipelines'dataset has been created at the lab scale testbed that includes information related to Modbus control packets, crc rates, and pump measurements, along with network traffic captured on a gas pipeline system at Mississippi State University's SCADA lab[65], [66]. Table 4.3 depicts the major features of the gas pipeline dataset. The output label represents 1 normal event and 6 attack events (including 2 response injection attacks, 2 command injection attacks, a code injection attack, and a Denial of Service attack) [63]. For binary classification, the output labels of 6 attack vectors are combined into one class and represented as an attack

event along with the normal event. The gas pipeline's dataset consists of a total of 26 features with 61156 normal events and 35,863 attack events.

4.3.3 Water Storage Dataset

The water storage dataset has also been generated on a lab-scale testbed at the same place where the gas pipeline dataset has been created [62]. The features of water storage and gas pipelines are almost similar that includes network traffic, process control, and measurement features for normal and attack events. The water storage parameters don't have features regarding the measurement of PID (process ID) such as rate, setpoint, cycle time, and deadband, instead, it includes features related to the level of the water in the tank. Table 4.3 depicts the description of the features of the water storage dataset. The dataset of the water storage tank consists of a total of 23 features with 172,415 normal events and 63,764 attack events. The features of the water storage and gas pipeline datasets are comparatively the same.

4.4 Experiments and Results

The experimental study and result analysis presented in this section are divided into three categories based on the placement of intrusion detection systems. This report highlights the results based on major contributions. However, the additional supporting results are presented in the attached papers (papers 1 and 2) in the Appendix section.

4.4.1 IDS for Plant Floor

Feature Selection: WFI scoring model based on Gradient Boosting

Generally, when we have a big model with hundreds or thousands of features, the feature selection approach is used to choose the most promising features and remove irrelevant features while retraining the model. Also, by analyzing the importance of each feature manually, we can get an idea of what the model is doing, and whether the model is working well. Here, we derive the importance of each feature by applying the WFI scoring method on Gradient Boosting trained model. Furthermore, all

the features are depicted as a percentage rating of how often the feature is used in determining the output label.

The feature importance scores reflect information gained by each feature during the construction of a decision tree. During experiments, we observe that 50% of the 128 features are not contributing to making any decision. The WFI score of such features is zero. While, out of the remaining 50% of features, 15 features provide a significant contribution in making decisions during the construction of the decision tree. The WFI score of those features has high values in the range of 1 to 10. The rest of the 45 features have feature importance scores between 0 and 1. These 45 additional features contribute comparatively less and have a large drop in feature importance score. Altogether the entire dataset is divided into three levels of information gain groupings, namely, most promising, slightly contributing, and irrelevant features. According to [67], feature extraction creates a subset of the given features which not only reduces the noise but also improves the classifiers' performance. Therefore, we have tested 15 datasets of four different categories (binary, three-class, seven-class & Multi-class) of the power grid systems created by the Oak Ridge National Laboratories using the most promising features [63]. To identify these best features, we use the WFI scoring model along with the concept of Num_trees.

Furthermore, to increase the execution speed, we perform feature extraction on binary datasets. We repeat the entire process by taking the various parameter value of Num_trees to collect various observations. From that, we have identified the best features by taking common important features from the estimations. Here, Num_trees refers to the number of estimators whereas n refers to the total number of features. We have used four estimators, namely, 100, 500, 700, and 1000, and initially, the dataset consists of $n = 128$ features.

Figure 4.4 represents the relative importance of each attribute on the binary dataset by considering four estimators. The high vertical bars represent the most promising and common features in all four estimators. In this experiment, all estimators use the top 15 features for each ensemble. In Table 4.5. we observe the most promising features across all 15 datasets. Also, to validate the strength of the selected features, the same 15 ones are applied to all four categories (Binary, three classes, seven classes, and Multi-class) of intrusion classification. It can be observed that each dataset has

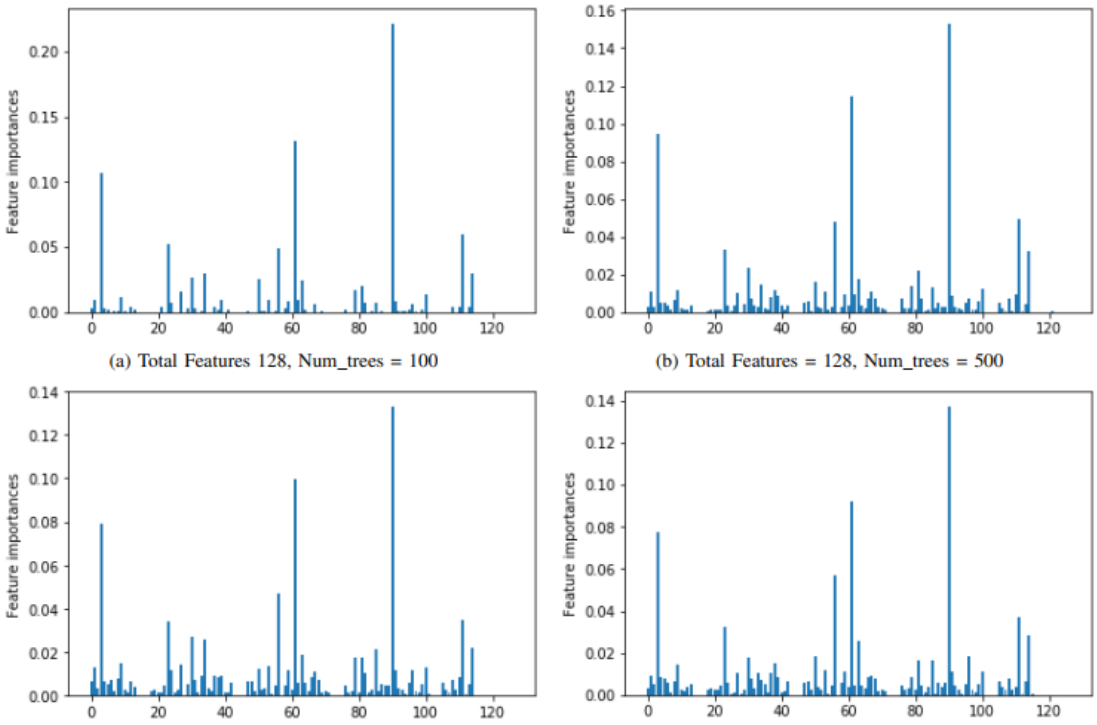


Figure 4.4: Represents the relative importance of each attribute of the dataset with 5000 records; computed by considering four estimators Num_trees = 100,500,700,1000

features	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
D1	R2-PA3:VH	R1-PM10:I	R1-PA1:VH	R2-PM1:V	R2-PA10:IH	R2-PA5:IH	R2-PM10:I	R3-PA5:IH	R1-PM5:I	R3-PA1:VH	R2-PM5:I	R1-PA5:IH	R4-PA1:VH	R3-PA7:VH	R1-PA7:VH
D2	R2-PM3:V	R4-PA2:VH	R1-PA2:VH	R2-PM1:V	R4-PA1:VH	R1-PA5:IH	R4-PM2:V	R1-PA1:VH	R3-PM7:V	R2-PA3:VH	R2-PA7:VH	R4-PM1:V	R3-PA5:IH	R3-PM5:I	R1-PA7:VH
D3	R3-PA4:IH	R2-PM10:I	R3-PA2:VH	R2-PA2:VH	R2-PM5:I	R4-PA1:VH	R2-PA4:IH	R3-PM2:V	R2-PA5:IH	R1-PM5:I	R3-PA3:VH	R2-PA3:VH	R3-PM5:I	R1-PA7:VH	R4-PM5:I
D4	R2-PA7:VH	R4-PM5:I	R4-PA2:VH	R4-PA3:VH	R4-PA7:VH	R1-PA5:IH	R4-PM2:V	R2-PA2:VH	R2-PA5:IH	R4-PA1:VH	R1-PM5:I	R4-PA5:IH	R1-PA3:VH	R1-PA2:VH	R2-PM5:I
D5	R3-PA7:VH	R1-PA5:IH	R4-PM2:V	R4-PA4:IH	R4-PA5:IH	R4-PM5:I	R3-PM6:I	R3-PA10:IH	R2-PA5:IH	R3-PA3:VH	R1-PA3:VH	R4-PM4:I	R4-PA1:VH	R2-PA2:VH	R4-PA7:VH
D6	R4-PA1:VH	R3-PM2:V	R3-PA2:VH	R4-PM3:V	R1-PA2:VH	R4-PA7:VH	R2-PA10:IH	R4-PA2:VH	R2-PA5:IH	R1-PM10:I	R1-PA7:VH	R4-PM2:V	R3-PA5:IH	R4-PM5:I	R1-PM5:I
D7	R4-PA6:IH	R1-PA7:VH	R1-PM5:I	R1-PA1:VH	R2-PM7:V	R1-PA6:IH	R4-PA7:VH	R3-PA5:IH	R3-PA6:IH	R4-PA1:VH	R4-PA3:VH	R3-PM2:V	R4-PM7:V	R2-PA3:VH	R3-PA3:VH
D8	R4-PA7:VH	R1-PM2:V	R1-PA2:VH	R2-PA3:VH	R1-PA5:IH	R2-PA1:VH	R1-PM5:I	R1-PA3:VH	R3-PA5:IH	R3-PA6:IH	R3-PA2:VH	R4-PA3:VH	R4-PM2:V	R4-PA1:VH	R4-PA5:IH
D9	R2-PA2:VH	R4-PM7:V	R2-PM5:I	R4-PA1:VH	R3-PA2:VH	R1-PA3:VH	R4-PA7:VH	R3-PA5:IH	R1-PM2:V	R1-PA2:VH	R4-PA5:IH	R2-PA3:VH	R3-PA3:VH	R4-PA2:VH	R4-PM2:V
D10	R3-PA4:IH	R1-PA1:VH	R1-PA7:VH	R4-PA5:IH	R4-PA7:VH	R2-PM1:V	R1-PA5:IH	R4-PA1:VH	R4-PM5:I	R4-PM7:V	R3-PM2:V	R2-PA5:IH	R4-PA2:VH	R4-PM2:V	R3-PA5:IH
D11	R2-PA4:IH	R3-PA5:IH	R4-PM1:V	R1-PM5:I	R2-PM5:I	R2-PA1:VH	R4-PM5:I	R2-PM7:V	R1-PA2:VH	R2-PA6:IH	R2-PA5:IH	R4-PA2:VH	R2-PM1:V	R4-PM7:V	R4-PM2:V
D12	R4-PA3:VH	R4-PA7:VH	R2-PA3:VH	R1-PA2:VH	R2-PM5:I	R1-PM5:I	R3-PM2:V	R2-PA5:IH	R3-PA5:IH	R4-PA1:VH	R4-PA2:VH	R4-PM5:I	R1-PA:ZH	R1-PA3:VH	R3-PM5:I
D13	R3-PA3:VH	R3-PA2:VH	R2-PA3:VH	R3-PA5:IH	R4-PA6:IH	R1-PA1:VH	R4-PA7:VH	R4-PA2:VH	R4-PM2:V	R3-PM2:V	R2-PA5:IH	R1-PA7:VH	R1-PA3:VH	R4-PA3:VH	R4-PA1:VH
D14	R3-PM10:I	R1-PA1:VH	R2-PA5:IH	R4-PM7:V	R1-PM2:V	R4-PA5:IH	R1-PA2:VH	R2-PA6:IH	R3-PA3:VH	R3-PA6:IH	R1-PA7:VH	R4-PA3:VH	R1-PA6:IH	R1-PA3:VH	R4-PM2:V
D15	R4-PA1:VH	R4-PM7:V	R2-PM5:I	R1-PA5:IH	R3-PM3:V	R1-PA7:VH	R3-PA5:IH	R4-PM5:I	R2-PA3:VH	R2-PA5:IH	R4-PA7:VH	R1-PA1:VH	R3-PA3:VH	R3-PM2:V	R4-PM2:V

Figure 4.5: Best 15 Features of 15 Datasets for all the four categories - Binary, Three classes, Seven classes, and Multi-class

a different set of stronger features, a conclusion that points to an independent feature selection process for each dataset type.

Evaluation parameters

The choice of the evaluation parameters always depends on the nature of the dataset, whether it is a multi-class or just binary. Typically, datasets are imbalanced in nature, a property defined by having classes of different sizes. Hence to evaluate the efficiency of the proposed GBFS-based framework, our approach not only relies on the accuracy of the classifier but also incorporates other assessment parameters like Detection Rate (True Positive Rate also called Recall & True Negative Rate), Precision, F1 Score and Miss Rate (False Negative Rate).

The assessment metrics, namely, accuracy, recall, precision, and false negative rate depend on the following four parameters, namely, True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) [38]. TP refers to the number of actual attacks which are classified as attacks, TN refers to the number of normal events classified as normal events, FP refers to the number of normal events misclassified as attacks and FN refers to the number of attacks misclassified as normal events. The evaluation metrics are defined as follows, described from the basic four definitions.

- Accuracy is the percentage of all normal and attack vectors that are correctly classified:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

- Detection Rate (True Positive Rate (TPR) and True Negative Rate (TNR)) refers to the percentage of total relevant results correctly classified by the classifier

$$TPR = \frac{TP}{TP + FN} \text{ (attack vector)} \tag{4.2}$$

$$TNR = \frac{TN}{TN + FP} \text{ (normal event)} \tag{4.3}$$

- Precision or Positive Predictive Value (PPV) refers to the percentage of the results which are relevant.

$$PPV = \frac{TP}{TP + FP} \text{ (attack event)} \tag{4.4}$$

- F1 Score is simply the harmonic mean of precision and recall evaluating the outcome in a balanced mode

$$F1\_score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{4.5}$$

- Miss Rate (FNR/FPR) is derived by subtracting the value of TPR from 1.

$$FPR = 1 - TNR \text{ (attack)} \tag{4.6}$$

$$FNR = 1 - TPR \text{ (normal)} \tag{4.7}$$

Result Discussion

The purpose of the proposed GBFS-based feature selection framework is to generate a subset of the given attributes from the entire dataset using a WFI metric to reduce the noise and improve the performance of the classifier. The derived subset of the top 15 features may or may not contribute the same in the decision-tree classifiers. We have observed the results of a total of 8 decision tree-based machine learning techniques to validate our proposed methodology via multiple simulation trials. Overall 60 computations are performed to evaluate the performance of each classifier including the results of fifteen datasets of all four categories. Figure 4.6 represents the comparative analysis of the accuracy of eight tree-based classifiers of 15 datasets of each binary, three-class, seven-class, and multiclass categories.

Amongst all the eight classifiers, it was observed that XGBoost, random forest, and its variance have proven to be the most efficient. However, other tree-based classifiers also proved their efficiency ranging between 92 to 94 for Binary and three-state and 85 to 90 for seven-class and multiclass. XGBoost comes up with accuracy equal to 97.26, 96.09, 92.97, and 92.44 for binary, three-class, seven-class, and multiclass datasets, respectively. Similarly, all three variants of Random Forest also achieve very high accuracy such as 97.26, 97.24, and 97.17 for binary, 96.18, 96.38, and 96.50 for three-class, 94.43, 94.31, and 94.19 for seven classes and 92.46, 92.92, 91.92 for multiclass, respectively.

Since the GBFS-Random Forest and its variances are the most efficient classifiers to classify the normal and attack vectors with the nearly same range of accuracy, we have compared the execution speed of all three classifiers to identify the best among

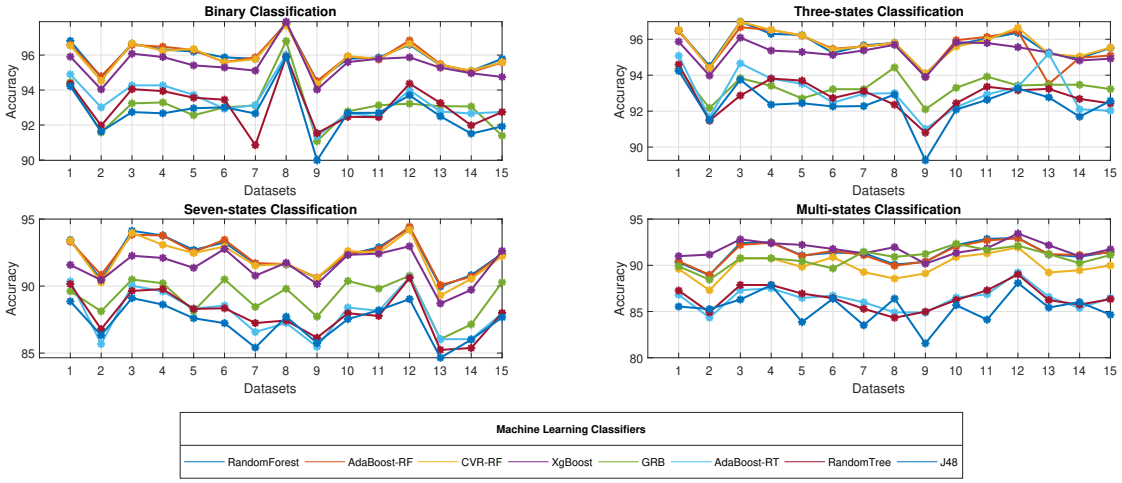


Figure 4.6: Comparative view of Different Machine Learning Classifiers for - four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets

them. As depicted in Figure 4.7, GBFS-Random Forest classified the various attack and normal events for all four categories in 1.5 seconds. GBFS-AdaBoost Random Forest took slightly more time than the GBFS-RF. GBFS-CVR-Random Forest took comparatively higher execution time as it uses the combined approach of boosting and ensemble of trees for the classification. However, by comparing the accuracy levels, we observe that the boosting does not improve the result much, in such case GBFS-RF is proven to be best amongst all three with high accuracy and less execution time.

We demonstrated that the 15 stochastic features shown in Figure 4.5 were the most promising features for all the decision tree-based classifiers by iteratively running all eight classifiers, for 15 datasets of all four categories. In each iteration, using 15 features, we retrained & re-tested all eight tree-based models to compute the general average trend of malicious and normal events by observing DR, FPR, and Execution Time.

Measure	Binary	Three-class	Seven-Class	Multi-class
Accuracy	97.26%	96.50%	94.12%	92.46%
FPR	0.037	0.067	0.019	0.003
Precision	0.9705	0.9887	0.9504	0.9250
Recall	0.9740	0.9676	0.9355	0.9240
F-Measure	0.9723	0.9781	0.9427	0.9244

Table 4.4: Performance evaluation metrics of Proposed GBFS Based Classifier

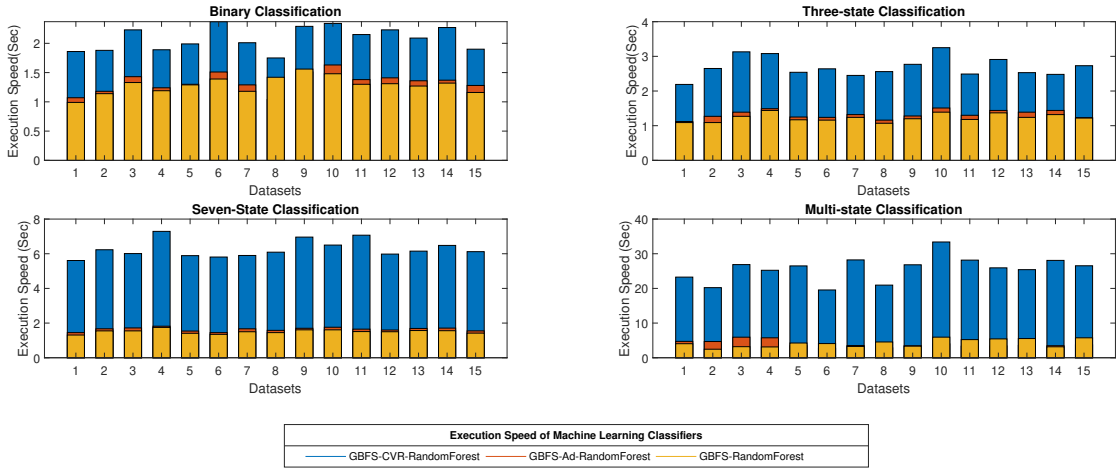


Figure 4.7: Comparative view of Execution speed of Three GBFS-based Random Forest variances to classify normal and attack events for four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets

All the selected classifiers maintain very high DR and lower FPR rates in all the computations as shown in Table 4.4. We have achieved 98.5% detection rate which truly differentiates attack and normal vectors with only 3.7% and 6.7% false positive rate for binary and three class classifications. Moreover, seven-class and multi-class classifiers have also outperformed as they gave around 94.42% and 92.5% for the detection rate. This validates the significance of our proposed methodology for feature selection. Real-time systems such as control and monitoring systems of industrial infrastructures/power grids need a methodology of feature extraction where processing time and storage space are always crucial.

4.4.2 IDS For Control Center

Feature Selection: Recursive Feature Elimination (RFE) method

The primary objective of the proposed model is to provide real-time intrusion detection for power-grid systems. Hence, our target is to build a fast and accurate model that captures any malicious event efficiently that may happen in the network. To fulfill both requirements, we have used the RFE-XGBoost based WFI scoring model for feature selection along with the majority vote-based ensemble method for prediction. The feature selection module improves the computational cost as we are targeting the

30 most consistent features out of 128 features of the given datasets. Furthermore, we have used the nine most powerful classifiers to classify normal and malicious events. For more accurate results, we have applied the majority vote-based ensemble method, which predicts the class label based on the majority of the class labels predicted by each of these classifiers.

These datasets used in our analysis are the publicly available datasets generated at the ORNL laboratory on a small power grid testbed [16]. For proper validation, experiments were computed for four different categories of the samples. Furthermore, the observations were carried out using 100,000 normal and attack events of each of these four categories, which were divided into 15 datasets. For fair distribution and assessment, each dataset was split randomly into two subsets, training (80%) and testing (20%). The training data was used for the algorithm training and the testing data was used to test the accuracy of the result. To avoid selection bias in the datasets and to reduce overfitting, we have used a 10-fold cross-validation technique during the training process. This method performs the training 10 times with different random selections (80/20) from the original dataset. This well-defined systematic approach circumvents the inadequacy of bias performance assessment. The proposed approach is implemented using Python on a Jupyter notebook using the Anaconda distribution platform on Windows 10 with an Intel Core i5-8300H 2.30GHz processor, 8 GB RAM, and Nvidia Geforce GTX 1060 GPU.

We have made observations based on the number of subsets of the features considering 15 binary datasets. Initially, we started with 128 features and reduced the number of features in each iteration based on the output of the WFI scoring model to compare the accuracy of the current set with the selected subset. To extract the gist of the features, we have applied WFI based scoring model which scores the importance of all features. This ranking defines how often the feature is used to determine the output label while constructing a tree. Figure 4.8 illustrates the comparative analysis of different features versus the accuracy graph of one of the 15 datasets. The classification with 30 features offers the highest accuracy during the classification of normal and attack events using the majority vote-based ensemble classifier.

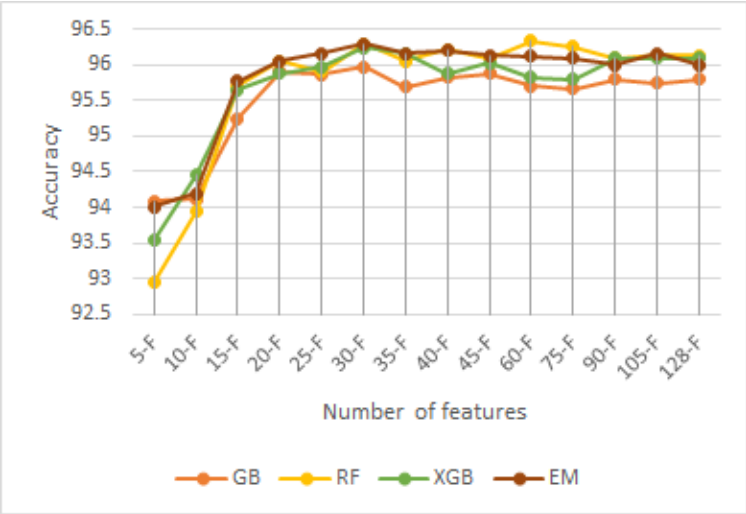


Figure 4.8: Comparative analysis of different features to evaluate the accuracy using RFE-XGBoost WFI scoring model.

Result Discussion

To evaluate the performance of the majority vote-based ensemble algorithm, we have computed the accuracy of fifteen datasets of all four categories using the nine most promising classifiers. The choice of these classifiers is carried out based on our preliminary results of the comparative analysis of various machine learning classifiers [6]. We have chosen nine heterogeneous classifiers to determine the efficiency of selected features via multiple simulation trials and observed the predictions of all the algorithms. After deriving the accuracy of all nine classifiers, the majority vote ensemble algorithm was applied to compare the prediction of the output labels. The comparison was carried out based on the majority class label voting classifier with "hard voting" to classify the input samples.

The ensemble algorithm predicts accurate outcomes by aggregating and applying the majority vote rule on the result of the different classifiers. We have incorporated heterogeneous classifiers, namely, random forest (RF), gradient boosting (GB), XGBoost (XGB), Extra Tree (ET), Decision Tree (DT), K-Nearest Neighbor (KNN), Naive Bayes (NB), Adaboost – Decision Tree (AdBoost-DT), and artificial neural network (ANN) to achieve performance improvement of the majority vote based ensemble model.

We have performed an overall 60 computations of each of the four categories

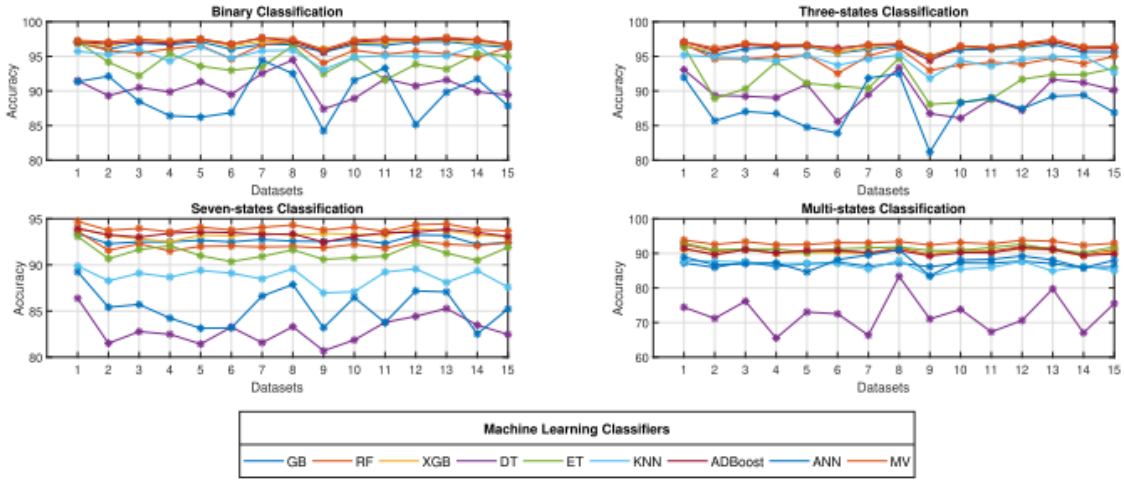
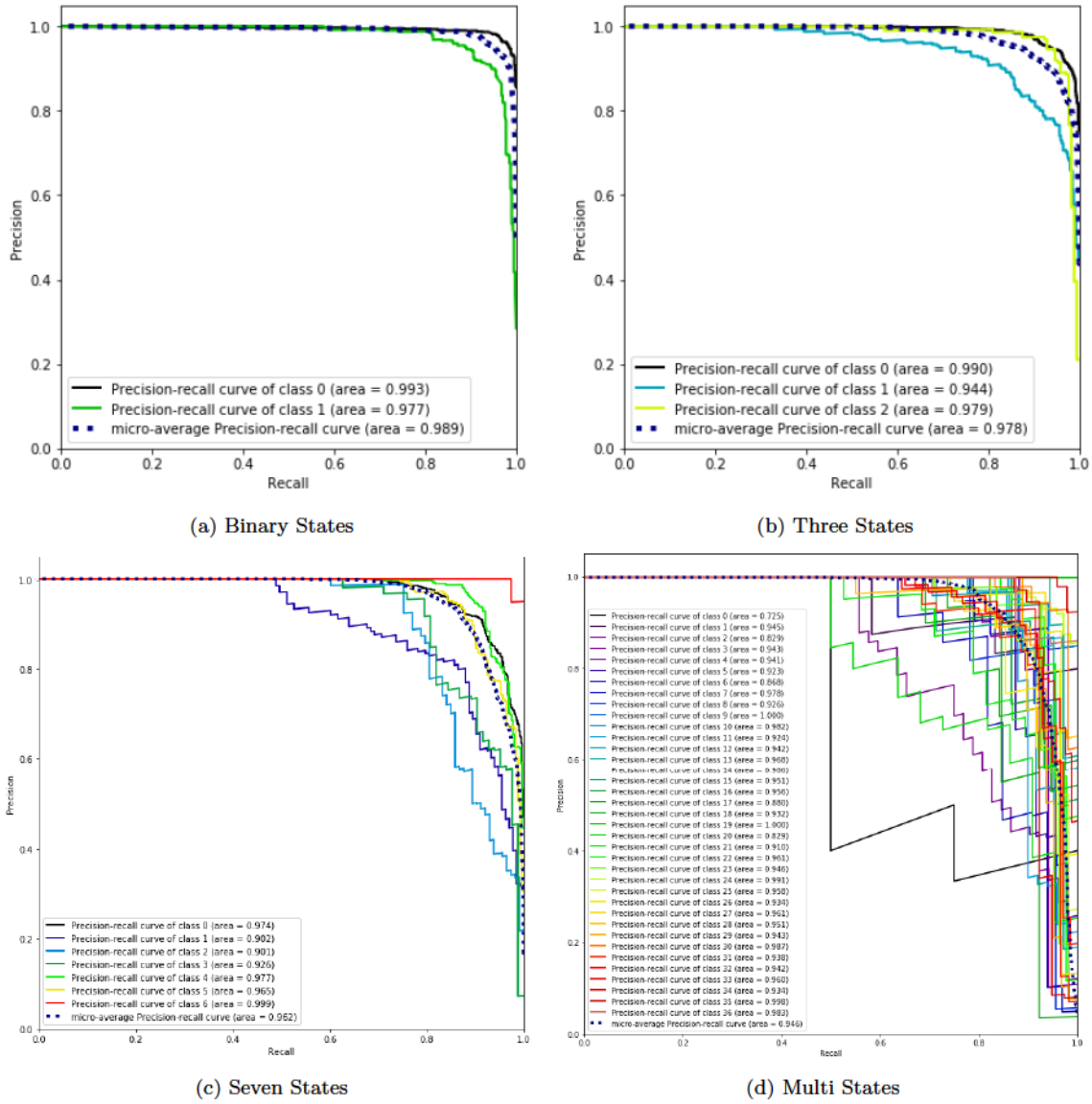


Figure 4.9: Comparative view of different Machine Learning classifiers for four categories for each of the fifteen datasets

(binary, three-states, seven-states, and multi-states) containing fifteen datasets to evaluate the performance of each of the ten classifiers. According to the analysis, the accuracy of the Naive Bayes algorithm is less compared with other classifiers for all four categories, namely, binary (around 52.34%), three class (58.21%), seven class (19.26%), and multi-class (13.2%). Figure 4.9 presents a comparative analysis of the accuracy of the remaining eight classifiers along with the majority vote-based ensemble algorithm. Among nine base classifiers random forest, gradient boosting, and XGBoost have mostly proven to be more efficient in the case of binary, three states, and seven states classification. However, for multi states classification, random forest, extra tree, and XGBoost are more promising than the other six classifiers.

In the case of imbalanced datasets, the PR plot is more informative than the ROC plot while evaluating classifiers [68]. Here we are not only targeting binary classification but also classifying multiple attack events. Hence, for more information retrieval, we have also analyzed PR curves in case of bias in the class distribution. The baseline of the PR curve is determined by the relation of precision and recall values. Figure 4.10 depicts the precision/recall for each threshold for a majority rule-based ensemble model by considering all four categories of the dataset. For all the four types, the majority rule-based ensemble classifier maintains a high detection rate. The proposed model has achieved 98.9%, 97.8%, 96.2%, and 94.6% of the average



(a) Binary States

(b) Three States

(c) Seven States

(d) Multi States

Figure 4.10: Precision-Recall Curves of RFE-based Majority vote ensemble method for four categories

precision-recall curve area for binary, three states, seven states, and multi-states, respectively. The exact percentage of each output label is depicted in Figure 4.10. The results indicate the model performs exceptionally well with all the categories to predict various types of class labels.

We have observed the importance of the various features in the previous section, where accuracy is measured by considering subsets of the features. In that, we have focused on the binary dataset. For further proof of concept, we have evaluated the accuracy of three other categories, namely, three-class, seven-class, and multi-class

Table 4.5:
Comparison of accuracy of majority vote ensemble algorithm with and without recursive feature elimination based feature selection

Classifiers	**Without Feature Selection** 128 features (Accuracy)	**RFE - Feature Selection** 30 features (Accuracy)
Binary	96.93	97.44
Three-Class	96.64	97.25
Seven Class	93.65	94.91
Multi-Class	92.23	93.08

datasets, by comparing all the 128 with 30 features. To extract the gist of the features, we have applied an RFE-based WFI scoring model, which scores the importance of all features recursively. This ranking defines how often the feature is used to determine the output label while constructing the tree. Table 4.5 illustrates the comparative analysis of four categories by considering 128 features versus 30 features extracted by RFE. The classification with 30 features offers the highest accuracy during the classification of normal and attack events using the majority vote ensemble classifier. In Table 4.5, we have presented the result of one of the 15 datasets. During experiments, we have also observed that the training time of multi states datasets with all the 128 features is unrealistic as it took more than 24 hours. Hence, feature selection is a crucial factor used to develop a better predictive model and make the model computationally efficient.

4.4.3 IDS for Intermediate SCADA Centers (sub-MTUs)

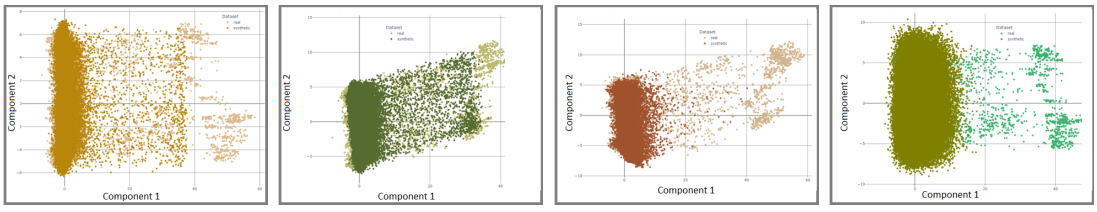
Generation of Synthetic Datasets

For efficient training of the ML model and for accurate prediction, we have generated the synthetic datasets from the existing original datasets for all three ICS applications using the CTGAN model. For that, we have used the SDV Python library that provides the utility of a GAN-based Deep Learning data synthesizer for tabular datasets. First, we created an instance of CTGAN and fit that instant for the given dataset. The CTGAN model learns the statistical properties of the dataset during the training process and accordingly generates synthetic data that captures the characteristics of the model. To control the learning behavior, CTGAN has many hyper-tuning parameters which will impact the performance of the model, both in

Table 4.6: Hyper-tuning parameters of CTGAN model

Dataset	**Hyper-tuning parameters**
Power-grids	epochs=500, batch_size=500, embedding_dim=256, generator_dim=(512,512), discriminator_dim=(512,512), generator_lr=0.0003, discriminator_lr=0.0003, discriminator_steps=5
Gas-pipeline, Water-storage	epochs=50, batch_size=150, embedding_dim=64, generator_dim=(128,128), discriminator_dim=(128,128), generator_lr=0.0001, discriminator_lr=0.0001, discriminator_steps=5

terms of the quality of the dataset and computational time. We have applied grid search to tune the parameters. Table 4.6 represents the hyper-tuning parameters that we have used to improve the overall score that represents the quality of the dataset. The SDV tool has provision to verify the quality of the synthetic data using evaluate function. This function evaluates the given dataset based on various statistical and probabilistic properties (such as KS Complement, ChiSquare Test, Logistic Detection test, Binary Decision Tree test, and BNLikelihood test)and accordingly returns the overall score of the model. We have used this function to verify the quality of the synthetic data before and after hyper-tuning. We obtained 85.74% overall score before hyper-tuning (using default parameters) and obtained 90.87% overall score after hyper-tuning for a power-grids dataset. Similarly, we obtained 93.27% for the gas pipeline dataset and 92.07% overall score based on statistical properties for the water storage dataset. We generated a total of 2,000,000 records (50% normal events, 50% attack events) for all three datasets.



(a) CopulaGAN (b) CTGAN (c) TVAE GAN (d) GaussianCopula GAN

Figure 4.11: Representation of power grid's data consist of the real and synthetic dataset (Fine-tuned post processed synthetic dataset)

Moreover, we have used a popular dimensionality reduction technique called Principal

Component Analysis (PCA) for visualizing the synthetic and real datasets. Figure 4.11 represents the transformation of 2 components to the latent space for the power system dataset using four tabular GAN models, namely, CopulaGAN, Conditional Tabular Generative Adversarial Network (CTGAN), Triplet-Based Variational Encoder Generative Adversarial Network (TVAE-GAN), and GaussianCopula GAN. This figure shows a similarity between real and synthetic datasets by overlapped data points. This visualization also validates the quality of the synthetic dataset.

Experimentation Methodology

Anomaly detection with vanilla autoencoders :

During these experiments, we used a vanilla autoencoder method with an encoder and a decoder model, that has layer size in decreasing and increasing fashion of neurons. There are two phases in the anomaly detection process, namely, the training phase, and, the inference phase. During the training phase, a network is trained with normal data to obtain a reconstruction of the given input. In the inference phase, the architecture is used to obtain a reconstruction of the data, and the attack traffic is identified as its reconstruction error that differs from the normal data reconstruction as shown in Figure 4.12.

Autoencoders work on the concept of latent space, which represents a vector of encoder and decoder networks. This vector is in form of reduced dimensionality and contains a distillation of the features of the original data, that allows good reconstruction. In the proposed experiments we have used latent spaces of 2 and 16 cells to focus on the important research question - what is the impact of different latent space sizes when one of the objectives is to focus on accurate prediction? To find the optimum network architecture, we performed a hyper-parameter tunning, where we focused on two complementary areas, namely, the network structure, and the latent vector dimension. We explored several combinations of deeper and shallower networks using Encoder-Decoder models. During analysis, we concluded that shallow architecture is sufficient to obtain promising results.

The result discussion of these experiments using vanilla autoencoders is further detailed in the following subsections.

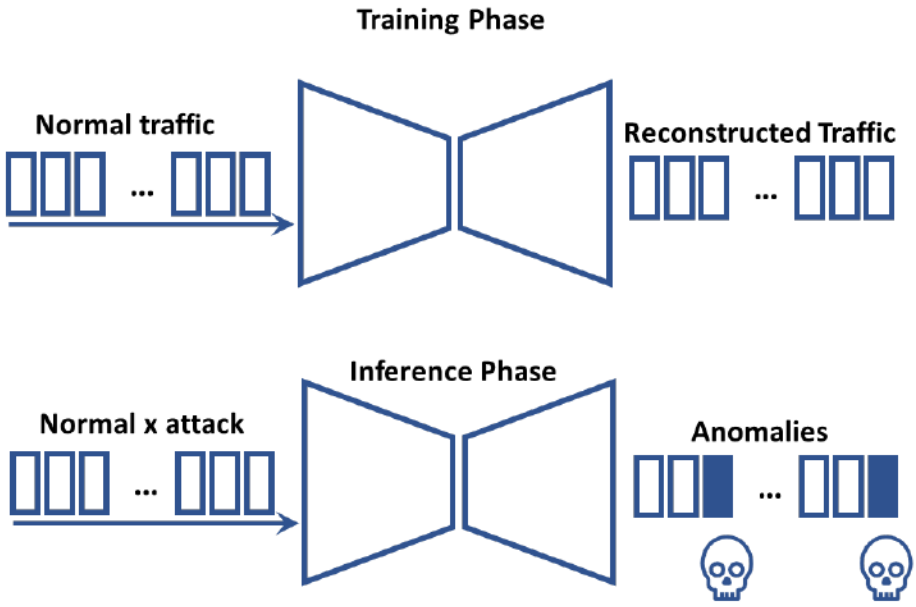


Figure 4.12: Models train with normal data and inference is made on normal and attack data

First Batch: Power grids SCADA Datasets

The first batch of experiments was performed on the power system dataset. One way to visualize the resulting quality is to obtain a two-dimensional image of the latent space using a PCA transformation. We perform a PCA transformation with 2 components on the latent space of the network. The purpose of visualizing the PCA distribution using two components is to verify the efficiency of the model in terms of discriminating between normal and attack events. Figure 4.13 represents the impact of the size of the latent space that indicates the separation of principal components. For this experiment, we have used the original power grid dataset.

Moreover, to overcome the limitation of the size of the dataset, we have used a set of synthetic data to explore two research questions: What is the impact of the synthetic data on the modeling and training process of autoencoders? How does synthetic data contribute to effectively improving the accuracy of the results? During the experiments with synthetic data, we trained the vanilla autoencoder model using synthetic normal events and tested them with real attack vectors. The purpose of this experiment is to verify the filtering behavior with real attack vectors. Figure 4.14 illustrates how the PCA transformation is applied to the latent space vector using synthetic data considering the same architectures and the same latent space sizes.

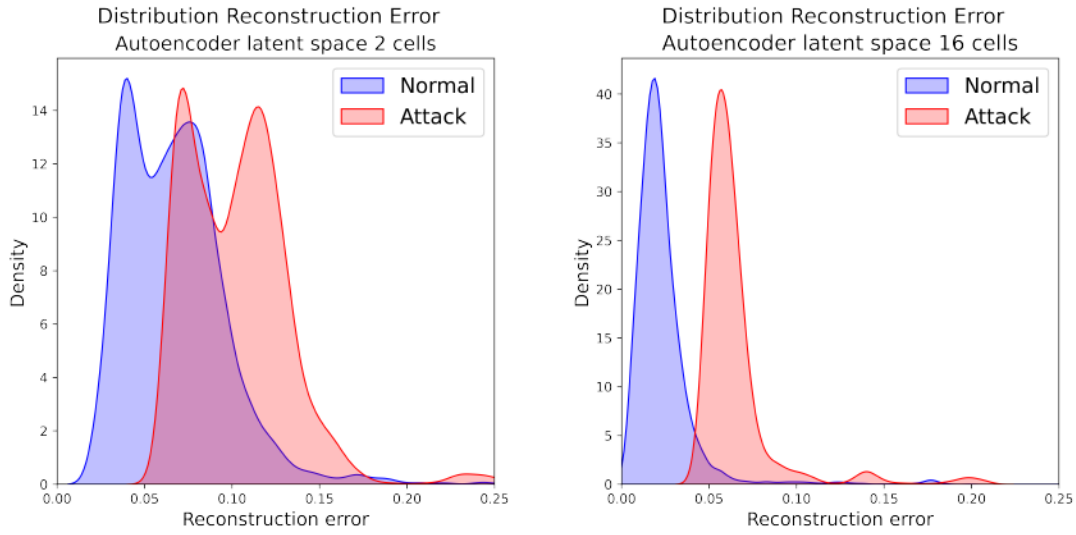


Figure 4.13: Vanilla Autoencoder - Using Real Power-grids Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space

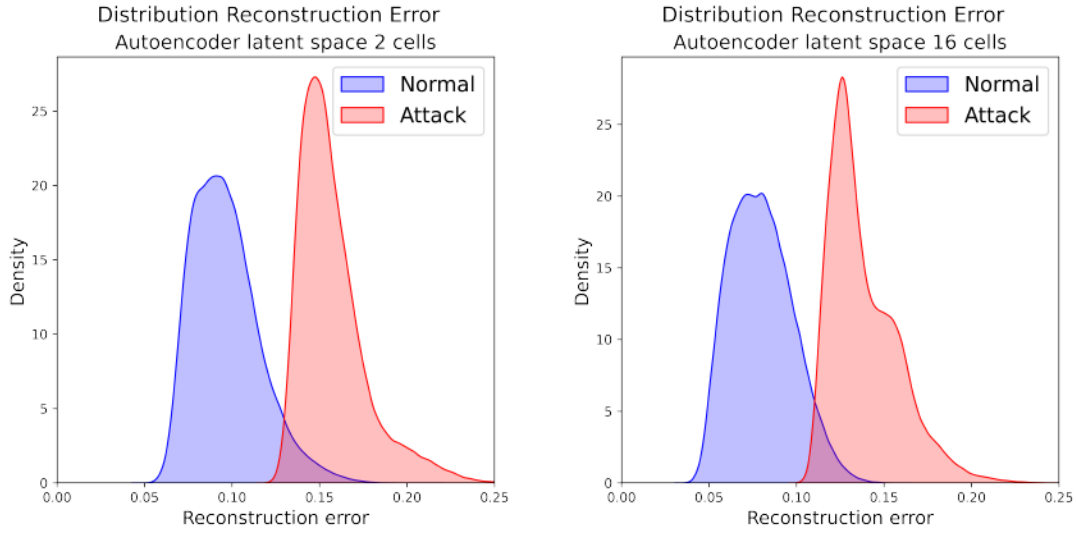


Figure 4.14: Vanilla Autoencoder - Using Synthetic Power-grids Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space

We conclude that the use of synthetic data supports more stable training, and optimizes the result in terms of false positive rate compare to using real datasets.

Second Batch: Gas Pipelines SCADA Datasets

In this experimentation, we use gas pipelines Pipeline datasets. Using the same autoencoder architectures (Latent space 2 and 16) we accomplish a good separation

of the two classes. In Fig. 4.15 we illustrate this result. We can see that even with a small latent space we obtain results that are close to 100% accuracy as the reconstruction error clearly identifies the normal class from the attack class.

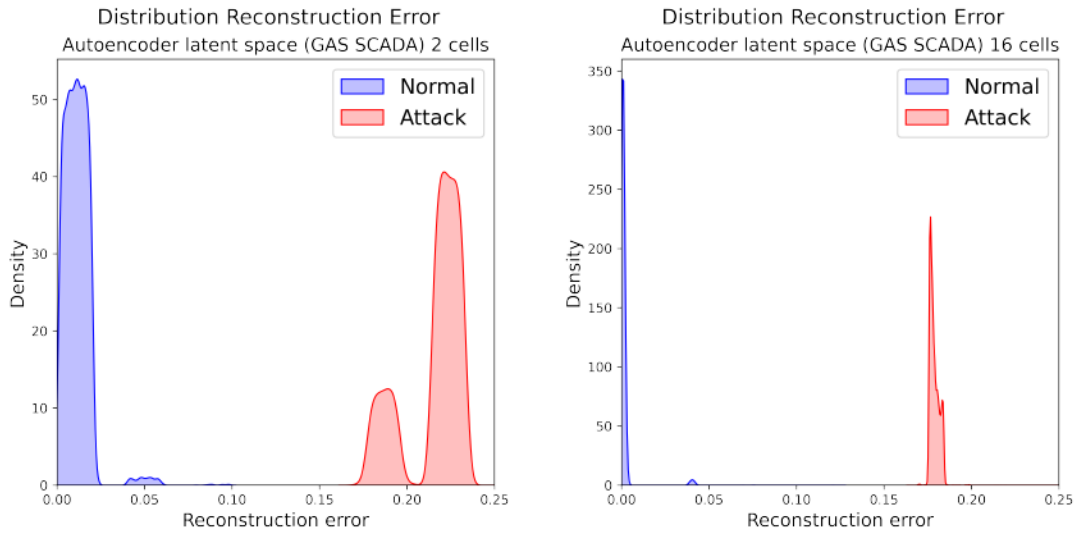


Figure 4.15: Vanilla Autoencoder - Using Real Gas Pipeline Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space

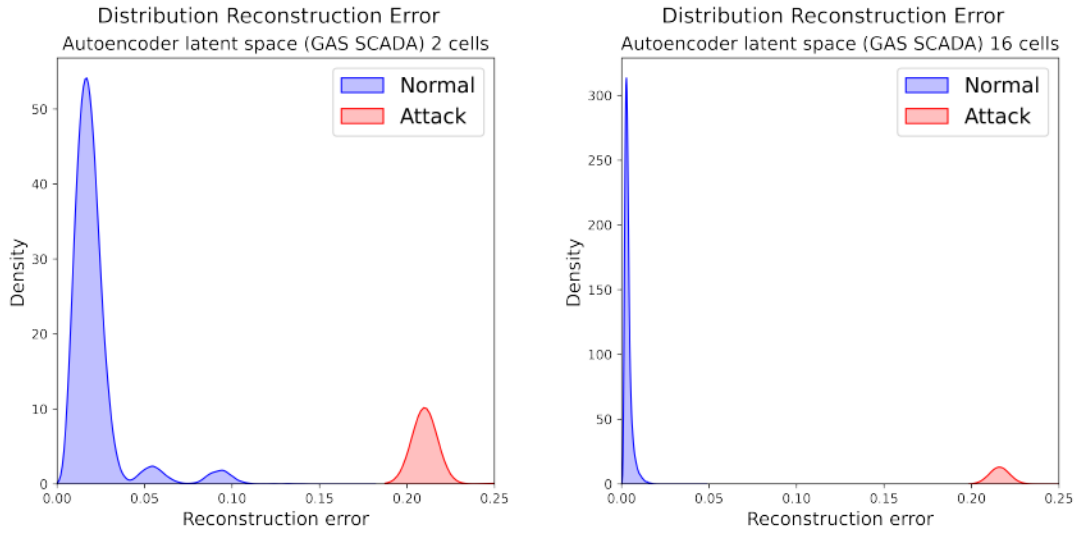


Figure 4.16: Vanilla Autoencoder - Using Synthetic Gas Pipeline Dataset: Error distribution of error reconstruction of Normal and Attack events with 2 and 16 cells in the latent space

To validate our proposed model further, we have run the experiments using gas

pipeline datasets. Using the same autoencoder architectures (latent space 2 and 16), we accomplished a good separation of the two classes, namely, normal and attack events. Figure 4.15 illustrates the result based on the original dataset of the gas pipeline testbed. The smaller latent space (LS=2) is sufficient to obtain perfect prediction accuracy as the reconstruction error identifies clearly the normal class from the attack class.

Moreover, we have validated the same approach using a synthetic dataset generated for the gas pipeline, and the results are quite similar as shown in Figure 4.16. We obtain a clear class separation with an accuracy result close to 94%. Furthermore, this approach validates that a synthetic normal dataset is as good as a real dataset to identify unknown attacks (the attack vectors used in this experiment for testing are from a real dataset). However, looking at the entire experimental results, we have concluded that the real dataset is sufficient to train the models for accurate prediction. The synthetic dataset is not required as it neither improves class separation nor accuracy. Accuracy is at 100% already with original normal traffic. The results indicate that the class separation between attack and normal traffic is simpler in the gas pipeline dataset as it doesn't have complex features to train and we can obtain perfect results using normal traffic of the original dataset with vanilla autoencoders.

Third Batch: Water Storage System's Dataset

We apply the same autoencoder architecture to water storage datasets. The structure of the dataset is simple as it only consists of 23 features. Moreover, the size of the real dataset is sufficient to train the model using real normal traffic. However, for the validation of synthetic data, we have generated the results based on real and synthetic datasets. In both cases, we obtain a perfect separation of the attack and normal classes as presented in Figure 4.17, and Figure 4.18. Like the gas pipeline dataset, the smaller latent space (LS=2) is sufficient to obtain perfect prediction accuracy as the reconstruction error identifies clearly the normal class from the attack class.

Anomaly detection with VAE autoencoders:

As discussed before in the Introduction section, the vanilla autoencoder uses a simple structure, while the variational autoencoder uses an advanced structure during the learning and reconstruction process. VAE autoencoders use the reparametrization

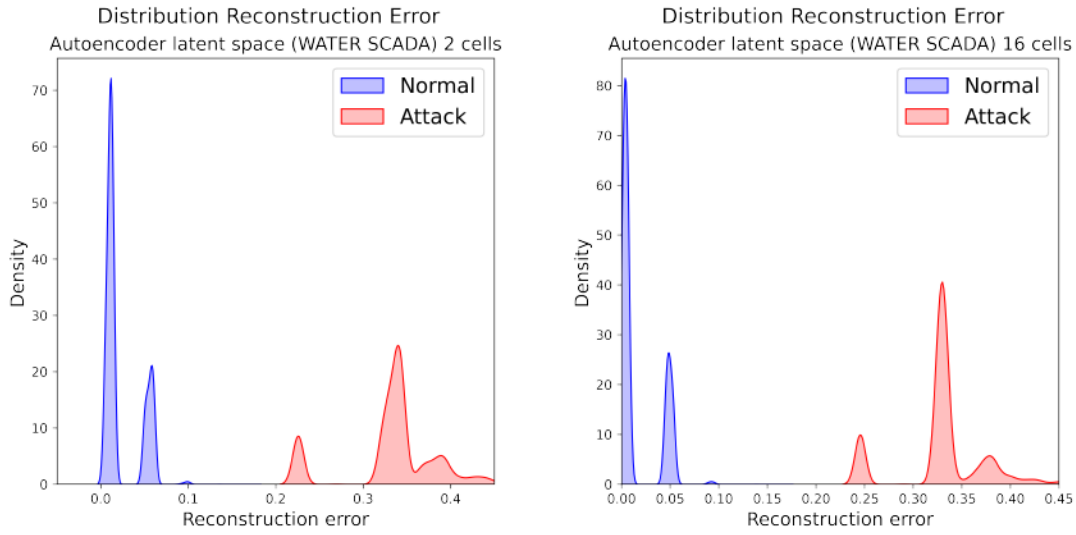


Figure 4.17: Vanilla Autoencoder: Using Real Dataset of Water Storage System: Error distribution of error reconstruction of normal and Attack events with 2 and 16 cells in the latent space

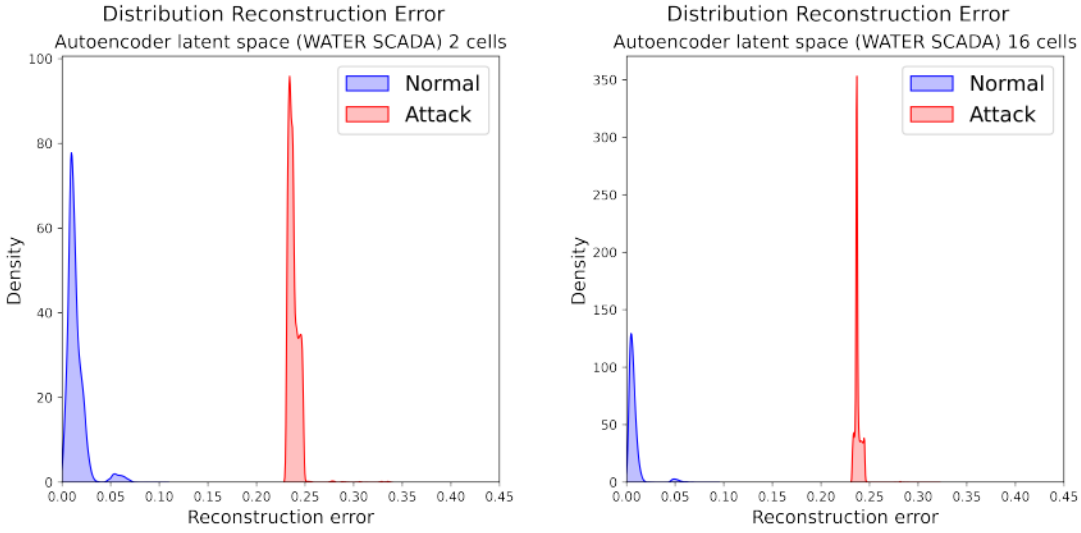


Figure 4.18: Vanilla Autoencoder: Using Synthetic Dataset of Water Storage System: Error distribution of error reconstruction of normal and Attack events with 2 and 16 cells in the latent space

trick, which is implemented using the average of the latent space and a sampling function that uses the gaussian distribution as input. Therefore, we decided to perform experiments with simple and complex autoencoders to see the difference in performance. We have performed the same set of experiments using variational

autoencoders that we have computed for vanilla autoencoders. To obtain a homogeneous comparison, we have selected 2 and 16 cells as latent space for the different VAE constructions. The network has been implemented using three layers, namely, encoder network, decoder network, and three hidden layers (128, 64, and 32 cells each) using *relu* activation. While training the model, we used 270 epochs and batch size = 32 for power grid datasets. Similarly, 80 epochs and 32 batch size are used for the gas pipeline dataset, and 70 epochs and 64 batch size are used for water storage datasets.

During the experiments, we formulated the implementation focusing on three main approaches to detect unknown attacks, namely, training (normal vectors) and testing (normal and attack vectors) the model using Original datasets, training & testing using the synthetic datasets, training using synthetic normal events, and validation using original attack vectors. We have trained the model using 80% of normal traffic. The validation is carried out using 20% of remaining normal traffic and 20% of attack vectors. The purpose of having this distribution is to maintain a balanced class structure.

As depicted in Table 4.7, we have computed the results in terms of precision(PR), recall(RC), and F1 score regarding normal and attack events for all three applications and we got promising results. Moreover, we have computed the number of anomalies detected during each cycle of the experiment. The results seem to be promising while considering latent space is equal to 16. We obtained an accuracy of around 98% for all three applications with original datasets. the accuracy has improved by 1% while using synthetic data for the training process and that is around 99%.

We have also tried other combinations to validate the efficiency of the model, such as validating the model using 20% normal events and all the attack records of the dataset (that reflects unbalanced datasets) and then verified the results by computing True Negative Rate (TNR) and False Negative Rate (FNR). For example, when we tried 4543 normal instances and 51543 attack vectors (a total of 56086 instances), we got 4400 True Positives (TP), 143 False Positives (FP), 51159 True Negatives (TN), 384 False Negatives (FN). In this case, we got TNR equal to 99.25%, and FNR equal to 8.45%.

In a nutshell, we observed promising results for all the experiments that we have performed to detect anomalies when a model has only knowledge of normal traffic.

Table 4.7: Experimentation results summary computed using Variational Autoencoders

Applications	Types of Dataset	Train 80% - N	Test 20% - N 20% - A	PR, RC, F1 (Normal)	PR, RC, F1 (Attack)	Anomaly Detected
Power Grids	Original	(18171, 128)	(9086, 128)	1.00, 0.97, 0.98	0.97, 1.00, 0.98	4641 / 4543
	Synthetic	(588638, 128)	(294320, 128)	1.00, 0.98, 0.99	0.98, 1.00, 0.99	144276 / 147160
	Syn(N), Real(A)	(80000, 128)	(40000, 128)	0.92, 0.98, 0.95	0.98, 0.91, 0.94	21514 / 20000
Gas Pipieline	Original	(48924, 26)	(24464, 26)	1.00, 0.97, 0.98	0.97, 1.00, 0.98	12650 / 12232
	Synthetic	(1600000, 26)	(800000, 26)	1.00, 0.98, 0.99	0.98, 1.00, 0.99	407905 / 400000
	Syn(N), Real(A)	(80000, 26)	(40000, 26)	1.00, 0.98, 0.99	0.98, 1.00, 0.99	20404 / 20000
Water Storage	Original	(137932, 23)	(68966, 23)	1.00, 0.95, 0.97	0.95, 1.00, 0.98	36261 / 34483
	Synthetic	(80000, 23)	(40000, 23)	1.00, 0.98, 0.99	0.98, 1.00, 0.99	20418 / 20000
	Syn(N), Real(A)	(80000, 23)	(40000, 23)	1.00, 0.98, 0.99	0.98, 1.00, 0.99	20409 / 20000

Variational autoencoder has proven an efficient technique for the detection of zero-day attacks for all the datasets (real and synthetic) of three SCADA applications.

4.5 Proposed IDS framework for Power Grid SCADA System

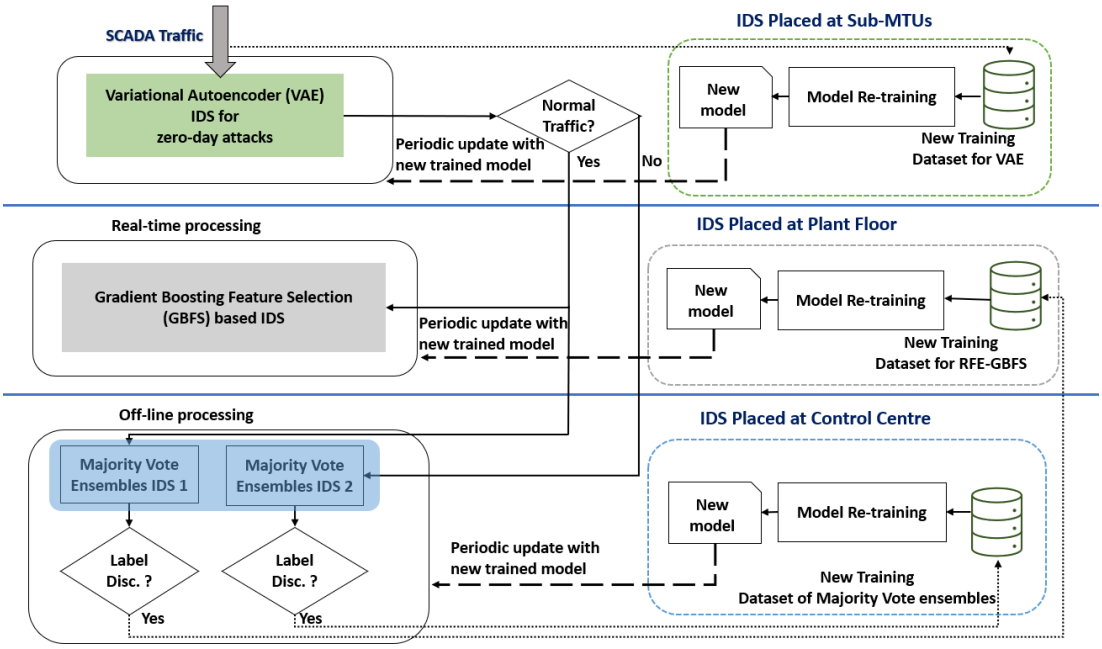


Figure 4.19: IDS framework for real-time SCADA systems for power grids

In distributive environments such as power grids, the availability of the most accurate intrusion detection system is a crucial factor. To achieve robust security along with availability, we have used defense-in-depth architecture by placing IDS at three different locations of the power grids, namely, IDS placed at Sub-MTUs (intermediate SCADA), IDS placed at the control center, and IDS placed at the plant floor. The proposed IDS framework for real-time industrial control systems for power grids is shown in Figure 4.19.

First, the SCADA power grid traffic is analyzed for zero-day attacks at intermediate SCADA centers using Variational Autoendoers (semi-supervised learning). In the case of attack vectors, the packet will be sent to the control center IDS for further evaluation. If the control center IDS also recognizes it as a malicious event, then the packet will be dropped and categorized as a type of known attack. However, in case of a discrepancy in prediction at the control center, the packet will be added to

the training dataset to retrain the supervised learning models. In this manner, the accuracy of supervised models will be increased by learning new features of zero-day events.

In the case of normal events at intermediated SCADA center, the traffic will be transmitted to both the supervised models for further assessments to detect any known attacks. The plant floor IDS uses a GBFS-based filtering model to detect intrusions. As IDS on the plant floor is high-speed and lightweight that is more compatible with detecting intrusions in real-time communication. However, for more accurate results, the output of this module is verified at the control center using the majority vote-based IDS with multiple classifiers. In case of a discrepancy in the output labels, the records will be added to a new training dataset to retrain the GBFS filtering model periodically. In this manner, we can achieve the most updated test model and replace the existing model with the recent model.

Through this approach the proposed framework achieves high computational speed and accurate prediction for live SCADA traffic that not only detects known attacks but is also used to filter zero-day attacks.

Table 4.8: Comparative analysis of various methods

Classifier	**Feature Selection**	**ML Model**	**Attack**	**Acc**
ADA-JRIP [35]	NA	Supervised	Known	94.55%
EMCT [38]	PCC	Supervised	Known	90.2%
GMMKM [39]	PCC	Supervised	Known	94.56%
Tree Based [6]	**GBFS**	**Supervised**	**Known**	**97.26%**
MV-EM [61]	**RFE-XG**	**Supervised**	**Known**	**98.24%**
PCA	**NA**	**Semi-supervised**	**Unknown**	**96.76%**
Vanila AE	**NA**	**Semi-supervised**	**Unknown**	**99.25%**
VAE	**NA**	**Semi-supervised**	**Unknown**	**98.92%**

To validate the effectiveness of the proposed scheme, we have compared the accuracy of our semi-supervised models with six published methods, namely, JRIP using Adaboost technique (AdaJRIP) [35], Expectation Maximization Clustering Technique (EMCT) [38], Gaussian Mixture – Kalman Filter Model (GMM-KF) using Pearson Correlation Coefficient (PCC) feature selection method [39] and GBFS based tree based classifiers [6]. All the methods utilize the same power-grids dataset for result computation.

Chapter 5

Robust & Secure Solution for SCADA communication - Methodology, Experiments & Result discussion

The research work reported in this chapter has resulted in two publications and one is under review:

- **D. Upadhyay**, M. Zaman, R. Joshi and S. Sampalli, "An Efficient Key Management and Multi-Layered Security Framework for SCADA Systems," in **IEEE Transactions on Network and Service Management,** " vol. 19, no. 1, pp. 642-660, March 2022, doi: 10.1109/TNSM.2021.3104531. *(Impact Factor: 4.19)*
- **D. Upadhyay**, N. Gaikwad, M. Zaman and S. Sampalli, "Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash Based Applications," in **IEEE Access**, vol. 10, pp. 112472-112486, 2022. *(Impact Factor: 4.43)*
- **D. Upadhyay**, H.Ohno, M. Zaman, B. Stacey and S. Sampalli, "Design and development of SCADA (Supervisory Control and Data Acquisition) test bench for validation of lightweight cipher,**" Computers & Security, Elsevier,** manuscript under preparation. *(Impact Factor: 5.105)*

5.1 Summary of the chapter

The protection of critical industrial infrastructure against cyber-attacks is crucial for ensuring public safety, security, and reliability. SCADA systems are used to control and monitor such industrial control systems. A robust solution to strengthen the security of these systems against cyber-attacks is a crucial requirement in the design of SCADA systems. Through this work, we aim to cover the protection of the industrial control system landscape by offering a low-cost and robust framework for SCADA networks, which protects them against various cyber-attacks. In this chapter, we have

proposed a session key agreement in addition to lightweight multi-layered encryption techniques. The framework combines both symmetric and asymmetric cryptography to achieve high computational speed by covering all the security mechanisms. This security model is proposed to enhance the security of various industrial sectors such as water and sewage plants, power stations, chemical plants, oil industries, product manufacturing units, and transportation systems. The successful deployment of this model will allow operators and technicians to monitor and control the plant devices remotely as it will protect the entire system from potential breaches.

5.2 Multi-layered Framework for Secure SCADA Communication

This section presents the proposed multi-layered framework for secure SCADA systems. The framework uses three levels for robustness, namely, symmetric key cryptography, cryptographically secure HMAC function, and a public key algorithm. The security features of each phase are illustrated in Figure 5.1.

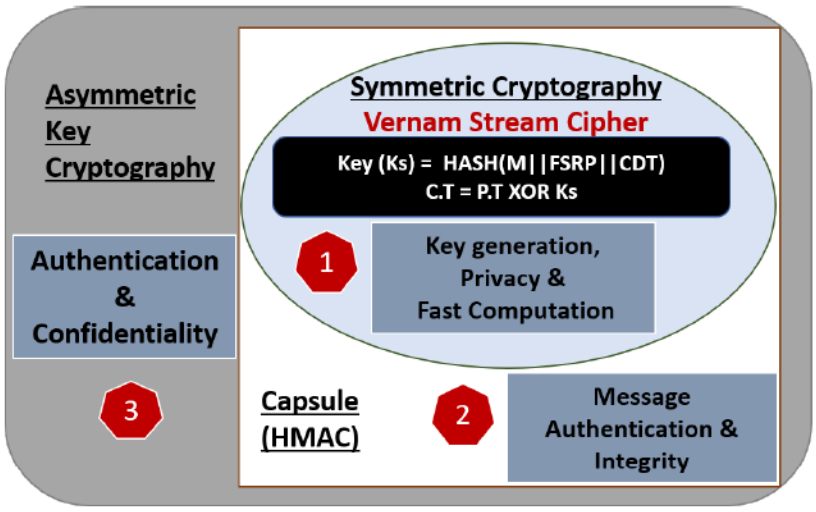


Figure 5.1: Multi-layered framework for secure SCADA communication

In our framework, a unique session key is generated for each connection between SCADA communication devices. The elements of this session key are securely shared using asymmetric key cryptography. This is called the key agreement stage. Furthermore, during this phase, the sender's authentication and recipient confidentiality are also validated using the private-public key pair. Moreover, HMAC is used for message

authentication and integrity. Once both the communication parties agree on the reliable key exchange, further communications take place using symmetric cryptography. The encryption of the original message is hashed, and subsequently, the symmetric keys are generated to encrypt the message using the lightweight Vernam cipher. After that, the cipher text and hash digest of this encrypted message are sent together over the communication channel. At the other end, the receiver device validates the message integrity using HMAC and then the cipher text is decrypted to receive sender's original message.

Since ICSs control field-site components at the plant floor, the activities related to controlling and monitoring of the elements should be done securely and efficiently [35]. For that, we have introduced two modules, namely, secure key exchange and secure information exchange. Moreover, secure information exchange consists of four methods, namely, Multi-Layered (ML) architecture, Random Prime Generator (RPG), Prime Counter (PC), and Hash Chaining (HC). While ML and HC offer very high security in SCADA networks, PC and HC are proposed for time-critical applications. The RPG offers medium level security and availability.

5.3 Secure Key Exchange

The key agreement refers to three stages, namely, key generation at the sender side, key distribution over the communication channel, and key extraction at the receiver side.

5.3.1 Key Generation

During the key generation phase, a sender (MTU or RTU) uses three main elements, namely, a Random Number (RN), Current Date & Time (CDT), and Fraction of Square Root of Prime number ($FSRP$). Here, CDT and $FSRP$ are used as secret elements to generate the session key. The choice of these two key elements is based on the property of generating true random numbers. CDT generates a random number every microsecond and to make it more random, we choose $FSRP$, which returns a non-terminating, non-repeating decimal number [69]. The session key (S_K) is derived by applying a hash function on both these elements by combining them, as in eq 5.1.

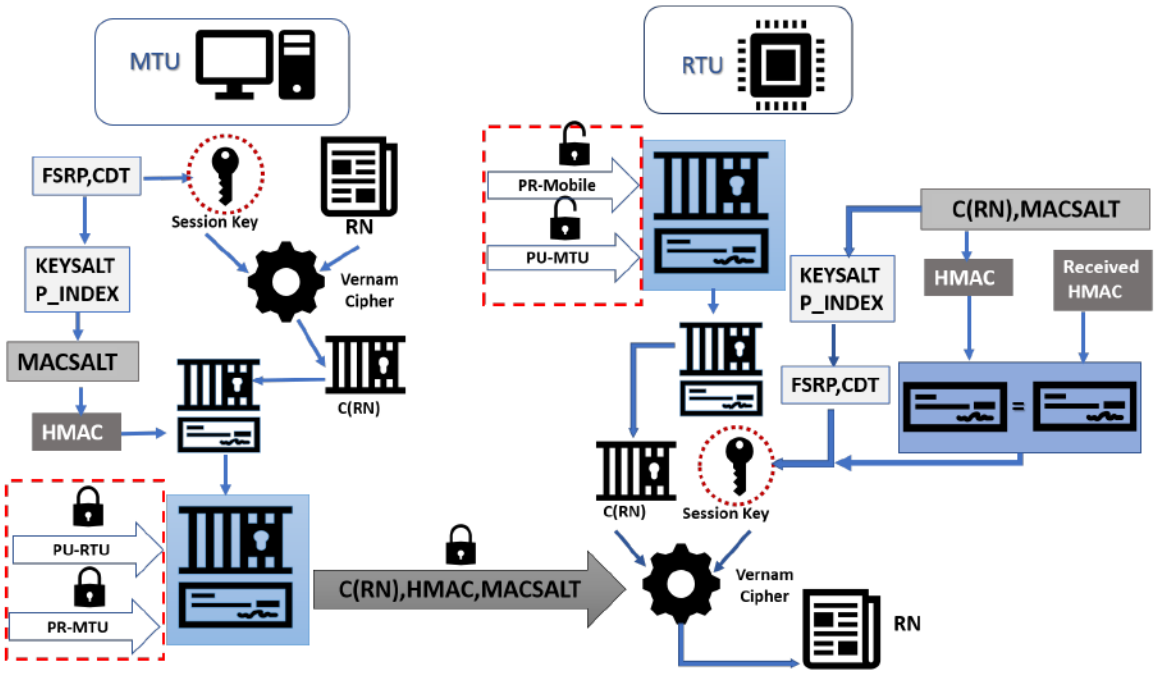


Figure 5.2: Secure Key exchange mechanism for SCADA systems

$$S_K = \text{HASH}(CDT \,||\, FSRP) \tag{5.1}$$

These session key elements are securely distributed using $MACSALT$. The index of $FSRP$ is combined with $KEYSALT$ to generate $MACSALT$, where $KEYSALT$ is derived by $XORing$ CDT and $FSRP$. The formulas are given in eq 5.2 & 5.3.

$$KEYSALT = CDT \oplus FSRP \tag{5.2}$$

$$MACSALT = KEYSALT \,||\, PRIME_{index} \tag{5.3}$$

Once S_K and $MACSALT$ are generated, RN is encrypted using S_K which generates cipher of random number $C(RN)$, as in eq 5.4.

$$C(RN) = RN \oplus S_K \tag{5.4}$$

In this process, the algorithm produces a hash not only from the encrypted RN but also from the CDT & $FSRP$ key elements. This derivation follows the procedure

of $HMAC$, as given in the eq 5.5 and is used to check message integrity.

$$HMAC_{sender} = \text{HASH}(C(RN), CDT \,||\, FSRP) \tag{5.5}$$

5.3.2 Key Distribution

The bundle of the $C(RN)$, $HMAC$ of $C(RN)$, and $MACSALT$ is securely sent over the communication channel using the private key of sender's (K_{spri}) and public key of receiver (K_{rpub}) that validate the sender's authentication and receiver's confidentiality as in eq 5.6.

$$K_{rpub}(K_{spri}(C(RN), HMAC_{sender}, MACSALT)) \tag{5.6}$$

5.3.3 Key Extraction

At the receiver side, the private key of the receiver and the public key of the sender is applied to validate authentication and confidentiality as in eq 5.7.

$$Krpr(Kspu(C(RN), HMAC_{sender}, MACSALT))) \tag{5.7}$$

The elements of $MACSALT$ are used to generate $FSRP$ and CDT. $PRIME_{index}$ is used to extract the value of $FSRP$ and CDT is obtained by XORing $FSRP$ and $KEYSALT$ as shown below in eq 5.8-5.10.

$$MACSALT = KEYSALT||PRIME_{index} \tag{5.8}$$

$$FSRP = FRAC(SQRT(PRIME_{index})) \tag{5.9}$$

$$KEYSALT = CDT \oplus FSRP \tag{5.10}$$

Finally, the session key is derived by applying hash on CDT and $FSRP$ as in eq 5.11.

$$CDT = FSRP \oplus KEYSALT \tag{5.11}$$

$$HMAC_{receiver} = HASH(C(RN), CDT||FSRP) \tag{5.12}$$

$HMAC$ is computed at the receiver using $C(RN)$, CDT & $FSRP$, as in eq.5.12 to compare with $HMAC_{sender}$ to check data integrity. The $HMAC$ of the sender and receiver are checked, if both are equal it moves to the next step, or else the message is discarded. The session key S_K is then validated using CDT and $FSRP$ as in eq.5.13. The session key is XORed with $C(RN)$ to get the RN as shown in eq.5.14.

$$S_K = \text{HASH}(CDT \,||\, FSRP) \tag{5.13}$$

$$RN = C(RN) \oplus S_K \tag{5.14}$$

The receiver will send an acknowledgment to the sender by encrypting $RN + 1$ using the same session key to validate secure key exchange. Figure 5.2 illustrates the secure key exchange mechanism between SCADA devices, namely, MTU and RTU.

The proposed scheme uses RN, CDT, and FSRP to generate the session key (Ks) for both communication devices, namely, MTU and RTU. However, during the key exchange, these elements are not transferred openly, rather RN is encrypted by the key generated using the combination of CDT & FSRP. Moreover, the modulo-2 operator (XOR) is applied on CDT and FSRP to generate the keysalt which will be shared over the communication channel along with an index of FSRP and the cipher text of RN. The index of FSRP is considered as the root of trust for the entire scheme. Furthermore, the Vernam cipher is used for symmetric key cryptography, which requires a fresh key for each message during the encryption and decryption process. This symmetric key is generated using the session key (Ks) and key parameters, namely, CDT and FSRP, depending on the proposed approaches. The FSRP can be generated using a random prime generator (Method 2) or a prime counter (Method 3). Furthermore, hash chaining (Method 4) can be combined with any of these approaches to generate a new fresh symmetric key for the Vernam cipher.

5.4 Secure Information Exchange

In SCADA systems, the field site components are controlled and monitored using short messages communicated between RTU and MTU. Based on the reading obtained from the field control devices, namely, RTU, PLC, and IED, the SCADA master (MTU) makes a proper decision and sends an appropriate signal to the field components to operate plant machinery. Generally, the control messages are short in length (typically 256 bits), which control the sensors and actuators of plant machinery. For example, in water management systems, the signals used during communication include OPEN/CLOSE the valve, SWITCH_ON/SWITCH_OFF the devices, the water level tank, etc. [55]. Such systems operate using short messages. Hence the average length of the control message consists of 24 to 32 characters (192 to 256 bits) for one frame.

The Vernam cipher requires the same length for key and message. Moreover, each communication message requires a distinct key for encryption and decryption. To generate such a unique key every time, we have proposed two main approaches, namely, multi-layered architecture, and hash chaining with FSRP. Moreover, both these approaches are further divided in the multiple methods to generate a unique value of FSRP, namely, random prime generator (RPG), and prime counter (PC). Figure 5.3 illustrates the symmetric key generation process used to encrypt and decrypt the message at both the communication endpoints. Both the sender and receiver negotiate RN (random number), CDT (current date and time), and the index number of FSRP (which acts as a seed for random prime generator/ prime counter) to generate session key (Ks). Using RPG/PC, both the sender and receiver generate a distinct FSRP for each message. Moreover, Blake2s (cryptographically secure hash function [70]) is applied on the session key and FSRP to generate the encryption key (Ke). Similarly, the receiver produces the decryption key (Kd) using the pre-shared Ks and the value of FSRP. Note that, the value of the symmetric key is not only depends on the previous key but also on the value of FSRP (which is generated using RPG/PC). In the case of our multi-layered architecture, instead of two parameters, both, MTU and RTU use three parameters, namely, CDT, FSRP, and Ks to generate the symmetric key. These parameters are exchanged securely using MACSalt and NTRUEncrypt public-key cryptography.

For our evaluation, the length of the key is 256 bits as Blake2s depends on a 32-byte word size. In the case of 256 bits < input string < 512 bits, we can replace Blake2s with Blake2b to generate the symmetric key, which consists of a 64-byte word size.

The complete process diagram of the proposed framework of secure SCADA systems is shown in Figure 5.3.

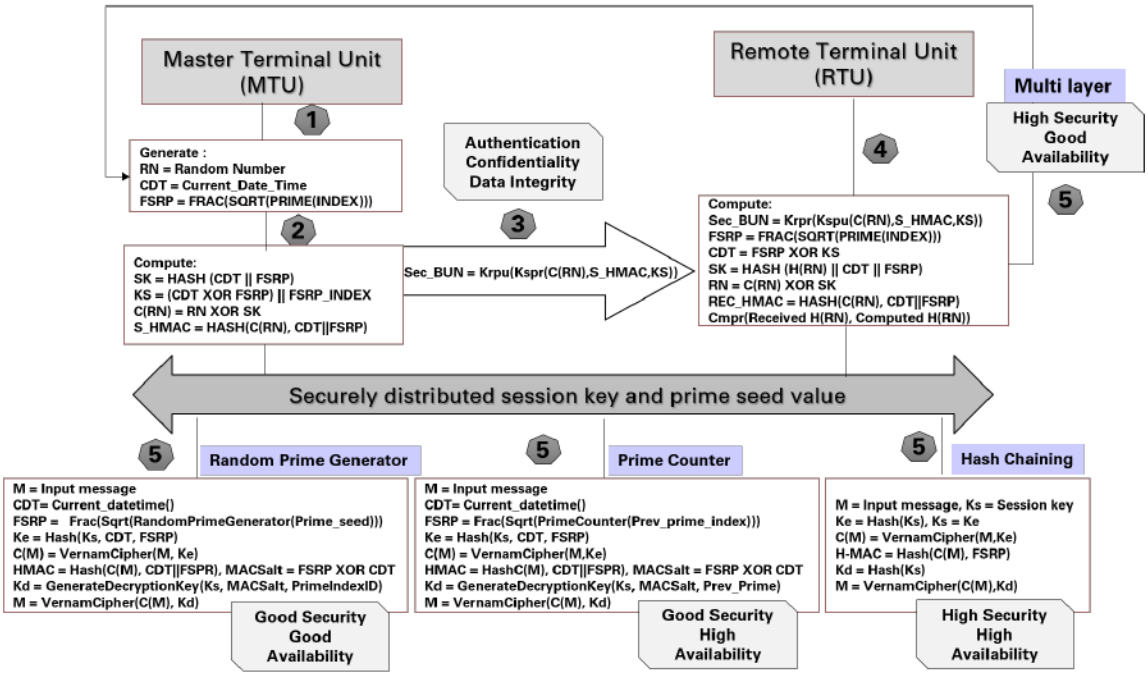


Figure 5.3: Complete process diagram of secure communication between MTU and RTU

The following section describes four methods to implement secure SCADA framework for information exchange.

5.4.1 Hybrid Multi-layered Architecture

We can use the same nomenclature of the session key agreement for further communication in which after successful distribution of the session key the message is communicated between two parties using both symmetric and asymmetric key cryptography. The data encryption and decryption are obtained using Vernam cipher. The key generator of the Vernam cipher follows the same procedure of session key derivation to generate the symmetric key at the sender and receiver sides. The symmetric key, HMAC and MACSALT are derived using FSRP and CDT. Further encrypted message C(M),

HMAC and MACSALT are shared securely using asymmetric key cryptography. Here, the complexity of the method is obtained by N * (Asymmetric Key + Symmetric Key) during each session which provides high security with moderate availability. N is the number of messages exchanged during the session.

The following methods describe the approach of symmetric key cryptography instead of using a combination of public-private key pairs. After secure session key and prime seed distribution, further encryption process can be carried out using one of the three symmetric key-based proposed methods as listed below.

In this method, the seed of the prime index value is used to determine FSRP using the next random prime number. Also, CDT and a hash of the input message h(M) are determined to generate a symmetric key, HMAC and MACSALT. This information is sent to the recipient over the communication channel. Using MACSALT and a random number prime generator, the receiver can generate the symmetric key to decrypt the data using the Vernam cipher. Here the complexity of the algorithm is measured by Asymmetric key + N * Symmetric key for every session where asymmetric and symmetric keys are used during session key distribution while the symmetric key is used during secure communication. However, this approach is comparatively less secure as the adversary could intercept the MACSALT to derive the keys such as FSRP and CDT. The Algorithm steps are shown in the paper # 3 attached after the Appendix.

5.4.2 Prime Counter

In this method, instead of random prime generator, we have used prime counter which significantly increases the execution speed. The rest of the steps are the same. The previous prime number is used to determine the next FSRP. Similarly, CDT and a hash of the input message h(M) are used to determine the symmetric key, HMAC, and MACSALT. This information is sent to the recipient. Using MACSALT and prime counter, the receiver can generate a symmetric key to decrypt the data using the Vernam cipher. In this approach, the adversary could also intercept the MACSALT to derive the essence of the keys such as FSRP and CDT. The complexity of an algorithm is measured by the Asymmetric key + N * Symmetric key for every session. Consequently, the model provides good security with high availability.

5.4.3 Hash Chaining

This proposed method is one of the robust solutions for SCADA systems which covers all the security mechanisms. This approach not only provides high security but also offers high availability. In this, the pre-shared session key is used as input of the hash function to generate the next symmetric key. Moreover, the previous FSRP is used to generate HMAC which can be derived independently at both the ends and is used to check the integrity of the message. The generated symmetric key is then used to encrypt and decrypt the message using the Vernam cipher. The complexity of this method is based on the Asymmetric + N * Symmetric key cryptography. In this proposed method, asymmetric key cryptography is used only once to distribute the session key, however, further communication takes place securely using symmetric key cryptography until the session has ended.

5.5 Experiments & Result Discussion

5.5.1 Algorithm selection of cipher suite for proposed framework

The choice of the algorithms to design the security framework generally depends on the nature of the application. The communication of SCADA systems relies on a real-time request-response mechanism. Moreover, SCADA field devices are equipped with micro-controllers for processing information and have limited computational power and resources. Consequently, identifying the most appropriate algorithms for the proposed scheme is one of our implementation's crucial steps. The identified algorithms for our cipher suite should provide faster execution speed and be suitable for deploying in an embedded system environment. The comparative analysis of various algorithms was carried out using wolfSSL and libntru 0.5 cryptosystems on Linux subsystem of Windows 10 with Intel Core i5-8300H 2.30GHz processor and 8 GB RAM. The wolfSSL is a lightweight and portable embedded SSL library that is specially meant for IoT, embedded, and RTOS environments [71]. The libntru 0.5 is an open-source library that supports the implementation of the public-key encryption scheme NTRUEncrypt in C language by following the IEEE P1363.1 standard [72]. Moreover, the proposed symmetric schemes are implemented on an integrated development environment for Python called IDLE on Windows 10 operating

system.

HASH Functions

In this framework, the hash function plays a vital role as it acts as a message authentication code and is used to generate a symmetric key. To identify the secure and computationally efficient function, we have compared various hash functions. Based on the comparative analysis of computational speed presented in Table 5.1, Blake seems to be most prominent.

Table 5.1: Comparative analysis of various Hash Functions

Algorithm	Ex.Speed (MB/sec)
MD5	340.729
RIPEMD	129.74
SHA	31.571
AES-256-CMAC	110.504
SHA2-256	152.863
SHA3-256	106.781
Blake2b	**172.2**
Blake2s	**169.78**

Symmetric Key Cryptography

Advanced Encryption Standard (AES) is the well-known symmetric key cryptography used to design secure systems. AGA has used AES as a symmetric key component in its standard protocol suite [73]. Nowadays, AES modes are preferred to secure the systems owing to better security and faster execution speed. 3DES is also used in traditional cryptosystems. In Table 5.2, we have compared the computational speed of various modes of AES and DES with the proposed hash-based Vernam Cipher. The computational speed of Vernam Cipher is calculated by adding the execution speed of Blake2S hash with the speed of Exclusive-OR operation. The comparative analysis shows that the hash-based symmetric key technique used in Vernam Cipher is faster than other algorithms.

Table 5.2: Comparative analysis of computational speed of various symmetric key algorithms

Algorithm	Ex.Speed (MB/sec)
AES-256-CBC-enc, AES-256-CBC-dec	94.565 , 88.169
AES-256-GCM-enc, AES-256-GCM-dec	25.596, 24.318
AES-256-ECB-enc, AES-256-ECB-dec	55.69, 63.067
AES-256-CFB	86.329
AES-256-OFB	71.146
AES-256-CTR	64.576
3DES	14.542
Vernam Cipher with Blake2s	**157.45**

Asymmetric Key Cryptography

Asymmetric key cryptography not only offers confidentiality but also ensures integrity, authentication, and non-repudiation during communication. Some public key algorithms such as Diffie-Hellman key exchange provide key distributions and secrecy, whereas some provide encryption and digital signatures such as RSA, ECC, and NTRU [74]. We have compared various well-established public key algorithms, namely, RSA, DH, ECC, and NTRU by considering the key size and total operations performed per second. According to the output results presented in Table 5.3, NTRU outperforms the other methods.

Table 5.3: Comparative analysis of the computational speed of various public key cryptography

Algorithm	Key Size	Mode	Ops/sec
RSA	1024	Key Generation	12
RSA	2048	Key Generation	3
DH	2048	Key Generation	913
DH	2048	Key Agreement	500
ECC	256	Key Generation	523
ECDHE	256	Key Agreement	500
NTRU	**1026**	**Key Generation**	**2153**
NTRU	**1499**	**Key Generation**	**698**
NTRU	**1615**	**Key Generation**	**801**
NTRU	**2066**	**Key Generation**	**345**

5.5.2 Computational speed of proposed framework

This section represents the calculation of the overall computational speed of the proposed framework. We have considered the execution time of the major four elements, namely, session key, symmetric key, HMAC, and asymmetric key. First, we have calculated the time to generate and extract the session key. After that, we computed the execution time of symmetric and asymmetric key generation, distribution, encryption, and decryption. Finally, we have calculated the overall time by combining it with the execution time to generate and extract the HMAC.

Total Execution Time

In order to achieve consistent results, we have measured the execution time of each cryptographic component. The execution time of these elements is listed in Table 5.4.

Table 5.4: Considerable parameters of different cryptographic components

Notation	Description	Cost in ms
Tskg	Time for a session key generation	0.06485
Tpc	Time to generate Prime number Counter	0.00997
Trpg	Time to generate random prime generator	0.5063
Tudt	Time to generate universal date and time	0.00010
Tse	Time for a symmetric encryption	0.000199
Tsd	Time for a symmetric decryption	0.000996
Thash	Time to generate HMAC	0.000001
Tex	Time for a session Key extraction	0.0992
Takg	Time for a asymmetric key generation	1.51025
Tae	Time for a asymmetric encryption	0.073
Tad	Time for a asymmetric decryption	0.1065

Moreover, Table 5.5 presents the mathematical equations that calculate the total execution time of all the four methods, namely, ML, RPG, PC, and HC. In hybrid approach, both symmetric and asymmetric algorithms are used to secure the information. In contrast, in the other three approaches, once the session key has been shared between two communication devices, only the symmetric key algorithm is used for performance improvement. Furthermore, the execution time of these three symmetric key algorithms is varied due to how they generate the keys to secure the information.

Table 5.6 represents the total execution time of all four proposed methods by

Table 5.5: Total execution time calculation

Method	Delay
Multi-layered Approach	N * (Tskg + Tex + Takg + Tae + Tad + Tsym + Tse + Tsd + Thash)
Symmetric Key Approach	Tskg + Tex + Takg + Tae + Tad + N * (Tsym + Tse + Tsd + Thash)
Random Prime Generator	Tsym = Trpg + Tudt
Prime Counter	Tsym = Tpc + Tudt
Hash Chaining	Tsym = Thash

Table 5.6: Total Execution Time in Seconds

BIT STREAM (256 bits)	Hybrid	RPG	PC	HC
1	0.0026	0.0052	0.0019	0.0016
10	0.0072	0.0437	0.0056	0.0026
50	0.0373	0.2806	0.0285	0.0046
100	0.0592	0.4111	0.0415	0.0136
500	0.3453	2.3248	0.2557	0.0635
1000 (32KB)	0.6203	6.5663	0.4410	0.1309
5000 (160KB)	3.3980	32.9295	2.5007	0.6867
10000 (320KB)	6.1889	50.8831	4.3941	1.3271
50000 (16MB)	30.2294	257.9262	21.2546	2.2456
100000 (32MB)	61.1824	560.3114	43.2326	10.4327

considering the major four parameters, namely, key generation, key extraction, encryption, and decryption. According to the results, the execution time of HC is lower than the other three methods and has proven the most efficient among all. Moreover, PC and ML approaches are more prominent than RPG. Comparatively, RPG takes more time because of its intricate design to generate a random prime number based on a seed value.

5.5.3 Calculation of Key storage cost & randomness evaluation

Since key generation, distribution, and extraction are periodic and costly operations, the SCADA network should have fewer stored keys on each field control device. For this reason, we have identified the storage cost of our proposed key management scheme. You can find a summary of the storage cost in the attached paper #3 by

considering the three types of communication, namely, point-to-point, broadcast, and multicast amongst MTU, Sub-MTU, and RTU. The total cost of keys is calculated at each SCADA location, where m denotes the number of sub-MTU's keys, and r represents the maximum number of RTU's keys.

Many cryptography applications may need to meet more robust random number generator requirements when the randomness of the keys is one of the most critical factors for that system. We have used the Vernam stream cipher for our proposed framework, which requires a distinct and random key to secure the information. In particular, the key generator's output must be unpredictable. Hence, we have evaluated the proposed symmetric key generator using the National Institute of Standards and Technology (NIST) statistical toolkit. Figure 5.4 presents all 16 tests and corresponding P-values for the proposed symmetric key generator for the Vernam cipher. Our proposed key generator passes all the statistical tests and is proven to be random.

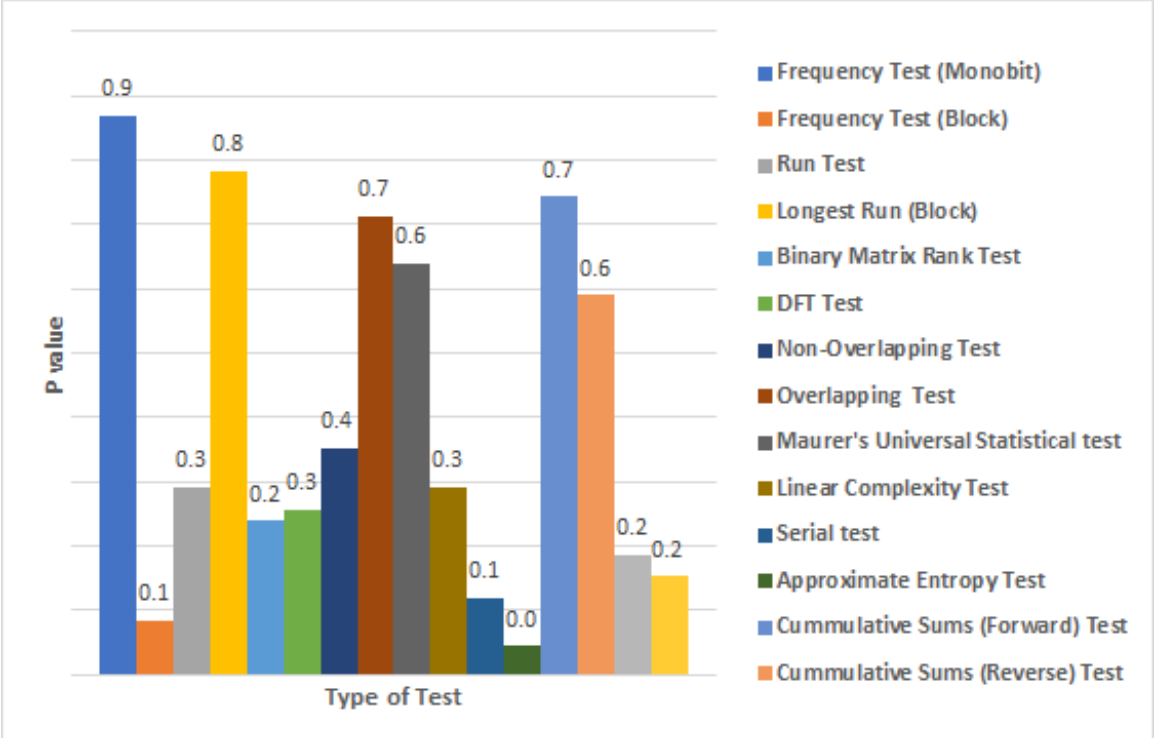


Figure 5.4: Randomness assessment of symmetric key

5.6 Performance Analysis

5.6.1 Security Analysis

In this section, the proposed framework is analyzed by considering various security mechanisms, namely, authentication, confidentiality, integrity, availability, and scalability. Moreover, the evaluation is extended by targeting various attacks and corresponding prevention mechanisms.

1. Message Integrity

    - Multi-layered hybrid architecture using symmetric and asymmetric key cryptography offers integrity.
    - Vernam stream cipher provides resistance to cryptography attacks [74].
    - Randomness of Key offers immunity to collision and preimage resistance attacks [75].
    - Dynamic Salt offers resistance to rainbow table attack and dictionary attack [75].
    - NTRU-based public key cryptography offers resistance to quantum attacks, brute force, and meet-in-the-middle attacks. It also prevents the system from data harvest attacks [76].
    - HMAC provides immunity against length extension attacks [77].

2. Authentication, Confidentiality

    - Public key of the sender and private key of the receiver of NTRU-based public key cryptography provides the sender's authentication and recipient's confidentiality.
    - HMAC offers message integrity and authentication.

3. High Availability – Faster execution

    - Once the session key distribution is established using a hybrid method, further communication will take place using symmetric key cryptography that increases the computation speed.

- Symmetric key generation using hash chaining and prime counter offers high execution speed.
- Use of Vernam stream cipher uses modulo operation for encryption and decryption which requires only 4 cycles in hardware implementation [78].
- NTRU is one of the fastest public key cryptography compared to well-known methods such as RSA and ECC [79].
- HMAC is derived using the same components used to generate the key. This reusability of elements reduces the computational time.

4. Scalability

- Same symmetric key cryptography (Vernam cipher) is used for both encryption and decryption.
- Authentication and confidentiality are established using public-private key pairs amongst communication parties.

Table 5.7: Comparative analysis of storage cost of keys

Key Management Schemes	MTU	Sub-MTU	RTU
SKE [4]	m(1+r)	1+r	1
SKMA [51]	m(1+r)	1+r	1
ASKMA [52]	2m+mr	r+logm	2+logr
ASKMA+ [14]	m	1+r+logm	1+logr
Symmetric [54]	r+1	-	2
Symmetric [54]	r+1	-	2
Hybrid [18]	m+2	2r+1	1+logr
CKMI [55]	2+r+m	2+r	1
Proposed Algo	**m+1**	**r+2**	**2**

5.6.2 Storage cost

The periodic session key agreement is a crucial step in SCADA communication that offers key refreshments. However, field control devices have limited power and memory requirements. Hence, an effective key agreement scheme with fewer stored keys can significantly improve the efficiency of SCADA networks. Many key management and

agreement schemes have been proposed to address the problem of key storage costs. We have compared the key storage cost of our proposed scheme with various published techniques, as presented in Table 5.7.

Table 5.8: Comparative Analysis of various cipher suites

Cipher Suite	Avg_Time (ms)
AGA_ECDHE, RSA, AES-128, GCM, SHA256	4.14
AGA_ECDHE, ECDSA, AES-128, GCM, SHA256	3.94
RSA, AES-128, CBC, SHA1	3.81
RSA, AES-128, CBC, SHA256	3.83
RSA, AES-256, CBC, SHA1	3.82
RSA, AES-256, CBC, SHA256	3.85
Multi-layered (NTRU, Vernam Cipher, Blake2s)	**2.61**
Random Prime Generator (NTRU, Vernam Cipher, Blake2s)	**5.25**
Prime Counter (NTRU, Vernam Cipher, Blake2s)	**1.91**
Hash Chaining (NTRU, Vernam Cipher, Blake2s)	**1.68**

5.6.3 Execution speed

Table 5.8 depicts the comparative analysis of the proposed scheme with various state-of-the-art techniques by implementing various cipher suites using the wolfSSL library. AGA has proposed two cipher suites for secure SCADA communication including the bundle of ECDHE, AES, RSA, and SHA256 and ECDHE, AES, ECC, and SHA256 for authentication, confidentiality, message integrity, and digital signature [73]. The cipher suite RSA, AES, CBC, and SHA is used in TLS communication, whereas we have used the NTRU, Vernam Cipher, and Blake2s for our proposed framework. The average execution time of our proposed cipher suite is comparatively better than other protocol standards.

5.7 Deployment of SCADA Security solution on a Hardware Test bench

This section covers the design and development of a small testbed of SCADA/IoT-based Industrial Control Systems along with the integration of the security module proposed in the previous section. The implementation consists of three major components,

namely, plant floor devices, control center elements, and communication protocols. We have developed three small demos, namely, control and monitor the speed of the servo motor using the HMI (Human Machine Interface) of VTSCADA from a remote location, water distribution system to control the water level and operate the valves and pumps, monitor the voltage level of the power systems. The simulation setup uses sensors, actuators, a Modbuspal simulator, and micro controls (Arduino, GBK Kit, Raspberry Pi) as plant components, while WSL and VTSCADA are used as control center devices. The communication is carried out using two communication protocols, namely, MQTT for IoT Devices, and Modbus for ICS components.

1. Plant Floor Components
    - Sensors & Actuators
    - GBK Kit, Arduino Micro-controller
    - Raspberry Pi, Onion-Omega2
    - Modbus Simulator
2. Control center Components
    - WSL Terminal, virtual Machine for Linux OS: MTU
    - VTSCADA – HMI, data historian, Alarm System
3. Communication Protocols
    - MQTT (IoT Devices)
    - Modbus (ICS Devices)

Figure 5.5 demonstrates the concept of testbed design for secure SCADA communication. We have considered the communication between PLCs and MTU where PLC is situated on the plant floor and MTU at the control center. At the control center, VTSCADA Light is used for various purposes such as controlling and monitoring the plant floor devices remotely via HMI, maintaining the log files at reporting server, and generating alerts for suspicious events. Here, we have configured the MTU Server in the Windows subsystem of Linux (WSL). The HMI is connected to MTU Server using an HDMI cable (wired connection). The proposed security module is deployed in the

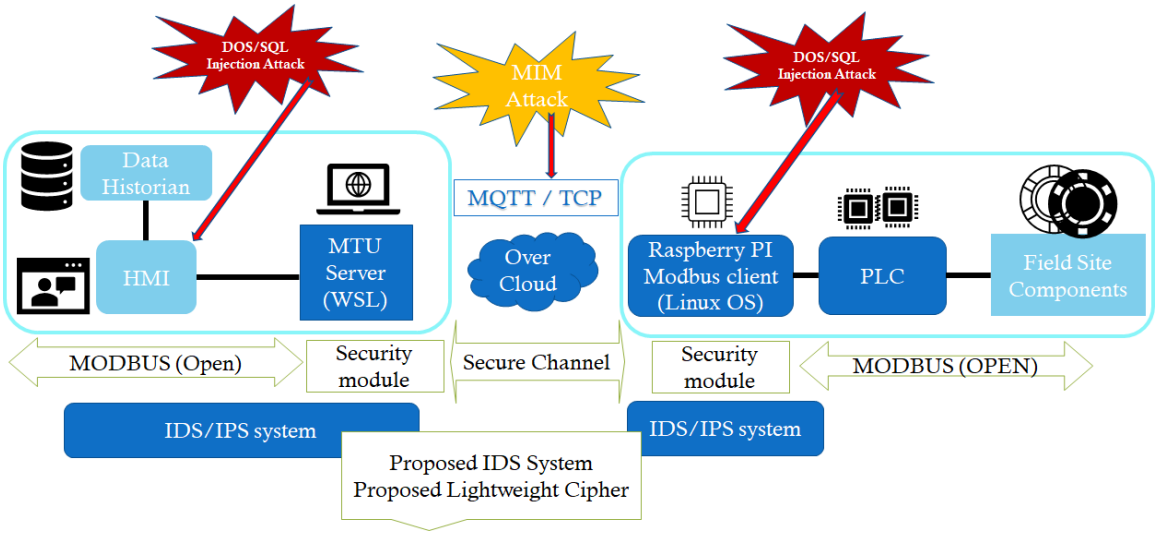


Figure 5.5: Test bench design for secure SCADA communications

MTU Server for secure communication with plant floor devices. However, owing to legacy inherited security loopholes of Modbus protocol, the internal intruder has the ability to intercept the traffic between MTU and HMI. To protect the system from such internal attacks, deployment of IDS is essential. We have proposed lightweight IDS systems for the control center and plant floor in one of our previous modules.

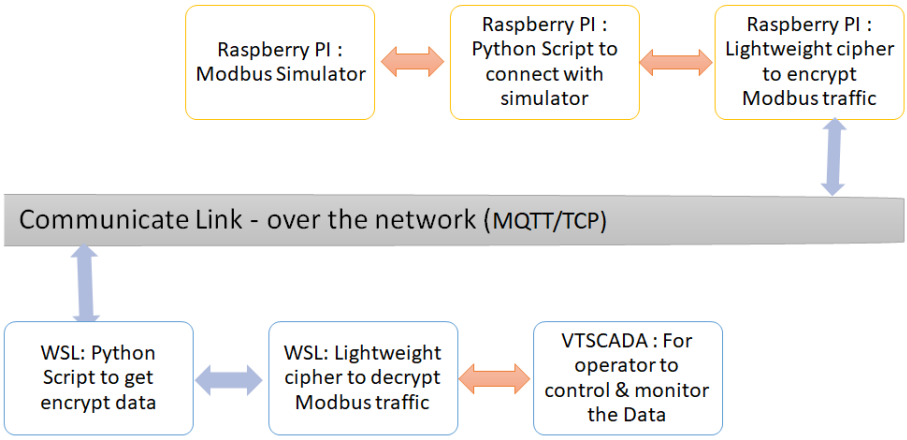


Figure 5.6: Flow Diagram of Proposed Framework for SCADA Test bench

At the plant floor, the same security module is deployed in Raspberry Pi which acts as a secure system-in-chip component for field site devices such as PLCs, IEDs, and RTUs. The MTU can communicate with field site components to control and monitor the plant floor devices via our proposed lightweight security module deployed

in the Raspberry Pi. SCADA field site components include legacy inherited security weaknesses and have very less computational and memory capabilities. Such devices cannot handle the security requirements, and hence additional components will need to be added between the field site components device and the network to handle the security protocol requirement. Raspberry Pi is an affordable and easily obtainable platform for prototyping, performing ladder logic testing, and integrating PLCs, RTUs, and IEDs in real time for various industrial control systems and cybersecurity concepts. Though raspberry Pi and field site components used a wired connection, there is a chance of interception owing to Modbus protocol and this could be prevented by deploying plant floor IDS proposed in the previous module.

Figure 5.6 represents the flow diagram of the proposed SCADA test bench. Secure communication is established using defense-in-depth architecture between field site components and the control center. This hardware test bench is currently under construction to experimentally evaluate the proposed framework.

5.7.1 Case Study 1: Controlling Servo Motor remotely using VTSCADA and MQTT Protocol

Components: Grove Beginner Kit (GBK), Servo Motor, Rotary Potentiometer, LED & Buzzer (Alarm System), VTSCADA (Control Center), MQTT Protocol

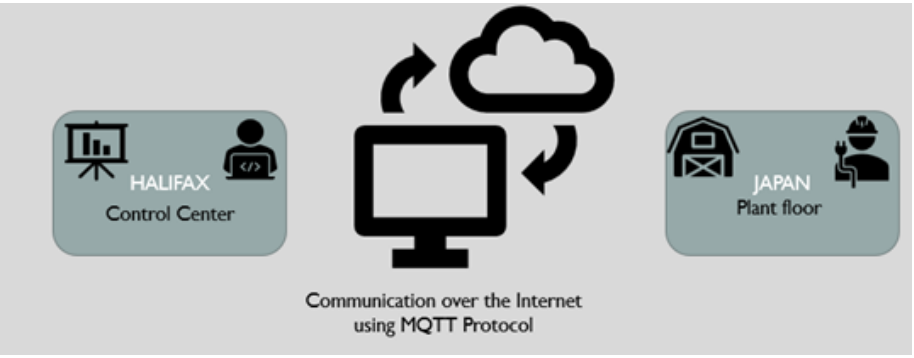


Figure 5.7: Layout of testbed

- Controlling the components from HMI to the Plant floor

  1. Sending data from VTSCADA to Rotate the servo motor.

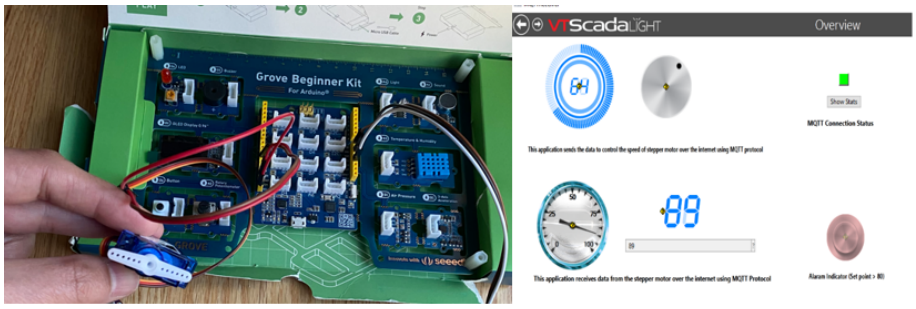


Figure 5.8: Experimental Setup

2. Setup the Alarm system according to the set points
   - If the input is greater than 80 then the LED will be ON & Buzzer starts buzzing
   - Alert the people working on the plant floor.
   - Prevent malicious activities – Data modification, MiTM attacks.

- Sending Data from Plant Floor to Control Center

1. Sending data from Rotary Potentiometer to VTSCADA
2. Setup the Alarm system according to the set points on HMI
   - If the input is greater than 80 then Buzzer starts buzzing and indicates on the alarm page
   - Indicates the high alert to technicians/operators at the control center.

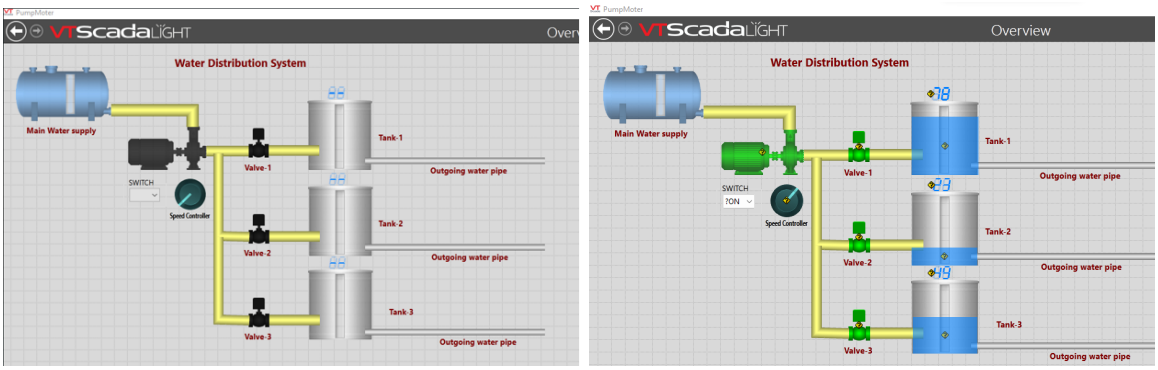
5.7.2 Case Study 2: Water Distribution System

Devices Used: Sensors, Actuators, Arduino, Raspberry Pi, WSL, VTSCADA
Communication Protocol: MQTT
Security Module: deployed in Raspberry Pi

We have designed a small testbed for water management & distribution systems. In that, the communication takes place using the MQTT protocol amongst all the devices. We have used water-level sensors which are connected to the Arduino micro-controller to measure the level of the water. Here, we have considered three tanks consisting of three water level sensors. Further, the Arduino controller is

Figure 5.9: Sensors and Microcontrollers

connected to the Raspberry Pi to establish secure communication. The status information of the level of the water has then transmitted securely to the WSL of the windows which acts as an MTU Server. At the SCADA server, the information is decrypted using the same security module and displayed on HMI (VTSCADA) as shown in Figure 5.10.



(a) Initial Stage (b) After Simulation

Figure 5.10: Human Machine Interface (VTSCADA)

5.7.3 Case Study 3: Voltage Level Indicators for Power system

Plant Floor Devices: ModbusPal Simulator (Automation tool – Voltage regulators, PLC (4 Registers, 2 Coils), Raspberry Pi

Control Center Devices: MTU Server (WSL), VTSCADA (HMI, Alarm System, Reporting logs)

Communication Protocol: Modbus - TCP/IP (Port – 502)

Security Module: Python Code - Hash Chaining Lightweight Cipher

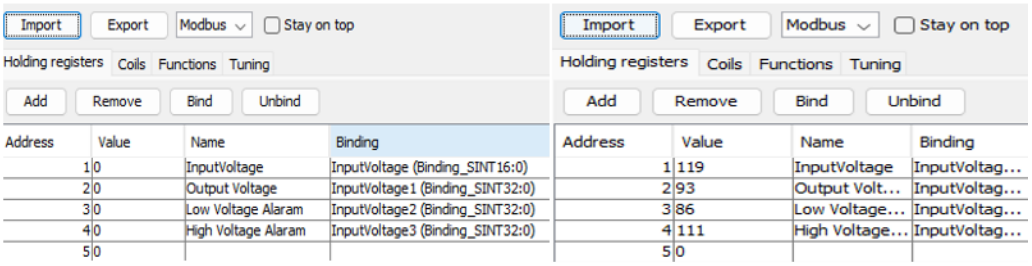


Figure 5.11: Modbuspal Simulator - Register values

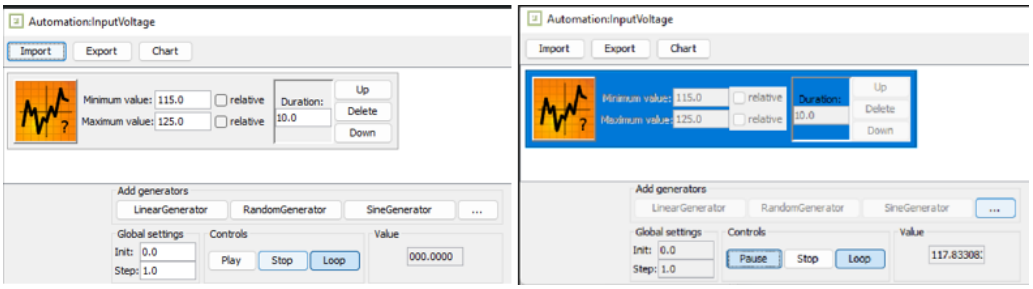


Figure 5.12: Automation tool for input Voltage generator

Using automation tool of the Modbuspal simulator, we have generated the random input voltage for one of the four registers of PLC. Another three registers contain the values of output voltage, low voltage alarm, and high voltage alarm respectively as shown in Figure 5.11. The values assigned to these registers are dynamic and generated by the simulator as depicted in Figure 5.12.

The Modbus protocol doesn't have security features and hence the data can be visible during transmission as shown in Figure 5.14. We have captured the Modbus traffic using the loopback address. Further, the data has been encrypted using the proposed security algorithm (hash chaining method) at Raspberry Pi before it leaves the system as shown in Figure 5.15. The encrypted status information of registers is sent to MTU. The received data is decrypted at MTU as shown in Figure 5.16 and further transmitted to the HMI of VTSCADA as demonstrated in Figure 5.13.

The simulator works on Modbus TCP/IP protocol using port number 502. We have also created 2 coils to enable output & for safety override. The status information

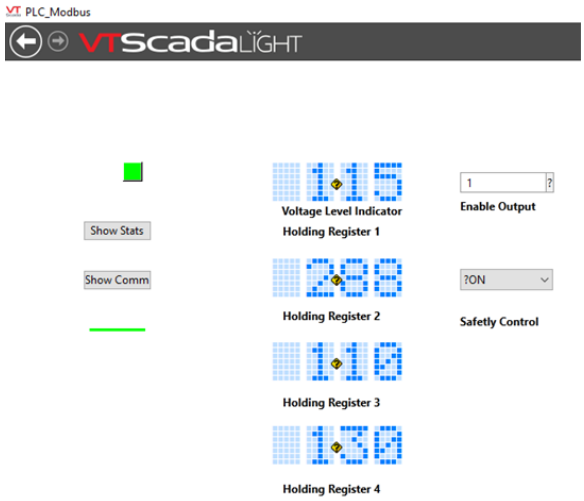


Figure 5.13: HMI view of Voltage Indicator on VTSCADA

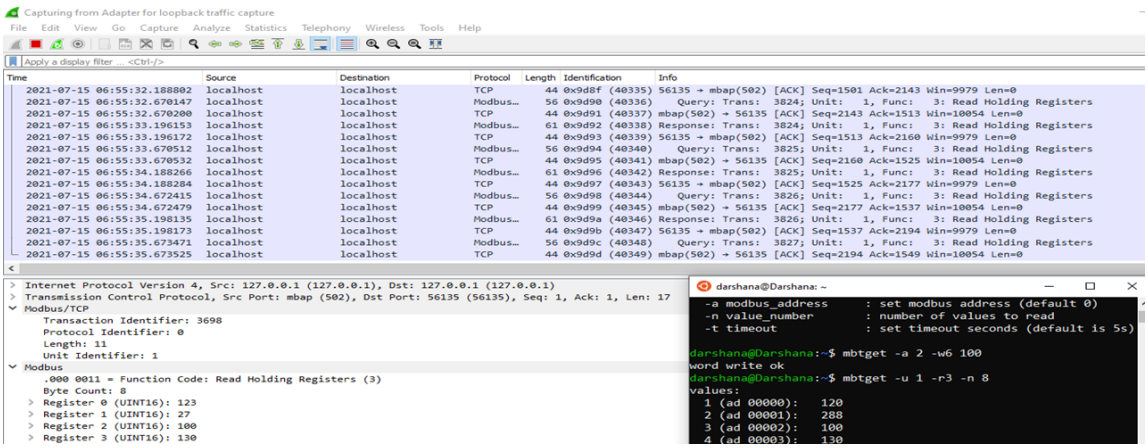


Figure 5.14: Modbus traffic captured using loopback address

is securely communicated between PLC and MTU by deploying security modules at both the communication ends, namely, raspberry pi at the plant floor, and WSl at the control center. The purpose of this simulation is to demonstrate secure communication between PLC and MTU over the internet.

5.7.4 Performance evaluation of proposed security methods on Raspberry Pi

This section represents the measurement of computational speed and memory utilization of Prime Counter and Hash Chaining (two of the most promising proposed security

Figure 5.15: Encyption applied on PLC data

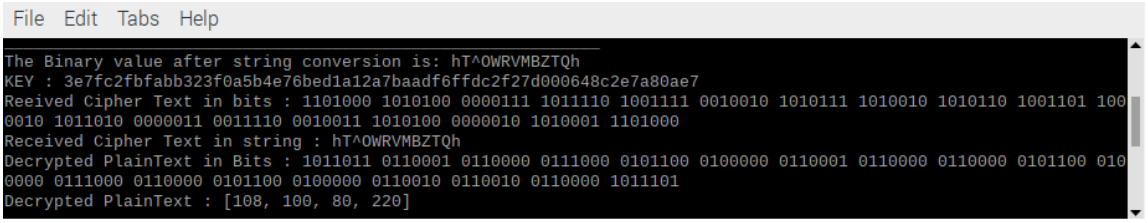


Figure 5.16: Decryption of received data at MTU

methods - described at beginning of this chapter) on hardware (Raspberry Pi, version 3 & 4). To calculate the execution time of each method we have considered the overall time of each module to generate and extract the symmetric key along with the encryption and decryption time taken by the Vernam stream cipher.

In Table 5.9, we present the time of two proposed symmetric key cryptography methods (in milliseconds) for various sizes of input streams. Based on the results, hash chaining seems to be the most efficient in terms of computational speed. Moreover, Raspberry Pi 4 has proven efficient in terms of processing power compared to Raspberry Pi 3. The average execution time to extract the MAC and to distribute the MAC along with the session key is shown in Figure 5.17.

Moreover, we have computed the average execution time of each phase of the proposed module by considering 10,000 binary strings. As depicted in Table 5.10, we have considered four different phases, namely, symmetric key generation, data

Table 5.9: Total execution time of proposed security algorithms on Raspberry Pi

Bit Strings	Prime Counter Time (MS)		Hash chaining Time (MS)	
	Raspberry PI - 3	Raspberry PI - 4	Raspberry PI - 3	Raspberry PI - 4
1	1.960	0.923	0.678	0.275
10	19.052	7.214	6.536	2.413
50	82.516	31.728	34.630	12.144
100	160.039	57.815	66.740	26.14
500	802.070	309.366	303.382	121.661
1000	1441.089	540.111	600.563	236.421
5000	8019.959	3119.473	3226.179	1216.03
10000	15775.525	6140.058	6319.000	2438.607

Table 5.10: Computational speed of various phases of proposed security algorithms on Raspberry Pi (in milli seconds)

Phases	Prime Counter Time (MS)		Hash Chaining Time (MS)	
	Raspberry Pi 3	Raspberry Pi 4	Raspberry Pi 3	Raspberry Pi 4
Key Generation	0.772	0.472	0.172	0.062
Data Encryption	0.211	0.111	0.1412	0.121
Data Decryption	0.823	0.132	0.2411	0.111
Key Distribution	1.831	0.83	1.676	0.82

Table 5.11: Comparison of memory utilization of proposed security algorithms on Raspberry Pi (in bytes)

Memory Parameters	Prime Counter		Hash chaining	
	Raspberry Pi 3	Raspberry Pi 4	Raspberry Pi 3	Raspberry Pi 4
RSS	39931904	30081024	39997440	30195712
VMS	53587968	38297600	53583872	38039552
Shared	10436608	8167424	10518528	8278016
Text	3760128	3760128	3760128	3760128
Data	29802496	22618112	29798400	22360064

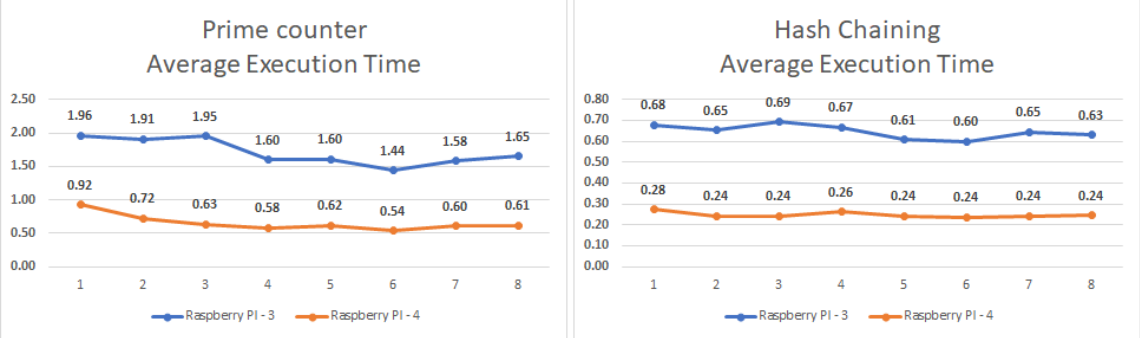


Figure 5.17: Average execution time of proposed security module on Raspberry Pi

encryption, data decryption, and session key distribution along with MAC to compare the execution time of two methods on two versions of Raspberry Pi.

For memory utilization, we have used psutil (Python system and process utilities) tool, a cross-platform library. This tool retrieves information on running processes and system utilization in Python such as CPU, memory, disks, networks, and sensors. It is mainly used for system profiling and monitoring the running processes. Particularly, we have used the memory_info function to retrieve the process memory utilization from total physical memory. The RSS (resident set size) represents non-swapped physical memory utilization. VMS (Virtual Memory Size) is the total amount of virtual memory a process used during computation. Shared memory is used by multiple processes. The text represents the amount of memory devoted to executable code. And, the data field represented in Table 5.11 indicates the amount of physical memory devoted to other than executable code. The size of these parameters is in Bytes.

Chapter 6

Concluding Remarks

The proposal covers three major elements of the security of SCADA systems, namely, vulnerability assessment, intrusion detection and secure communication.

For vulnerability assessment, Onion Omega2 (a small embedded Linux server) is taken as a case study component that is used for building SCADA/IoT systems. While it provides efficient functionality, it is important to be aware of its vulnerabilities and built-in security features. We have identified product-level vulnerabilities of Onion Omega2 using scanners and penetration tools. This helped us to identify the threats and vulnerabilities of Onion Omega2 and measure the level of risk. The vulnerabilities include missing patches, insecure system configurations, and other security-related updates. The identified vulnerabilities can either be fixed by the vendor and/or network administrator/engineer.

We have presented GBFS based WFI Scoring model for plant floor and RFE-XGBoost-based feature selection approach along with the majority vote-based ensemble method to classify normal and attack events at a control center. The experimental results reveal that the proposed framework fares well in terms of accuracy, detection rate, precision, and recall. Moreover, the proposed model outperforms some of the state-of-the-art published techniques. The model offers a blend of effectiveness with precision, as it uses a limited number of stable features, and the classification is carried out based on combined predictions of the nine most promising classifiers. Moreover, this combination requires limited computational cost, which is one of the crucial factors for mission-critical applications.

A robust solution to strengthen the security of ICSs against cyber-attacks is a crucial requirement in the design of SCADA systems. Through this work, we aim to cover the protection of the industrial control system landscape by offering a low-cost and robust framework for SCADA networks, which protects them against various cyber-attacks. In this section, we have proposed a session key agreement in addition

to lightweight multi-layered encryption techniques. The framework combines both symmetric and asymmetric cryptography to achieve high computational speed by covering all the security mechanisms. This security model is proposed to enhance the security of various industrial sectors such as water and sewage plants, power stations, chemical plants, oil industries, product manufacturing units, and transportation systems. The successful deployment of this model will allow operators and technicians to monitor and control the plant devices remotely as it will protect the entire system from potential breaches.

Bibliography

[1] D. Upadhyay, S. Sampalli, and B. Plourde, "Vulnerabilities' assessment and mitigation strategies for the small linux server, Onion Omega2," *Electronics*, vol. 9, no. 6, p. 967, 2020.

[2] D. Upadhyay and S. Sampalli, "SCADA (Supervisory Control and Data Acquisition) systems: Vulnerability assessment and security recommendations," *Computers & Security*, vol. 89, p. 101666, 2020.

[3] Y. Cherdantseva, P. Burnap, A. Blyth, P. Eden, K. Jones, H. Soulsby, and K. Stoddart, "A review of cyber security risk assessment methods for scada systems," *Computers & security*, vol. 56, pp. 1–27, 2016.

[4] A. Rezai, P. Keshavarzi, and Z. Moravej, "Key management issue in scada networks: A review," *Engineering science and technology, an international journal*, vol. 20, no. 1, pp. 354–363, 2017.

[5] F. M. Salem, E. Ibrahim, and O. Elghandour, "A lightweight authenticated key establishment scheme for secure smart grid communications," *Journal homepage: http://iieta. org/journals/ijsse*, vol. 10, no. 4, pp. 549–558, 2020.

[6] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids," *IEEE Transactions on Network and Service Management*, 2020, early access, doi: 10.1109/TNSM.2020.3032618.

[7] K. Poulsen, "Feature importance of feature selection," 2003, Accessed on 9/10/2019. [Online]. Available: https://www.securityfocus.com/news/6767

[8] B. Krebs, "Cyber incident blamed for nuclear power plant shutdown," washington Post. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05

[9] B. Kesler, "The vulnerability of nuclear facilities to cyber attack," *Strategic Insights*, vol. 10, no. 1, pp. 15–25, spring 2011.

[10] "SANS and electricity information sharing and analysis center (e-isac). analysis of the cyber attack on the ukrainian power grid," Accessed 2019-09-28. [Online]. Available: http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[11] N. Kshetri and J. Voas, "Hacking power grids: A current problem," *Computer*, vol. 50, no. 12, pp. 91–95, December 2017.

[12] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting scada cyber security: A survey of techniques," *Computers & Security*, vol. 70, pp. 436 – 454, 2017.

[13] B. Barrett, "Security news this week: An unprecedented cyberattack hit us power utilities," Accessed 2019-11-14. [Online]. Available: www.wired.com/story/power-grid-cyberattack-facebook-phone-numbers-security-news/

[14] D. Choi, S. Lee, D. Won, and S. Kim, "Efficient secure group communications for scada," *IEEE Transactions on power delivery*, vol. 25, no. 2, pp. 714–722, 2010.

[15] T. Pramod and N. Sunitha, "Polynomial based scheme for secure scada operations," *Procedia Technology*, vol. 21, pp. 474–481, November, 2015.

[16] A. Rezai, P. Keshavarzi, and Z. Moravej, "Secure scada communication by using a modified key management scheme," *ISA transactions*, vol. 52, no. 4, pp. 517–524, 2013.

[17] A. Rezai, P. Keshavarzi, and Z. Moravej, "Advance hybrid key management architecture for scada network security," *Security and communication networks*, vol. 9, no. 17, pp. 4358–4368, 2016.

[18] D. Choi, H. Jeong, D. Won, and S. Kim, "Hybrid key management architecture for robust scada systems," *Journal of information science and engineering*, vol. 29, no. 2, pp. 281–298, 2013.

[19] R. Jiang, R. Lu, C. Lai, J. Luo, and X. Shen, "Robust group key management with revocation and collusion resistance for scada in smart grid," in *2013 IEEE global communications conference (GLOBECOM)*. IEEE, 2013, pp. 802–807.

[20] A. Rezai, P. Keshavarzi, and Z. Moravej, "A new key management scheme for scada networks," $2^{n}dInternationalSymposiumonComputinginScienceEngineering, p.\ 04, 2011$.

[21] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135 812–135 831, 2019.

[22] Y. Sun, L. Sun, Z. Shi, W. Yu, and H. Ying, "Vulnerability finding and firmware association in power grid," in *2019 Fifth Conference on Mobile and Secure Services (MobiSecServ)*. IEEE, 2019, pp. 1–5.

[23] C.-C. Sun, C.-C. Liu, and J. Xie, "Cyber-physical system security of a power grid: State-of-the-art," *Electronics*, vol. 5, no. 3, p. 40, 2016.

[24] G. Yadav and K. Paul, "Architecture and security of scada systems: A review," *International Journal of Critical Infrastructure Protection*, vol. 34, p. 100433, 2021.

[25] A. Abou el Kalam, "Securing scada and critical industrial systems: From needs to security mechanisms," *International Journal of Critical Infrastructure Protection*, vol. 32, p. 100394, 2021.

[26] K. Coffey, R. Smith, L. Maglaras, and H. Janicke, "Vulnerability analysis of network scanning on scada systems," *Security and Communication Networks*, vol. 2018, 2018.

[27] S. Andersson and O. Josefsson, "On the assessment of denial of service vulnerabilities affecting smart home systems," 2019.

[28] M. Andrews, "Guest editor's introduction: The state of web security," *IEEE Security & Privacy*, vol. 4, no. 4, pp. 14–15, 2006.

[29] J. Steven, "Adopting an enterprise software security framework," *IEEE Security & Privacy*, vol. 4, no. 2, pp. 84–87, 2006.

[30] S. Samtani, S. Yu, H. Zhu, M. Patton, J. Matherly, and H. Chen, "Identifying scada systems and their vulnerabilities on the internet of things: A text-mining approach," *IEEE Intelligent Systems*, vol. 33, no. 2, pp. 63–73, 2018.

[31] R. Leszczyna, "Approaching secure industrial control systems," *IET Information Security*, vol. 9, no. 1, pp. 81–89, 2015.

[32] J. F. Van der Maelen Urıa and C. A.-R. Alvarez, "Using spline functions for obtaining accurate partial molar volumes in binary mixtures," *Computers & chemistry*, vol. 22, no. 2-3, pp. 225–235, 1998.

[33] J. Suaboot, A. Fahad, Z. Tari, J. Grundy, A. N. Mahmood, A. Almalawi, A. Y. Zomaya, and K. Drira, "A taxonomy of supervised learning for idss in scada environments," *ACM Computing Surveys (CSUR)*, vol. 53, no. 2, pp. 1–37, 2020.

[34] M. Altaha, J.-M. Lee, M. Aslam, and S. Hong, "An autoencoder-based network intrusion detection system for the scada system." *J. Commun.*, vol. 16, no. 6, pp. 210–216, 2021.

[35] R. C. Borges Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *2014 7th International Symposium on Resilient Control Systems (ISRCS)*, Aug 2014, pp. 1–8.

[36] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 650–662, June 2015.

[37] ——, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov 2015.

[38] M. Keshk, N. Moustafa, E. Sitnikova, and G. Creech, "Privacy preservation intrusion detection technique for scada systems," in *2017 Military Communications and Information Systems Conference (MilCIS)*, Nov 2017, pp. 1–6.

[39] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2019.

[40] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[41] M. Abirami, U. Yash, and S. Singh, "Building an ensemble learning based algorithm for improving intrusion detection system," in *Artificial Intelligence and Evolutionary Computations in Engineering Systems*. Springer, 2020, pp. 635–649.

[42] X. Li, M. Zhu, L. T. Yang, M. Xu, Z. Ma, C. Zhong, H. Li, and Y. Xiang, "Sustainable ensemble learning driving intrusion detection model," *IEEE Transactions on Dependable and Secure Computing*, pp. 1–1, 2021.

[43] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82 512–82 521, 2019.

[44] N. Moustafa, M. Keshk, K.-K. R. Choo, T. Lynar, S. Camtepe, and M. Whitty, "Dad: A distributed anomaly detection system using ensemble one-class statistical learning in edge networks," *Future Generation Computer Systems*, vol. 118, pp. 240–251, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167739X21000212

[45] Y. Wang, Y. Shen, and G. Zhang, "Research on intrusion detection model using ensemble learning methods," in *2016 7th IEEE International Conference on Software Engineering and Service Science (ICSESS)*, 2016, pp. 422–425.

[46] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2018, pp. 1–6.

[47] F. Idrees, M. Rajarajan, M. Conti, T. M. Chen, and Y. Rahulamathavan, "Pindroid: A novel android malware detection system using ensemble learning methods," *Computers & Security*, vol. 68, pp. 36–46, 2017. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404817300640

[48] A. H. Mirza, "Computer network intrusion detection using various classifiers and ensemble learning," in *2018 26th Signal Processing and Communications Applications Conference (SIU)*. IEEE, 2018, pp. 1–4.

[49] R. Singh, M. Kalra, and S. Solanki, "A hybrid approach for intrusion detection based on machine learning," in *2019 International Conference on Intelligent Sustainable Systems (ICISS)*. IEEE, 2019, pp. 187–192.

[50] M. F. Moghadam, M. Nikooghadam, A. H. Mohajerzadeh, and B. Movali, "A lightweight key management protocol for secure communication in smart grids," *Electric Power Systems Research*, vol. 178, p. 106024, 2020.

[51] R. Dawson, C. Boyd, E. Dawson, and J. Gonzalez Nieto, "Skma-a key management architecture for scada systems," in *ACSW Frontiers 2006: Proceedings of the 4th Australasian Symposium on Grid Computing and e-Research (AusGrid 2006) and the 4th Australasian Information Security Workshop (Network Security)(AISW-NetSec 2006)[CRPIT, Volume 54]*. Australian Computer Society Inc, 2006, pp. 183–192.

[52] D. Choi, H. Kim, D. Won, and S. Kim, "Advanced key-management architecture for secure scada communications," *IEEE Transactions on Power Delivery*, vol. 24, no. 3, pp. 1154–1163, 2009.

[53] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 2, pp. 375–381, 2011.

[54] D. Kang, J. Lee, B. Kim, and D. Hur, "Proposal strategies of key management for data encryption in scada network of electric power systems," *International Journal of Electrical Power & Energy Systems*, vol. 33, no. 9, pp. 1521–1526, 2011.

[55] P. TC, T. GS, S. Iyengar, and N. Sunitha, "Ckmi: Comprehensive key management infrastructure design for industrial automation and control systems," *Future Internet*, vol. 11, no. 6, p. 126, 2019.

[56] T. M.D. Hadley, K.A. Huston, "AGA-12, Part 2 Performance Test Results," Accessed 2020-10-12. [Online]. Available: https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/9-AGA-12_Part_2_Performance.pdf

[57] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, "Novel anonymous key establishment protocol for isolated smart meters," *IEEE Transactions on Industrial Electronics*, vol. 67, no. 4, pp. 2844–2851, 2019.

[58] N. Saxena, B. J. Choi, and R. Lu, "Authentication and authorization scheme for various user roles and devices in smart grid," *IEEE transactions on Information forensics and security*, vol. 11, no. 5, pp. 907–921, 2015.

[59] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Computers & Electrical Engineering*, vol. 52, pp. 114–124, 2016.

[60] D. Pliatsios, P. Sarigiannidis, T. Lagkas, and A. G. Sarigiannidis, "A survey on scada systems: secure protocols, incidents, threats and tactics," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1942–1976, 2020.

[61] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Intrusion detection in scada based power grids: Recursive feature elimination model with majority vote ensemble algorithm," *IEEE Transactions on Network Science and Engineering*, vol. 8, no. 3, pp. 2559–2574, 2021.

[62] U. Adhikari, S. Pan, T. Morris, R. Borges, and J. Beaver, "Industrial control system (ics) cyber attack datasets," datasets used in the experimentation. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[63] T. H. Morris, Z. Thornton, and I. P. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," in *7th Annual Southeastern Cyber Security Summit*, 2015, Huntsvile, AL. June 3 - 4, 2015.

[64] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner, "An evaluation of machine learning methods to detect malicious scada communications," in *2013 12th International Conference on Machine Learning and Applications*, vol. 2, Dec 2013, pp. 54–59.

[65] J. Hsu, D. Mudd, and Z. Thornton, "Mississippi state university project report-scada anomaly detection," 2014.

[66] P. Nader, P. Honeine, and P. Beauseroy, "l_p-norms in one-class classification for intrusion detection in scada systems," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2308–2317, 2014.

[67] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Transactions on Smart Grid*, vol. 10, no. 2, pp. 2326–2329, March 2019.

[68] K. M. Ting, *Precision and Recall*. Boston, MA: Springer US, 2010, pp. 781–781. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_652

[69] R. Zazkis, "Representing numbers: Prime and irrational," *International Journal of Mathematical Education in Science and Technology*, vol. 36, no. 2-3, pp. 207–217, 2005.

[70] t. f. e. From Wikipedia, "Blake (hash function)," Accessed 2021-05-12. [Online]. Available: https://en.wikipedia.org/wiki/BLAKE_(hash_function)

[71] wolfSSL, "Embedded tls library for applications, devices, iot, and the cloud," Accessed 2020-08-12. [Online]. Available: https://www.wolfssl.com/download

[72] libNTRU, "The ntru project," Accessed 2020-08-12. [Online]. Available: https://tbuktu.github.io/ntru/

[73] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, and S. Shenoi, "Security strategies for scada networks," in *International Conference on Critical Infrastructure Protection*. Springer, 2007, pp. 117–131.

[74] H. Delfs, H. Knebl, and H. Knebl, *Introduction to cryptography*. Springer, 2002, vol. 2.

[75] M. Stevens *et al.*, "Attacks on hash functions and applications," *Mathematical Institute, Faculty of Science, Leiden University*, vol. 3, 2012.

[76] H. Wang, Z. Ma, and C. Ma, "An efficient quantum meet-in-the-middle attack against ntru-2005," *Chinese Science Bulletin*, vol. 58, no. 28-29, pp. 3514–3518, 2013.

[77] S. N. Kumar, "Review on network security and cryptography," *International Transaction of Electrical and Computer Engineers System*, vol. 3, no. 1, pp. 1–11, 2015.

[78] S. Ghosh, M. LeMay, M. R. Sastry, and D. M. Durham, "Processor hardware and instructions for sha3 cryptographic operations," Apr. 16 2020, uS Patent App. 16/709,837.

[79] J. Hermans, F. Vercauteren, and B. Preneel, "Speed records for ntru," in *Cryptographers' Track at the RSA Conference*. Springer, 2010, pp. 73–88.

Appendix A

Selected Publications from the thesis

The following figure A.1 maps the publications to the contributions of this thesis.

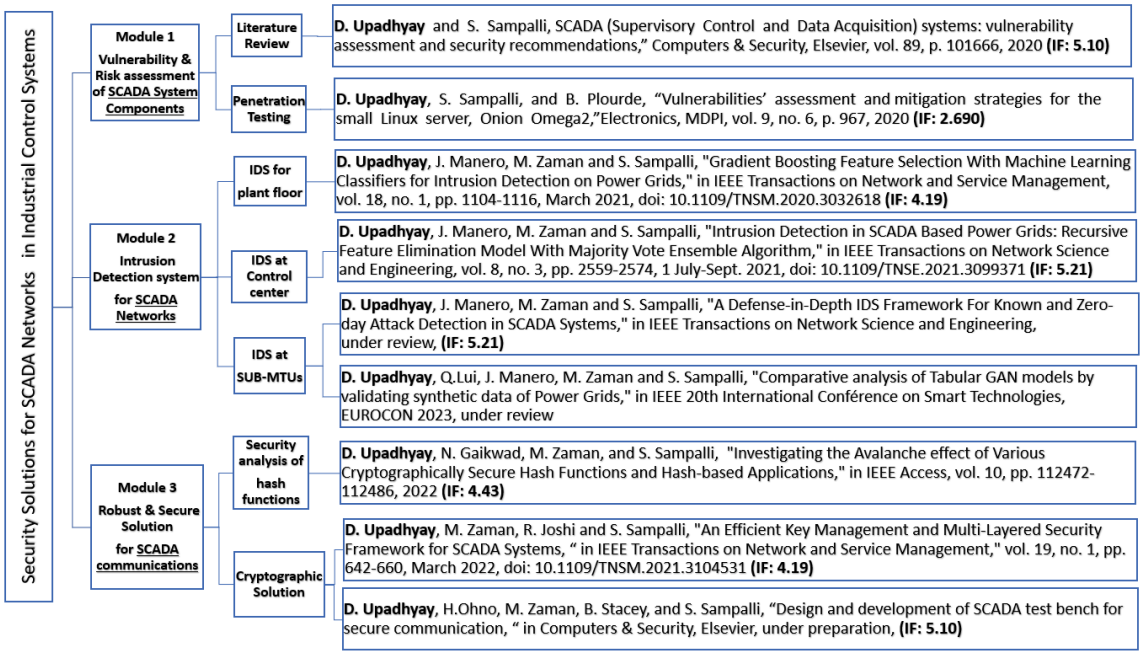


Figure A.1: Publication history based on SCADA Security modules along with impact factor

List of Publications from thesis

1. **D. Upadhyay** and S. Sampalli, "SCADA (Supervisory Control and Data Acquisition) systems: vulnerability assessment and security recommendations,"**Computers & Security, Elsevier,** vol. 89, p. 101666, 2020. *(Impact Factor: 5.105)*

2. **D. Upadhyay**, S. Sampalli, and B. Plourde, "Vulnerabilities' assessment and mitigation strategies for the small linux server, Onion Omega2," **Electronics, MDPI**, vol. 9, no. 6, p. 967, 2020. *(Impact Factor: 2.69)*

3. **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids," in **IEEE Transactions on Network and Service Management,** vol. 18, no. 1, pp. 1104-1116, March 2021, doi: 10.1109/TNSM.2020.3032618. *(Impact Factor: 4.19)*

4. **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model With Majority Vote Ensemble Algorithm," in **IEEE Transactions on Network Science and Engineering,** vol. 8, no. 3, pp. 2559-2574, 1 July-Sept. 2021, doi: 10.1109/TNSE.2021.3099371. *(Impact Factor: 5.21)*

5. **D. Upadhyay**, J. Manero, M. Zaman and S. Sampalli, "A Defense-in-Depth IDS Framework For Known and Zero-day Attack Detection in SCADA Systems," in **IEEE Transactions on Network Science and Engineering,** manuscript under review. *(Impact Factor: 5.21)*

6. **D. Upadhyay**, Q. Lui, J. Manero, M. Zaman and S. Sampalli, "Comparative analysis of Tabular GAN models by validating synthetic data of Power Grids," in 20^{th} **IEEE International Conference on Smart Technologies, EUROCON 2023**, manuscript under review.

7. **D. Upadhyay**, M. Zaman, R. Joshi and S. Sampalli, "An Efficient Key Management and Multi-Layered Security Framework for SCADA Systems," in **IEEE Transactions**

on Network and Service Management, " vol. 19, no. 1, pp. 642-660, March 2022, doi: 10.1109/TNSM.2021.3104531. *(Impact Factor: 4.19)*

8. **D. Upadhyay**, N. Gaikwad, M. Zaman and S. Sampalli, "Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash-Based Applications," in **IEEE Access**, vol. 10, pp. 112472-112486, 2022. *(Impact Factor: 4.43)*

9. **D. Upadhyay**, H.Ohno, M. Zaman, B. Stacey and S. Sampalli, "Design and development of SCADA test bench for secure communication," in **Computers & Security, Elsevier,** manuscript under preparation, *(Impact Factor: 5.105)*

Gradient Boosting Feature Selection With Machine Learning Classifiers for Intrusion Detection on Power Grids

Darshana Upadhyay, Jaume Manero, Marzia Zaman, and Srinivas Sampalli, *Member, IEEE*

***Abstract*—Smart grids rely on SCADA (Supervisory Control and Data Acquisition) systems to monitor and control complex electrical networks in order to provide reliable energy to homes and industries. However, the increased inter-connectivity and remote accessibility of SCADA systems expose them to cyber attacks. As a consequence, developing effective security mechanisms is a priority in order to protect the network from internal and external attacks. We propose an integrated framework for an Intrusion Detection System (IDS) for smart grids which combines feature engineering-based preprocessing with machine learning classifiers. Whilst most of the machine learning techniques fine-tune the hyper-parameters to improve the detection rate, our approach focuses on selecting the most promising features of the dataset using Gradient Boosting Feature Selection (GBFS) before applying the classification algorithm, a combination which improves not only the detection rate but also the execution speed. GBFS uses the Weighted Feature Importance (WFI) extraction technique to reduce the complexity of classifiers. We implement and evaluate various decision-tree based machine learning techniques after obtaining the most promising features of the power grid dataset through a GBFS module, and show that this approach optimizes the False Positive Rate (FPR) and the execution time.**

***Index Terms*—SCADA systems, power grids, random forest, gradient boosting, feature selection, cyber security, network intrusions.**

I. Introduction

POWER grids are the basic infrastructure that support our economies and daily lives by providing and sustaining a continuous supply of electricity. They play a fundamental role in connecting our industries and homes with locations far away from where the electricity is generated, while assuring the quality of the electricity supply at the point of consumption.

Manuscript received May 1, 2020; revised October 4, 2020 and August 22, 2020; accepted October 13, 2020. Date of publication October 22, 2020; date of current version March 11, 2021. The authors gratefully acknowledge the support by the Natural Sciences and Engineering Research Council (NSERC), Canada through a Collaborative Research Grant. The associate editor coordinating the review of this article and approving it for publication was F. De Turck. *(Corresponding author: Srinivas Sampalli.)*

Darshana Upadhyay and Srinivas Sampalli are with the Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 4R2, Canada (e-mail: srini@cs.dal.ca).

Jaume Manero is with the Department of Computer Science, Technical University of Catalonia, 08034 Barcelona, Spain.

Marzia Zaman is with the Research and Development Department, Cistel Technology, Ottawa, ON K2E 7V7, Canada.

Digital Object Identifier 10.1109/TNSM.2020.3032618

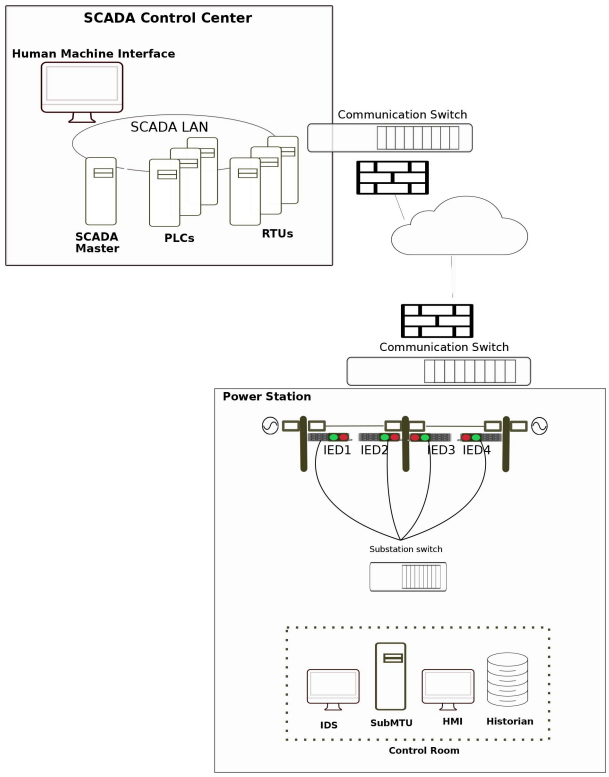


Fig. 1. SCADA System Architecture for Power Grids Legend: PLCs: Programmable Logic Controllers, RTUs: Remote Terminal Units, HMI: Human Machine Interface, IEDs: Intelligent Electronic Devices.

These systems are complex and distributed in nature and comprise several components such as power lines, transformers, sensors, phasor measurement units (PMUs) and sub-stations connected to supervisory control and data acquisition (SCADA) systems for real time monitoring, management and control. Figure 1 illustrates the block diagram of a SCADA architecture for a power grid, showing SCADA components such as SCADA Master, HMI, PLCs, RTUs, and various power grid components such as IEDs, substation switch and control room components.

Generally, the sensors and PMUs at power stations monitor different attributes of electrical signals continuously and transmit that to the field control devices such as PLC, RTU, or IED. Communication between the field control devices and the SCADA master takes place via communication links and switches. The SCADA master is located at the control center.

The field control devices supply digital status information to the SCADA Master to determine acceptable parameter ranges. This information will then be transmitted back to the field device(s) where action may be taken to optimize the performance of the system. Moreover, the status information is stored in a data historian and displays it on an HMI (Human Machine Interface), which provides centralized monitoring and system control.

Originally, power grids were designed to generate and distribute the electricity in an efficient and timely manner, rather than focusing on security aspects of the critical infrastructure of the system. However, the increase of inter connectivity and remote accessibility places power grids under the risk of internal and external attacks.

Real-time cyber attacks can disrupt entire power grids. For example, in 2003 the Davis-Besse nuclear power plant near Oak Harbor, Ohio was infected by a Slammer worm that traveled from a consultant's network to the process control network and generated unwanted traffic [1]. As a result, the plant personnel could not access the safety parameter display system for around five hours which showed sensitive data about the reactor core, temperature, and radiation sensors of the power plant. In 2006 the Browns Ferry nuclear plant in Athens, Alabama was shut down after the failure of critical reactor components and controllers due to a cyber attack on their internal network [2]. In 2008, the second unit of the Hatch nuclear power plant in Baxley, Georgia experienced an automatic shutdown due to routine software update to a single computer on the plant floor. The update was performed to synchronize data between the plant and business networks [2]. Another incident in an Iranian nuclear plant was reported in 2011 where the plant process was interrupted due to the Stuxnet worm. This attack was initiated by connecting an infected USB drive to the Programmable Logic Controller (PLC) at the plant floor [3]. The Ukraine power plant cyber attack was reported in 2015 [4]. This was the first known successful attack on power grids where attackers were able to disrupt electricity supply to the end users. Thus, power grid attacks are one of the most critical issues in industrial control systems and it is important to protect them by applying adequate safety measures [5].

General safeguards include defense-in-depth architecture which separates the control and corporate network traffic, strong access control and authentication mechanisms, restricted perimeters using DMZ (demilitarized zone), vulnerability assessment and risk management systems [6]. However, these safeguards are difficult to deploy and maintain owing to legacy-inherited security loopholes and restrictions [7]. Therefore, these relevant preemptive measures are not sufficient to protect the power grids from cyber attacks. Additional protection layer is also required which detects and prevents the system from malicious events and threats.

Generally, packet filtering and identification of threats are key to securing these systems. However, traditional firewalls do not always fulfill all the security requirements of critical infrastructures. For example, in 2019, the western U.S. power grid infrastructure was hacked. The intruders created periodic blind spots for grid operators for about 10 hours, by identifying a vulnerability in the firewall configuration [8]. Therefore, the design and development of sophisticated and accurate intrusion detection and prevention systems are one of the primary objectives to secure power grids.

Researchers and security experts have proposed various intrusion detection and prevention approaches to ensure secure and safe operations of power grids. A signature-based approach is used for pattern matching to determine frequent signatures of malicious packets [9]. In this approach the signature of every incoming packet is compared with all the stored signatures to identify threats. This approach is valid for known intrusions but is unable to identify zero-day attacks [9].

More recently, data mining, clustering and statistical signal processing approaches have been used for anomaly detection. These techniques are effective compared to pattern-matching, but usually generate a high level of false-positive alarms [10]. Therefore, there is a need for better techniques that detect intrusions from real incoming traffic. Machine Learning and Deep Learning have stronger pattern recognition capabilities than standard approaches. These techniques train and test the model according to real network traffic to detect anomalies with better precision and generate a smaller number of false-positive alerts. Some of the most prominent machine learning techniques include decision trees, Bayesian, genetic algorithms, neural-networks and support vector machines [11].

Decision tree algorithms, which make decisions using bias and variance analysis mechanisms are one of the powerful supervisory machine learning techniques. Furthermore, ensemble methods use the principle of combining weak learners to obtain a stronger predictive model for better prediction and performance. Ensembles can be obtained by boosting, which is a specific mechanism where learners gradually learn from the previous weak learners to reduce the overall loss function. Moreover, Gradient Descent is used to optimize the overall tree selection. This combined approach provides a powerful method for identification and pattern recognition capabilities for structured data [12].

Our proposed approach uses the Gradient Boosting algorithm as the base classifier to detect malicious activities in power grids. To solve the classification and regression problems, the ensemble Gradient Boosting algorithm has proven to be more efficient than traditional boosting approaches [13]. The ensemble Gradient Boosting algorithm is an ensemble learning method based on a combination of additive models (weak learners), which can gradually learn from the previous misclassifications to create a stronger learning model [14]. This algorithm has been complemented with a feature selection process that increases the overall performance by extracting the most relevant features from the input data.

The proposed technique has been developed using various library functions of the open source library scikit-learn [15]. The library offers various classification, regression, and clustering algorithms. Table I summarizes the general scientific meanings of the software implementation terms used in this article.

The major contributions of this article are as follows.

1) We use the gradient boosting weighted feature importance scoring model and tune the Num_trees parameters

TABLE I
GENERAL SCIENTIFIC MEANINGS OF SOFTWARE IMPLEMENTATION TERMS

Term	Description
scikit- learn	This is a Python module integrated with a wide range of machine learning techniques for both supervised and unsupervised learning. We have used various functions of scikit-learn library for the implementation and comparative analysis of the proposed methodology.
Num_trees	This term indicates the number of trees that we want to build for the average prediction. For the proposed feature selection technique, we have tuned this parameter by creating 100, 500, 700, and 1000 trees at each iteration.
Model based feature selection	This refers to the meta-transformer technique, which uses the WFI scoring model to remove insignificant features according to the threshold value. In this paper, we have used Grading Boosting as a base model, and the threshold value is set to 0.5 to remove the unimportant features.

to identify the top important features. To make it more efficient, we merge these two concepts to select the most Promising and common features from the existing datasets that reduce the overhead and increase the execution speed for SCADA based power grids.

2) We derive 15 most promising features from the binary class and apply the same features to the rest of the three categories, namely, three class, seven class, and multi class, to evaluate the performance of the feature selection module.
3) We evaluate eight different tree-based algorithms to validate the effectiveness of the selected features for the classification of various power system attacks.
4) We perform a comparative analysis of eight tree-based classifiers and identify the top three tree-based classifiers according to multiple performance metrics.
5) We compare the accuracy of proposed methodology with published state-of-the-art techniques.

The rest of this article is organized as follows. Section II describes related research in the area of power grid security by considering various attacks and protection schemes. The proposed intrusion detection system framework based on Gradient Boosting Feature Selection is introduced in Section III. Section IV covers algorithm conceptualization and mathematical proof of our approach. Section V describes the proposed mechanism of feature selection by combining regularization strategies with Weighted Feature Importance metrics. Section VI presents the complete experimental setup, evaluations, result-analysis and comparative studies of various tree-based machine learning techniques performed on power grid datasets. Conclusion and future work are provided in Section VII.

II. BACKGROUND AND RELATED WORK

Many researchers have proposed different types of intrusion detection systems (IDSs) according to the need of securing various components in power grids. For example, one approach is specifically focused on security of the RTU and the PLC, as these devices are easy targets for cyber attacks [16]. A real-time attack with malware running on a PLC was demonstrated by black hat researchers in 2016 [17].

Malicious cyber-attacks have costly consequences in power grids, and as a result the grid operators are increasingly investing in IDSs. IDSs are typically based on the principle that attacks show different behavior and patterns from the normal traffic [18]. In this sense the classification problem can be reduced to a pattern recognition activity. To identify malicious behavior, identifying a pattern that differs from the normal flow is required. The traditional approach is to develop a signature of the attack and recognize this signature. This method requires extensive manual work as the signature is manually added to the database when the attack is identified and its signature extracted. A more sophisticated approach is to use machine learning to perform the pattern recognition process [11].

Feature selection is also known as dimensionality reduction, which is used to improve the accuracy of estimators and boost the performance of the high-dimensional datasets. The feature selection techniques are mainly categorized into four types, namely, Variance Threshold (VT), Univariate Feature Selection (UFS), Recursive Feature Elimination (RFE) and Model based feature selection. VT is a simple baseline approach that removes all the variance which does not meet the threshold, whereas UFS follows the method of a statistical test to identify the best features [19]. In the UFS approach, the features are selected by either comparing false positive rates or obtaining scores or percentile of the given features [20]. Moreover, the configurable strategy of UFS allows a combination of two approaches, namely, univariate selection and hyper-parameter search estimator.

On the other hand, RFE selects the features recursively by comparing the outcome of a larger set with the smaller set while training the dataset [21]. This technique is more efficient in terms of estimators' accuracy scores but computationally costlier than VT and UFS. Model based feature selection method is a meta-transformer that uses the WFI scoring model to remove unimportant features according to the threshold value [22]. This is comparatively faster than other techniques as feature importance score is obtained during tree construction. Moreover, this method can easily be merged with other estimators, such as tuning the parameters.

To identify top features, we have used a gradient boosting based WFI scoring model to discard the irrelevant features along with Num_trees to tune the parameters. This approach improves not only the accuracy of the tree-based classifiers but also the execution speed.

Several machine learning approaches have been tested to filter malicious packets, for instance, K-nearest neighbours (k-NN) is quite effective, since its main characteristic, is being a 'lazy learner' - it does not contain the trained model but builds it real-time by learning from the nearest neighbours - is very well aligned to this task. However it has high performance requirements and may have fitting issues for imbalanced small datasets [23]. Other tested approaches such

as Support Vector Machines (SVM), which maps the inputs into another dimensional space, offer good results, but are costly to train. Neural network approaches have also shown strong representational capabilities, but have not yet been widely applied in commercial applications [11].

In the classification field, and for structured data inputs, the gradient boosting family of algorithms shows improved representation capabilities [24]. This approach combines boosting with decision trees techniques. Specifically they combine random tree refinements with boosting techniques' optimization. Variants like Gradient Tree Based Boosting (GTBM) or the recently developed XGBoosting (Extreme Gradient Boosting) are becoming tools of choice in many applications [24]. However their effectiveness has not been widely studied on various IDS applications, which is the main motivation for this work.

Furthermore, power grid SCADA systems rely on real time request response mechanisms to operate the sub-station components accurately by consuming minimal CPU and battery resources. For such time-critical systems, the deployed intrusion detection system should act as quickly to capture malicious activities using minimal resources in a given time period for larger-scale deployments. Our proposed model leverages all the competencies for such systems. The model offers a combination of efficiency with precision, as it reaches high accuracy levels while using a limited amount of resources. This combination makes this model a good fit for mission critical applications or for large sets of disseminated SCADA devices, that have limited computing availability for filtering mechanisms, and both these properties fit very well with the power system scenario.

III. Framework for a GBFS Based Intrusion Detection System

This section presents the proposed framework for an intrusion detection system that distinguishes normal and malicious events by analyzing SCADA traffic on power grids. The proposed framework operates in three phases, namely, preprocessing the data, feature selection, and anomaly detection using a classification approach. The elements for each phase are illustrated in Figure 2.

During the data preprocessing phase, data cleansing, feature mapping and feature normalization are applied to the raw dataset to obtain filtered data. Then the Gradient Boosting Feature Selection approach is applied on filtered data to select the most promising features from the entire dataset dynamically. Since power grids use a complex mix of SCADA systems to control field-site components, network monitoring devices such as SNORT and Syslog are used to capture the different types of features [25].

Usually, real-time data obtained from sensors or real-time systems always presents some consistency issues, the signal is lost, or the measuring devices get off the scale readings at some point. For this reason, we need to do a data cleansing operation to remove incorrect data. We remove infinities and NaN values, looking for empty sequence points that will be avoided by the algorithms.

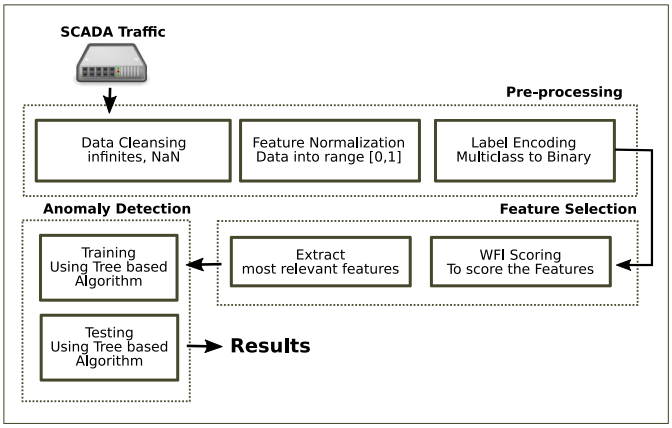


Fig. 2. Framework for a GBFS Based Intrusion Detection System.

Furthermore, in order to extract the relevant features, we apply a Gradient Boosting Feature Selection which uses Weighted Importance Feature extraction method to select the most promising features. This approach helps to improve the computational speed and also assists in providing a precise outcome for anomaly detection. Moreover, reduction in features helps in consuming less memory while training and testing the dataset during classification to classify normal and attack events.

IV. Gradient Boosting and XG Boosting Theory

In this work, we have used the combination of two main concepts, namely, the gradient boosting WFI scoring model and Num_trees for feature selection, and XGBoost as one of the classification methods.

One of the most efficient techniques of the tree-based ensemble method is called boosting, which stores the labels and weights of the leaf nodes that make the prediction interpretations easy to handle. Gradient boosting [26] is a practical approach proposed by Chen and Guestrin [24] and is considered as one of the algorithms of choice in machine learning. We can obtain a strong learner by combining weak learners during the gradient boosting process. In this technique, the classification is dependent on the residuals of the previous iteration where the impact of each feature is evaluated sequentially until a target accuracy is obtained. The residuals are calculated by a Loss function $\mathcal{L}(\phi)$ that is optimized using gradient descent. The final result $\phi(X)$ is obtained by the addition of the results of the K sequential classifier functions f_k as follows:

$$\hat{Y} = \Phi(X) = \sum_{k=1}^{K} f_k(X) \quad f_k \in F \tag{1}$$

where f_k is a decision tree, and K is the total number of iterations in the boosting algorithm.

XGBoosting has two enhancements, an improvement over Gradient Descent and a more sophisticated regularization strategy. The regularization factor to the cost function controls the optimization process and manages the overfitting factor. In this, the function to optimize in Step t is called the regularization term $\Omega(f_t)$ and we use it in the following equation to

calculate a Loss function $\mathcal{L}(\phi)_t$ at step t.

$$\mathcal{L}(\phi)_t = \sum l(f_{t-1} + f_t) + \Omega(f_t). \quad (2)$$

Without the regularization factor, the tree will split until it learns all the features of the training set, which may result in overfitting. By using a regularization function the training stops when the function identifies that the model is good enough based on the learning score, which avoids the chance of overfitting.

During optimization, the regularization term is improved by approximation using a short Taylor series decomposition. For complete details of XGBoost, we refer the reader to the original article written by Chen and Guestrin [24].

A. Using Weighted Feature Importance (WFI) for Feature Selection

Gradient boosting uses a powerful metric, called *feature / importance*, to retrieve the scores of each attribute according to importance after the boosted tree is constructed. This scoring model provides the importance of each attribute in terms of making key decision while constructing decision trees. Generally, feature importance provides a score that defines the significant role of each attribute. This importance is computed explicitly by comparing and ranking all the features amongst one another in the dataset. The importance of a single decision tree is calculated by the amount of each attribute split point, weighted by the number of observations from that node. This split point is used to improve the performance and efficiency of the algorithm.

In particular, purity (Gini Index) is used to select the split points or to identify a more specific error function. The feature importance of each tree is averaged across all the decision trees within the model. The Model based feature selection class is used to transform a dataset into subsets by using the most promising features. The focal point of this approach is to embed the preprocessing with this model using WFI to reduce the training time by removing irrelevant features from the given dataset. Once the most promising ones are derived through the GBFS technique, we can effectively use them for training and testing the model.

V. A Novel Weighted Featured Selection Algorithm for Intrusion Detection

A. Power System—Testbed Description

This section describes the overall approach with regard to multilevel multiple attack vector classification of power system disturbances. To evaluate the performance of the GBFS based proposed algorithm, three publicly available datasets are used [27]. These datasets were created at Oak Ridge National Laboratories (ORNL) using the power system testbed.

The power system testbed configuration has been implemented using power generators- G1,G2 and IEDs - R1 through R4, to control the breakers BR1 through BR4, on or off, respectively. To fulfill the simulation requirements, the three-bus two-line transmission system is created [28]. Each one of the four IEDs uses a distance protection scheme to trip the respective breaker in case of fault detection, whether the nature of the fault is valid, or faked since they do not have smart logic to detect the difference between original and fake faults. Furthermore, operators can manually trip the breakers by issuing commands in case of maintenance on the lines or other system components [25].

B. Dataset

The datasets include measurement related to normal, disturbance, control and cyber attack behaviours with regards to electrical transmission system in the power grid [29]. There are three publicly available datasets and two of them are derived using the third main dataset consisting of fifteen sets with 37 power event scenarios in each dataset. The datasets are randomly sampled and categorised into three major classes; Binary, Three-class and Multiclass. Furthermore, we have derived a fourth dataset named Seven-class of fifteen sets from the multiclass dataset, consisting of seven power event scenarios in each.

The experiments were carried out using 4 different categories of the datasets where the Binary dataset has two output labels, namely, normal and attack, The Three-class dataset has three output labels - one additional label to binary dataset is no event. The Seven-class dataset has seven output labels as follows: 1 natural SLG (Single Line Ground) fault event owing to short-circuit in a power line, 1 data injection attack, 2 remote tripping command injection attacks and, 3 relay setting change attacks. The 37 scenarios of Multiclass dataset are divided mainly in three categories - Natural events, 1 No event and 28 Attack events. 8 Natural events categorized in 6 SLG faults and 2 Line maintenance events. Furthermore, no event indicates normal operation load changes and 28 attack events are mainly divided into 3 major attack events termed as Data Injection, Remote Tripping Command Injection and Attack on Relay Setting. These attacks are further subcategorized in 6 data-injection SLG fault replay attacks, 4 command injection attacks against single IED (relay), 2 command injection attacks against two IEDs, 10 relay setting change attacks on a single IED, 4 relay setting change attacks on two IEDs, and 2 relay disable and line maintenance attacks [30]. Moreover, these authentic datasets are used in various experiments related to power system cyber-attacks classification [27]. All the attacks scenarios are simulated by assuming that the intruder is an internal entity, which is capable enough to launch various attacks by issuing malicious commands from the substation switch [25].

Each power grid dataset consists of 128 features. To derive these features, 4 phasor measurement units (PMUs) are used to measure the electrical signals on an electrical power grid using common time source to maintain time synchronization. Each PMU measures 29 features, hence in total 116 PMU measurement carried out using 4 PMUs. These features are referred as R# - signal_Reference which indicates the index of PMU and type of measurement. For example, R1-PA1:VH represents the Phase A voltage phase angle measured by PMU R1 [27]. Also, 16 more columns are additionally inserted by control panel logs, snort alerts and relay logs where relay and PMU are integrated together [30]. The last column represents the marker to label different events. The description of all the features is

TABLE II
DESCRIPTION OF FEATURES

Feature	Description
PA1:VH-PA3:VH	Phase A-C Voltage Phase Angle
PM1:V-PM3:V	Phase A-C Voltage Magnitude
PA4:IH-PA6:IH	Phase A-C Current Phase Angle
PM4:I-PM6:I	Phase A-C Current Magnitude
PA7:VH-PA9:VH	Pos.-Neg.-Zero Voltage Phase Angle
PM7:V-PM12:V	Pos.-Neg.-Zero Voltage Magnitude
PA10:VH-PA12:VH	Pos.-Neg.-Zero Current Phase Angle
PM10:V-PM12:V	Pos.-Neg.-Zero Current Magnitude
F	Frequency for relays
DF	Frequency Delta (dF/dt) for relays
PA:Z	Apparent impedance seen by relays
PA:ZH	Apparent impedance Angle seen by relays
S	Status Flag for relays

shown in Table II. Also, each set of 15 sets consist average 294 "no event"İinstances, 1221 natural events instances and 3711 attack vectors across the classification schemes [25].

C. Regularization Strategies

Generally, boosting algorithms play a vital role in controlling the bias-variance trade-off. The objective of the gradient boosting algorithm is to generate an optimal combination of the trees while training the model using the concept of binomial deviance theorem. In addition to minimizing the loss function to the smallest possible degree, it is necessary to tune the hyper-parameters carefully, since complex trees overfit and simple trees can move the model to under-fitting. The majority of the tuning parameters are divided into two categories, one is specifically meant for construction and efficiency of each individual tree, and the other type of boosting parameters are used to boost the operation in the model. Owing to this fact, we have tuned the hyper-parameters by extensive grid search, taking learning rate, sub-samples and Num_trees into consideration.

We have analyzed the effect of different regularization strategies on various datasets by implementing a grid search. Figure 3 illustrates the effect of boosting parameters of one of the 15 binary datasets. According to the results depicted in the graph, regularization via shrinkage (learning rate $= 0.1$) improves the performance significantly, as compared to without shrinkage (learning rate & subsample $= 1.0$) and in the case of stochastic gradient boosting (combination with learning rate and subsample < 1.0).

D. Feature Selection

Generally, when we have a big model with hundreds or thousands of features, the feature selection approach is used to choose the most promising features and to remove irrelevant features while retraining the model. Also, by analyzing the importance of each feature manually, we can get an idea of what the model is doing, and the model is working well. Here, we derive the importance of each feature by applying WFI scoring method on Gradient Boosting trained model. Furthermore, all the features are depicted as a percentage rating of how often the feature is used in determining the output label. To make the list of features easier to read, we have

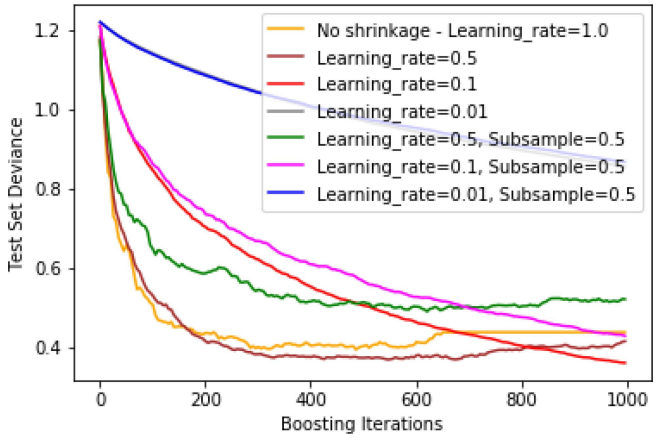


Fig. 3. Different Regularization strategies applied on a binary classification. A hyper parameter optimization (learning rate $= 0.1$) improves the result significantly, with small learning rates more trees are required for convergence.

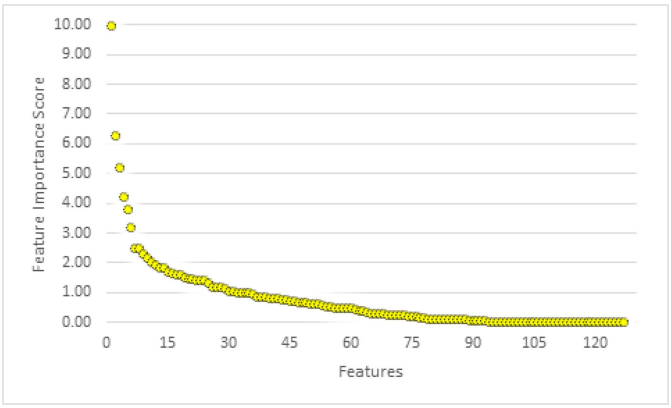


Fig. 4. WFI scoring model to rank the features.

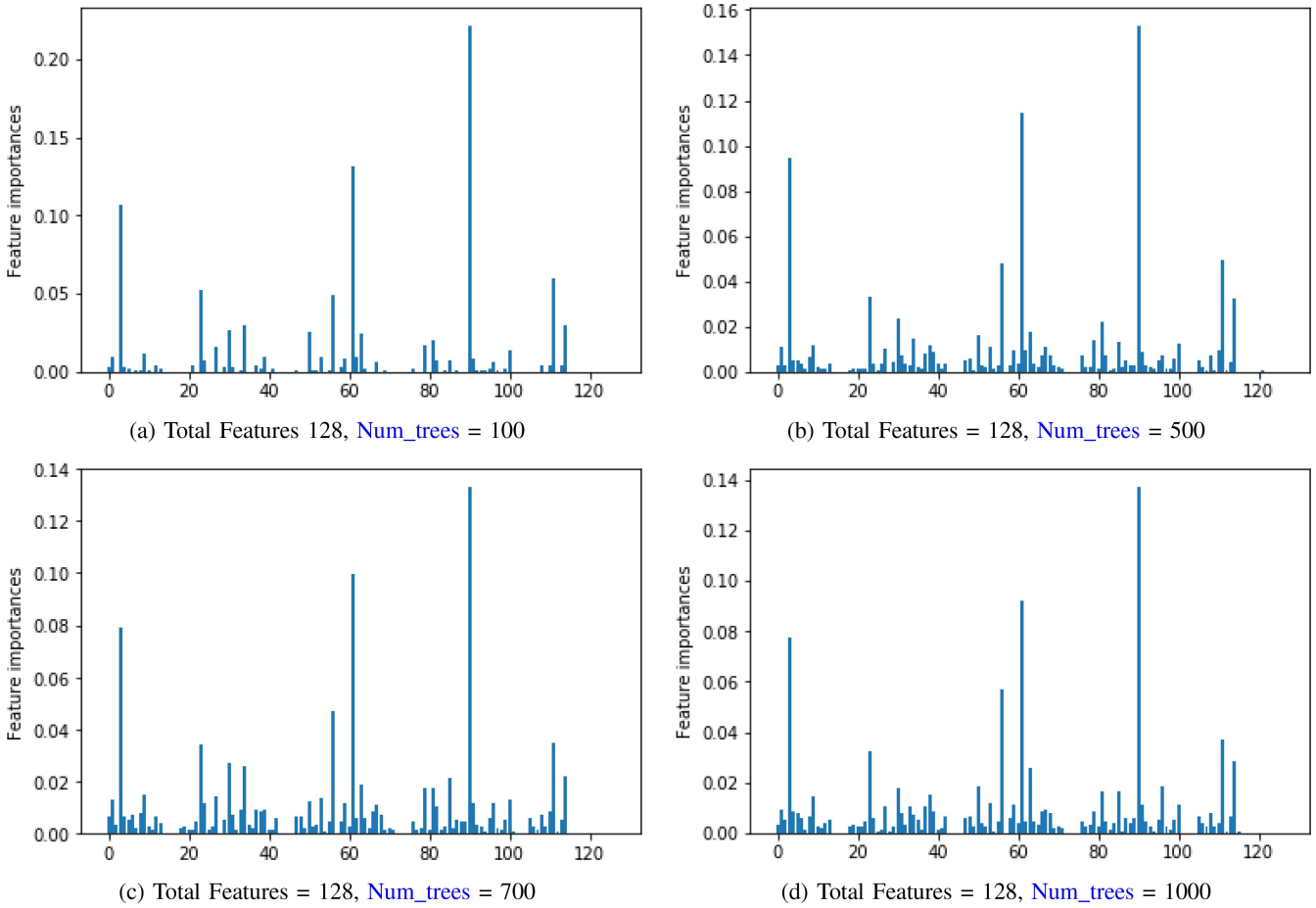
sorted them from most important to least important as shown in Figure 4.

The feature importance scores reflect information gain by each feature during the construction of a decision tree. During experiments, we observe 50% of the 128 features are not contributing to making any decision. The WFI score of such features is zero. While, out of the remaining 50% of features, 15 features provide a significant contribution in making decisions during the construction of decision-tree. The WFI score of those features has high values in the range of 1 to 10. The rest of the 45 features having feature importance scores between 0 and 1. These 45 additional features contribute comparatively less and have a large drop in feature importance score. Altogether the entire dataset is divided into three levels of information gain groupings, namely, most promising, slightly contributing, and irrelevant features.

According to [31], feature extraction creates a subset of the given features which not only reduces the noise but also improves the classifiers' performance. Therefore, we have tested 15 datasets of four different categories (binary, three-class, seven-class & Multi-class) of power grid system created by the Oak Ridge National Laboratories using the most promising features [27]. To identify these best features, we use the WFI scoring model along with concept of Num_trees.

TABLE III
GRADIENT BOOSTING FEATURE SELECTION (BEST 15 FEATURES OF 15 DATASETS FOR ALL THE FOUR CATEGORIES - BINARY, THREE CLASSES, SEVEN CLASSES AND MULTI-CLASS)

features	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	f11	f12	f13	f14	f15
D1	R2-PA3:VH	R1-PM10:I	R1-PA1:VH	R2-PM1:V	R2-PA10:IH	R2-PA5:IH	R2-PM10:I	R3-PA5:IH	R1-PM5:I	R3-PA1:VH	R2-PM5:I	R1-PA5:IH	R4-PA1:VH	R3-PA7:VH	R1-PA7:VH
D2	R2-PM3:V	R4-PA2:VH	R1-PA2:VH	R2-PM1:V	R4-PA1:VH	R1-PA5:IH	R4-PM2:V	R1-PA1:VH	R3-PM7:V	R2-PA3:VH	R2-PA7:VH	R4-PM1:V	R3-PA5:IH	R3-PM5:I	R1-PA7:VH
D3	R3-PA4:IH	R2-PM10:I	R3-PA2:VH	R2-PA2:VH	R2-PM5:I	R4-PA1:VH	R2-PA4:IH	R3-PM2:V	R2-PA5:IH	R1-PM5:I	R3-PA3:VH	R2-PA3:VH	R3-PM5:I	R1-PA7:VH	R4-PM5:I
D4	R2-PA7:VH	R4-PM5:I	R4-PA2:VH	R4-PA3:VH	R4-PA7:VH	R1-PA5:IH	R4-PM2:V	R2-PA2:VH	R2-PA5:IH	R4-PA1:VH	R1-PM5:I	R4-PA5:IH	R1-PA3:VH	R1-PA2:VH	R2-PM5:I
D5	R3-PA7:VH	R1-PA5:IH	R4-PM2:V	R4-PA4:IH	R4-PA5:IH	R4-PM5:I	R3-PM6:I	R3-PA10:IH	R2-PA5:IH	R3-PA3:VH	R1-PA3:VH	R4-PM4:I	R4-PA1:VH	R2-PA2:VH	R4-PA7:VH
D6	R4-PA1:VH	R3-PM2:V	R3-PA2:VH	R4-PM3:V	R1-PA2:VH	R4-PA7:VH	R2-PA10:IH	R4-PA2:VH	R2-PA5:IH	R1-PM10:I	R1-PA7:VH	R4-PM2:V	R3-PA5:IH	R4-PM5:I	R1-PM5:I
D7	R4-PA6:IH	R1-PA7:VH	R1-PM5:I	R1-PA1:VH	R2-PM7:V	R1-PA6:IH	R4-PA7:VH	R3-PA5:IH	R3-PA6:IH	R4-PA1:VH	R4-PA3:VH	R3-PM2:V	R4-PM7:V	R2-PA3:VH	R3-PA3:VH
D8	R4-PA7:VH	R1-PM2:V	R1-PA2:VH	R2-PA3:VH	R1-PA5:IH	R2-PA1:VH	R1-PM5:I	R1-PA3:VH	R3-PA5:IH	R3-PA6:IH	R3-PA2:VH	R4-PA3:VH	R4-PM2:V	R4-PA1:VH	R4-PA5:IH
D9	R2-PA2:VH	R4-PM7:V	R2-PM5:I	R4-PA1:VH	R3-PA2:VH	R1-PA3:VH	R4-PA7:VH	R3-PA5:IH	R1-PM2:V	R1-PA2:VH	R4-PA5:IH	R2-PA3:VH	R3-PA3:VH	R4-PA2:VH	R4-PM2:V
D10	R3-PA4:IH	R1-PA1:VH	R1-PA7:VH	R4-PA5:IH	R4-PA7:VH	R2-PM1:V	R1-PA5:IH	R4-PA1:VH	R4-PM5:I	R4-PM7:V	R3-PM2:V	R2-PA5:IH	R4-PA2:VH	R4-PM2:V	R3-PA5:IH
D11	R2-PA4:IH	R3-PA5:IH	R4-PM1:V	R1-PM5:I	R2-PM5:I	R2-PA1:VH	R4-PM5:I	R2-PM7:V	R1-PA2:VH	R2-PA6:IH	R2-PA5:IH	R4-PA2:VH	R2-PM1:V	R4-PM7:V	R4-PM2:V
D12	R4-PA3:VH	R4-PA7:VH	R2-PA3:VH	R1-PA2:VH	R2-PM5:I	R1-PM5:I	R3-PM2:V	R2-PA5:IH	R3-PA5:IH	R4-PA1:VH	R4-PA2:VH	R4-PM5:I	R1-PA:ZH	R1-PA3:VH	R3-PM5:I
D13	R3-PA3:VH	R3-PA2:VH	R2-PA3:VH	R3-PA5:IH	R4-PA6:IH	R1-PA1:VH	R4-PA7:VH	R4-PA2:VH	R4-PM2:V	R3-PM2:V	R2-PA5:IH	R1-PA7:VH	R1-PA3:VH	R4-PA3:VH	R4-PA1:VH
D14	R3-PM10:I	R1-PA1:VH	R2-PA5:IH	R4-PM7:V	R1-PM2:V	R4-PA5:IH	R1-PA2:VH	R2-PA6:IH	R3-PA3:VH	R3-PA6:IH	R1-PA7:VH	R4-PA3:VH	R1-PA6:IH	R1-PA3:VH	R4-PM2:V
D15	R4-PA1:VH	R4-PM7:V	R2-PM5:I	R1-PA5:IH	R3-PM3:V	R1-PA7:VH	R3-PA5:IH	R4-PM5:I	R2-PA3:VH	R2-PA5:IH	R4-PA7:VH	R1-PA1:VH	R3-PA3:VH	R3-PM2:V	R4-PM2:V



(a) Total Features 128, Num_trees = 100

(b) Total Features = 128, Num_trees = 500

(c) Total Features = 128, Num_trees = 700

(d) Total Features = 128, Num_trees = 1000

Fig. 5. Represents the relative importance of each attribute of the dataset with 5000 records;computed by considering four estimators Num_trees = 100,500,700,1000.

Furthermore, to increase the execution speed, we perform feature extraction on binary datasets. We repeat the entire process by taking the various parameter value of Num_trees to collect various observations. From that we have identified best features by taking common important features from the estimations as shown in the Algorithm 1. Here, Num_trees refers to the number of estimators whereas n refers to the total number of features. We have used four estimators, namely, 100, 500, 700, 1000 and initially dataset consist of $n = 128$ features.

Figure 5 represents the relative importance of each attribute on the binary dataset by considering four estimators. The high vertical bars represent the most promising and common features in all four estimators. In this experiment, all estimators use the top 15 features for each ensemble. In Table III we observe the most promising features across all 15 datasets. Also, to validate the strength of the selected features, the same 15 ones are applied to all four categories (Binary, three classes, seven classes and Multi-class) of intrusion classification. It can be observed that each dataset has a different set of stronger features, a conclusion that points to independent feature selection process for each dataset type. The most important features which contribute in determining the intrusions are Voltage

Algorithm 1: Weighted Feature Importance Based on a Gradient Boosting Feature Selection Model

Input: Training power-grid dataset PD
Output: Selected feature subset Selected PD
Initialize: Current power-grid dataset
Current-PD $= \{1, 2, \ldots, n\}$;
begin
  i $\leftarrow$ 0
  Num_trees $\leftarrow \{100, 500, 700, 1000\}$
  Num_trees $\leftarrow$ Num_trees (i)
  while *Features(Num_trees $>$ 0)* **do**
    (1) Create GB model on value Num_trees
    (2) Evaluate Ranking with WFI scoring
    (3) Remove features lower importance
    (4) Store the features in Scored-PD
    Num_trees $\leftarrow$ Num_trees (i + 1)
  end
  (5) Compare features of Scored-PD from all Num_trees
  (6) Take common features of Scored-PD Selected-PD $\leftarrow$ Scored-PD
end

Phase Angles, Voltage Magnitude, Current Phase Angles and Current Magnitudes according to the attack location on PMUs.

VI. Experiments

A. Evaluation parameters

The choice of the evaluation parameters always depends on the nature of the dataset, whether it is a multi-class or just binary. Typically, datasets are imbalanced in nature, a property defined by having classes of different sizes. Hence to evaluate the efficiency of the proposed GBFS based framework, our approach does not only relies on the accuracy of the classifier but also incorporates other assessment parameters like Detection Rate (True Positive Rate also called Recall & True Negative Rate), Precision, F1 Score and Miss Rate (False Negative Rate).

The assessment metrics, namely, accuracy, recall, precision and false negative rate depend on the following four parameters,namely, True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) [32]. TP refers to the number of actual attacks which are classified as attacks, TN refers to the number of normal events classified as normal events, FP refers to the number of normal events misclassified as attacks and FN refers to the number of attacks misclassified as normal events. The evaluation metrics are defined as follows, described from the basic four definitions.

- Accuracy is the percentage of all normal and attack vectors that are correctly classified:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

- Detection Rate (True positive Rate (TPR) and True Negative rate (TNR)) refers to the percentage of total relevant results correctly classified by the classifier

$$TPR = \frac{TP}{TP + FN} \ \text{(attack vector)} \tag{4}$$

$$TNR = \frac{TN}{TN + FP} \ \text{(normal event)} \tag{5}$$

- Precision or Positive Predictive Value (PPV) refers to the percentage of the results which are relevant.

$$PPV = \frac{TP}{TP + FP} \ \text{(attack event)} \tag{6}$$

- F1 Score is simply the harmonic mean of precision and recall evaluating the outcome in balanced mode

$$F1\_score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{7}$$

- Miss Rate (FNR/FPR) is derived by subtracting the value of TPR from 1.

$$FPR = 1 - TNR \ \text{(attack)} \tag{8}$$

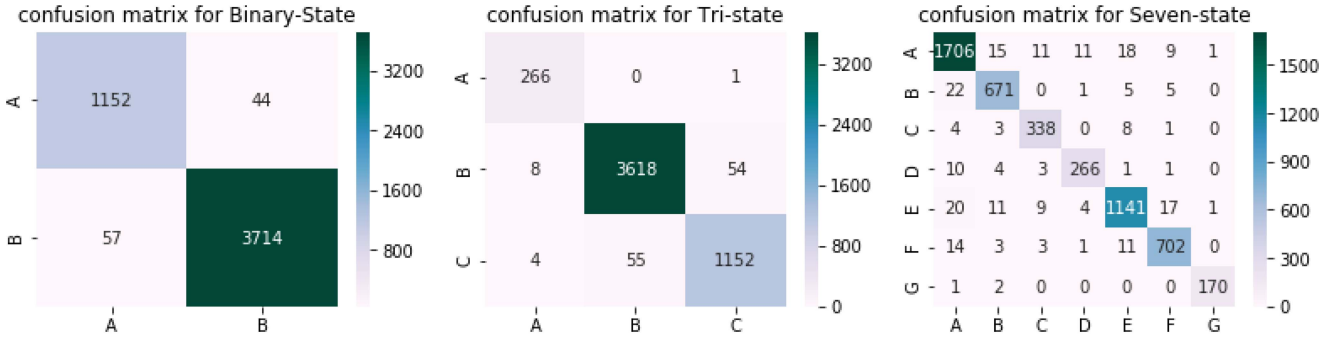
$$FNR = 1 - TPR \ \text{(normal)}\,. \tag{9}$$

B. Experimental results

Our target is to develop a model in such a way that it can be easily deployed in a real-time power grid. For that, the model should be fast and smart in identifying malicious events that occur in the network. Therefore, we target the most relevant features to classify normal and attack vectors. To compute the most promising features, we have used a WFI scoring model of Gradient Boosting feature selection. We have applied the GBFS approach on the binary dataset by considering multiple values of Num_trees = 100, 500, 700 and 1000 to identify the most common amongst all. From our observations, we conclude that mostly in each estimation the top 15 features remain the same.

We conclude, experimentally, that high accuracy values comply with a small learning rate, hence we decided to set the value of Num_trees = 1000 along with learning rate = 0.1. After computing 15 features of 15 sets of a binary dataset, we used the same features to compute the three-class, seven-class and multi-class dataset to detect the various attacks as all the four datasets of 15 sets have the same input measurement records - only the output label differs according to the category of the dataset. Table III represents the 15 features of 15 sets for all the four categories. Also, the primary goal of choosing a binary dataset to compute the promising features is to achieve faster execution speed and precise outcome in terms of detection rate as it only contains normal and attack vectors. Moreover, we have applied the same features to the rest of the three categories as basically all the categories are containing both malicious and normal events.

The datasets are well suited for ensemble classifiers since each set of the 15 datasets is produced at different attack locations by ORNL, and each consists of approximately 5500 records. The significance of the features depends on the location of the attacks on PMUs. Hence, the automatic stepwise feature selection is one of the crucial points for classification, which can be effectively handled by tree-based ensemble classifiers. Furthermore, for the proof of the concept, we have



(a) cf2 (b) cf3 (c) cf7

Fig. 6. Confusion matrices, 2,3,7 output labels.

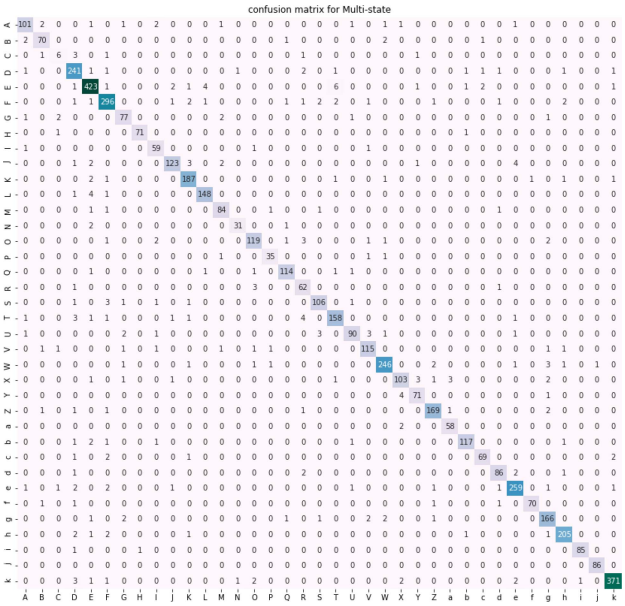


Fig. 7. Confusion Matrix for the 37 output labels.

evaluated the accuracy of other machine learning techniques such as Naive Bayes, Support Vector Machine (SVM), Simple Logistic Regression (SLR), One Rule (OneR), Decision Table (DT) and Artificial Neural Network (ANN) for all the four categories as mentioned in Table III. We focus on tree-based ensemble classifiers since they give the best accuracy.

To evaluate the efficiency of the top 15 features in terms of detection rate and execution speed, we have applied various classifiers on all the 15 datasets of four categories. As the computed features are generated using the GBFS technique, we specifically target decision tree classifiers with a combination of boosting approaches such as GB, XGBoost, Random Forest(RF), AdaBoost Random Forest(AdaBoost-RF), ClassificationViaRegration- Random Forest(CVR-RF), Random Tree, AdaBoost Random Tree and J48.

The proposed framework is programmed using Python on a Jupyter Notebook (Anaconda distribution) on Windows 10 with Intel Core i5-8300H 2.30GHz processor, 16 GB RAM and Nvidia Geforce GTX 1060 6go GPU. The results of classification of various classifiers are also validated using a WEKA platform [33]. The experiments are computed using

TABLE IV
COMPARATIVE ANALYSIS (ACCURACY) OF VARIOUS MACHINE LEARNING TECHNIQUES

Classifiers	Naïve Bayes	SVM	SL	OneR	DT	ANN
Binary	54.17	70.2	78.04	81.87	89.86	88.34
Three-State	50.62	69.46	69.85	75.02	81.27	80.43
Seven-State	20.04	37.45	42.55	57.12	80.32	61.97
Multi-State	11.59	20.81	32.76	40.22	73.72	61.13

random samples of 100,000 normal and attack observations for each of the four categories divided into 15 sets. The training and testing set of the model is obtained using 10-fold cross-validation methodology to measure the accuracy without biasing the normal or malicious output classes.

To assess the performance of each classifier, we have computed the following performance metrics: accuracy, detection rate, false-positive rate, F1 score and execution speed of 15 datasets of all the four categories. The results of performance metrics are derived from the confusion matrix during each classification. Figure 6 represents the example of one of the best confusion matrix of binary, three-class and seven-class classifier, respectively. Similarly, Figure 7 depicts the most promising confusion matrix of the multi-class classifier which can differentiate the total of 37 various attacks and normal events. By analyzing the confusion matrix, we can differentiate normal and attack vector in terms of True Positive, True Negative, False Positive and False Negative.

C. Result Discussion

The purpose of the proposed GBFS based feature selection framework is to generate a subset of the given attributes from entire dataset using a WFI metric to reduce the noise and improve the performance of the classifier. The derived subset of the top 15 features may or may not contribute same in the decision-tree classifiers. We have observed the results of total 8 decision tree-based machine learning techniques to validate our proposed methodology via multiple simulation trials. Overall 60 computations are performed to evaluate the performance of each classifier to include the results of fifteen datasets of all the four categories. Figure 7 represents the comparative analysis of the accuracy of eight decision

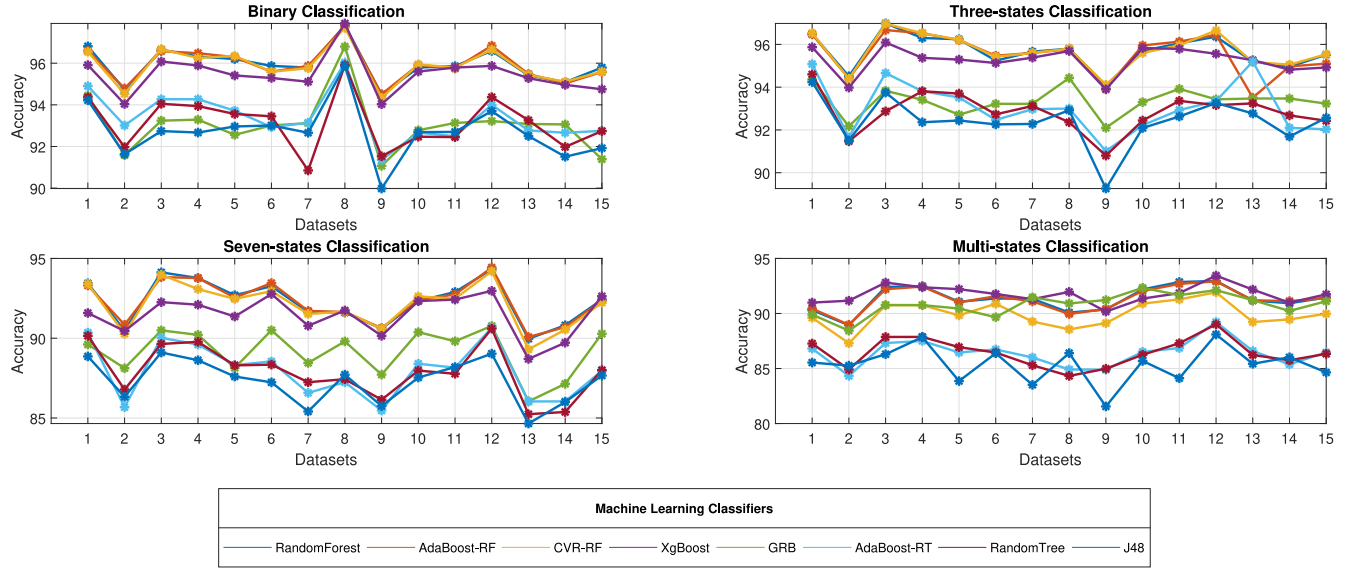


Fig. 8. Comparative view of Different Machine Learning Classifiers for - four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets.

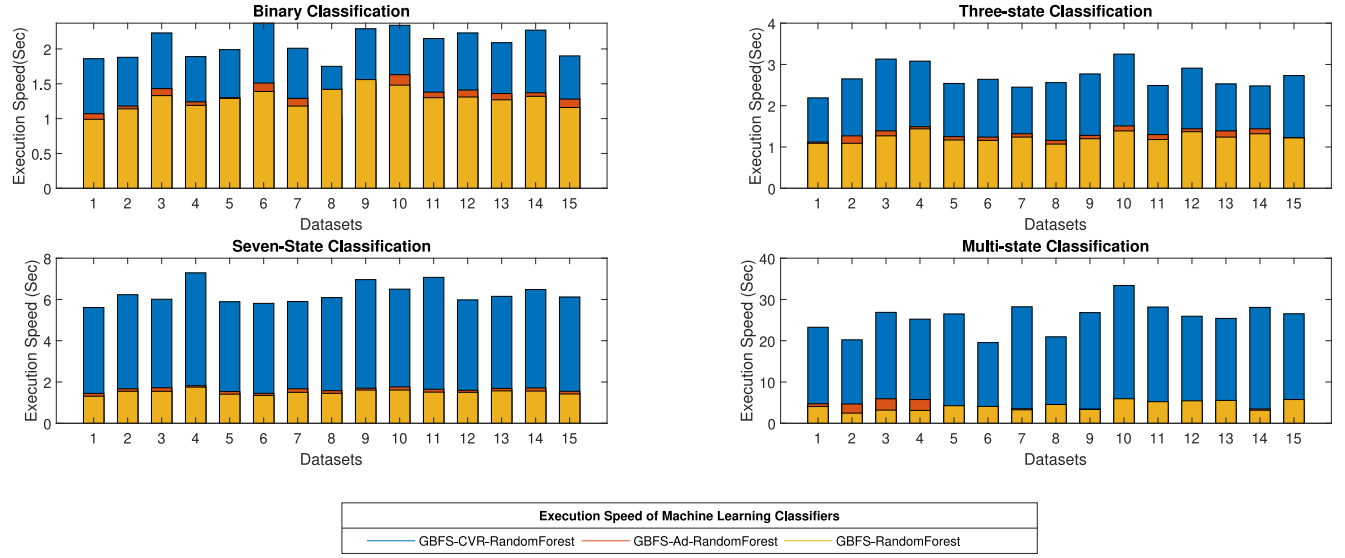


Fig. 9. Comparative view of Execution speed of Three GBFS based Random Forest variances to classify normal and attack events for four categories (binary, three-state, seven-state and multi-state) for each of 15 datasets.

tree-based classifiers of 15 datasets of each binary, three-class, seven-class and multiclass categories.

Amongst all the eight classifiers, it was observed that XGBoost, random forest and its variance have proven to be most efficient. However, other tree-based classifiers also proved their efficiency ranging between 92 to 94 for Binary and three-state and 85 to 90 for seven class and multiclass. XGBoost comes up with accuracy equal to 97.96, 96.09, 92.97, 92.44 for binary, three-class, seven class and multiclass datasets, respectively. Similarly, all three variants of Random Forest also achieve very high accuracy such as 97.82, 97.78 and 97.70 for binary, 97.18, 97.18 and 97.01 for three-class, 94.43, 94.31 and 94.19 for seven class and 92.96, 92.92, 91.92 for multiclass, respectively. Since the GBFS-Random Forest and its variances are the most efficient classifiers to classify the normal and attack vectors with nearly same range of accuracy, we have compared the execution speed of all the three classifiers to identify the best among them. As depicted in Figure 9, GBFS-Random Forest classified the various attack and normal events for all the four categories in 1.5 seconds. GBFS-AdaBoost Random Forest took slightly more time than the GBFS-RF. GBFS-CVR-Random Forest took comparatively higher execution time as it uses the combined approach of boosting and ensemble of trees for the classification. However, by comparing the accuracy levels, we observe that the boosting does not much improve the result much, in such case GBFS-RF is proven to be best amongst all three with high accuracy and less execution time.

TABLE V
PERFORMANCE EVALUATION METRICS OF PROPOSED GBFS BASED CLASSIFIER

Measure	Binary	Three-class	Seven-Class	Multi-class
Accuracy	97.96%	96.80%	94.42%	92.96%
FPR	0.037	0.067	0.019	0.003
Precision	0.9705	0.9887	0.9504	0.9250
Recall	0.9740	0.9676	0.9355	0.9240
F-Measure	0.9723	0.9781	0.9427	0.9244

TABLE VI
COMPARATIVE ANALYSIS OF OVERALL PERFORMANCE OF VARIOUS TECHNIQUES AND PROPOSED GBFS BASED CLASSIFIER

Classifier	**Data Cleaning**	**Feature Selection**	**Features (%)**	**Classes**	**Accuracy**
ADA-JRIP [25]	NA	NA	100%	2	94.55%
				3	94.61%
				37	85.85%
CPM [34],[28]	Applied	NA	100%	7	93.00%
				25	90.40%
EMCT[32]	Applied	PCC	25%	2	70.60%
			50%		76.3%
			75%		83.5%
			100%		90.2%
GMMKM[35]	Applied	PCC	25%	2	94.56%
			50%		95.83%
			75%		96.82%
			100%		97.27%
Tree Based	Applied	GBFS	**12%**	**2**	**97.66%**
				3	**96.50%**
				7	**94.12%**
				37	**92.46%**

We demonstrated that the 15 stochastic features shown in Table III were the most promising features for all the decision tree-based classifiers by iteratively running all the eight classifiers, for 15 datasets of all the four categories. In each iteration, using 15 features, we retrained & re-tested all the eight decision-tree based models to compute the general average trend of malicious and normal events by observing DR, FPR and execution Time.

All the selected classifiers maintain very high DR and lower FPR rate in all the computations as shown in Table V. We have achieved 98.5% of detection rate which truly differentiates attack and normal vectors with only 3.7% and 6.7% of false positive rate for binary and three class classification. Moreover, seven-class and multi-class classifiers have also outperformed as they gave around 94.42% and 92.5% for the detection rate.

This validates the significance of our proposed methodology for feature selection. Real-time systems such as control and monitoring systems of industrial infrastructures/power grids need a methodology of feature extraction where processing time and storage space are always crucial.

To validate the efficiency of the proposed methodology, we have compared GBFS based decision tree algorithm with four published methods, namely AdaBoost-JRIP (AdaJRIP) [25], Common Path Mining [28], [34], Expectation Maximization Clustering Technique (EMCT) [32] and Gaussian Mixture–Kalam Filter Model (GMM-KF) using Pearson Correlation Coefficient (PCC) feature selection method [35], by considering various performance evaluation factors such as whether proper pre-processing is applied or not; to accelerate the process, whether feature selection approach is incorporated or not and if applied how many features are selected to evaluate the accuracy for various output classes.

It can be seen from Table VI that our proposed framework outperforms compared to those of the published techniques and accomplishes the highest accuracy with the 97.66%, 96.50% , 94.12% , 92.46% with only 12% of the features for all the four categories of the power system datasets. Note that the results mentioned in the table refer to the highest accuracy achieved during the classification of the attacks and normal events by various tree based classifiers.

TABLE VII
COMPARATIVE ANALYSIS OF VARIOUS FEATURE SELECTION METHODS

Feature Selection Method	Classifier	Features	Class	Accuracy	Execution Time (sec)
Chi-Square	RF	106	2	96.69	4.6
PCA	RF	27	2	92.57	4.3
GBFS	RF	15	2	97.66	1.2

Moreover, in order to show the efficiency, we have compared our proposed scheme with two well-known feature selection methods, namely, Chi-Square and Principal Component Analysis (PCA), in terms of the number of features, accuracy and execution time for a binary class using Random Forest (RF) classifier as shown in Table VII.

As mentioned earlier, data cleansing was performed to accelerate the process of classification using various machine learning algorithms. However, the technique in [25] has obtained comparatively low results with various well-known machine-learning algorithms such as OneR, SVM, Random Forest, Naive Bayes, JRIP and AdaBoost-JRIP owing to disregarding preprocessing before applying the classification approach on the power system dataset. As per our observations, the given dataset needs to be refined by removing infinite values before mapping and scaling the records. The features R1:PA:Z, R2:PA:Z, R3:PA:Z, R4:PA:Z, represent apparent impedance of the relay associated with IEDs of the given power system dataset comprising of infinite values and should be removed. However, in our proposed methodology, the top 15 features of any of the sets does not rely on impedance of relay attribute such as R1:PA:Z, R2:PA:Z, R3:PA:Z and R4:PA:Z. Hence we are essentially not deleting any row records of the given dataset.

Proper sanitization converts the type of the features from nominal to numeric which makes a huge impact in taking decision to classify the events of the given dataset by various classifiers. To demonstrate the impact of preprocessing and feature selection we have computed the results with and without preprocessing and with and without feature selection by applying all the eight decision-tree based classifiers on the power system dataset as mentioned in Table VIII.

The first two columns represent the accuracy and execution speed computed by eight decision-tree based classifiers without applying pre-processing on the dataset. In this case, all the classifiers have failed to achieve high accuracy and better execution speed because in order to predict the outcome,

TABLE VIII
COMPARATIVE ANALYSIS OF VARIOUS TREE-BASED CLASSIFIERS BASED ON PRE-PROCESSING AND FEATURE SELECTION METHODOLOGY

Algorithm	Cl	W/o Pre-Proc		W Pre-Proc		15 features	
		Acc (%)	Ex.sp Sec	Acc (%)	Ex.sp Sec	Acc (%)	Ex.sp Sec
XgBoost	2	71.14	8.14	96.23	3.46	97.91	1.98
GB	2	70.21	7.29	95.68	3.34	96.80	1.16
RF	3	72.66	8.53	96.01	5.35	97.18	1.47
ADA-RF	3	73.16	8.94	96.68	6.22	97.18	2.14
CVR-RF	7	59.22	9.98	93.20	8.36	94.42	5.14
J48	7	36.81	1.21	87.34	0.90	89.10	0.30
RT	37	27.73	0.35	88.25	0.11	90.01	0.04
ADA-RT	37	27.22	0.57	89.12	0.12	90.22	0.06

the classifier applies the modeling algorithm on both numerical and categorical inputs. At each iteration the decision-tree makes the decision by considering both the type of data in the dataset,that results in a long prediction time and low accuracy rate. Hence, proper sanitizing is the primary step for the classification.

In contrast, the third and forth columns of the table represents the results computed by the eight classifiers by applying proper pre-processing on the entire dataset of 128 features. The pre-processing includes feature mapping, feature normalization and feature encoding techniques which improves the accuracy and execution speed.

Finally, we have combined pre-processing with feature selection to select the fifteen most promising features from the dataset before applying the classifier, which not only improves the accuracy but also improves the execution time. In a nutshell, our approach combines both pre-processing and feature selection, which has proven best amongst all the three approaches for all the decision-tree based classifiers.

VII. CONCLUSION AND FUTURE WORK

This article presented a GBFS based feature selection approach to identify the most promising features for anomaly detection in power grids. The overall framework consists of three key components. Initially, during data preprocessing, the features are mapped and scaled to a specific range. To accelerate the execution speed and learning efficiency, a GBFS based feature selection approach is applied on filtered data to compute the most promising features from the entire dataset dynamically according to network/SCADA traffic. The dynamic approach of selecting the features from the entire dataset hides largely all the sensitive information of the power grid system. Finally, these reconstructed datasets are used by decision-tree based algorithms that classify the various attacks and normal events. The experimental results reveal the efficiency of the framework in terms of accuracy, detection rate, miss rate and execution speed compared to the original dataset. Moreover, the proposed GBFS based model outperforms some state-of-the-art techniques described in published works.

In the future, we plan to extend this work by combining the results of several classifiers to achieve an accurate outcome by applying majority vote ensemble method. This method predicts the output label based on the majority of the output labels predicted by each classifier. This will further improve the efficiency of the prediction and provides the most accurate output label in terms of normal and attack events. We will target various classifiers, namely, Random Forest, Gradient Boosting, XGBoost, Artificial Neural Network, Naïve Base, and Decision Table for ensemble learning by referring to preliminary results from this article. This approach will help to generate a better predicting model compared to a single model using "hard voting" based majority rule ensemble technique.

REFERENCES

[1] K. Poulsen. (2003). *Feature Importance of Feature Selection*. Accesed: Oct. 9. 2019. [Online]. Available: https://www.securityfocus.com/news/6767

[2] B. Krebs, *Cyber Incident Blamed for Nuclear Power Plant Shutdown*, Washington Post, Washington, DC, USA, 2008. Accessed: Oct. 4, 2020. [Online]. Available: https://www.waterfall-security.com/wpcontent/uploads/2009/11/CyberIncidentBlamedForNuclear PowerPlantShutdownJune08.pdf

[3] B. Kesler, "The vulnerability of nuclear facilities to cyber attack," *Strategic Insights*, vol. 10, no. 1, pp. 15–25, 2011.

[4] *SANS and Electricity Information Sharing and Analysis Center (E-ISAC). Analysis of the Cyber Attack on the Ukrainian Power Grid*. Accesed: Sep. 28, 2019. [Online]. Available: http://www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[5] N. Kshetri and J. Voas, "Hacking power grids: A current problem," *Computer*, vol. 50, no. 12, pp. 91–95, Dec. 2017.

[6] D. Upadhyay and S. Sampalli, "SCADA (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations," *Comput. Security*, vol. 89, Oct. 2020, Art. no. 101666.

[7] S. Nazir, S. Patel, and D. Patel, "Assessing and augmenting SCADA cyber security: A survey of techniques," *Comput. Security*, vol. 70, pp. 436–454, Jul. 2017.

[8] B. Barrett. *Security News This Week: An Unprecedented Cyberattack Hit U.S. Power Utilities*. Accesed: Nov. 11, 2019. [Online]. Available: www.wired.com/story/power-grid-cyberattack-facebook-phone-numbers-security-news/

[9] A.-S. K. Pathan, *The State of the Art in Intrusion Prevention and Detection*. Boston, MA, USA: Auerbach, 2014.

[10] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *Int. J. Elect. Power Energy Syst.*, vol. 99, pp. 45–56, Jul. 2018.

[11] C.-F. Tsai *et al.*, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[12] Z. Xu *et al.*, "Gradient boosted feature selection," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, 2014, pp. 522–531. [Online]. Available: http://doi.acm.org/10.1145/2623330.2623635

[13] X. Lin, X. Zhang, and X. Xu, "Efficient classification of hot spots and hub protein interfaces by recursive feature elimination and gradient boosting," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 56, pp. 1525–1534, Sep./Oct. 2020.

[14] Machinelearning. (2019). *Feature Importance of Feature Selection*. Accesed: Apr. 19, 2019. [Online]. Available: https://machinelearningmastery.com/feature-importance-and-feature-selection-with-xgboost-in-python/

[15] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[16] E. D. Knapp and R. Samani, *Applied Cyber Security and the Smart Grid: Implementing Security Controls into the Modern Power Infrastructure*, 1st ed. London, U.K.: Syngress, 2013.

[17] R. Spenneberg, M. Brüggemann, and H. Schwartke. (2006). *PLC-blaster: A Worm Living Solely in the PLC*. [Online]. Available: https://www.blackhat.com/docs/asia-16/materials/asia-16-Spenneberg-PLC-Blaster-A-Worm-Living-Solely-In-The-PLC-wp.pdf

[18] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th ed. Upper Saddle River, NJ, USA: Prentice-Hall, 2013.

[19] S. J. Mousavirad *et al.*, "High-dimensional multi-level maximum variance threshold selection for image segmentation: A benchmark of recent population-based metaheuristic algorithms," in *Proc. Genet. Evol. Comput. Conf. Companion (GECCO)*, 2020, pp. 1608–1613. [Online]. Available: https://doi.org/10.1145/3377929.3398143

[20] Z. Zhu, Y.-S. Ong, and M. Dash, "Wrapper–filter feature selection algorithm using a memetic framework," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 37, no. 1, pp. 70–76, Jan. 2007.
[21] P. M. Granitto *et al.*, "Recursive feature elimination with random forest for PTR-MS analysis of agroindustrial products," *Chemometr. Intell. Lab. Syst.*, vol. 83, no. 2, pp. 83–90, 2006.
[22] F. Pan *et al.*, "Feature selection for ranking using boosted trees," in *Proc. 18th ACM Conf. Inf. Knowl. Manag. (CIKM)*, 2009, pp. 2025–2028. [Online]. Available: https://doi.org/10.1145/1645953.1646292
[23] S. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 667–671, 2005.
[24] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Disc. Data Min. (KDD)*, New York, NY, USA, 2016, pp. 785–794.
[25] R. C. B. Hink *et al.*, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, pp. 1–8.
[26] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, pp. 1189–1232, Mar. 2000.
[27] T. H. Morris, Z. Thornton, and I. P. Turnipseed, "Industrial control system simulation and data logging for intrusion detection system research," in *Proc. 7th Annu. Southeastern Cyber Security Summit*, Jun. 2015, pp. 179–186.
[28] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.
[29] U. Adhikari *et al. Industrial Control System (ICS) Cyber Attack Datasets Datasets Used in the Experimentation*. Accessed: Oct. 4, 2020. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets
[30] J. M. Beaver, R. C. Borges-Hink, and M. A. Buckner, "An evaluation of machine learning methods to detect malicious scada communications," in *Proc. 12th Int. Conf. Mach. Learn. Appl.*, vol. 2, Dec. 2013, pp. 54–59.
[31] R. Punmiya and S. Choe, "Energy theft detection using gradient boosting theft detector with feature engineering-based preprocessing," *IEEE Trans. Smart Grid*, vol. 10, no. 2, pp. 2326–2329, Mar. 2019.
[32] M. Keshk *et al.*, "Privacy preservation intrusion detection technique for scada systems," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2017, pp. 1–6.
[33] I. H. Witten *et al.*, *Data Mining, Fourth Edition: Practical Machine Learning Tools and Techniques*, 4th ed. San Francisco, CA, USA: Morgan Kaufmann, 2016.
[34] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Trans. Ind. Informat.*, vol. 11, no. 3, pp. 650–662, Jun. 2015.
[35] M. Keshk *et al.*, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, early access, Mar. 25, 2019, doi: 10.1109/TSUSC.2019.2906657.

Darshana Upadhyay received the master's degree in computer science from Nirma University, Ahmedabad, India. She is currently pursuing the Ph.D. degree with the Faculty of Computer Science, Dalhousie University, Canada. She also served as a Lecturer with Nirma University. For her master's thesis, she has completed a novel project in the area of linear-feedback shift register design for secure systems. Her primary research includes algorithm conceptualization, hardware design in the field of embedded systems, vulnerability assessments, and intrusion detection techniques for IoT/SCADA-based systems. She has awarded the Gold Medal for securing the first position during her graduate study. She is the co-recipient of the Indo-Canadian Shastri Research Grant in the field of wireless security and intrusion detection systems. She has been invited to be one of the Women in International Security-Canada's 2020 Emerging Thought Leaders.

Jaume Manero received the Ph.D. degree in artificial intelligence from the Technical University of Catalonia, Barcelona, Spain. He is working with the Barcelona Supercomputing Center and his Ph.D. dissertation was "Deep learning architectures applied to wind time-series forecasting." He is a Visiting Research Scientist with Dr. Srini Sampalli's MYTech Lab (Emerging Wireless Technologies) where he is working on how deep learning can impact in the development of cyber-security applications.

Marzia Zaman received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Memorial University of Newfoundland, Canada, in 1993 and 1996, respectively. She started her career with Nortel Networks, Ottawa, Canada, in 1996, where she joined the Software Engineering Analysis Lab and later joined the Optera Packet Core project as a Software Developer. In addition, she has many years of industry experience as a Researcher and a Software Designer with Accelight Networks, Excelocity, Sanstream Technology, and Cistel Technology. Since 2009, she has been working closely with the Centre for Energy and Power Electronics Research, Queen's University, Canada, and one of its industry collaborators, Cistel Technology, on multiple power engineering projects. Her research interests include renewable energy, wireless communication, IoT, cyber security, machine learning, and software engineering.

Srinivas Sampalli (Member, IEEE) received the Bachelor of Engineering degree from Bangalore University and the Ph.D. degree from the Indian Institute of Science, Bengaluru, India. He is currently a Professor and a National 3M Teaching Fellow with the Faculty of Computer Science, Dalhousie University. He has led numerous industry-driven research projects on Internet of Things, wireless security, vulnerability analysis, intrusion detection and prevention, and applications of emerging wireless technologies in healthcare. He currently oversees and runs the Emerging Wireless Technologies (MYTech) Lab and has supervised over 150 graduate students in his career. His primary joy is in inspiring and motivating students with his enthusiastic teaching. He has received the Dalhousie Faculty of Science Teaching Excellence Award, the Dalhousie Alumni Association Teaching Award, the Association of Atlantic Universities' Distinguished Teacher Award, a Teaching Award instituted in his name by the students within his Faculty, and the National 3M Teaching Fellowship, Canada's most prestigious teaching acknowledgment. Since September 2016, he holds the honorary position of the Vice President (Canada) of the International Federation of National Teaching Fellows, Canada, a consortium of national teaching award winners from around the world.

Intrusion Detection in SCADA Based Power Grids: Recursive Feature Elimination Model With Majority Vote Ensemble Algorithm

Darshana Upadhyay, Jaume Manero, Marzia Zaman, and Srinivas Sampalli, *Member, IEEE*

***Abstract*—We propose an integrated framework for an intrusion detection system for SCADA (Supervisory Control and Data Acquisition)-based power grids. Our scheme combines RFE-XGBoost (Recursive Feature Elimination-eXtreme Gradient Boosting) based feature selection with a majority vote ensemble method. RFE selects features recursively based on Weighted Feature Importance (WFI) scores during the training process, while the majority vote ensemble method predicts the output label based on a total of nine heterogeneous classifiers - three bagging ensembles, namely, Random Forest (RF), Extra Tree (ET), and Decision Tree (DT), three boosting ensembles, namely, XGBoost (XGB), Gradient Boosting (GB), and AdaBoost-Decision Tree (AdB-DT) along with artificial neural network (ANN), Naive Bayes (NB), and k-nearest neighbors (KNN). This leads to a more accurate solution as a result of the combination of the most useful features and prediction from multiple heterogeneous classifiers. Experimental results show that our approach increases the accuracy, precision, recall, F1 score, and decreases the miss rate as compared to previous approaches. The model is also evaluated for four different class categories, namely binary, three-class, seven class and multi-class, using Precision Recall (PR) and Receiver Operating Characteristic (ROC) plot. In addition, an end-to-end IDS framework is proposed for efficient and accurate detection of intrusions.**

***Index Terms*—SCADA systems, power grids, recursive feature elimination, majority vote, ensemble method, feature selection, cyber security, network intrusions.**

I. Introduction

POWER grids are the underlying infrastructure that support our economies and daily lives by providing and sustaining a continuous supply of electricity. They play a fundamental role in connecting industries and homes with far away locations from where the power is originally generated. Furthermore, they assure the quality of the electricity supply at the point of consumption. In the past, power grids were isolated systems. The field devices of such systems were managed locally on the plant floor. However, as technology advanced, energy system devices were gradually monitored and controlled remotely. Currently, SCADA (Supervisory Control and Data Acquisition) systems play a vital role in the management of power grid components efficiently.

Current power grids comprise of multiple substations and control centers and widely spread in large geographical areas. Each substation consists of various components such as power lines, transformers, sensors, actuators, and phasor measurement units (PMUs), along with supervisory control and data acquisition (SCADA) elements for monitoring the system components remotely. Fig. 1 shows the block diagram of a SCADA architecture for power systems. A SCADA network segment typically includes a SCADA master, HMI (Human Machine Interface), and data historian placed at the control center, communication links, and various field control devices such as Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and IEDs (Intelligent Eletronic Devices). The sensors and actuators located at power grids frequently supply digital status information to the field control devices. These devices further communicate this information to MTU, where the server will process the data according to acceptable parameter ranges. This information will then be transmitted back to field control device to improve the performance and to avoid hazards. The SCADA master also stores the status information on the data historian and displays it on the HMI for centralized control and monitoring of the power grids.

However, this evolution has connected power systems to the Internet, which, in turn, can expose them to various cyber-attacks such as False Data Injection (FDI) attacks, Denial of Service (DoS), or Man-In-the-Middle (MIM) attacks [1], [2]. FDI manipulates the energy measurement parameters, either by identifying the backdoors that bypass the system or by using privileges of authorized personnel [3]. The cyber attack against the Ukrainian power plant in 2015 is one example of such attacks in which nearly 250,000 people were left without electricity for many hours [4]. Another example is the attack on the Davis-Besse nuclear power plant in Oak Harbor, USA [5] which was infected by the SQL Slammer worm. The worm infected the entire power system with a DoS attack launched

Manuscript received January 26, 2021; revised May 29, 2021; accepted July 15, 2021. Date of publication July 26, 2021; date of current version September 16, 2021. This work was supported by the Natural Sciences and Engineering Research Council (NSERC), Canada through a Collaborative Research Grant. Recommended for acceptance by Dr. Fan Wu. *(Corresponding author: Srinivas Sampalli.)*

Darshana Upadhyay and Srinivas Sampalli are with the Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 4R2, Canada (e-mail: darshana@dal.ca; srini@cs.dal.ca).

Jaume Manero is with the Technical University of Catalonia, 08034 Barcelona, Spain, and also with the Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 4R2, Canada (e-mail: jaume.manero@dal.ca).

Marzia Zaman is with the Research & Development Department, Cistel Technology Inc., Ottawa, ON K2E 7V7, Canada (e-mail: marzia@cistel.com).

Digital Object Identifier 10.1109/TNSE.2021.3099371

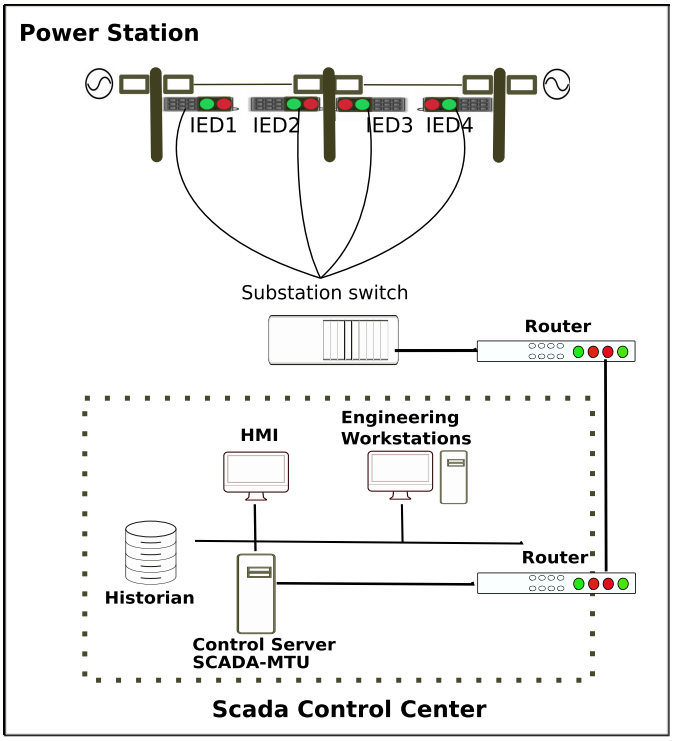


Fig. 1. SCADA System Architecture for Power System. Legend: HMI: Human Machine Interface, IEDs: Intelligent Electronic Devices, MTU: Master Terminal Unit.

by exploiting the vulnerabilities of the SCADA system. Such attacks on a nation's power grid can lead to catastrophic consequences [6].

Power grids face significant challenges pertaining to the security and privacy of the data. One of the challenges in securing power systems is the deployment of safeguards and the management of the network because of legacy-inherited security weaknesses and limitations. Although many security controls including defense in depth architecture, access control, authentication mechanisms, confidentiality, integrity techniques, and firewalls have been developed to protect critical infrastructure, the rapid evolution of hacking techniques can easily expose the integrity of the system's data and devices [7]. For example, in March 2019, the operators at a power grid center in the US lost communication with multiple sites of power generators due to a known firewall vulnerability [8].

Researchers have proposed intrusion detection techniques to secure SCADA based power grids. Hink *et al.* provide a comparative analysis of various machine learning techniques using a power grids dataset and identify Adaboost-JRIP as one of the best classifiers [9]. However, the authors do not filter and reduce the dimension of the dataset. Hence, they are unable to achieve good accuracy and execution speed. Pan *et al.* have focused on hybrid IDS using data mining, where they have used common path mining to identify the location of attacks [10], [11]. Further, in [12], the authors apply Pearson Correlation Coefficient (PCC) for feature selection and extract 75% of features. They use an Expectation Maximization Clustering Technique (EMCT) to classify the events. Using this approach, they improve the execution speed but do not achieve better accuracy for a multi-class dataset. Moreover, this technique is enhanced by combining PCC with the Gaussian Mixture - Kalman Filter Model (GMM-KF) in [14]. The authors are able to reduce the percentage of the features to 25 and achieve good accuracy and execution speed. However, this experiment is limited to a binary dataset. Moustafa *et. al.* [13] have used ICA - Independent Component Analysis feature selection and Beta Mixture Hidden Markov (BMHM) classification model. The authors have obtained promising results in regards to accuracy. However, they have worked on a subset of the features, and hence we could not identify the exact number of features used in this paper. We have recently proposed WFI based GBFS model for feature selection and extracted 12% of the most promising features in [15]. Our target was to achieve high execution speed and a better predictive model for real-time SCADA communication. The proposed GBFS model has been further verified with different machine learning algorithms. We have identified that the proposed solution is suitable for tree-based classifiers. Note that all these experimental studies use the power grid dataset created by Oak Ridge National Laboratories (ORNL). Table I summarizes the literature on IDSs for power grids.

In our earlier work [15], we have proposed a computationally efficient intrusion detection framework for power grids, which not only improves the computational cost but also provides privacy preservation. In that approach, we have determined the most significant features using a Weighted Feature Importance (WFI) based gradient boosting scoring model [15]. Furthermore, we have applied eight tree-based algorithms on multilevel multiple datasets to classify various attacks and normal events to validate the efficiency of derived features [15]. In particular, the most promising features were detected by considering multiple values of number of trees while training the model to apply the WFI scoring concept. From our preliminary results, we have identified three bagging ensembles, namely, Random Forest (RF), Extra Tree (ET), and Decision Tree (DT), three boosting ensembles XGBoost (XGB), Gradient Boosting (GB) and AdaBoost-Decision Tree (AdB-DT) as the most promising classifiers. Moreover, we have identified the accuracy of other machine learning classifiers such as artificial neural network (ANN), Naive Bayes (NB), and k-nearest neighbors (KNN).

In this paper, we have enhanced the feature selection and classification module. The feature selection approach is extended by incorporating Recursive Feature Elimination (RFE) method. In that, the GBFS model is improved by replacing the gradient boosting with XGBoost as we found XGBoost is the most promising classifier amongst all the tree-based classifiers in our previous work. Hence, XGBoost can be a better fit to score the features using the WFI technique while training the dataset. Moreover, we have replaced the concept of evaluating number of trees while training the model with RFE approach. This approach helps us achieve a better predictive model by searching all the stable features instead of the most promising features while constructing the tree.

Another enhancement has been applied to the classification model by using a majority vote based ensemble method consisting of six tree-based classifiers along with artificial neural network (ANN), Naive Bayes (NB), and k-nearest neighbors

TABLE I
LITERATURE REVIEW OF PUBLISHED INTRUSION DETECTION SYSTEMS FOR POWER GRIDS

Attributes	Machine Learning for Power System Disturbance and Cyber-attack Discrimination [9]	Developing a Hybrid Intrusion Detection System Using Data Mining for Power Systems [10], [11]	Machine Learning for Power System Disturbance and Cyber-attack Discrimination [12]	A New Threat Intelligence Scheme for Safeguarding Industry 4.0 Systems [13]	An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems [14]	Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids [15]
Feature Section Method	Not Applied	Not Applied	PCC (Pearson's correlation coefficient)	ICA (Independent component analysis)	PCC	GBFS
Features (%)	100%	100%	75%	compared various subset of features	25%	12%
Classification Technique	Adaboost - JRIP	Common Path Mining	Expectation Maximization Clustering Algo	Beta mixture-hidden Markov models (MHMMs)	Gaussian Mixture-Kalman Filter Model (GMM-KF)	Tree-Based (XGBoost)
Algorithm used	Machine Learning	Data mining and pattern recognition	Maximum likelihood estimation	Statistical common estimation method	Bayesian filtering algorithm	Machine Learning
Measure used	Accuracy	Accuracy	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure	Accuracy, Precision, Recall, F-measure, Training time
Dataset used	Oak Ridge National Laboratories (ORNL) - power grid dataset [16]					
Pros & Cons	Data are not pre-processed properly, Low accuracy and execution speed	Improvement in identification of attacks, provide location of attack.Moderate accuracy and execution speed	Improved execution speed but compromising in accuracy with multi-class datasets	Improved accuracy, no observations regarding execution speed	Improvement in accuracy, Tested for binary classification, No multi attack vectors classification	Significantly improved accuracy and execution speed

(KNN). We have replaced the single tree base classifier with a majority vote based ensemble method. The selection of various heterogeneous classifiers is based on our preliminary results [15]. This approach will determine the output label based on the majority of the class labels predicted by all the nine classifiers. The selected nine algorithms in this model work on different analogies, such as tree based, naive based, lazy learner, and neural networks based prediction. Consequently, the output label is calculated using a majority of heterogeneous predictions, which turned into a robust predictive model.

The concept of voting is used to average the output values based on the prediction of different classifiers. This process produces relatively uncorrelated output predictions of various classifiers which significantly reduces the error rate. Moreover, if output labels are highly correlated, in that case also this approach can easily detect a minor error. Furthermore, decision tree-based classifiers are good candidates for this approach as small perturbations generate totally different structures and splits. Hence combining prediction of such models using majority vote significantly improves the efficiency of the classification process. However, the execution speed and training time of this model could be higher than the single classifier. Hence, we have suggested an end to end machine learning based Intrusion Detection System framework for power grid SCADA security which utilize both the models as depicted in Figure 9 and described in Section VII. The objective and major contributions of this work are listed below.

Objective: The aim of this work is to propose a robust intrusion detection system for power grids which is compatible with time critical systems and has the capability to detect intrusions accurately using effective features of the network traffic. Moreover, the proposed model is a good fit for the control center to serve large-scale SCADA systems.

Contributions:

1) We use RFE-XGBoost based weighted feature importance scoring model to identify the most promising features. RFE selects the features recursively based on the weighted importance score of each feature by comparing a previously trained model with the current model. Through this approach, the most stable features of the dataset are determined which will be useful to achieve a better predictive model.
2) We derive 30 most promising features out of 128 features of the binary class, which significantly reduces the dimension of the dataset. Furthermore, the same features are used to the rest of the three categories, namely, three class, seven class, and multi-class to train the model to evaluate the efficiency of the RFE based feature selection model.
3) For the performance improvement, we apply the majority vote ensemble algorithm by considering nine heterogeneous classifiers to predict the output based on the majority of the class labels predicted by each of these nine classifiers.
4) We propose a deployment model of the IDS in SCADA-based power grids which reflects real-time traffic

monitoring by introducing placement of IDS models at the different locations, namely, plant floor, and control center.

For performance assessment and validation, we compare the accuracy of a total of nine classifiers along with the majority vote ensemble classifier. Moreover, we examine one of the classifiers of each method of bagging, boosting, and voting ensembles in terms of Precision-Recall (PR) and Receiver Operating Characteristic (ROC) plot. To validate the efficiency of the selected features and majority vote classifier, we evaluate the various performance metrics, namely, precision, recall, F1 score and miss rate of our proposed scheme. We also compare the accuracy of the majority vote ensemble method with existing bagging and boosting based ensemble techniques. Further, we compare the accuracy of the proposed methodology with published state-of-the-art techniques.

The rest of the paper is structured as follows. Section II describes the background and related work in the area of power grid security. The proposed intrusion detection framework and process diagram are described in Section III. Section IV covers algorithm conceptualisation and mathematical proof of RFE based feature selection and the majority vote ensemble method. Section V describes the experimental results and discussions. The proposed placement of IDS framework in SCADA based power grids is described in Section VI. Concluding remarks are provided in Section VII.

II. Background and Related Work

A. Power Grid Intrusion Detection Systems

The development of power grids has motivated researchers to propose various types of intrusion detection techniques to ensure security [17]. Generally, an IDS can be classified into two categories, namely, host-based and network-based. Host-based IDSs monitor the hosts on a network by collecting and monitoring various event logs of targeted devices. For example, a host based IDS for SCADA systems focuses explicitly on securing components such as RTUs and IEDs [12]. The IDS is responsible for identifying attacks against an IED of the substation by recording sequential events [18]. Network-based IDSs monitor the entire network traffic to detect malicious activities. This type of IDS can be further categorized into rule-based and anomaly-based IDS [19]. Rule-based IDSs are used in SCADA power grids for in-depth protocol analysis. This model works on the signature-based approach for pattern matching to analyze the input data for malicious packets [20]. In this approach, the signature of every incoming packet is compared with all the stored signatures to detect the threats. However, this approach works mainly for known threats but is unable to detect zero-day attacks [20]. Furthermore, the anomaly is detected based on packet loggers and packet sniffing tools to match the incoming traffic. This method is also less efficient for unidentified traffic [21].

More recently, data mining, clustering, data visualization, and statistical signal processing approaches have been used for intrusion detection. These techniques are more effective than rule-based intrusion detection, but typically produce a high level of false alerts [22]. Therefore, there is a need for more sophisticated methods that deal with real-time traffic monitoring. Machine learning-based techniques such as K-nearest neighbor (KNN), Hidden Markov models, and Support Vector Machines (SVM) have been used for detecting intrusion from real incoming traffic. KNN, also known as lazy learner, learns from nearest neighbors at run time [23]. However, this approach may be overfit for imbalanced small datasets. The support vector machine maps the input into another dimensional space, which offers promising results but is costly to train. Both these techniques require learning of expected anomaly but are sensitive to noise presented in the training datasets [24]. Similarly, Artificial Neural Network (ANN) needs a large dataset to learn, which probably takes a long training time and is not widely used for small datasets [22].

For small and imbalanced datasets, tree-based classifiers have proven to be one of the most efficient techniques [25]. Decision tree algorithms are one of the powerful supervisory machine learning techniques. They make decisions using bias and variance analysis to predict the labels. Furthermore, ensemble methods use the principle of combining weak learners to obtain a more reliable predictive model for better prediction and performance.

Ensembles can be obtained by boosting, which is a specific mechanism where learners gradually learn from the previous weak learners to reduce the overall loss function. This combined approach provides a powerful methodology for identification and pattern recognition for structured data [25]. XGBoost leverages the capabilities of boosting with ensembles. Moreover, we have identified that XGBoost is promising classifier amongst all the tree-based classifiers based on our preliminary results for ORNL dataset [15]. Furthermore, this method is not widely studied on power-grid based IDS applications. Consequently, we have decided to use the XGBoost model in RFE based feature selection scheme to obtain precise features. Further, it is also used as one of the classifiers to predict the output label in the majority vote ensemble method. We have listed pros and cons of each machine learning algorithms that we have used to generate majority vote based model in Table II.

IDSs for real-time systems such as SCADA-based power grids require low computational cost with high accuracy and execution speed. Such an IDS can be developed using a hybrid approach that combines the feature selection model along with an efficient classification scheme [26] which is the motivation for our proposed framework.

B. Ensemble Methods

A machine learning ensemble consists of a combination of several algorithms to obtain a result with better accuracy than from an individual classifier [27]. The ensemble is a statistical artifact known for over a hundred years based on the principle of *Wisdom of the Crowds* [28]. It was originally proposed by Sir Francis Galton who made a contest for observing a crowd in a cattle fair and showed that he was able to determine the weight of an ox by averaging the individual guesses from each

TABLE II
COMPARISON OF MACHINE LEARNING METHODS

Methods	Trees	k-NN	Artificial Neural Network	Naive Bayes
Description	Random Forest, XGBoost, Extra Trees and Adaboost are ensembles based on the decision trees	k-Nearest Neighbour looks for distance similarity between classes	Artificial Neural Networks are powerful classifiers able to work with complex patterns, able to represent non-linearity	Naive Bayes is a probabilistic classifier method
Characteristics	Decision trees, by bootstrapping (Random Forest) or by applying boosting (XGBoost, Ada Boosting) are easy to train and adaptable to different kinds of data	k-NN obtains good results when the data structure can be represented in a feasible representation space	consists of large number of "neuron" as processing elements, contains weighted connections to represent the distribution of data, acquire knowledge through learning process	Naive Bayes, based on Bayes theorem, has an assumption of feature independence, and requires prior probability estimates
Pros & Cons	Good fit for small datasets, Efficiently handled automatic step-wise feature selection, Does not required normalization and scaling	Effective for specific types, Needs homogeneous features, Flexible to choose the distance, No training is required, learn from neighbours	Expensive training for easy sets of data, but with complex data has superior ability to represent it	If probabilities are known obtains best results at low computational cost

person in a more precise way than the prediction of an individual.

There are three major classes of ensembles: bagging, boosting, and voting. Bagging and boosting use the same learner algorithm for prediction of the output labels. The difference between the two methods lies in how they generate successive subsets during classification. In boosting the datasets are randomly created, whereas in bagging, the elements are weighted, and not all of them have the same probability for selection [29], [30]. The third class, namely, stacking (voting) leverages several different algorithms working with the same data [31]. In a nutshell, bagging is used to decrease the model's variance; boosting works on the model's bias and voting achieves better performance by combining prediction of classification algorithms. The brief comparison of each of these three methods is listed in Table III.

TABLE III
COMPARISON OF ENSEMBLE METHODS: BOOSTING, BAGGING, AND STACKING

Method	Characteristics	Result
Bagging	homogeneous learners, parallel fitting, generate bootstrapping examples	focus on reducing variance Example : Random Forest
Boosting	homogeneous learners, sequential fitting, each time with wrongly classified samples	focus on reducing bias, model can be improved with gradient descent approach. Example : XGBoost
Stacking	heterogeneous learners, parallel fitting, meta-model combines learner results	focus on reducing bias Example : Majority Vote

In machine learning approaches, bagging is a powerful method to develop ensembles. The proposed method in [32] represents an example of a case study of neighbouring wind turbines based on bagging. This work was initially developed by Kramer *et al.* [33]. Bagging or bootstrapping aggregation consists of building independent predictors which extract the different samples from the training set and average the output by the prediction algorithms.

To achieve the best results, the predictors should be different or without correlation [27]. The voting ensemble creates multiple models and combines them to produce improved results. It is a more accurate classifier compared to the single predictive model.

Over a past few years, many Intrusion Detection Systems for various communication technologies have been proposed to detect the threats more accurately based on ensemble learning [34], [35], [36], [37], [38], [39], [40], [41].

One of the IDSs [34] is developed for imbalanced data samples (KDDcup99), where it is seen that J48 and Random Forest work best for big sample classes while others such as Bayesian network and Random tree seem to be a good fit for small samples. Therefore, the authors [34] propose a solution based on ensemble learning by applying a majority vote classifier to improve the performance of classification. Further, this work is improved by combining the prediction of Bagging and Boosting using ensemble techniques with tree base algorithms as the base classifier in [35].

In [36], the authors propose a novel approach that combines permission and intents supplements with an ensemble method for accurate malware detection for cellular phone communication. Moreover, in [37], authors execute anomaly detection over the communication networks by combining the prediction of three different types of classifiers, namely, neural networks, decision trees, and logistic regression using a weighted majority voting scheme.

The research work in [38] focuses on developing an IDS for network administrators by combining supervised and unsupervised learning techniques using ensemble method. This approach has been tested on various datasets like KDD Cup 99, NSK-KDD, and Kyoto 2006+ and is able to classify around 95% of the incoming traffic correctly [38]. In [39], the authors propose sustainable ensemble learning to improve the detection rate by aggregating multiclass regression models such that ensemble learning adapts to different attacks. Cloud-based solutions for distributed anomaly detection systems can be found in [40]. In [41], the authors propose a Gaussian

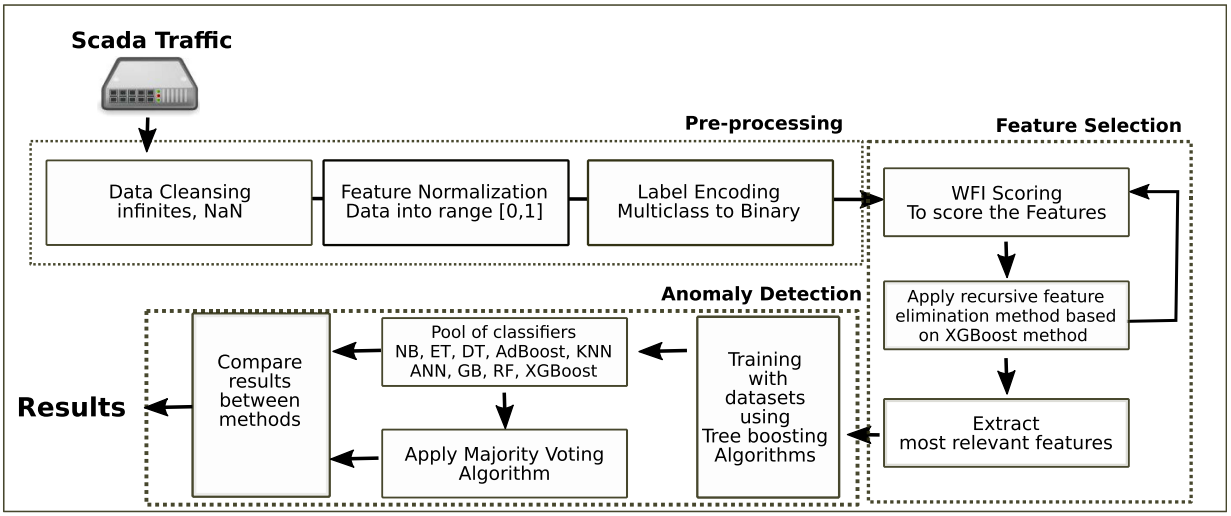


Fig. 2. Process diagram of the proposed framework for intrusion detection in power grids.

mixture based anomaly detection technique that relies on ensemble one-class statistical learning model that is designed to effectively recognize zero day attacks in real-time using the concept of edge networks.

III. Proposed Framework for Intrusion Detection System in Power Grids

This section presents the proposed scheme for an intrusion detection system for power grids to classify traffic into attacks and normal events by analyzing SCADA traffic. This novel approach uses the RFE-XGBoost based feature selection method to determine the most consistent features from the dataset based on feature importance scores. Furthermore, the majority vote ensemble method identifies accurate outcomes during classification. This combined approach accomplishes two significant aspects of real-time traffic monitoring namely, accuracy and computational speed. The entire framework is divided into three phases - data preprocessing, feature selection, and anomaly detection, as illustrated in Fig. 2.

The data cleaning, feature mapping, and feature normalization are done in the preprocessing phase to obtain streamed and sanitized data. Since the power grid is part of a large industrial control systems that use complex SCADA infrastructure to control the substation equipment, network monitoring devices such as SNORT, Wireshark and Syslog are used to obtain the different types of features from the communication data [16]. Usually, streaming data that is obtained from sensors or actuators in real-time systems has reliability issues, such as lost signal or wrong observations due to failures in measuring devices which result in their inability to interpret the scale readings. For this reason, the data cleansing operation is a critical process to remove incorrect data (like infinities or NaN data). In this phase we remove empty sequences that otherwise will generate issues such as inaccurate and faulty inferences with the algorithms. Moreover, the power grid records are collected at four PMUs (Phasor Measurement Units) which are situated at different locations in the power substation. Various internal attacks were launched by the ORNL to generate the IDS dataset for power grids reflecting the diverse nature of records. Another transformation that we performed in the data in this phase is the data normalization, to improve the training stability in the classifiers, especially for Artificial Neural Networks. For this normalization a standard scaler, a method that normalizes the records by considering zero mean and unit variance, was used.

In the feature selection phase, the importance of each feature is identified using the WFI scoring model. The recursive feature elimination approach is then applied to the binary dataset to eliminate irrelevant features recursively. Once the model determines the most consistent features, in the anomaly detection phase, the nine classifiers, namely NB, ET, DT, RF, GB, XGBoost, ADBoost, KNN, and ANN are used to predict the output labels. Finally, the majority vote-based ensemble method predicts the class label for input samples based on the majority of the class labels predicted by each of these nine classifiers. The voting classifier uses “hard voting” to classify the input sample based on the majority class label.

IV. Concept of Methodology

A. Majority Voting Algorithm

There are two main categories of majority-based ensemble methods, namely, voting and averaging [42]. Generally, voting is used for classification, while averaging is used for regression. We have used a voting based ensemble method to detect intrusions. In this method, we can create multiple base models using a training dataset. The output of each base model acts as the input of the majority vote base ensemble algorithm. These base models are created using different splits of the same training dataset along with other classifiers. The majority vote classifier predicts the output label based on the prediction of multiple base models. To calculate the overall error, we assume that the probability of each base model being correct is $(1-\epsilon)$, where ϵ is the classifier error. We assume that the

Algorithm 1. Majority vote ensemble training algorithm for n classifiers

Data:
Dataset **Train** $= \langle \mathbf{X}, \widehat{\mathbf{Y}} \rangle$, **Test** $= \langle \mathbf{x}, \widehat{\mathbf{y}} \rangle$,
Size of Test Dataset: m
Classifiers $\mathbf{C} = \langle c_i | i \in 1 \cdots n \rangle$
begin
  for $i : 1$ **to** n **do**
    p_i train predictor (c_i) on Test Dataset
  end
  for i 1 **to** m **do**
    for j 1 **to** n **do**
      Apply predictor(c_j) to sample x_i
    end
    best prediction = more classifier votes
    $\hat{y} \leftarrow$ best prediction
  end
end
Result:
Predictions: $\widehat{\mathbf{y}} = \langle y_i | i \in 1 \cdots m \rangle$

classification errors are independent, and we can also obtain the probability of the majority vote error by applying binomial distribution. The probability of obtaining k valid predictions out of n (k over 50% or $k > n/2$) is achieved using binomial distribution as follows:

$$\text{Probability}(X = k) = \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k} \tag{1}$$

We obtain the total probability by adding all the individual probabilities for each k:

$$\text{Total Probability} = \sum_{k > \frac{n}{2}}^{n} \binom{n}{k} \epsilon^k (1 - \epsilon)^{n-k} \tag{2}$$

If $\epsilon < \frac{1}{2}$, and the predictions from the classifier are considered as independent, the error is, in principle, smaller, as when $n \to \infty, \epsilon \to 0$. With the majority vote strategy, we can obtain better accuracy than with the direct or linear-averaged approach. The majority vote model gives the same weight to each one of the votes using a *democratic approach* (see Algorithm 1)

If we observe that some inputs are more potent than others, then we can quantify and adjust this contribution. For instance, with a Bayesian model averaging (BMA) where the weighting is adjusted after training by reviewing the individual contributions to accuracy one by one.

The other type of ensemble approach are short term algorithms where ensembles are applied for short term energy demand forecasting [43], [44]. The use of ensembles combined with deep or machine learning algorithms is a promising area of research as the ability to run multiple algorithms in parallel is efficient, and the combination of models with different strengths generates better results.

B. Feature Selection

Through feature selection, we can select the subset of relevant features for the appropriate model construction. This will avoid the bane of dimensionality and enhances the generalization of the model by reducing overfitting [45]. However, due to this approach, some feature information may be lost, but that does not impact the overall performance of the model; instead, the selected features are more representative to model the classifier. Moreover, the samples with hundreds of features will increase the computation cost and decrease the classification performance. Therefore, our first target is to identify the subset of the relevant features of the dataset which are highly related to the class but are not related to each other.

To identify the most consistent features of power grid datasets, we have used the WFI based scoring model by ranking the elements. This method extracts the feature importance score of each feature by considering the improvement in impurity while splitting the individual tree. The irrelevant features are removed recursively according to the scoring model using the RFE approach on XGBoost algorithm. We have used binary datasets to extract the most relevant features instead of multi-class datasets, as sometimes the WFI scoring model has a bias towards multiple categories of the dataset [46]. However, the extracted features are applicable to all four categories of the datasets.The features are carefully removed without losing much of the information to generate the feature subset using RFE approach. In each iteration, XGBoost is trained with a selected feature subset to measure the accuracy.

During the process of feature selection, the current subset is replaced by the selected set of features when the accuracy of the current subset is increased by more than 0.5%. This way, we can achieve consistent elements from the entire dataset. The steps of the RFE-XGBoost algorithm are shown in Algorithm 2.

V. Experiments and Results

A. Datasets

To determine the performance of the proposed approach, we have used three public benchmark datasets [16]. These datasets were created at Oak Ridge National Laboratories (ORNL) by setting up a power grid testbed [10]. This testbed was configured using various power grid components, namely, power generators - G1 and G2, IEDs - R1 to R4, breakers - BR1 to BR4 and a three-bus two-line transmission system. In the case of fault detection, the IED trips the corresponding breaker depending upon the nature of the fault. However, these IEDs are not smart enough to differentiate between original and fake failures. Moreover, operators can also manually trip the breakers and other system components during system maintenance [9].

The datasets derived from this power grid testbed contain measurement related to normal, disturbance, control, and cyber-attack behaviors captured during electrical transmission [11]. These datasets are randomly sampled and classified into three main categories, namely, binary, three state, and multi state. Initially, the multi state dataset is constructed during the experiment, and consists of a total of 37 scenarios. These scenarios are mainly divided into three categories, namely, 8 natural events, one no event, and

Algorithm 2. Recursive Feature Elimination based on XGboost WFI scoring model

Data:
Training power-grid data-set: PD
begin
Initialize:
Current features $Curr\_PD$ = {1,2,3,...,n}
Ranked features Sel_PD = $Curr\_PD$
Set standard deviation SD = 0.5
Set proportion of features to be deleted = $SetProp$
Build XGBoost model based on $Curr\_PD$
Compute the initial accuracy $Acc(Curr\_PD)$
while *Features(Curr_PD) != Empty* **do**
Evaluate the ranking criteria
Rank features of $Curr\_PD$ in ascending order by WFI scoring model
Remove features = SetProp(min($Score$))
Store the features in Sel_PD
Build $XGBoost$ model based on the rank features Sel_PD
Compute the accuracy $Acc(Sel\_PD)$
if $Acc(Curr\_PD) + SD < Acc(Sel\_PD)$ **then**
$Curr\_PD \leftarrow Sel\_PD$
else if $Acc(Curr\_PD) == Acc(Sel\_PD)$ & $Ftr(Curr\_PD) > Ftr(Sel\_PD)$**then**
$Curr\_PD \leftarrow Sel\_PD$
end
$Best\_PD \leftarrow Cur\_PD$
end
Result: ranked feature subset $Best\_D$

28 attack events. The eight natural events are further divided into 6 SLG faults events and 2 line maintenance events, as listed in Table IV. Moreover, the 28 attack events are subcategorized into three major attack events, namely, Data Injection, Remote Tripping Command Injection, and Attack on Relay Settings. These include 6 SLG fault replay attacks, 4 command injection attacks against single IED, 2 command injection attacks against 2 IEDs, 10 relay setting change attacks on a single IED, 4 relay setting change attacks on 2 IEDs, and 2 relay disable and line maintenance attacks as listed in Table IV. These attack scenarios are simulated using the concept of an internal intruder, who can launch different attacks by issuing malicious injections from the substation [10]. Moreover, we have derived a seven-states dataset from the multi-states dataset. The dataset of each category is sub-sampled into fifteen sets. Table IV gives the summary of various output labels according to the four categories of the dataset.

The datasets of the power grids consist of a total of 128 features. These features are derived using 4 Phasor Measurement Units (PMUs), which measure electrical signals of substation using a common time source for effective time synchronization. A total of 106 PMU measurements are carried out using 4 PMUs, where each PMU measures 29 features of a particular location. These features are referred to as R# (signal Reference), which indicate the index of PMU and type of measurement. For example, R2-PA2: IH represents the phase A - current phase angle measured by PMU located at R2 [16]. Twelve different categories indicate phase angles and magnitude of voltage and current. The detailed description of the features is given in [15]. Furthermore, 16 more features are derived using control panel logs, snort alerts, and relay logs [10]. The last column refers to a marker that labels different normal and malicious events. Each set consists of around 5000 instances that include 294 no events, 1221 natural events, and 3711 attack events approximately, which represents that the given datasets are imbalance in nature.

TABLE IV
DESCRIPTION OF THE OUTPUT LABELS OF THE VARIOUS CATEGORIES OF THE DATASETS

Categories	Output Labels
Binary (2)	Normal, Attack
Three States (3)	Normal, Attack, No Event
Seven States (7)	1-Natural SLG Fault, 1-Data injection attack 2- Remote Tripping Command injection attack 3- relay setting change attacks
Multi States (37)	1 – No event 8 – Natural events - 6 SLG faults - 2 Line maintenance events 28 – Attack events - 6 data injection SLG fault replay attacks - 4 command injection attacks against single IED - 2 command injection attacks against two IEDs - 10 relay setting change attacks on single IED - 4 relay setting change attacks on two IEDs - 2 relay disable and line maintenance attacks

B. Evaluation Methodology

The primary objective of the proposed model is to provide a real-time intrusion detection for power-grid systems. Hence, our target is to build a fast and accurate model that captures any malicious event efficiently that may happen in the network. To fulfill both requirements, we have used RFE-XGBoost-based WFI scoring model for feature selection along with the majority vote-based ensemble method for classification. The feature selection module improves the computational cost as we are targeting the 30 most consistent features out of 128 features of the given datasets. Furthermore, we have used nine most powerful classifiers to classify the normal and malicious events. For more accurate results, we have applied the majority vote-based ensemble method, which predicts the class label based on majority of the class labels predicted by each of these classifiers.

These datasets used in our analysis are the publicly available datasets generated at the ORNL laboratory on a small power grid testbed [16]. For proper validation, experiments were computed for four different categories of the samples. Furthermore, the observations were carried out using 100,000 normal and attack events of each of these four categories, which were divided into 15 datasets. For fair distribution and assessment, each dataset was split randomly into two subsets, training (80%) and testing (20%). The training data was used for the algorithm training and the testing data was used to test the accuracy of the result. To avoid selection bias of the datasets and to reduce the overfitting, we have used 10-fold cross-

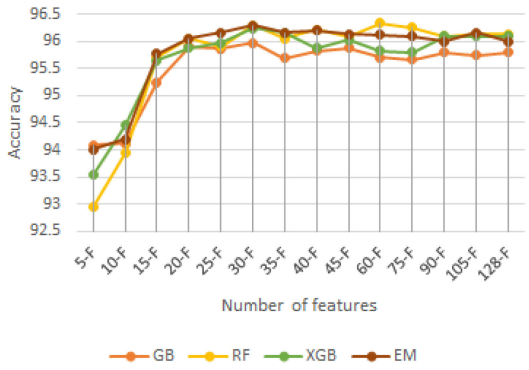


Fig. 3. Comparative analysis of different features to evaluate the accuracy using RFE-XGBoost WFI scoring model.

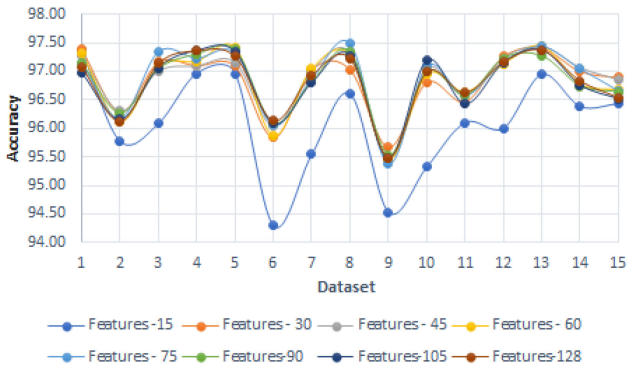


Fig. 4. Different number of features are evaluated to measure the accuracy of 15 binary datasets using RFE-XgBoost WFI scoring model.

validation technique during the training process. This method performs the training 10 times with different random selections (80/20) from the original dataset. This well-defined systematic approach circumvents the inadequacy of bias performance assessment. The proposed approach is implemented using Python on a Jupyter notebook using the Anaconda distribution platform on Windows10 with Intel Core i5-8300H 2.30GHz processor, 8 GB RAM, and Nvidia Geforce GTX 1060 GPU.

C. Evaluation of Feature Selection

We have made observations based on the number of subsets of the features considering 15 binary datasets. Initially, we started with 128 features and reduced the number of features in each iteration based on the output of the WFI scoring model to compare the accuracy of the current set with the selected subset. To extract the gist of the features, we have applied WFI based scoring model which scores the importance of all features. This ranking defines how often the feature is used to determine the output label while constructing tree.

Fig. 3 illustrates the comparative analysis of different features versus accuracy graph of one of the 15 datasets. The classification with 30 features offers the highest accuracy during classification of normal and attack events using the majority vote-based ensemble classifier.

Fig. 4 demonstrates the accuracy of different 15 datasets according to the various subsets of the total features. The accuracy increased significantly up to 30 features, after that there is no substantial improvement in accuracy. Hence we have extracted most 30 features of each dataset and consider the same features for all the four categories, namely, binary, three-class, seven-class and multi-class datasets.

D. Result Discussion

To evaluate the performance of the majority vote based ensemble algorithm, we have computed the accuracy of fifteen datasets of all the four categories using nine most promising classifiers. The choice of these classifiers is carried out based on our preliminary results of the comparative analysis of various machine learning classifiers [15]. We have chosen nine heterogeneous classifiers to determine the efficiency of selected features via multiple simulation trials and observed the predictions of all the algorithms. After deriving the accuracy of all the nine classifiers, the majority vote ensemble algorithm was applied to compare the prediction of the output labels. The comparison was carried out based on the majority class label voting classifier with "hard voting" to classify the input samples.

The ensemble algorithm predicts accurate outcomes by aggregating and applying the majority vote rule on the result of the different classifiers. We have incorporated heterogeneous classifiers, namely, random forest (RF), gradient boosting (GB), XGBoost (XGB), Extra Tree (ET), Decision Tree (DT), K-Nearest Neightbour (KNN), Naive Bayes (NB), Adaboost - Decision Tree (AdBoost-DT), and artificial neural network (ANN) to achieve performance improvement of the majority vote based ensemble model.

We have performed overall 60 computations of each of the four categories (binary, three-states, seven-states, and multi-states) containing fifteen datasets to evaluate the performance of each of ten classifiers. According to the analysis, the accuracy of the Naive Bayes algorithm is less compared with other classifiers for all the four categories, namely, binary (around 52.34%), three class (58.21%), seven class (19.26%), and multi class (13.2%). Fig. 5 presents a comparative analysis of the accuracy of the remaining eight classifiers along with the majority vote-based ensemble algorithm. Among nine base classifiers random forest, gradient boosting and XGBoost have mostly proven to be more efficient in the case of binary, three states, and seven states classification. However, for multi states classification, random forest, extra tree and XGBoost are more promising than the other six classifiers.

Moreover, the majority vote ensemble classifier outperforms by taking advantage of prediction logic of other nine classifiers. The accuracy of the majority vote based ensemble method is higher and more precise than the other nine classifiers with accuracy around 98.24% for binary, 97.95% for three states, 95.91% for seven states, and 93.78% for multi states datasets, approximately.

To validate the effectiveness of the proposed scheme, we have compared the accuracy of majority vote based ensemble algorithm with five published methods, namely AdaBoost-JRIP (AdaJRIP) [9], Common Path Mining [10], [11],

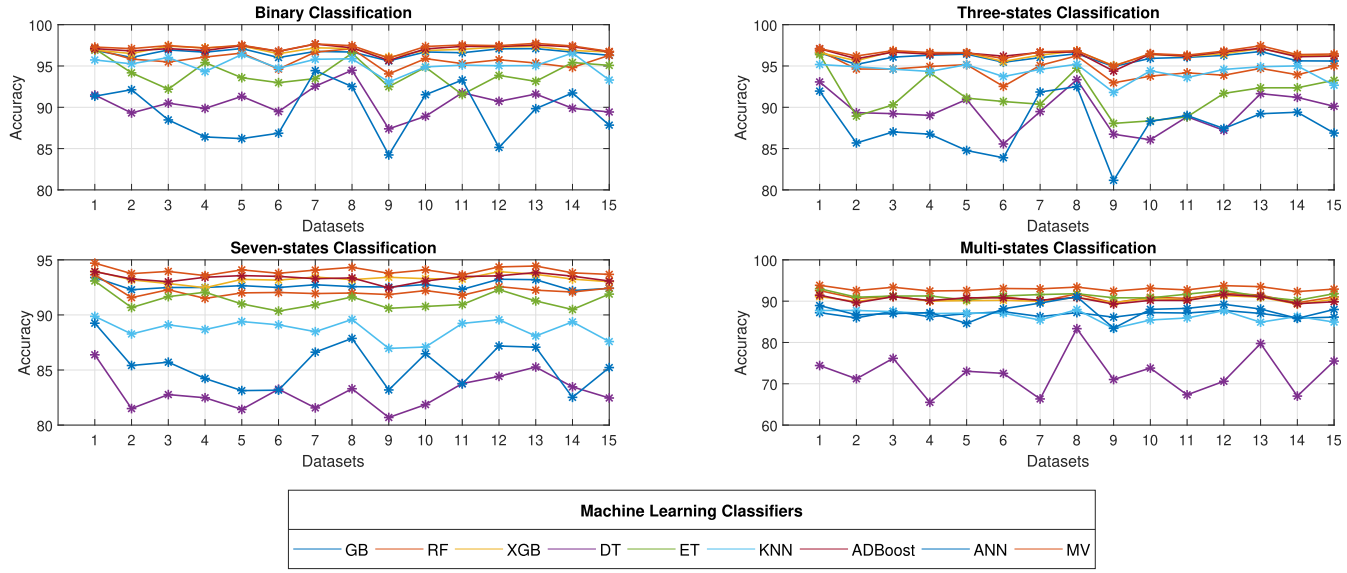


Fig. 5. Comparative view of different Machine Learning classifiers for four categories for each of the fifteen datasets.

TABLE V
COMPARATIVE ANALYSIS OF OVERALL PERFORMANCE OF VARIOUS TECHNIQUES AND PROPOSED MAJORITY VOTE ENSEMBLE METHOD BASED CLASSIFIER

Classifier	Data Cleaning	Feature Selection	Features (%)	Classes	Accuracy
ADA-JRIP [9]	NA	NA	100%	2	94.55%
				3	94.61%
				37	85.85%
CPM [10],[11]	Applied	NA	100%	7	93.00%
				25	90.40%
EMCT [12]	Applied	PCC	25%	2	70.60%
			50%		76.3%
			75%		83.5%
			100%		90.2%
GMMKM [14]	Applied	PCC	25%	2	94.56%
			50%		95.83%
			75%		96.82%
			100%		97.27%
Tree Based [15]	Applied	GBFS	12%	2	97.26%
				3	96.50%
				7	94.12%
				37	92.46%
MV-EM	Applied	RFE-XG	**25%**	**2**	**98.24%**
				3	**97.95%**
				7	**95.91%**
				37	**93.78%**

Expectation Maximization Clustering Technique (EMCT) [12], Gaussian Mixture - Kalman Filter Model (GMM-KF) using Pearson Correlation Coefficient (PCC) feature selection method [14] and GBFS based tree based classifiers [15].

Furthermore, we have also compared various performance evaluation factors such as whether proper pre-processing is applied on datasets; whether feature selection approach is incorporated and if applied how many features are selected to evaluate the accuracy by considering four states of dataset. Table V shows that the proposed framework outperforms compared to other published techniques. The model accomplishes significant accuracy for all the four categories by selecting only 25% of the features. Note that the results mentioned in the table refer to the highest accuracy achieved during the classification by the majority vote based ensemble algorithm.

TABLE VI
COMPARISON OF ACCURACY OF DIFFERENT ENSEMBLE TECHNIQUES

Categories	Bagging ensembles			Boosting ensembles			Stacking
Classifiers	**DT**	**RF**	**ET**	**GB**	**AdB(DT)**	**XGB**	**Majority**
Binary	91.32	96.08	96.54	96.76	95.21	96.93	98.24
Three-Class	93.07	96.84	96.4	96.59	95.26	96.61	97.95
Seven Class	86.38	93.65	93.08	93.4	92.16	93.97	95.91
Multi-Class	74.42	92.56	93.02	87.22	91.59	91.1	93.78

Bagging generally considers homogeneous weak learners to train the model sequentially. However, the learning process occurs independently, and prediction is made by averaging all the parallel models. On the other hand, boosting learns sequentially by considering errors from previous ones. In both these methods, homogeneous learners are used. In contrast, stacking often considers heterogeneous weak learners to train the meta-model to predict the output based on different model predictions. We have discussed the literature pertaining to various ensemble methods in Section II, namely, bagging, boosting and stacking. To demonstrate the efficiency of our proposed approach, we have compared bagging and boosting based ensemble methods with the majority vote based ensemble technique which refers to stacking approach.

As shown in Table VI, bagged DT, RF and ET are examples of bagging ensembles whereas boosting ensembles include GB, AdB-DT and XGB. Furthermore, we have designed the majority vote ensemble technique by applying the predictions of nine heterogeneous classifiers. Table VI represents the promising results compared to other ensemble techniques in terms of accuracy. Moreover, the other three non-ensemble classifiers, namely, NB, KNN and ANN have less accuracy; 54.29%, 94.32%, 88.47% for binary, 58.21%, 93.72%, 87.02% for three state, 21.57%, 89.10%, 84.23% for seven

TABLE VII
SPECIFICATION OF EACH MODEL ARCHITECTURE

Model	Main Parameters
Gradient Boosting	estimators = 100, max depth = 12, min split = 12
Random Forest	estimators = 100, criterion ='gini', max depth = 12, min samples leaf = 1, min split = 2
XGBoost	estimators = 100, max depth = 12, random state = 3
Decission Tree	max depth = 12, min split = 2, random state = 3
Extra Tree	estimators = 100, max depth = 12, min split = 2
AdaBoost	classifier = DT, max depth = 12, min split = 2
ANN	dense = 512, epochs=50 batch=16, opt=adam, act=relu
KNN	neighbours = 3

state, and 13.18%, 87.77%, 83.13% for multi state as compared to the majority vote ensemble method.

We have denoted the specification of each model in Table VII. These parameters are achieved using a grid search while training the model for hyper-parameter tuning, which improves the efficiency of each classifier. Here, estimators refer to the number of trees created in the model during the training process. At the same time, maximum depth (max depth) represents the node expansion until all leaves contain less than the value defined in the minimum samples split (min split). For the ANN model, we have created 512 hidden layers with 50 epochs each by considering a batch size equal to 16. Furthermore, we have considered 'adam' optimizer for weight optimization and 'relu' as activation function for the hidden layer. KNN decides the output label by considering the prediction of three nearest neighbors.

Using an ROC plot, we can visualize the trade-offs between the true positive rate (TPR) also known as sensitivity and false positive rate (FPR). Further, the Area Under the Curve (AUC) presents the degree of separability, which defines the capability to differentiate the classes. Fig. 6 shows the ROC curves of four classifiers created by the 10-fold cross-validation method. We have examined RF, GB, XGBoost and Majority Vote, which represent the different categories of ensemble technique, namely, bagging, boosting, and voting ensembles.

Moreover, we have presented the ROC curve of one of the fifteen datasets of each of the four categories. ROC curve qualifies the model according to the total area under the curve for each classifier. The metric falls between 0 and 1, with a higher value indicates better classification performance. The graphs in Fig. 6 compare the AUC of four classifiers. The green curve represents the majority vote-based ensemble method is contributing to the high AUC scores for all four classes. This means that the majority vote based model is better at achieving a blend of precision and recall. Furthermore, random forest and XGBoost contribute slightly better than gradient boosting for all four categories. However, gradient boosting is comparatively lower in terms of AUC scores specifically for the multi states dataset.

In the case of imbalanced datasets, the PR plot is more informative than the ROC plot while evaluating classifiers [47]. Here we are not only targeting binary classification but also classifying multiple attack events. Hence, for more information retrieval, we have also analyzed PR curves in case of bias in the class distribution. The baseline of the PR curve is determined

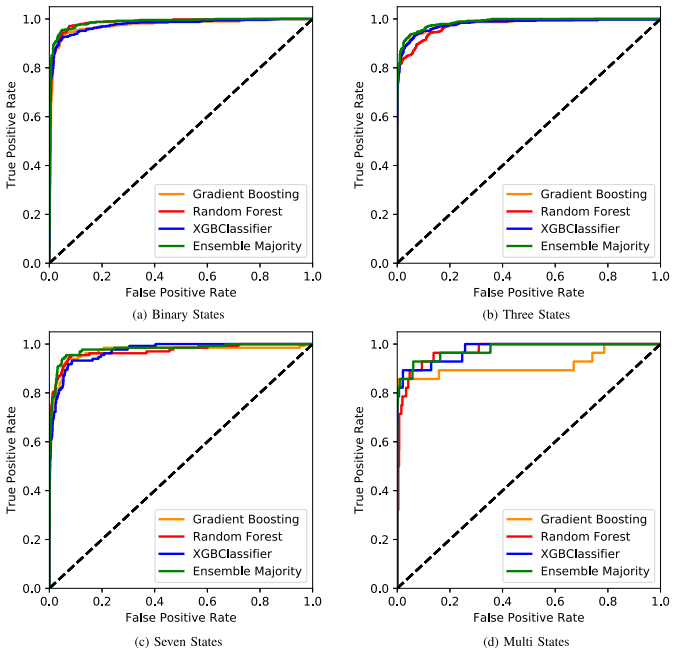
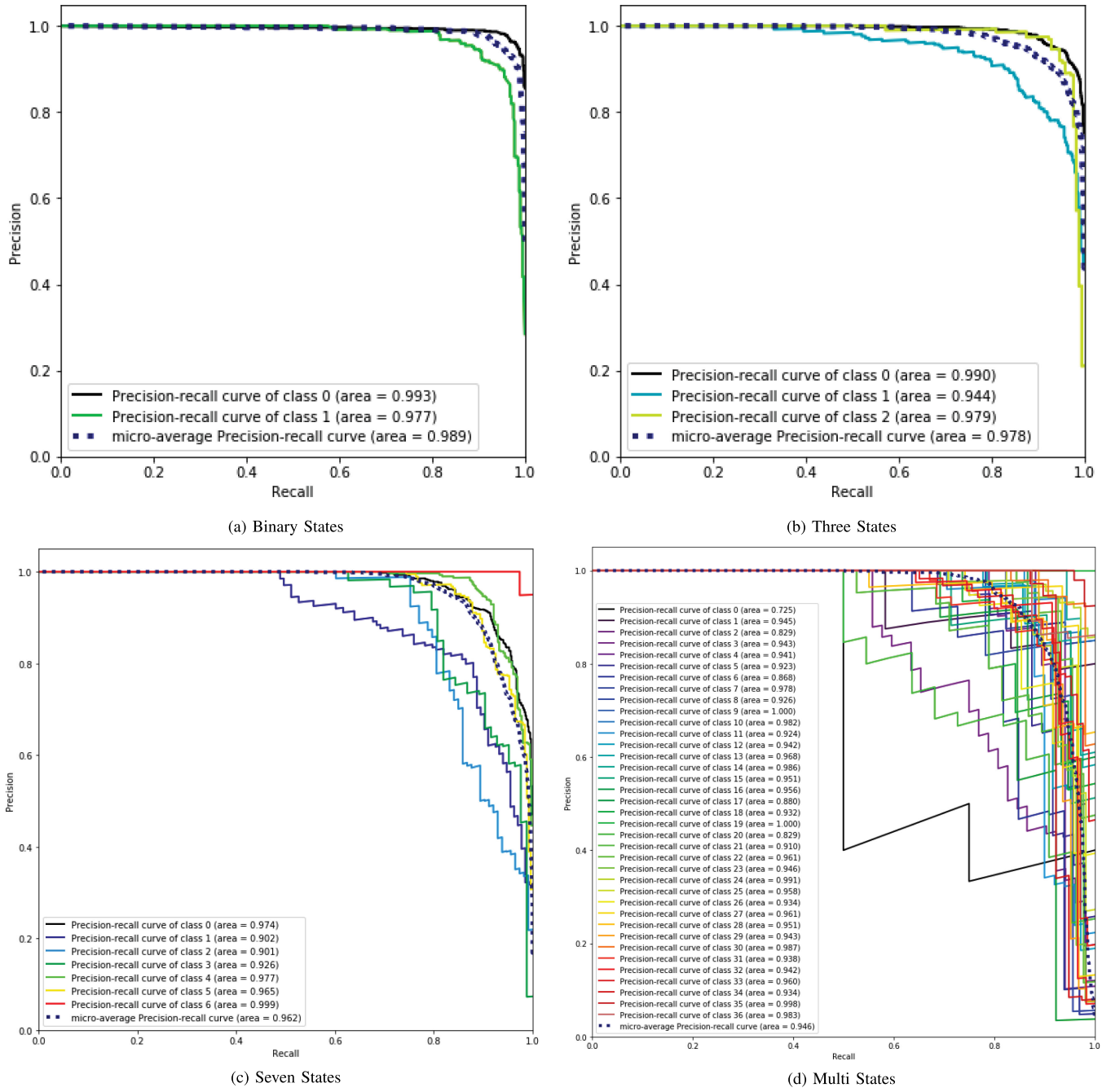


Fig. 6. ROC Curves of three types (bagging, boosting and stacking) of Machine Learning Classifiers for four categories.

by the relation of precision and recall values. Fig. 7 depicts the precision/recall for each threshold for a majority rule-based ensemble model by considering all the four categories of the dataset. For all the four types, the majority rule-based ensemble classifier maintains a high detection rate. The proposed model has achieved 98.9%, 97.8%, 96.2%, and 94.6% of the average precision-recall curve area for binary, three states, seven states, and multi-states, respectively. The exact percentage of each output label is depicted in Fig. 7. The results indicate the model performs exceptionally well with all the categories to predict various types of class labels.

Precision defines the ratio of the number of true positives, divided by the total number of true positives and false positives, which describes the efficiency of the model in terms of prediction of the positive class. Recall represents the ratio of the number of true positives divided by the total number of true positives and false negatives. While F measure is used to combined the precision and recall to determine the harmonic mean of those parameters. For the precise assessment, we have measured the efficiency of our proposed model not only by evaluating the accuracy of the classification but also by considering other factors such as, recall, precision, F1 score and miss rate. We have achieved high precision, recall, and F measure for RFE based majority vote ensemble method for all the four categories. The results of these performance metrics are illustrated in Table VIII. We have evaluated the results for all the 15 datasets of all four categories. However, we have depicted the most promising result of all the observations in Table VIII. Furthermore, for a more detailed view, we have represented the results of all the simulation trials in Fig. 8, which consist of all the 15 datasets of binary, three states, seven states, and multi states categories. As shown in the figure, we have achieved around 97% detection rate, which offers significant classification of



(a) Binary States

(b) Three States

(c) Seven States

(d) Multi States

Fig. 7. Precision-Recall Curves of RFE based Majority vote ensemble method for four categories.

attack and normal events for binary and three states categories, with only 3% miss rate. Furthermore, the seven class and multi-class output labels are also accomplished with 93% detection rate with around 7% miss rate.

We have observed the importance of the various features in the previous section, where accuracy is measured by considering subsets of the features. In that, we have focused on the binary dataset. For further proof of concept, we have evaluated the accuracy of three other categories, namely, three class, seven class, and multi class datasets, by comparing all the 128 with 30 features. To extract the gist of the features, we have applied an RFE based WFI scoring model, which scores the importance of all features recursively. This ranking defines how often the feature is used to determine the output label while constructing the tree. Table IX illustrates the comparative analysis of four categories by considering 128 features

TABLE VIII
PERFORMANCE EVALUATION METRICS OF PROPOSED RECURSIVE FEATURE ELIMINATION BASED MAJORITY VOTE ENSEMBLE METHOD

Measure	Binary	Three-class	Seven-Class	Multi-class
Precision	97.40%	97.21%	95.38%	94.04%
Recall	96.63%	96.1%	93.46%	92.79%
F-Measure	96.99%	96.6%	94.26%	93.04%
Miss Rate	3.37%	3.90%	6.54%	7.21%

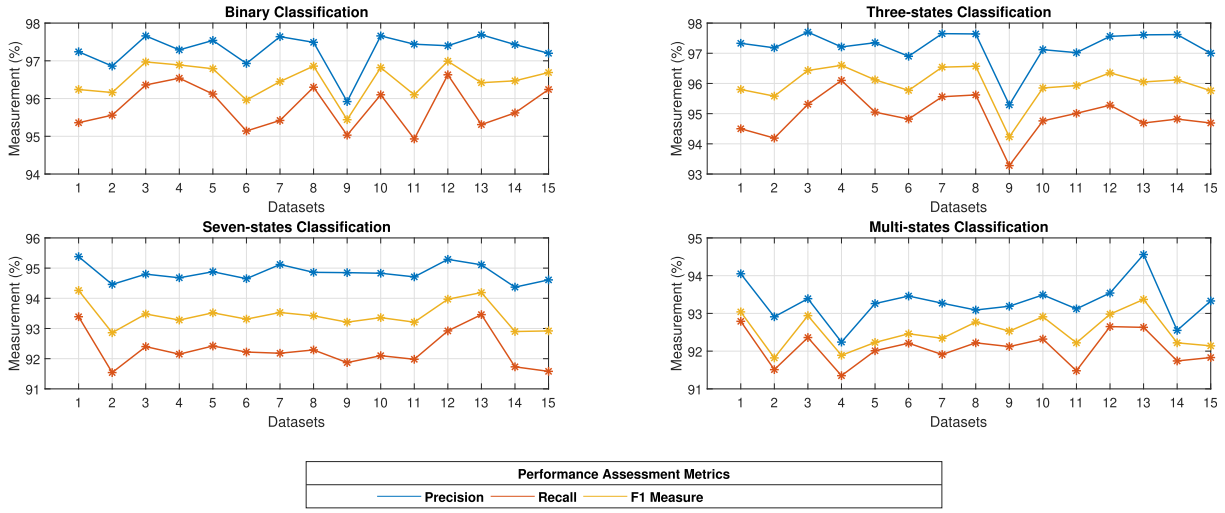


Fig. 8. Result of various performance measurements (Precision, Recall, F1 Measure) of RFE based majority vote ensemble method for four categories of fifteen datasets.

TABLE IX
COMPARISON OF ACCURACY OF MAJORITY VOTE ENSEMBLE ALGORITHM WITH AND WITHOUT RECURSIVE FEATURE ELIMINATION BASED FEATURE SELECTION

Classifiers	**Without Feature Selection** 128 features (Accuracy)	**RFE - Feature Selection** 30 features (Accuracy)
Binary	96.93	97.44
Three-Class	96.64	97.25
Seven Class	93.65	94.91
Multi-Class	92.23	93.08

TABLE X
EXECUTION TIME OF RANDOM SAMPLE OF 5300 RECORDS

Phase	Execution time (secs)
Pre-Processing time	0.0186300039
Feature Selection time	1.646741629
Training Time	306.036

TABLE XI
COMPARATIVE ANALYSIS OF TRAINING TIME OF VARIOUS CLASSIFIERS (RANDOM SAMPLE OF 5300 RECORDS)

Various Classifiers	Training time (sec)
Gradient Boosting	101.2503724
Random Forest	9.819750547
XGBoost	7.018202782
Decision Tree	0.892643929
Extra Tree	3.217407703
K-nearest neighbors	3.426834106
Naive Bayes	0.041889906
AdaBoost (DT)	42.29490685
Artificial neural network	240.9818392
Majority Vote	306.0361404

versus 30 features extracted by RFE. The classification with 30 features offers the highest accuracy during the classification of normal and attack events using the majority vote ensemble classifier. In Table IX, we have presented the result of one of the 15 datasets. During experiments, we have also observed that the training time of multi states datasets with all the 128 features is unrealistic as it took more than 24 hours. Hence, feature selection is a crucial factor used to develop a better predictive model and make the model computationally efficient.

The detection time is determined using real-time data classification based on incoming traffic (generally based on one observation). Intrusion detection systems should provide an immediate response to potential attacks. To improve the performance of such systems we need to eventually train the module based on the behavior of real-time traffic and accordingly need to deploy the model in a real-time environment. Since training involves computational time and resources, it is mostly performed using high-performance infrastructure (generally offline on the plant floor or at the control center). In contrast, the intrusion detection inference engine (trained model) is used to classify the observation of real-time traffic and deployed in hardware that is connected to the communication network.

We have conducted experiments to determine the execution time of each of the four phases, namely, pre-processing, feature selection, training time, and testing time of the proposed technique by taking random samples from the original dataset (5300 records out of 100,000 records). The execution times reported refer to the implementation of the proposed approach on Windows10 with Intel Core i5-8300H 2.30GHz processor, 8 GB RAM, and Nvidia Geforce GTX 1060 GPU. The execution times of all the modules of the proposed algorithm are listed in Table X.

The above refers to the training time of the three phases. Generally, the preprocessing, feature selection, and training of the model are performed frequently at certain time intervals at the plant floor/control center offline using high computational resources. However, to address the detection rate of the proposed scheme we need to target real-time classification. In principle, the filtering mechanism of the proposed algorithm should be incorporated in edge computing devices such as smart routers and smart switches. This will significantly reduce the detection rate as filtering (preprocessing) is

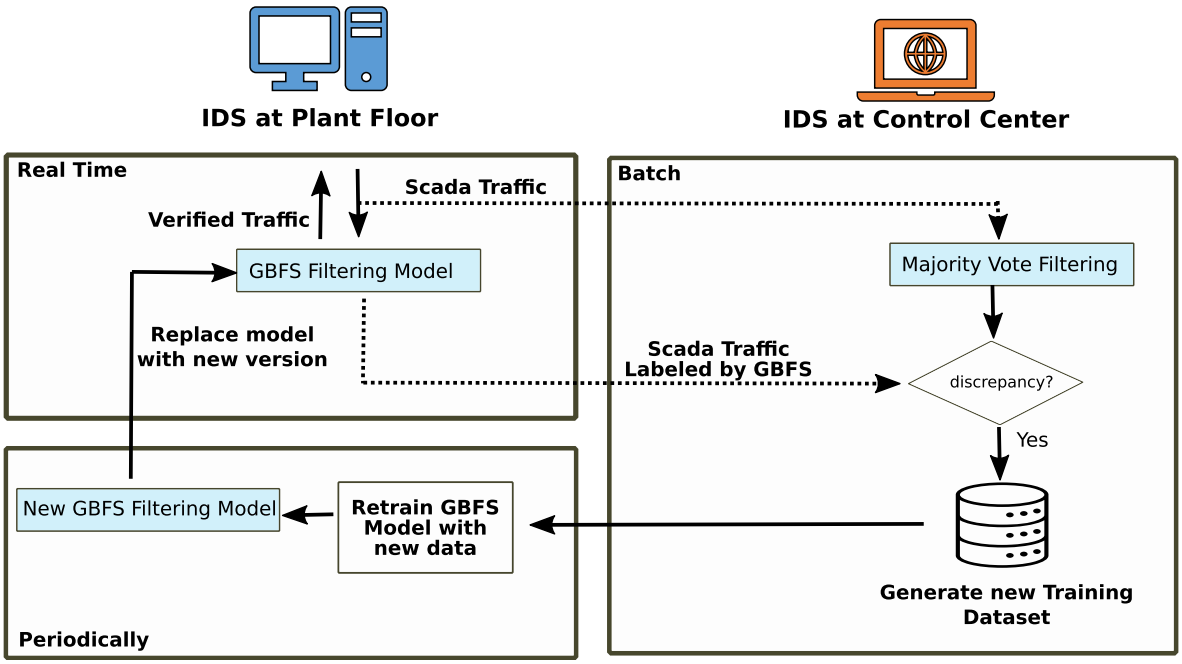


Fig. 9. IDS framework for real-time SCADA systems for power grids.

computed on hardware. These distributed computing devices make the detection rate low (nanoseconds) which also avoids the requirement of a powerful CPU or the support of a GPU. Thus, the execution time to classify normal/attack events by our proposed model is comparatively low which is adequate for a real-time intrusion detection system.

Furthermore, we have compared the training time of various classifiers with the majority vote-based ensemble method as depicted in Table XI. While the majority vote ensemble method takes more time compared to the single classifier, to balance the execution time, and to obtain high performance, we have proposed a real-time IDS for SCADA systems as discussed in Section VI. In particular, we have deployed a majority vote-based IDS on the control center that monitors all the plant floor IDSs which work on a single GBFS based classifier. This approach maintains the performance of IDS for real-time SCADA systems to distinguish attacks and normal events during live data streaming with the standard available hardware.

VI. Proposed IDS Framework for Power Grid SCADA System

We have extended our previous GBFS based model with RFE based majority vote ensemble method by combining the results of several classifiers to achieve an accurate outcome. The purpose of the previous model is to achieve accurate classification without deteriorating the performance of the system using prediction of a single classifier. However, majority vote ensemble method predicts the output label based on the majority of the output labels predicted by each classifier. This will further improve the efficiency of the prediction and provides the most accurate output label in terms of normal and attack events. For that, we have targeted various heterogeneous classifiers, namely, Random Forest, Gradient Boosting, XGBoost, Artificial Neural Network, Naïve Base, and Decision Table for ensemble learning by referring to preliminary results from this paper [15]. This approach will generate a better predicting model than a single model using a hard voting based majority rule ensemble technique.

In distributive environments such as power grids, the availability of the most accurate intrusion detection system is a crucial factor. This is achieved by replacing the existing deployed model with the most recent ones, which enhances the capability of IDS and is accomplished by training the model frequently according to the live traffic. The training time plays a significant role in real time detection as shorter execution time develops the model quickly. We have proposed the IDS framework for real-time SCADA systems for power grids, as shown in Fig. 9. In this approach, we place two different IDSs at two different locations, one at the plant floor and another at the control center. The plant floor IDS analyzes the SCADA traffic using the GBFS based filtering model as it is more compatible in detecting the intrusions in real-time communication. However, for more accurate results, the output of this module is verified at the control center using the majority vote-based IDS with multiple classifiers. In case of a discrepancy in the output labels, the records will be added to a new training dataset to retrain the GBFS filtering model periodically. This way, we can achieve the most updated test model and replace the existing model with the recent model. Through this approach the proposed framework achieves high computational speed and accurate prediction for live SCADA traffic of power grids.

VII. Conclusions

This paper presents a RFE- XGBoost based feature selection approach along with the majority vote-based ensemble method for intrusion detection in power grids. The proposed framework comprises of three key elements, namely, data preprocessing, feature selection, and anomaly detection. Initially, during data preprocessing, the features are mapped and scaled to a specific range. The RFE-XGBoost based feature selection approach is subsequently applied on filtered data to compute the most stable features from the entire dataset. This approach enhances the learning efficiency. Furthermore, the selection of the features is carried out dynamically according to network traffic. In the subsequent stage, these reconstructed datasets are used by nine heterogeneous classifiers to predict the various attacks and normal events. Finally, the majority vote-based ensemble algorithm is applied to predict the output based on the majority of the class labels predicted by each of the nine classifiers.

The experimental results reveal that the proposed framework fares well in terms of accuracy, detection rate, precision, and recall. Moreover, the proposed model outperforms some of the state-of-the-art published techniques. The model offers a blend of effectiveness with precision, as it uses the limited number of stable features, and the classification is carried out based on combined predictions of nine most promising classifiers. Moreover, this combination requires limited computational cost, which is one of the crucial factors for mission-critical applications. Thus the proposed model has the potential to leverage the competencies of real-time SCADA systems for power grids.

Acknowledgments

Thanks to Brian Stacey and Rohit Joshi of Cistel Technologies for their valuable feedback.

References

[1] B. Krebs, "Cyber incident blamed for nuclear power plant shutdown," Washington Post. [Online]. Available: http://www.washingtonpost.com/wp-dyn/content/article/2008/06/05

[2] B. Kesler, "The vulnerability of nuclear facilities to cyber attack," *Strategic Insights*, vol. 10, no. 1, pp. 15–25, 2011.

[3] H. Xu, Y. Lin, X. Zhang, and F. Wang, "Power system parameter attack for financial profits in electricity markets," *IEEE Trans. Smart Grid*, vol. 11, no. 4, pp. 3438–3446, Jul. 2020.

[4] "SANS and Electricity Information Sharing and Analysis Center (e-isac). analysis of the cyber attack on the ukrainian power grid," Accessed: Sept. 28, 2019. [Online]. Available: http:// www.nerc.com/pa/CI/ESISAC/Documents/E-ISAC_SANS_Ukraine_DUC_18Mar2016.pdf

[5] K. Poulsen, "Feature importance of feature selection," 2003. Accessed: Sept. 10, 2019. [Online]. Available: https://www.securityfocus.com/news/6767

[6] N. Kshetri and J. Voas, "Hacking power grids: A current problem," *Computer*, vol. 50, no. 12, pp. 91–95, Dec. 2017.

[7] D. Upadhyay and S. Sampalli, "Scada (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations," *Comput. Secur.*, vol. 89, 2020, Art. no. 101666.

[8] B. Sussman, "Revealed: Details of 'first of its kind' disruptive power grid attack," Accessed: Mar. 21, 2020. [Online]. Available: https://www.secureworldexpo.com/industry-news/first-u.s.-power-grid-attack-details

[9] R. C. Borges Hink *et al.*, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst.*, Aug. 2014, pp. 1–8.

[10] S. Pan, T. Morris, and U. Adhikari, "Classification of disturbances and cyber-attacks in power systems using heterogeneous time-synchronized data," *IEEE Trans. Ind. Inform.*, vol. 11, no. 3, pp. 650–662, Jun. 2015.

[11] S. Pan, T. Morris, and U. Adhikari, "Developing a hybrid intrusion detection system using data mining for power systems," *IEEE Trans. Smart Grid*, vol. 6, no. 6, pp. 3104–3113, Nov. 2015.

[12] M. Keshk *et al.*, "Privacy preservation intrusion detection technique for scada systems," in *Proc. Mil. Commun. Inf. Syst. Conf.*, Nov. 2017, pp. 1–6.

[13] N. Moustafa, E. Adi, B. Turnbull, and J. Hu, "A new threat intelligence scheme for safeguarding industry 4.0 systems," *IEEE Access*, vol. 6, pp. 32 910–32 924, 2018.

[14] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 1, pp. 66–79, Jan.–Mar. 2021.

[15] D. Upadhyay *et al.*, "Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 1104–1116, Mar. 2021.

[16] U. Adhikari *et al.*, "Industrial control system (ics) cyber attack datasets," datasets used in the experimentation. [Online]. Available: https://sites.google.com/a/uah.edu/tommy-morris-uah/ics-data-sets

[17] S. Ghosh and S. Sampalli, "A survey of security in scada networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.

[18] E. D. Knapp and R. Samani, *Applied Cyber Security and the Smart Grid: Implementing Security Controls into the Modern Power Infrastructure*, 1st ed. Syngress Publishing, 2013.

[19] S.-J. Kim *et al.*, "Network anomaly detection for m-connected scada networks," in *Proc. 8th Int. Conf. Broadband Wireless Comput., Commun. Appl.*, 2013, pp. 351–354.

[20] A.-S. K. Pathan, *The State of the Art in Intrusion Prevention and Detection*. Boston, MA, USA: Auerbach Publications, 2014.

[21] Y. Yang *et al.*, "Rule-based intrusion detection system for scada networks," in *Proc. 2nd IET Renewable Power Gener. Conf.*, Sep. 2013, pp. 1–4.

[22] C.-C. Sun, A. Hahn, and C.-C. Liu, "Cyber security of a power grid: State-of-the-art," *Int. J. Elect. Power Energy Syst.*, vol. 99, pp. 45–56, 2018.

[23] S. Tan, "Neighbor-weighted k-nearest neighbor for unbalanced text corpus," *Expert Syst. Appl.*, vol. 28, no. 4, pp. 667–671, 2005.

[24] L. Maglaras, "Intrusion detection in scada systems using machine learning techniques," Ph.D. dissertation, Dept. Comput. Informat., Univ. Huddersfield, U.K., 2018.

[25] Z. Xu *et al.*, "Gradient boosted feature selection," in *Proc. 20th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining (KDD '14)*, 2014, pp. 522–531.

[26] R. Singh, M. Kalra, and S. Solanki, "A hybrid approach for intrusion detection based on machine learning," in *Proc. Int. Conf. Intell. Sustain. Syst.*, 2019, pp. 187–192.

[27] L. Rokach, "Ensemble-based classifiers," *Artif. Intell. Rev.*, vol. 33, no. 1, pp. 1–39, 2010.

[28] J. Surowiecki, *The Wisdom of Crowds*. New York, NY, USA: Anchor Books, 2005.

[29] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[30] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. ser. Springer Series in Statistics. Berlin, Germany: Springer, 2009.

[31] D. H. Wolpert, "Stacked generalization," *Neural Netw.*, vol. 5, no. 2, pp. 241–259, 1992.

[32] J. Heinermann and O. Kramer, "Machine learning ensembles for wind power prediction," *Renewable Energy*, vol. 89, pp. 671–679, 2016.

[33] O. Kramer, F. Gieseke, and B. Satzger, "Wind energy prediction and monitoring with neural computation," *Neurocomput.*, vol. 109, pp. 84–93, Jun. 2013.

[34] Y. Wang, Y. Shen, and G. Zhang, "Research on intrusion detection model using ensemble learning methods," in *Proc. 7th IEEE Int. Conf. Softw. Eng. Service Sci.*, 2016, pp. 422–425.

[35] N. T. Pham *et al.*, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proc. Australas. Comput. Sci. Week Multiconference*, 2018, pp. 1–6.

[36] F. Idrees *et al.*, "Pindroid: A novel android malware detection system using ensemble learning methods," *Comput. Secur.*, vol. 68, pp. 36–46, 2017.

[37] A. H. Mirza, "Computer network intrusion detection using various classifiers and ensemble learning," in *Proc. 26th Signal Process. Commun. Appl. Conf.*, 2018, pp. 1–4.
[38] M. Abirami, U. Yash, and S. Singh, "Building an ensemble learning based algorithm for improving intrusion detection system," in *Artif. Intell. Evol. Comput. Eng. Syst*. Springer, 2020, pp. 635–649.
[39] X. Li *et al.*, "Sustainable ensemble learning driving intrusion detection model," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 4, pp. 1591–1604, Jul./Aug. 2021.
[40] X. Gao, C. Shan, C. Hu, Z. Niu, and Z. Liu, "An adaptive ensemble machine learning model for intrusion detection," *IEEE Access*, vol. 7, pp. 82512–82521, 2019.
[41] N. Moustafa *et al.*, "Dad: A distributed anomaly detection system using ensemble one-class statistical learning in edge networks," *Future Gener. Comput. Syst.*, vol. 118, pp. 240–251, 2021.
[42] N. Littlestone and M. Warmuth, "The weighted majority algorithm," *Inf. Computation*, vol. 108, no. 2, pp. 212–261, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0890540184710091
[43] A. Kusiak, H. Zheng, and Z. Song, "Short-term prediction of wind farm power: A data mining approach," *IEEE Trans. Energy Convers.*, vol. 24, no. 1, pp. 125–136, Mar. 2009.
[44] S. Hassan, A. Khosravi, and J. Jaafar, "Examining performance of aggregation algorithms for neural network-based electricity demand forecasting," *Int. J. Elect. Power Energy Syst.*, vol. 64, pp. 1098–1105, 2015.
[45] X. Lin, X. Zhang, and X. Xu, "Efficient classification of hot spots and hub protein interfaces by recursive feature elimination and gradient boosting," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 17, no. 5, pp. 1525–1534, Sep.–Oct. 2020.
[46] C. Strobl *et al.*, "Bias in random forest variable importance measures: Illustrations, sources and a solution," *BMC Bioinf.*, vol. 8, no. 1, p. 25, 2007.
[47] K. M. Ting, *Precision and Recall*. Boston, MA, USA: Springer, 2010, pp. 781–781. [Online]. Available: https://doi.org/10.1007/978-0-387-30164-8_652

Darshana Upadhyay received the master's degree in computer science from Nirma University, Ahmedabad, India. She is currently working toward the Ph.D. degree at the Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada. Prior to starting her Ph.D., she worked as a Lecturer at Nirma University, Ahmedabad, India. For her master's thesis, she has completed a novel project in the area of linear feedback shift register design for secure systems. Her primary research includes algorithm conceptualization, hardware design in the field of embedded systems, vulnerability assessments, and intrusion detection techniques for IoT/SCADA based systems. She was the co-recipient of the Indo-Canadian Shastri research grant in the field of wireless security and intrusion detection systems. She has been invited to be one of the Women in International Security - Canada's 2020 Emerging Thought Leaders. She was awarded the 2020–2021 Citizenship Award from the Faculty of Computer Science, Dalhousie University, for being known as a congenial, reliable, mature person who is respected by peers, and for building a community atmosphere within the faculty. She was also awarded the Gold Medal for securing the first position during her master's degree.

Jaume Manero received the Ph.D. degree in artificial intelligence from the Technical University of Catalonia, Barcelona, Spain. He is currently with the Barcelona Supercomputing Center, Barcelona, Spain. His Ph.D. dissertation was deep learning architectures applied to wind time series forecasting. He is also a Visiting Research Scientist with Dr. Srini Sampalli's MYTech Lab (Emerging Wireless Technologies) where he is working on how deep learning can impact in the development of cyber-security applications.

Marzia Zaman received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Memorial University of Newfoundland, St. John's, NL, Canada, in 1993 and 1996, respectively. In 1990, she joined the Nortel Networks, Ottawa, ON, Canada, where she joined the Software Engineering Analysis Lab and later joined the Optera Packet Core project as a Software Developer. In addition, she has many years of industry experience as a Researcher and Software Designer with Accelight Networks, Excelocity, Sanstream Technology, and Cistel Technology, Ottawa, ON, Canada. Since 2009, she has been with the Centre for Energy and Power Electronics Research, Queen's University, Canada and one of its industry collaborators, Cistel Technology, on multiple power engineering projects. Her research interests include renewable energy, wireless communication, IoT, cyber security, machine learning, and software engineering.

Srinivas Sampalli (Member, IEEE) received the bachelor of engineering degree from Bangalore University, Bangalore, India and the Ph.D. degree from the Indian Institute of Science, Bangalore, India, and is currently a Professor and National 3M Teaching Fellow in the Faculty of Computer Science, Dalhousie University, Halifax, NS, Canada. He has led numerous industry-driven research projects on Internet of Things, wireless security, vulnerability analysis, intrusion detection and prevention, and applications of emerging wireless technologies in healthcare. He currently oversees and runs the Emerging Wireless Technologies Lab and has supervised over 150 graduate students in his career. His primary joy is in inspiring and motivating students with his enthusiastic teaching. He is the recipient of the Dalhousie Faculty of Science Teaching Excellence Award, the Dalhousie Alumni Association Teaching Award, the Association of Atlantic Universities' Distinguished Teacher Award, a teaching award instituted in his name by the students within his Faculty, and the 3M National Teaching Fellowship, Canada's most prestigious teaching acknowledgement. Since September 2016, he holds the honorary position of the Vice President (Canada), of the International Federation of National Teaching Fellows, a Consortium of National Teaching Award winners from around the world.

An Efficient Key Management and Multi-Layered Security Framework for SCADA Systems

Darshana Upadhyay, Marzia Zaman, Rohit Joshi, and Srinivas Sampalli, *Member, IEEE*

***Abstract*—Supervisory Control and Data Acquisition (SCADA) networks play a vital role in industrial control systems. Industrial organizations perform operations remotely through SCADA systems to accelerate their processes. However, this enhancement in network capabilities comes at the cost of exposing the systems to cyber-attacks. Consequently, effective solutions are required to secure industrial infrastructure as cyber-attacks on SCADA systems can have severe financial and/or safety implications. Moreover, SCADA field devices are equipped with microcontrollers for processing information and have limited computational power and resources. This makes the deployment of sophisticated security features challenging. As a result, effective lightweight cryptography solutions are needed to strengthen the security of industrial plants against cyber threats. In this paper, we have proposed a multi-layered framework by combining both symmetric and asymmetric key cryptographic techniques to ensure high availability, integrity, confidentiality, authentication and scalability. Further, an efficient session key management mechanism is proposed by merging random number generation with a hashed message authentication code. Moreover, for each session, we have introduced three symmetric key cryptography techniques based on the concept of Vernam cipher and a pre-shared session key, namely, random prime number generator, prime counter, and hash chaining. The proposed scheme satisfies the SCADA requirements of real-time request response mechanism by supporting broadcast, multicast, and point to point communication.**

***Index Terms*—SCADA Systems, random number generator, symmetric key cryptography, public key algorithm, cyber security, network attacks, key management.**

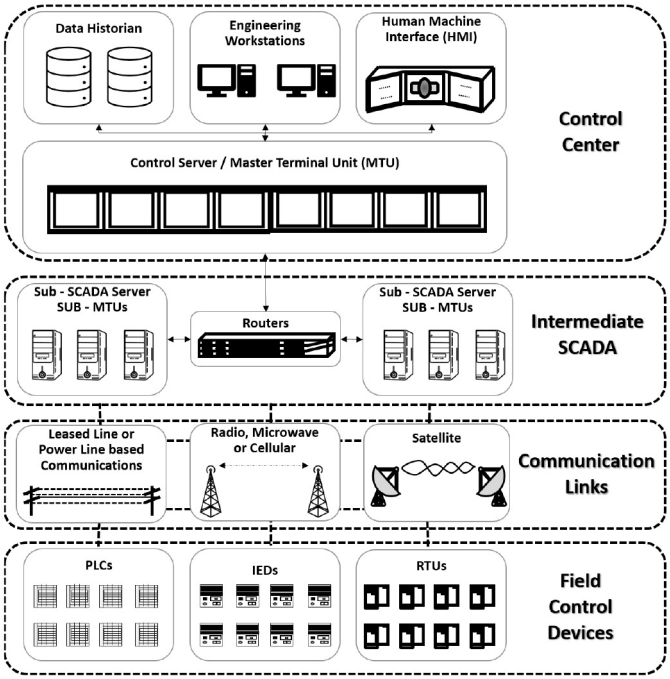


Fig. 1. Block diagram of a SCADA system, Legend: MTU: Master Terminal Unit, PLCs: Programmable Logic Controllers, RTUs: Remote Terminal Units, IEDs: Intelligent Electronic Devices.

I. Introduction

THERE has been a surge in the deployment of Supervisory Control and Data Acquisition (SCADA) systems to control and monitor industrial infrastructure over the Internet [1]. Organizations such as oil and natural gas, power stations, water & sewage systems, chemical plants, manufacturing units, railway, and other transportation use SCADA systems to monitor and control their infrastructure such as oil pipelines, solar panels, water pipelines, boilers, railway tracks, and plant floor components across open access networks [2], [3].

A SCADA system typically includes a control server (also known as Master Terminal Unit (MTU)), SUB-MTUs, communication links (e.g., satellite, radio or microwave links,cellular network, switched or lease lines and power-lines), and geographically dispersed field control devices, namely, Programmable Logic Controllers (PLCs), Remote Terminal Units (RTUs), and Intelligent Electronic Devices (IEDs) [2], [4]. The block diagram of a typical SCADA system is depicted in Figure 1.

For continuous monitoring and control of plant floor devices, sensors and actuators are used to measure different attributes of machinery and transmit that information to field devices [5]. Further, the field control devices, namely, PLCs, RTUs, and IEDs supply digital status information to the MTU (typically placed at the remote location) to determine the acceptable ranges according to parameters set in the server. This information will then be transmitted back to the field control device(s) where actions may be taken to optimize the

Manuscript received October 29, 2020; revised March 9, 2021 and June 22, 2021; accepted August 4, 2021. Date of publication August 17, 2021; date of current version March 11, 2022. The authors gratefully acknowledge the support in part by the Natural Sciences and Engineering Research Council (NSERC), Canada, through a Collaborative Research Grant. The associate editor coordinating the review of this article and approving it for publication was J. Zhang. *(Corresponding author: Srinivas Sampalli.)*

Darshana Upadhyay and Srinivas Sampalli are with the Faculty of Computer Science, Dalhousie University, Halifax, NS B3H 1W5, Canada (e-mail: srini@cs.dal.ca).

Marzia Zaman and Rohit Joshi are with the Research and Development Department, Cistel Technology Inc., Ottawa, ON K2E 7V7, Canada.

Digital Object Identifier 10.1109/TNSM.2021.3104531

performance of the system. Moreover, the status information is stored in a database and is displayed on a Human Machine Interface (HMI) at the control center, where operators can interact with the plant floor machinery for centralized monitoring and system control [6]. Large SCADA networks such as those on a power plant require hundreds of field devices and dedicated subsystems to reduce the load on the centralized server [2].

SCADA communication messages have sensitive information as they are used to monitor and control the plant floor devices. For example, in water and sewage systems, the communication messages are used to raise and lower water tank levels or open and close the safety valves. Since these control devices are operated and monitored remotely, they can make them high-value targets for attackers to launch various cyber-attacks that can compromise the control systems, communication, and emergency services. Consequently, one of the critical aspects of the SCADA systems is secure transmission of messages so that they cannot be tampered during the communication. Moreover, the SCADA devices must be authenticated and maintain confidentiality of the information during the transmission so that no interceptor can misuse the system.

In the last few years, many key management techniques have been published to secure SCADA communication, namely, SCADA key establishment (SKE), SCADA Key Management Architecture (SKMA), Advanced SCADA Key Management Architecture (ASKMA), Hybrid Key Management Architecture (HKMA) and Advanced Hybrid SCADA Key Management Architecture (AHSKMA), Limited Self-Healing key distribution (LiSH) [7], [8], [9], [10], [11], [12]. These techniques fall under two main categories, namely, centralized key management and decentralized key management schemes. Moreover, each of these categories uses three approaches to generate and extract the session key, namely, symmetric, asymmetric, and hybrid. The drawback of the centralized scheme is that if the key distribution center (KDC) is down, the communication is cut off, which is not acceptable in SCADA systems. In a decentralized approach, the keys are created using keying material and may only affect the single communication link in case of a breakdown.

The symmetric key based approach is efficient in terms of message integrity and high availability, but does not provide authentication and confidentiality. On the other end, asymmetric key provides message integrity, authentication, and privacy, but may compromise availability. Hence, hybrid techniques are more suitable for SCADA systems. A few key management techniques have been proposed using hybrid methods. For example, Rezai *et al.* [10] propose an advanced Hybrid key management architecture (HSKMA), which improves the key management architecture proposed by Choi *et al.* [11]. However, it uses a centralized KDC to distribute the keys. Moreover, the communication between the MTU and the sub-MTU is established using Elliptic-Curve Cryptography (ECC) based asymmetric key cryptography while the sub-MTU and the RTU communicate using Rivest–Shamir–Adleman (RSA) asymmetric key cryptography. The same approach has been used to enhance the scheme proposed by Rezai *et al.* [13] using a decentralized system in [9]. In this scheme, the master keys are refreshed using ECC and symmetric cryptography is used for encryption, decryption, and session key updates. However, this scheme does not validate the message integrity and authentication. Moreover, none of the previous methods has practical implementation proof that it provides immunity against quantum attacks [14]. Furthermore, it has been known that RSA does not guarantee perfect forward secrecy [11]. In summary, none of the techniques covers all the security aspects.

The forgoing discussion brings in the need for an effective cryptography solution that will prevent these systems from potential breaches. The objective of this paper is to propose a robust & low-cost security framework for automated industries to mitigate various security flaws and cyber-attacks. The proposed work aims to offer a multi-layered security framework for industrial infrastructures by combining both symmetric and asymmetric key cryptography techniques. This novel approach follows a layered architecture, where the MTU and sub-MTU can communicate using a hybrid technique for an entire session while the sub-MTU and RTU can communicate using symmetric key cryptography once the session key is securely exchanged. Also, we have proposed a novel approach to generate symmetric keys using vernam cipher rather than using existing methods such as 3DES, AES, etc. Furthermore, the proposed scheme satisfies SCADA requirements of real-time request-response mechanism by supporting broadcast, multicast, and point-to-point communication.

A. Contributions of the Paper

1) We propose a secure session-key agreement scheme according to SCADA protocol standards to ensure the security amongst MTU, sub-MTUs and RTUs. For that, a true random number generator based on current date and time (CDT) and a fraction of the square root of a prime number (FSRP) are used to generate the session key. Moreover, these elements are shared by XORing them to enhance the privacy of the shared secrets. Furthermore, the dynamic HMAC is derived using the value of FSRP. Moreover, using these same elements, the HMAC is derived to validate the integrity of the message. This reusability of the elements increases the computational speed of session key, symmetric key and HMAC derivation. The randomness of key and HMAC offers immunity against various attacks such as correlation attacks, length extension attacks, etc.
2) We propose a novel approach to generate symmetric keys in the Vernam cipher by combining prime counter and hash chaining techniques. The mathematical property of the fraction square root of prime number (FSRP) is used, which returns a non-terminating, non-repeating irrational number. In a recent publication by Manjunatha *et al.* [15], the authors propose Vulgar fractions to generate a complex key with secured seed exchange for the Vernam cipher. This fraction is generated by dividing a small number by a large

prime number, resulting in a fraction number [15]. For example, frac(1/7) = 0.1428571428571 generates long strings with a repetitive sequence of digits. However, our proposed approach advances that method by generating completely random and non-repeating decimal numbers using the concept of FSRP. For example frac(sqrt (7)) = 0.645751311064590590501615753639 3 returns long strings without repetitive sequence of digits.

3) We propose a multi-layered framework by integrating the concept of symmetric and asymmetric key cryptography that ensures various security mechanisms, namely, authentication, confidentiality, message integrity, availability, and scalability for SCADA systems. The proposed method for symmetric key cryptography is based on the Vernam cipher, which provides protection against all the cryptographic attacks while the NTRU based post-quantum public-key algorithm resists quantum and data harvest attacks.
4) We identify an efficient cipher suite by comparing and analyzing various private and public key algorithms for the proposed framework by considering multiple factors, namely, prevention mechanism against classical and quantum attacks, key storage cost, the randomness of key and computational speed. The proposed cipher suite overcomes the weaknesses of the cipher suite offered by the American Gas Association (AGA) security standards [14], [16].

B. Outline of the Paper

The rest of this paper is organized as follows. Section II describes related research in the areas of key management and encryption schemes foe SCADA systems. Section III, presents the reasoning of choice of the algorithms. The proposed multi-layered framework for secure SCADA communication is introduced in Section IV, which covers secure key and information exchange. Section V presents the complete experimental setup which includes algorithm selection for cipher suites, computational speed of proposed framework, randomness evaluation of symmetric key, and calculation of the cost of the keys. Section VI presents the comparative studies with the state-of-the-art techniques in terms of security analysis, storage cost, and execution speed. Section VII concludes the paper.

II. Background and Related Work

A. Literature Survey

SCADA networks are typically configured using proprietary protocols such as Modbus, IEC 61850, IEC 60870, DNP3, and Profinet, which do not support secure data communication. Moreover, the remote procedure call (RPC) follows open link communication and one of the real-time examples of the consequent vulnerability was the Blaster worm [2]. Furthermore, many network sniffing tools are freely available to view and gather the network traffic [17]. Therefore, secure data transmission is one of the important requirements for SCADA systems. Key management and encryption play a vital role in securing SCADA communication. Typically, in a SCADA communication, the MTU sends control signals to the RTUs to control the plant floor devices, which require three types of communication, namely, broadcast, multicast, and point to point. However, controller RTUs may need to operate other field RTUs. In case of an emergency shutdown, to acquire the clock information or synchronization, MTUs broadcast the signal to all the control devices such as RTUs, IEDs, and PLCs. To operate a specific substation device, the MTU requires multicast communication, whereas monitoring and controlling the plant for machinery typically requires point-to-point communication. Therefore, while designing a secure framework for SCADA networks, it is crucial to cover all three types of communication.

During the last two decades, many key management schemes have been proposed, which typically fall into two categories, namely, centralized key distribution such as [4], [7], [18], [19], and decentralized key distribution scheme such as [9], [20], [21], [22]. In the centralized scheme, the Key Distribution Center (KDC) plays a vital role in generating and distributing the secret keys to establish secure communication between the communication parties. In contrast, the decentralized scheme requires pre-shared keying material that is used to create the session key. Once the session key is derived using keying essence, further communication takes place using that key. Furthermore, some key management schemes use the public key-based technique to establish secure transmission. Although this method is time-consuming and power-consuming, various research studies suggest that ECC is a suitable public-key cryptosystem [4], [9], [11].

Sandia Labs proposed a SCADA key establishment (SKE) method for managing cryptographic keys in the network [7]. This scheme is proposed for point-to-point communication amongst MTU, sub-MTU, and RTU and uses the symmetric key technique to establish secure communications between sub-MTUs and RTUs, while sub-MTUs and MTUs communicate using public key cryptography. For the symmetric key, the session key is generated using three types of keys, namely, long term key (LTK), general seed key (GSK), and general key (GK) [7]. KDC assigns public and private key pair to each sub-MTU and MTU. However, this method does not support broadcast, multicast, and RTU to RTU communication. Moreover, it increases the overall key storage overhead and complexity as the long-term keys are managed manually. In [19], the authors propose a SCADA Key Management Architecture (SKMA) for secure session key management, which enhances the capability of SKE. While the SKE uses both a public key algorithm and a symmetric key algorithm, the SKMA uses only symmetric encryption algorithm. SKMA generates a session key using a pseudorandom function, keyed by the node-node key, and a timestamp that is based on the duration of the session. SKMA uses key establishment protocol based on ISO 11770-2 mechanism [8]. However, the scheme does not provide secure message broadcasting but supports RTU-RTU communication. Moreover, it does not provide any confidentiality and integrity.

Advanced SCADA Key Management Architecture (ASKMA) supports both message broadcasting and secure communications. Furthermore, evenly spreading the total amount of computation across the high power nodes (MTU or

SUB-MTU) significantly avoids the performance bottleneck and keeps minimal burden on the low power nodes (RTU). It uses the LKH (Logical Key Hierarchy protocol) to construct a logical tree of symmetric keys. Each member knows all the symmetric keys from its leaf to the root, and if any new node joins the group, LKH updates the entire set of symmetric keys from its leaf to the root. Although the overall performance of ASKMA has many advantages, it can be less efficient during the multicast communication process. To solve this issue, ASKMA+ was proposed [7]. ASKMA+ divides the key structure into two classes, by applying the IoLus framework to construct each class as a logical key hierarchy (LKH) structure. Through this key structure, the authors proposed a more efficient key-management scheme supporting efficient multicast communication by considering the number of keys stored in a remote terminal unit (RTU). However, ASKMA+ does not address the availability issue in SCADA.

To satisfy the availability requirement, Hybrid Key Management Architecture (HKMA) and Advanced Hybrid Scada Key Management Architecture (AHSKMA) were proposed [10], but there is a chance that field devices will stop working during the replacement of field control devices. To solve this issue, Choi *et al.* propose a hybrid key management scheme [11]. A centralized key distribution (CKD) protocol is applied between the sub-MTU and MTU, and LKH protocol is applied between sub-MTU and RTU. However, if the centralized key distribution server breaks down, the entire approach fails to execute the protocol. Rezai *et al.* [9] also use a hybrid key management method using ECC. Jiang *et al.* [12] propose Limited Self-Healing key distribution (LiSH), which offers revocation capabilities along with collusion-resistance for group communication in SCADA systems. The LiSH+ is used to address the dynamic revocation mechanism, which enhances the base method of LiSH. Kang *et al.* [21] propose a scheme for radial SCADA systems based on a pre-shared session key that relies on symmetric key cryptography. This solution enhances the performance of the radial SCADA system by using the master key concept.

AGA-12, Part 2, provides security features offering a new security protocol standard [23]. It uses cipher suites to secure communication amongst SCADA field devices, which covers authentication, confidentiality, and integrity. However, it fails to provide faster execution. Furthermore, it does not offer prevention against quantum and Denial of Service (DoS) attacks. In addition, AGA-12 uses the RSA algorithm for encryption, which was recently cracked and also does not provide key management [14]. The other security standards, such as IEC 62210, IEC 62351, fail to offer security against man-in-the-middle (MiM) attacks and also lack strong key management. A novel key distribution method was proposed for smart grids in [24] which uses identity-based cryptography. This method adopts a hybrid approach to counteract man-in-the-middle and replay attacks. However, this method does not cover the authentication of the SCADA components. The authors in [25] introduce the authentication and authorization roles for SCADA devices using attribute-based access control. The hybrid Diffie-Key exchange, along with the authentication scheme, was proposed in [26]. This scheme uses RSA and AES for session key generation and encryption. However, it does not provide high availability.

B. Research Gaps

Originally, the objective of SCADA systems was to focus on accurate and efficient process execution at the plant floor rather than aiming to secure communication. While accessing the plant machinery remotely through SCADA systems accelerates the industrial processes, it compromises the security by exposing the systems to the outside world [24]. Consequently, unauthorized parties such as hackers, intelligent foreign agents, and corporate saboteurs, can exploit the weaknesses to compromise industrial systems. Typically, general safeguards include restricted perimeters, patch management, strong cryptography and most importantly, separation of the control network and corporate network through the defense-in-depth mechanism [1], [22]. However, these security guards are difficult to deploy owing to legacy-inherited security weaknesses, and that significantly increases the chances of possible exploitation during real-time communication [2], [27].

Moreover, SCADA field devices such as PLCs, RTUs, and IEDs have resource and computational power limitations that make the deployment of sophisticated security features challenging [9]. Furthermore, availability, integrity, and confidentiality are the three fundamental security requirements of SCADA communication [19]. To circumvent threats against these security requirements, a robust security framework for key management schemes and lightweight encryption techniques are needed [9], [20]. Although many key management and encryption techniques have been proposed, few methods exist for secure key exchange for point-to-point communication while some are specifically intended for broadcast and multicast communication. Furthermore, none of the schemes satisfy all the requirements of secure SCADA communication and real-time request-response mechanism. Some private key based methods offer integrity and availability, while some public key based methods provide authentication and confidentiality. Hence, neither private nor public key based approach alone is sufficient [4], [28]. The development of a secured SCADA framework with hybrid efficient key management scheme and lightweight cipher is the primary research gap that is addressed in this paper.

C. Proposed Solution

The proposed system aims to provide a multi-layered security framework for industrial infrastructures by combining both symmetric and asymmetric key cryptography techniques. This novel approach covers major security aspects of the systems, namely availability, integrity, confidentiality, authentication and scalability. For that, an efficient session key management mechanism has been proposed besides lightweight ciphers by merging the concept of random number generator and Hashed Message Authentication Code (HMAC). Moreover, for each session, three symmetric key cryptography techniques are introduced, namely, random prime number generator,

prime counter, and hash chaining based on the concept of Vernam cipher and pre-shared session key. Furthermore, the proposed scheme satisfies SCADA requirements such as real-time request response mechanism by supporting broadcast, multicast, and point to point communication.

III. Reasoning of Choice of Algorithms

Many SCADA-based industrial systems, such as water & sewage control, energy and power plants, and gas pipelines, rely on real-time communication with limited computational resources. We have used the Vernam cipher for symmetric key cryptography because it is proven to offer an absolute secure solution theoretically, is easy to implement, and accelerates encryption & decryption by using low power and memory [29]. Therefore, it is an appropriate solution for embedded system devices. Moreover, the modulo-2 operator (XOR) used in the Vernam cipher provides faster execution and the flexibility in design of the onboard hardware [15]. By employing these features of the Vernam cipher, we can protect the data with low computational power and memory utilization.

The Vernam cipher provides complete secrecy as the key is unique and completely random for each message. An amount of time that is necessary to break any cipher and tamper with the data is based on the size and nature of the symmetric key. However, in the Vernam cipher, as the keys are random and unique for every message, an eavesdropper will be unable to guess the key even with unlimited computing power. Even asymmetric ciphers such as RSA can be broken with unlimited time and processing power [14]. Furthermore, the frequency analysis of the Vernam cipher is evenly distributed, and hence cryptanalysis will not produce any meaningful information.

The focus of the proposed framework is to provide high security along with high availability since SCADA communication depends on real-time request-response mechanisms. We can replace digital signature and asymmetric key cryptography by applying HMAC in symmetric key cryptography. This approach provides message authentication and integrity without compromising the execution speed during the communication between MTU and RTU.

Typically, HMAC depends on a shared secret key, which is exchanged using a trusted channel (in our case, we have used NTRU-based asymmetric key cryptography) between the sender and receiver to agree on the same key before starting the information exchange. The same secret key is combined with the MAC to generate HMAC at both the communication devices. However, the cryptographic strength of the HMAC depends on the size of the secret key, since brute force attacks are the most common attacks against HMAC.

In a typical key distribution scenario, the secret key is distributed over the trusted channel. Instead, in our proposed approach, we exchange the parameters of FSRP & CDT such as the index of FSRP and keysalt which are used to generate the secret key. Moreover, these parameters are reusable, and are not only used to generate the session key but also are applied to produce the key for HMAC. Furthermore, the key used in HMAC depends on the value of FSRP, which is generated by a random prime generator or a prime counter to produce a new key for each message. This makes brute force attacks computationally infeasible as the secret key used in HMAC is dynamically generated.

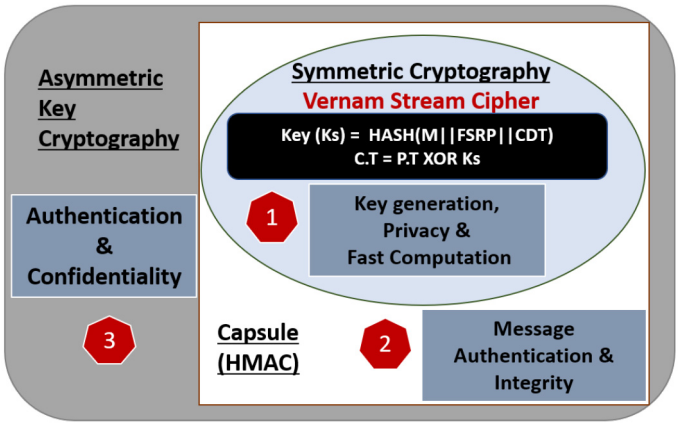


Fig. 2. Multi-layered framework for secure SCADA communication.

IV. Multi-Layered Framework for Secure SCADA Communication

This section presents the proposed multi-layered framework for secure SCADA communication. The framework uses three levels for robustness, namely, symmetric key cryptography, cryptographically secure HMAC function, and a public key algorithm. The security features of each phase are illustrated in Figure 2.

In our framework, a unique session key is generated for each connection between SCADA communication devices. The elements of this session key are securely shared using asymmetric key cryptography. This is called the key agreement stage. Furthermore, during this phase, the sender's authentication and recipient confidentiality are also validated using the private-public key pair. Moreover, HMAC is used for message authentication and integrity. Once both the communication parties agree on the reliable key exchange, further communications take place using symmetric key cryptography. The encryption of the original message is hashed, and subsequently, the symmetric keys are generated to encrypt the message using the lightweight Vernam cipher. After that, the cipher text and hash digest of this encrypted message are sent together over the communication channel. At the other end, the receiver device validates the message integrity using HMAC and then the cipher text is decrypted to receive sender's original message.

Since ICSs control field-site components at the plant floor, the activities related to controlling and monitoring of the elements should be done securely and efficiently [30]. For that, we have introduced two modules, namely, secure key exchange and secure information exchange. Moreover, secure information exchange consists of four methods, namely, Multi-Layered (ML) architecture, Random Prime Generator (RPG), Prime Counter (PC), and Hash Chaining (HC). While ML and HC offer very high security in SCADA networks, PC and HC

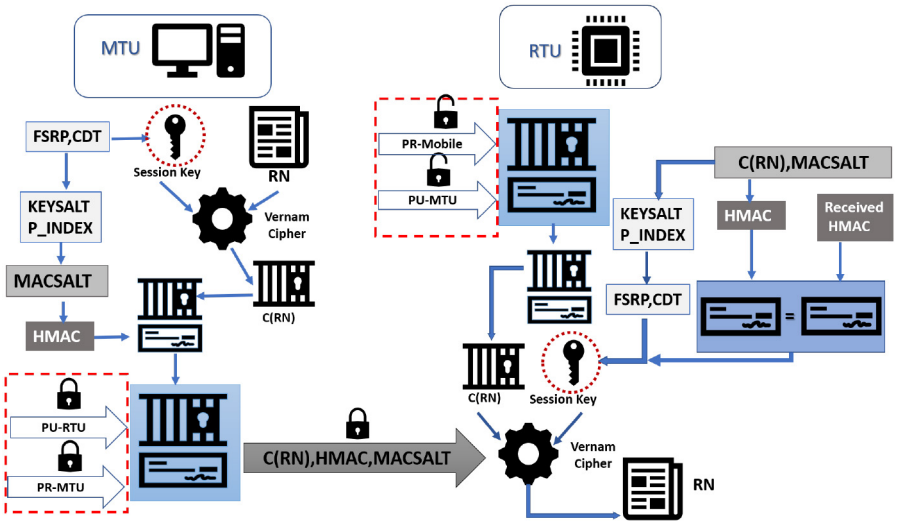


Fig. 3. Secure Key exchange mechanism for SCADA systems.

are proposed for time-critical applications. The RPG offers medium level security and availability.

A. Secure Key Exchange

The key agreement refers to three stages, namely, key generation at the sender side, key distribution over the communication channel, and key extraction at the receiver side.

1) Key Generation: During the key generation phase, a sender (MTU or RTU) uses three main elements, namely, a Random Number (*RN*), Current Date & Time (*CDT*), and Fraction of Square Root of Prime number (*FSRP*). Here, *CDT* and *FSRP* are used as secret elements to generate the session key. The choice of these two key elements is based on the property of generating true random numbers. *CDT* generates a random number every microsecond and to make it more random, we choose *FSRP*, which returns a non-terminating, non-repeating decimal number [31]. The session key (S_K) is derived by applying a hash function on both these elements by combining them, as in eq. (1).

$$S_K = \text{HASH}(CDT||FSRP) \tag{1}$$

These session key elements are securely distributed using *MACSALT*. The index of *FSRP* is combined with *KEYSALT* to generate *MACSALT*, where *KEYSALT* is derived by *XORing CDT* and *FSRP*. The formulas are given in eq. (2) & (3).

$$KEYSALT = CDT \oplus FSRP \tag{2}$$

$$MACSALT = KEYSALT||PRIME_{index} \tag{3}$$

Once S_K and *MACSALT* are generated, *RN* is encrypted using S_K which generates cipher of random number *C*(*RN*), as in eq. (4).

$$C(RN) = RN \oplus S_K \tag{4}$$

In this process, the algorithm produces a hash not only from the encrypted *RN* but also from the *CDT* & *FSRP* key elements. This derivation follows the procedure of *HMAC*, as given in the eq. (5) and is used to check message integrity.

$$HMAC_{sender} = \text{HASH}(C(RN), CDT||FSRP). \tag{5}$$

2) Key Distribution: The bundle of the *C*(*RN*), *HMAC* of *C*(*RN*), and *MACSALT* is securely sent over the communication channel using the private key of sender's (K_{spri}) and public key of receiver (K_{rpub}) that validate the sender's authentication and receiver's confidentiality as in eq. (6).

$$K_{rpub}\big(K_{spri}(C(RN), HMAC_{sender}, MACSALT)\big). \tag{6}$$

3) Key Extraction: At the receiver side, the private key of receiver and public key of sender is applied to validate authentication and confidentiality as in eq. (7).

$$Krpr(Kspu(C(RN), HMAC_{sender}, MACSALT))) \tag{7}$$

The elements of *MACSALT* are used to generate *FSRP* and *CDT*. $PRIME_{index}$ is used to extract the value of *FSRP* and *CDT* is obtained by XORing *FSRP* and *KEYSALT* as shown below in eq. (8)-(10).

$$MACSALT = KEYSALT||PRIME_{index} \tag{8}$$

$$FSRP = FRAC(SQRT(PRIME_{index})) \tag{9}$$

$$KEYSALT = CDT \oplus FSRP \tag{10}$$

Finally, the session key is derived by applying hash on *CDT* and *FSRP* as in eq. (11).

$$CDT = FSRP \oplus KEYSALT \tag{11}$$

$$HMAC_{receiver} = HASH(C(RN), CDT||FSRP) \tag{12}$$

HMAC is computed at the receiver using *C*(*RN*), *CDT* & *FSRP*, as in eq. (12) to compare with $HMAC_{sender}$ to check data integrity. The *HMAC* of the sender and receiver are checked, if both are equal it moves to the next step, else the message is discarded. The session key S_K is then validated using *CDT* and *FSRP* as in eq. (13). The session key is XORed with *C*(*RN*) to get the RN as shown in eq. (14).

$$S_K = \text{HASH}(CDT||FSRP) \tag{13}$$

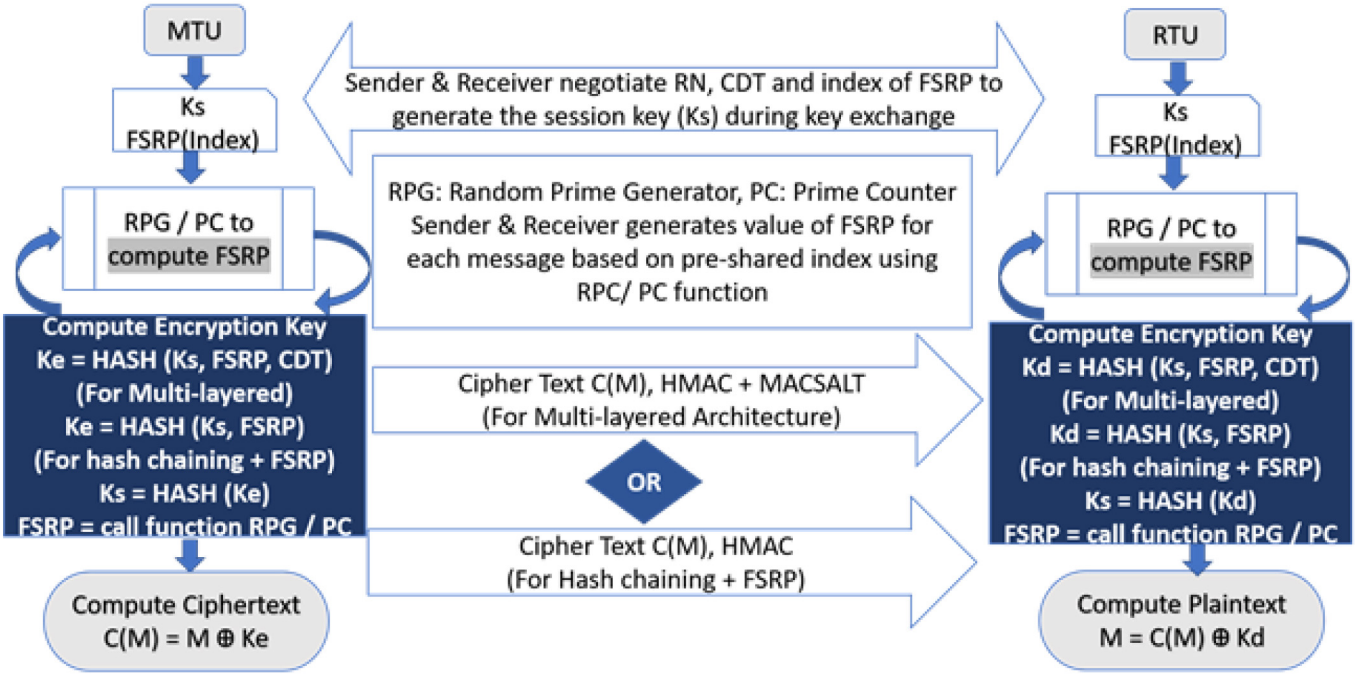


Fig. 4. Process diagram of encryption and decryption of data in secure SCADA communication.

$$RN = C(RN) \oplus S_K \tag{14}$$

The receiver will send an acknowledgement to the sender by encrypting $RN + 1$ using the same session key to validate secure key exchange. Figure 3 illustrates the secure key exchange mechanism between SCADA devices, namely, MTU and RTU.

The proposed scheme uses RN, CDT, and FSRP to generate the session key (Ks) for both the communication devices, namely, MTU and RTU. However, during the key exchange, these elements are not transferred openly, rather RN is encrypted by the key generated using the combination of CDT & FSRP. Moreover, the modulo-2 operator (XOR) is applied on CDT and FSRP to generate the keysalt which will be shared over the communication channel along with an index of FSRP and the cipher text of RN. The index of FSRP is considered as the root of trust for the entire scheme. Furthermore, the Vernam cipher is used for symmetric key cryptography, which requires a fresh key for each message during the encryption and decryption process. This symmetric key is generated using the session key (Ks) and key parameters, namely, CDT and FSRP, depending on the proposed approaches. The FSRP can be generated using a random prime generator (Method 2) or a prime counter (Method 3). Furthermore, hash chaining (Method 4) can be combined with any of these approaches to generate a new fresh symmetric key for the Vernam cipher.

B. Secure Information Exchange

In SCADA systems, the field site components are controlled and monitored using short messages communicated between RTU and MTU. Based on the reading obtained from the field control devices, namely, RTU, PLC, and IED, the SCADA master (MTU) makes a proper decision and sends an appropriate signal to the field components to operate plant machinery. Generally, the control messages are short in length (typically 256 bits), which control the sensors and actuators of plant machinery. For example, in water management systems, the signals used during communication include OPEN/CLOSE the valve, SWITCH_ON/SWITCH_OFF the devices, RAISE/LOW the water level tank, etc. [22]. Such systems operate using short messages. Hence the average length of the control message consists of 24 to 32 characters (192 to 256 bits) for one frame.

The Vernam cipher requires the same length for key and message. Moreover, each communication message requires a distinct key for encryption and decryption. To generate such a unique key every time, we have proposed two main approaches, namely, multi-layered architecture, and hash chaining with FSRP. Moreover, both these approaches are further divided in the multiple methods to generate a unique value of FSRP, namely, random prime generator (RPG), and prime counter (PC). Figure 4 illustrates the symmetric key generation process used to encrypt and decrypt the message at both the communication endpoints. Both the sender and receiver negotiate RN (random number), CDT (current date and time), and the index number of FSRP (which acts as a seed for random prime generator/ prime counter) to generate session key (Ks). Using RPG/PC, both the sender and receiver generate a distinct FSRP for each message. Moreover, Blake2s (cryptographically secure hash function [32]) is applied on the session key and FSRP to generate the encryption key (Ke). Similarly, the receiver produces the decryption key (Kd) using the pre-shared Ks and the value of FSRP. Note that, the value of the symmetric key not only depends on the previous key but also on the value of FSRP (which is generated using RPG/PC). In the case of our multi-layered architecture, instead of two parameters, both, MTU and RTU use three parameters, namely, CDT,

Algorithm 1: Multi-Layered (Hybrid)

Input: M = Input Message
begin
  Sender:
  while *(Session!=END)* **do**
    (1) Generate *CDT*
    (2) Generate *FSRP*
    (3) $K_e \leftarrow$ HASH(K_s, *CDT*, *FSRP*)
    (4) $C(M) \leftarrow M \oplus K_e$, $K_s \leftarrow K_e$
    (5) $HMAC_S \leftarrow$ HASH($C(M)$, *CDT* || *FSRP*)
    (6) *KEYSALT* $\leftarrow$ *FSRP* $\oplus$ *CDT*
    (7) *MACSALT* $\leftarrow$ *KEYSALT* || $Index$
    (8) $Bundle \leftarrow K_{rpub}(K_{spri}(C(M), HMAC_S,$ *MACSALT*))
  end
  Receiver:
  while *(Session != END)* **do**
    (1) $Bundle \leftarrow K_{rpri}(K_{spub}(C(M), HMAC_S,$ *MACSALT*))
    (2) $FSRP = Frac(Sqrt(PRIME(Index)))$
    (3) $CDT \leftarrow KEYSALT \oplus FSRP$
    (4) $HMAC_R \leftarrow$ HASH($C(M)$, *CDT* || *FSRP*)
    if *(($HMAC_S$, $HMAC_R$) == TRUE)* **then**
      (5) $K_d \leftarrow$ HASH(K_s,*CDT*,*FSRP*)
      (6) $M \leftarrow C(M) \oplus K_d$, $K_s \leftarrow K_d$
    else
      (7) Discard *M*
    end
  end
end

Algorithm 2: RPG & Prime Counter

Input: M = Input Message, K_s = Session Key
  (a) $FSRP = frac(Sqrt(RPG(Seed)))$ OR (b) $FSRP = frac(Sqrt(PC(Index)))$
begin
  Sender:
  while *(Session!=END)* **do**
    (1) Generate *CDT*
    (2) $K_e \leftarrow$ HASH(Ks, *CDT*, *FSRP*)
    (3) $C(M) \leftarrow M \oplus K_e$, $K_s \leftarrow K_e$
    (4) $HMAC_S \leftarrow$ HASH($C(M)$, *CDT* || *FSRP*)
    (5) *MACSALT* $\leftarrow$ *FSRP* $\oplus$ *CDT*
    (6) $Index \leftarrow Index + 1$
    (7) Transmit $C(M)$, *HMAC*,*MACSALT*
  end
  Receiver:
  while *(Session != END)* **do**
    (1) *CDT* $\leftarrow$ *MACSALT* $\oplus$ *FSRP*
    (2) $HMAC_R \leftarrow$ HASH($C(M)$, *CDT* || *FSRP*)
    if *(($HMAC_S$, $HMAC_R$) == TRUE)* **then**
      (3) $Kd \leftarrow$ HASH(Ks, *CDT*, *FSRP*)
      (4) $K_s \leftarrow K_d$, $Index \leftarrow Index + 1$
      (5) $M \leftarrow C(M) \oplus K_d$
    else
      (6) Discard M
    end
  end
end

FSRP, and Ks to generate the symmetric key. These parameters are exchanged securely using MACSalt and NTRUEncrypt public-key cryptography.

For our evaluation, we assume that the length of the key is 256 bits as Blake2s depends on a 32 byte word size. In the case of 256 bits $<$ input string $<$ 512 bits, we can replace Blake2s with Blake2b to generate the symmetric key, which consists of a 64 byte word size.

The following section describes four methods to implement secure SCADA framework for information exchange.

1) Hybrid Multi-Layered Architecture: We can use the same nomenclature of session key agreement for further secure communication in which after successful distribution of the session key the message is communicated between two parties using both symmetric and asymmetric key cryptography. The data encryption and decryption are obtained using Vernam cipher. The key generator of the Vernam cipher follows the same procedure of session key derivation to generate the symmetric key at the sender and receiver sides. The symmetric key, HMAC and MACSALT are derived using FSRP and CDT. Further encrypted message C(M), HMAC and MACSALT are shared securely using asymmetric key cryptography. Here, the complexity of the method is obtained by N * (Asymmetric Key + Symmetric Key) during each session which provides high security with moderate availability. N is the number of messages exchange during the session. The steps of this approach are shown in Algorithm 1.

The following methods describe the approach of symmetric key cryptography instead of using a combination of public-private key pairs. After secure session key and prime seed distribution, further encryption process can be carried out using one of the three symmetric key based proposed methods as listed below.

2) Random Prime Number Generator: In this method, the seed of the prime index value is used to determine FSRP using next random prime number. Also, CDT and hash of the input message h(M) are determined to generate symmetric key, HMAC and MACSALT. This information is sent to the recipient over the communication channel. Using MACSALT and random number prime generator, the receiver can generate the symmetric key to decrypt the data using the Vernam cipher. Here the complexity of algorithm is measured by Asymmetric key + N * Symmetric key for every session where asymmetric and symmetric key are used during session key distribution while the symmetric key is used during secure communication. However, this approach is comparatively less secure as the adversary could intercept the MACSALT to derive the keys such as FSRP and CDT. Algorithm 2 summarizes the above process.

3) Prime Counter: In this method, instead of random prime generator, we have used prime counter which significantly increases the execution speed. The rest of the steps are same

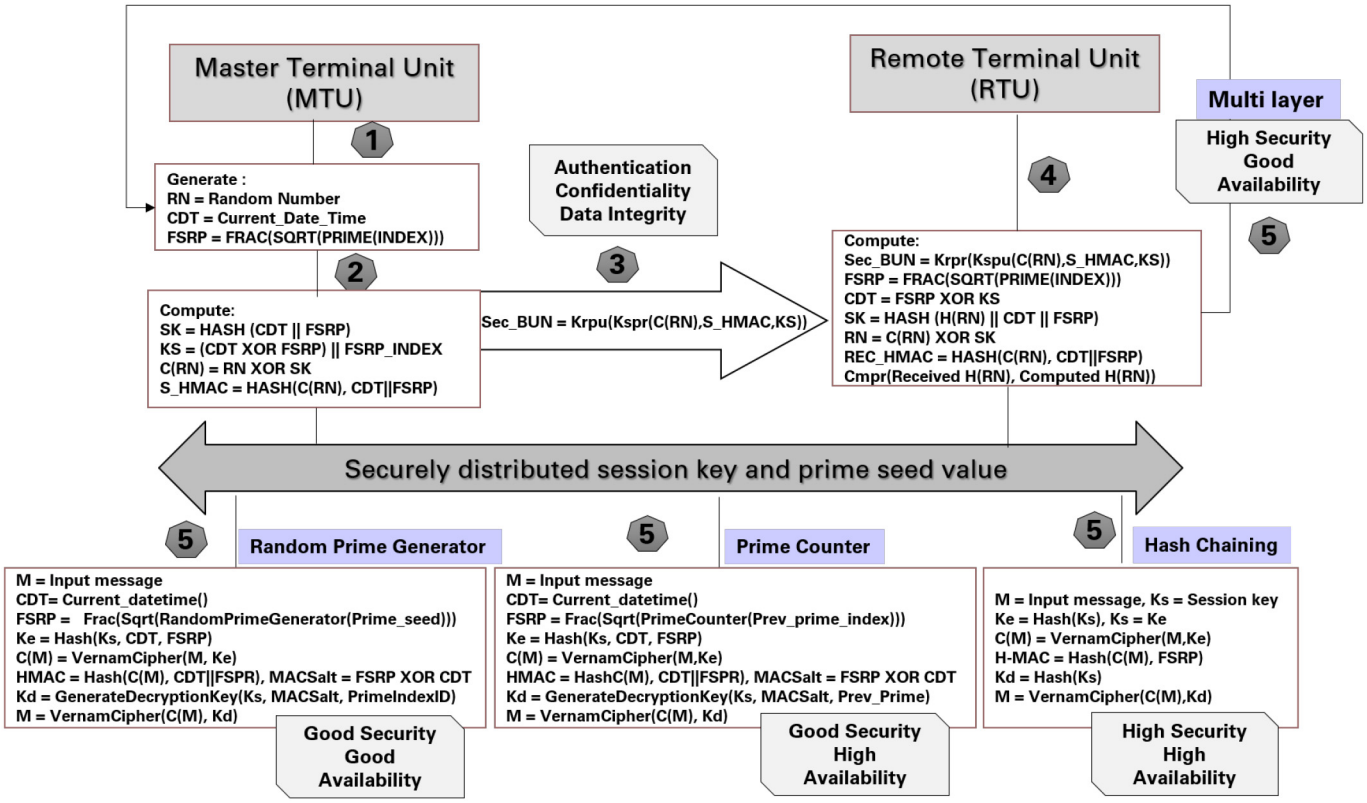


Fig. 5. Complete process diagram of secure communication between MTU and RTU.

Algorithm 3: HASH Chaining

Input: M = Input Message, K_s = Session Key
$FSRP = Frac(Sqrt(PC(Index)))$

begin
  Sender:
  while *(Session!=END)* **do**
    (1) $K_e \leftarrow$ HASH(K_s,*FSRP*)
    (2) $C(M) \leftarrow M \oplus K_e$, $K_s \leftarrow K_e$
    (3) $HMAC_S \leftarrow$ HASH($C(M)$,*FSRP*)
    (4) $Index \leftarrow Index + 1$
    (5) Transmit $C(M)$, *HMAC*
  end
  Receiver:
  while *(Session != END)* **do**
    (1) $HMAC_R \leftarrow$ HASH($C(M)$, *FSRP*)
    (2) $Index \leftarrow Index + 1$
    if *(($HMAC_S$, $HMAC_R$) == TRUE)* **then**
      (3) $K_d \leftarrow$ HASH(K_s, *FSRP*), $K_s \leftarrow K_d$
      (4) $M \leftarrow C(M) \oplus K_d$
    else
      (5) Discard *M*
    end
  end
end

and hence we have highlighted the difference in red font in Algorithm 2. The previous prime number is used to determine next FSRP. Similarly CDT and hash of the input message h(M) are used to determine symmetric key, HMAC and MACSALT. This information is sent to the recipient. Using MACSALT and prime counter, the receiver can generate symmetric key to decrypt the data using Vernam cipher. In this approach the adversary could also intercept the MACSALT to derive the essence of the keys such as FSRP and CDT. The complexity of algorithm is measured by Asymmetric key + N * Symmetric key for every session. Consequently, the model provides good security with high availability.

4) Hash Chaining: This proposed method is one of the robust solutions for SCADA systems which covers all the security mechanisms. This approach not only provides high security but also offers high availability. In this, the pre-shared session key is used as input of the hash function to generate the next symmetric key. Moreover, the previous FSRP is used to generate HMAC which can be derived independently at both the ends and is used to check the integrity of the message. The generated symmetric key is then used to encrypt and decrypt the message using the Vernam cipher, as mentioned in the Algorithm 3. The complexity of this method is based on the Asymmetric + N * Symmetric key cryptography.

The complete process diagram of the proposed framework of secure SCADA systems is shown in Figure 5.

V. Experiments

A. Algorithm Selection of Cipher Suite for Proposed Framework

The choice of the algorithms to design the security framework generally depends on the nature of the application.

The communication of SCADA systems relies on a real-time request-response mechanism. Moreover, SCADA field devices are equipped with micro controllers for processing information and have limited computational power and resources. Consequently, identifying the most appropriate algorithms for the proposed scheme is one of our implementation's crucial steps. The identified algorithms for our cipher suite should provide faster execution speed and be suitable for deploying in an embedded system environment. The comparative analysis of various algorithms was carried out using wolfSSL and libntru 0.5 cryptosystems on Linux subsystem of Windows 10 with Intel Core i5-8300H 2.30GHz processor and 8 GB RAM. The wolfSSL is a lightweight and portable embedded SSL library that is specially meant for IoT, embedded, and RTOS environments [33]. The libntru 0.5 is an open source library that supports the implementation of the public-key encryption scheme NTRUEncrypt in C language by following the IEEE P1363.1 standard [34]. Moreover, the proposed symmetric schemes are implemented on an integrated development environment for Python called IDLE on Windows 10 operating system.

1) HASH Functions: In this framework, the hash function plays a vital role as it acts as a message authentication code and is used to generate a symmetric key. To identify the cryptographically secure and computationally efficient function, we have compared various hash functions. Based on the comparative analysis of computational speed presented in Table I, Blake seems to be most prominent. There are three flavors of Blake's hash function, namely, Blake, Blake2, and Blake3. Furthermore, Blake2 is subcategorized in two types, namely, Blake2s and Blake2b. Blake2b is designed for 64 bits of word length while Blake2s and Blake3 are designed for 32 bits. Both the categories of Blake2 are cryptographically secure hash functions and used to target various applications such as cloud storage intrusion detection, version control systems, and Internet of Things. Moreover, it is computationally efficient like MD5, and provides security similar to SHA-3 [35]. We can also take advantage of Blake2 in multicore architectures for parallel processing. Furthermore, Blake2 uses 32% less RAM than Blake and has proven efficient MAC function [36]. These features make Blake2 a suitable candidate for SCADA systems. For the framework implementation, we have used Blake2s as one of our proposed cipher suite elements. A new version of Blake, namely Blake3, has been released recently [37]. Blake3 is comparatively faster than Blake2s as it uses seven rounds, whereas Blake2s uses ten rounds to compute the hash function [38]. One scope for future work for our research would be to implement our framework using Blake3.

2) Symmetric Key Cryptography: Advanced Encryption Standard (AES) is the well-known symmetric key cryptography used to design secure systems. AGA has used AES as a symmetric key component in its standard protocol suite [16]. Nowadays, AES modes are preferred to secure the systems owing to better security and faster execution speed. 3DES is also used in traditional cryptosystems. In Table II, we have compared the computational speed of various modes of AES and DES with the proposed hash-based Vernam Cipher. The computational speed of Vernam Cipher is calculated by adding the execution speed of Blake2S hash with the speed of Exclusive-OR operation. The comparative analysis shows that the hash-based symmetric key technique used in Vernam Cipher is faster than other algorithms.

TABLE I
COMPARATIVE ANALYSIS OF VARIOUS HASH FUNCTIONS

Algorithm	Ex.Speed (MB/sec)
MD5	340.729
RIPEMD	129.74
SHA	31.571
AES-256-CMAC	110.504
SHA2-256	152.863
SHA3-256	106.781
Blake2b	**172.2**
Blake2s	**169.78**

TABLE II
COMPARATIVE ANALYSIS OF COMPUTATIONAL SPEED OF VARIOUS SYMMETRIC KEY ALGORITHMS

Algorithm	Ex.Speed (MB/sec)
AES-256-CBC-enc, AES-256-CBC-dec	94.565 , 88.169
AES-256-GCM-enc, AES-256-GCM-dec	25.596, 24.318
AES-256-ECB-enc, AES-256-ECB-dec	55.69, 63.067
AES-256-CFB	86.329
AES-256-OFB	71.146
AES-256-CTR	64.576
3DES	14.542
Vernam Cipher with Blake2s	**157.45**

3) Asymmetric Key Cryptography: Asymmetric key cryptography not only offers the confidentiality but also ensures integrity, authentication, and non-repudiation during communication. Some public key algorithms such as Diffie-Hellman key exchange provide key distributions and secrecy, whereas some provide encryption and digital signature such as RSA, ECC, and NTRU [39]. We have compared various well-established public key algorithms, namely, RSA, DH, ECC, and NTRU by considering the key size and total operations performed per second. According to the output results presented in Table III, NTRU outperforms the other methods. NTRU public-key cryptography is also known as NTRUEncrypt. This is constructed using a lattice-based technique by applying the concept of the shortest vector problem. It depends on the factoring of certain polynomials in a polynomial ring into a quotient of two minimal coefficients. Both encryption and decryption follow simple polynomial multiplication, which makes NTRU faster than other asymmetric key cryptosystems [40]. Moreover, the points mentioned below represent the capabilities of the NTRU based public key cryptography. Therefore, we have chosen the NTRU public key algorithm for our proposed cipher suite.

- NTRU is the highest performing public key cryptograpic system for embedded devices [41].
- NTRU decryption is more than 92 times faster than RSA decryption at an equivalent security level [42].
- NTRU is nearly 60% faster than RSA at encryption and TLS with a 370 times improvement in key generation time [42].

TABLE III
COMPARATIVE ANALYSIS OF THE COMPUTATIONAL SPEED OF VARIOUS PUBLIC KEY CRYPTOGRAPHY

Algorithm	Key Size	Mode	Ops/sec
RSA	1024	Key Generation	12
RSA	2048	Key Generation	3
DH	2048	Key Generation	913
DH	2048	Key Agreement	500
ECC	256	Key Generation	523
ECDHE	256	Key Agreement	500
NTRU	**1026**	**Key Generation**	**2153**
NTRU	**1499**	**Key Generation**	**698**
NTRU	**1615**	**Key Generation**	**801**
NTRU	**2066**	**Key Generation**	**345**

- NTRU encryption and decryption are faster than the best-performing ECC algorithms at equivalent security levels [42].
- NTRU is only around 20 times slower than a recent AES implementation [42].
- Both RSA and ECC are vulnerable to quantum computing attacks where NTRU offers resistance to that [43].
- NTRU accomplishes TLS authentication and key negotiation by combining classic cryptography which offers quantum-safe cryptography [43].
- Parallel implementation of NTRU is possible on top of the existing crypto infrastructure [41].

B. Computational Speed of Proposed Framework

This section represents the calculation of the overall computational speed of the proposed framework. We have considered the execution time of the major four elements, namely, session key, symmetric key, HMAC, and asymmetric key. First, we have calculated the time to generate and extract the session key. After that, we have computed the execution time of symmetric and asymmetric key generation, distribution, encryption, and decryption. Finally, we have calculated the overall time by combining it with execution time to generate and extract the HMAC.

1) Execution Time of Session Key Generation and Extraction: We have generated the session key, KEYSALT, and MACSALT using two random parameters CDT and FSRP, along with Blake2s HASH function. These parameters are securely exchanged between two communication SCADA devices and extracted back at the receiver side. The average execution time and total execution time to generate and extract these elements are shown in Figure 6 and Figure 7. We observed that it takes approximately 0.15 milliseconds average execution time to create and extract a 256-bit session key.

2) Execution Time of Symmetric Key Cryptography: This section presents the execution time of three symmetric key cryptography methods, namely, Random Prime Generator (RPG), Prime Counter (PC) and Hash Chaining (HC). To calculate the execution time of each method we have considered the overall time of each module to generate and extract the symmetric key along with encryption and decryption time taken by the Vernam stream cipher. In Table IV, we present the time of three proposed symmetric key cryptography methods

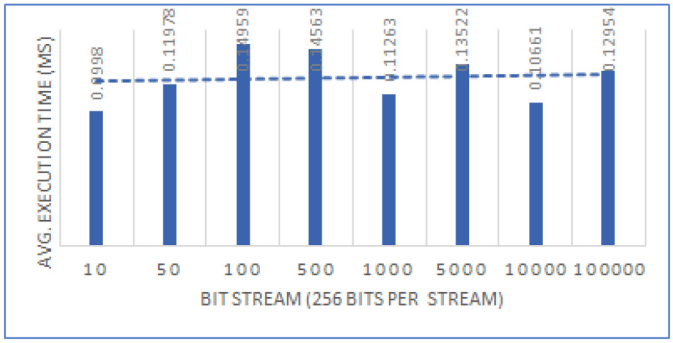


Fig. 6. Average Execution Time for Session Key Generation and extraction.

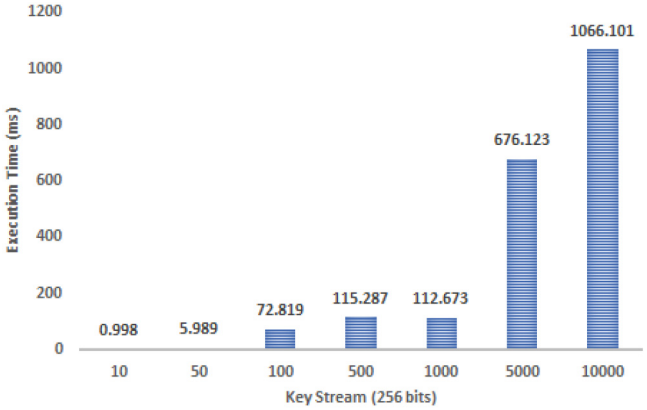


Fig. 7. Overall Time (Session Key).

TABLE IV
EXECUTION TIME OF PROPOSED SYMMETRIC KEY METHODS

Input stream (256 bits)	Random Prime (second)	Prime Counter (second)	Hash Chaining (second)
1	0.0050	0.000022	0.0000009
10	0.040	0.0004	0.0001
50	0.280	0.0027	0.0003
100	0.411	0.0040	0.0012
500	2.325	0.254	0.0062
1000	6.566	0.439	0.0129
5000	32.929	2.499	0.0685
10000	50.883	4.392	0.1325

(in seconds) for various sizes of input streams. Based on the results, hash chaining seems to be the most efficient in terms of computational speed amongst the three proposed methods.

3) Execution Time of NTRU Based Public Key Cryptography: We have compared NTRU based implementations based on security levels, namely, moderate, standard, high, and highest security. Each security level is defined considering the size of cipher text, a public key, and private key. In most applications, the standard security level is used to avoid lattice-based, brute force, and man-in-the-middle attacks. The observation is carried out using total execution time by considering key generation, encryption, and decryption as shown in Table V. Moreover, we have computed the average execution time of public-key pair generation, which is around 1.51 ms with an encryption time of 0.073 ms and a decryption time of 0.106 ms.

TABLE V
EXECUTION TIME OF NTRU BASED PUBLIC KEY CRYPTOGRAPHY

NTRU (INPUT=256 bits)	Key_Gen (ms)	Enc (ms)	Dec (ms)	Total (ms)
Moderate Security (CT=1022, Kpub=1026,Kpr=111)	0.468	0.03	0.047	0.545
Standard Security (CT=1495,Kpub=1499,Kpr=1339)	1.432	0.073	0.096	1.601
High Security (CT=1611,Kpub=1615,Kpr=301)	1.248	0.066	0.138	1.452
Highest Security (CT=2062,Kpub=2066,Kpr=227)	2.893	0.123	0.145	3.161
Average Execution Time (ms)	1.510	0.073	0.106	1.689

TABLE VI
CONSIDERABLE PARAMETERS OF DIFFERENT CRYPTOGRAPHIC COMPONENTS

Notation	Description	Cost in ms
Tskg	Time for a session key generation	0.06485
Tpc	Time to generate Prime number Counter	0.00997
Trpg	Time to generate random prime generator	0.5063
Tudt	Time to generate universal date and time	0.00010
Tse	Time for a symmetric encryption	0.000199
Tsd	Time for a symmetric decryption	0.000996
Thash	Time to generate HMAC	0.000001
Tex	Time for a session Key extraction	0.0992
Takg	Time for a asymmetric key generation	1.51025
Tae	Time for a asymmetric encryption	0.073
Tad	Time for a asymmetric decryption	0.1065

TABLE VII
TOTAL EXECUTION TIME CALCULATION

Method	Delay
Multi-layered Approach	N * (Tskg + Tex + Takg + Tae + Tad + Tsym + Tse + Tsd + Thash)
Symmetric Key Approach	Tskg + Tex + Takg + Tae + Tad + N * (Tsym + Tse + Tsd + Thash)
Random Prime Generator	Tsym = Trpg + Tudt
Prime Counter	Tsym = Tpc + Tudt
Hash Chaining	Tsym = Thash

4) Total Execution Time: In order to achieve consistent results, we have measured the execution time of each cryptographic components. The execution time of these elements is listed in Table VI.

Moreover, Table VII presents the mathematical equations that calculate the total execution time of all the four methods, namely, ML, RPG, PC, and HC. In hybrid approach, both symmetric and asymmetric algorithms are used to secure the information. In contrast, in the other three approaches, once the session key has been shared between two communication devices, only the symmetric key algorithm is used for performance improvement. Furthermore, the execution time of these three symmetric key algorithms is varied due to how they generate the keys to secure the information.

Table VIII represents the total execution time of all the four proposed methods by considering the major four parameters, namely, key generation, key extraction, encryption and decryption. According to the results, the execution time of HC is lower than the other three methods and has proven most efficient amongst all. Moreover, PC and ML approaches are more prominent than RPG. Comparatively, RPG takes more time because of its intricate design to generate a random prime number based on a seed value.

TABLE VIII
TOTAL EXECUTION TIME IN SECONDS

BIT STREAM (256 bits)	Hybrid	RPG	PC	HC
1	0.0026	0.0052	0.0019	0.0016
10	0.0072	0.0437	0.0056	0.0026
50	0.0373	0.2806	0.0285	0.0046
100	0.0592	0.4111	0.0415	0.0136
500	0.3453	2.3248	0.2557	0.0635
1000 (32KB)	0.6203	6.5663	0.4410	0.1309
5000 (160KB)	3.3980	32.9295	2.5007	0.6867
10000 (320KB)	6.1889	50.8831	4.3941	1.3271
50000 (16MB)	30.2294	257.9262	21.2546	2.2456
100000 (32MB)	61.1824	560.3114	43.2326	10.4327

C. Calculation of Key Storage Cost

Storage cost is another important parameter to evaluate the performance of SCADA networks. Field control devices such as RTUs, PLCs, and IEDs are typically located at the plant floor and remote from the MTU. Hence they require to update the session keys periodically. On the other hand, if field control devices have many static keys, and if any of them is compromised, it can expose the entire network communication. Consequently, the session key update process is a very crucial step. Since the key generation, distribution, and extraction are periodic and costly operations, the SCADA network should have fewer stored keys on each field control device. For this reason, we have identified the storage cost of our proposed key management scheme. Table IX summarizes the storage cost by considering the three types of communication, namely, point-to-point, broadcast, and multicast amongst MTU, Sub-MTU, and RTU. The total cost of keys is calculated at each SCADA location, where m denotes the number of sub-MTU's keys, and r represents the maximum number of RTU's keys.

D. Randomness Evaluation

Many cryptography applications may need to meet more robust random number generator requirements when the randomness of the keys is one of the most critical factors for that system. We have used the Vernam stream cipher for our proposed framework, which requires a distinct and random key to secure the information. In particular, the key generator's output must be unpredictable. Hence, we have evaluated the proposed symmetric key generator using the National Institute of Standards and Technology (NIST) statistical toolkit. We have configured this tool in the Linux subsystem of the Windows 10 operating system. This toolkit offers a total of sixteen different statistical tests to determine whether a generator is suitable for a particular cryptosystem. Each test evaluates the randomness based on specific criteria by considering the number of 1's and 0's in the binary stream and accordingly produces the P-value. If the test has P-value

TABLE IX
KEY MANAGEMENT : STORAGE COST OF KEYS

Types	MTU	SUB-MTU	RTU
Point to Point Communication			
MTU to SUB-MTU SUB-MTU to RTU RTU to SUB-MTU	Private Key of MTU= 1, (D) Public Key = m, (E) m = Number of Sub-MTUs	Private Key of Sub-MTU = 1, (D) Public Key = r, (E) r = Number of RTUs	Private Key of RTU= 1, (D) Public Key = 1, (E) Public key of Sub-MTU
RTU to RTU	NA	NA	Private Key of Controller RTU = 1, (D) Public Key = r, (E) r = Number of Slave-RTUs
Broadcast Communication			
MTU to RTU	Private Key of MTU= 1, (E)	Private Key of Sub-MTU = 1, (E) Public Key of MTU = 1, (D)	Public Key of sub-MTU = 1, (D)
RTU to RTU	NA	NA	Private Key of Controller RTU = 1, (E) Public Key of Controller RTU = 1, (D)
Multicast Communication			
MTU to RTU	Private Key of MTU= 1, (E)	Private Key of Sub-MTU = 1, (E) Public Key of MTU = 1, (D)	Public Key of sub-MTU = 1, (D)
RTU to RTU	NA	NA	Private Key of Controller RTU = 1, (E) Public Key of Controller RTU = 1, (D)
Total Keys	1(PK) + m(PUB-SUBMTU) = **m + 1**	1(PK)+r(PUB-RTUs) + 1(PUB-MTU) = **r + 2**	Controller RTU: **m + 2** 1 (PK) + 1 (PUB-SUBMTU) + m (PUB-RTU) Slave RTUs: **2** 1(PK) + 1(PUB-CRTU)

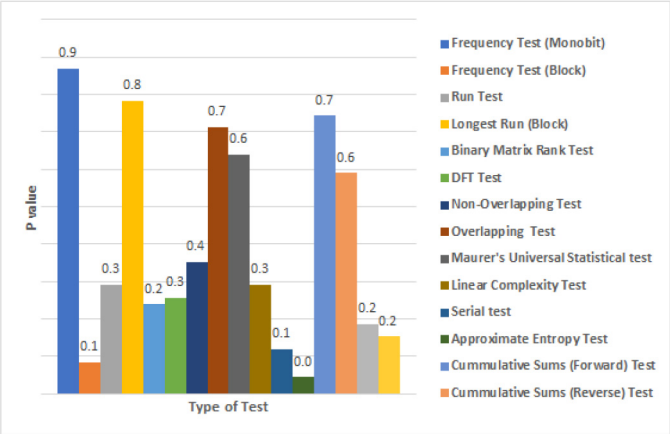


Fig. 8. Randomness assessment of symmetric key.

≥ 0.001, that means an input binary sequence would be random with a 99.9% confidence. Figure 8 presents all the 16 tests and corresponding P-values for the proposed symmetric key generator for Vernam cipher. Our proposed key generator passes all the statistical tests and proven to be random.

VI. PERFORMANCE ANALYSIS

A. Formal Analysis of Protocol

Researchers currently use two main approaches to verify security protocols, namely, provable security and the formal method approaches [44], [45], [46]. Provable security defines a rigorous framework to describe and prove cryptographic properties from a mathematical point of view. However, the formal method approach proposes a model to describe and analyze cryptographic protocols by abstracting basic properties. Dalal *et al.* [47] discusses various tools such as Avispa, ProVerif, and Scyther that are useful for the formal verification of the cryptographic protocols. Scyther outperforms the state-of-the-art Avispa tools. Although Scyther uses no abstraction techniques, it still offers a performance level similar to the abstraction-based ProVerif tool [47]. In Scyther, small (e.g., Needham-Schroeder, Yahalom, Otway-Rees) to medium-sized (e.g., TLS, Kerberos) protocols are usually verified in less than a second. Moreover, Scyther is currently the fastest protocol verification tool that does not use approximation methods [48].

Therefore, we have used the Scyther tool to formally verify our security protocol, which performs the evaluation under the cryptographic assumption. We define all the cryptographic functions completely. Moreover, the entire assessment is carried out by considering the presence of an adversary. This tool uses an unbounded model checking approach that demonstrates the soundness of a protocol for all the possible behaviors in the presence of an adversary [49]. The language used in Scyther is called Security Protocol Description Language (SPDL). It is also known as role-based language that describes the entire protocol using roles and sending/receiving events.

SPDL provides expressions for encryption and hashing. Furthermore, we can verify authentication, confidentiality and message integrity using claims in the Scyther. We have mainly focused on three types of goals, namely, non-injective synchronization, non-injective agreement, and secrecy for our proposed approach. We have generated a trace pattern route that represents the packet forwarding from RTU to MTU, as illustrated in Figures 9 and 10. Figure 11 illustrates the protocol design code for Scyther to analyze the attacks by considering all the participants, namely, MTU, RTU, and

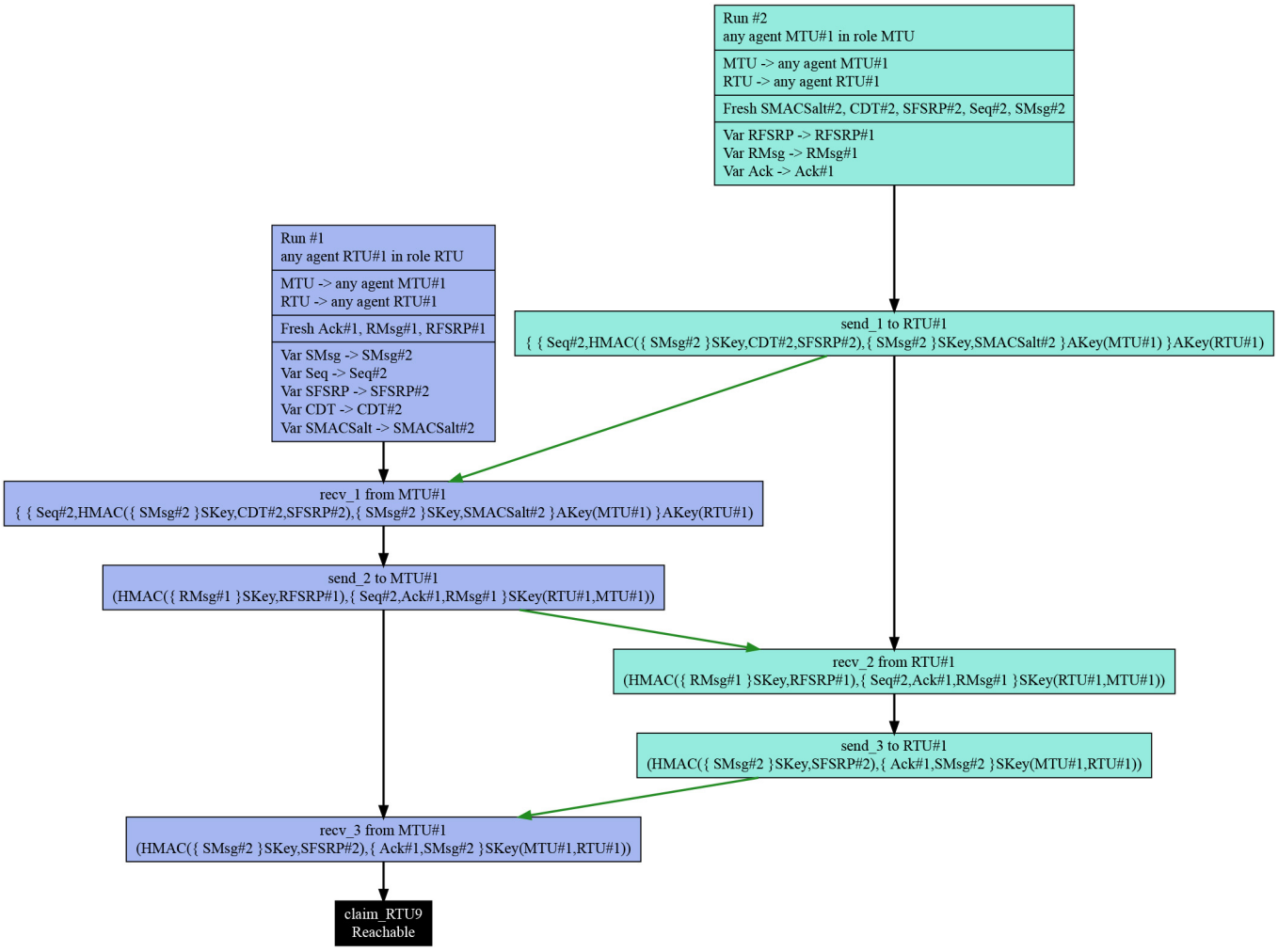


Fig. 9. Formal Analysis of proposed protocol for secure communication between MTU and RTU using Scyther Tool.

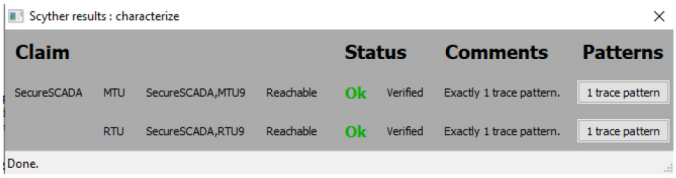


Fig. 10. Claims & trace pattern validation of proposed protocol for secure communication between MTU and RTU using Scyther Tool.

the attacker. We have verified the protocol using "automatic claim" and "verification claim" procedures. As illustrated in Figure 12, our proposed framework is resistant to all the attacks over the communication channel.

B. Attack Analysis on Hash Function

Generally, a hash function can be broken by three types of attacks, namely, collision attack, preimage resistance attack, and length extension attack [35], [50], [51], [52], [53]. A brief description of each attack is described below.

1) *Collision Attack:* This attack aims to identify two different inputs that will generate the same hash value to create a collision with transmitted data over the communication channel. For example, the attacker will try to find messages m1 & m2, leading to the same hash function, i.e., Hash(m1) = Hash(m2). In general, for two different precedes, p1 & p2, the intruder chooses two appendages m1 & m2 such that Hash(p1||m1) = Hash(p2||m2) which leads to the chosen-prefix collision attack.
2) *Preimage Resistance Attack:* This attack is intended to find out the message for the particular hash value. That means, given a hash value h, the attacker will find a message m such that Hash(m) = h.
3) *Length Extension Attack:* In this attack, an attacker can use Hash(m1) and the length of m1 to calculate Hash(m1||m2), where an attacker will control m2 without knowing the content of m1.

Three types of approaches are used to check the strength of the hash function to test if the given hash function can be broken practically, theoretically or partially as listed below [54].

1) *Practically Broken:* The attack has been demonstrated in practice and able to break the entire hash function.

Protocol description | Settings

```
secret AKey: Function;
secret  SKey: Function;
hashfunction HMAC;
protocol SecureSCADA(MTU,RTU)
{
    role MTU
    {
        fresh SMACSalt: Nonce; fresh SMsg: Nonce;
        fresh CDT: Nonce;fresh SFSRP: Nonce; fresh Seq: Nonce;
        var Ack: Nonce;var  RMsg: Nonce;var RFSRP: Nonce;

        send_1(MTU,RTU,
        {{Seq,HMAC({SMsg}SKey,CDT,SFSRP),{SMsg}SKey,SMACSalt}AKey(MTU)}AKey(RTU));
        read_2(RTU,MTU,
        HMAC({RMsg}SKey,RFSRP),{Seq,Ack,RMsg}SKey(RTU,MTU) );
        send_3(MTU,RTU,
        HMAC({SMsg}SKey,SFSRP),{Ack,SMsg}SKey(MTU,RTU) )

        claim_i1 (MTU,Secret,CDT);
        claim_i2 (MTU,Secret,SFSRP);
        claim_i3 (MTU,Secret,SMsg);
        claim_i4 (MTU,Secret,SMACSalt);
        claim_i5 (MTU,Alive);
        claim_i6 (MTU,Weakagree);
        claim_i7 (MTU,Niagree);
        claim_i8 (MTU,Nisynch);
    }
    role RTU
    {
        var SMACSalt: Nonce;var CDT: Nonce;var SFSRP: Nonce;var Seq: Nonce;var SMsg: Nonce;
        fresh Ack: Nonce;fresh RMsg: Nonce;fresh RFSRP: Nonce;

        read_1(MTU,RTU,
        {{Seq,HMAC({SMsg}SKey,CDT,SFSRP),{SMsg}SKey,SMACSalt}AKey(MTU)}AKey(RTU));
        send_2(RTU,MTU,
        HMAC({RMsg}SKey,RFSRP),{Seq,Ack,RMsg}SKey(RTU,MTU) );
        read_3(MTU,RTU,
        HMAC({SMsg}SKey,SFSRP),{Ack,SMsg}SKey(MTU,RTU) );

        claim_i1 (RTU,Secret,RMsg);
        claim_i2 (RTU,Secret,RFSRP);
        claim_i3 (RTU,Secret,Ack);
        claim_i4 (RTU,Secret,Seq);
        claim_i5 (RTU,Alive);
        claim_i6 (RTU,Weakagree);
        claim_i7 (RTU,Niagree);
        claim_i8 (RTU,Nisynch);
    }
}
```

Fig. 11. Claims & Protocol Design Code for Scyther for Analysis of attacks.

Scyther results : verify

Claim				Status		Comments
SecureSCADA	MTU	SecureSCADA,MTU1	Secret CDT	Ok	Verified	No attacks.
		SecureSCADA,MTU2	Secret SFSRP	Ok	Verified	No attacks.
		SecureSCADA,MTU3	Secret SMsg	Ok	Verified	No attacks.
		SecureSCADA,MTU4	Secret SMACSalt	Ok	Verified	No attacks.
		SecureSCADA,MTU5	Alive	Ok	Verified	No attacks.
		SecureSCADA,MTU6	Weakagree	Ok	Verified	No attacks.
		SecureSCADA,MTU7	Niagree	Ok	Verified	No attacks.
		SecureSCADA,MTU8	Nisynch	Ok	Verified	No attacks.
	RTU	SecureSCADA,RTU1	Secret RMsg	Ok	Verified	No attacks.
		SecureSCADA,RTU2	Secret RFSRP	Ok	Verified	No attacks.
		SecureSCADA,RTU3	Secret Ack	Ok	Verified	No attacks.
		SecureSCADA,RTU4	Secret Seq	Ok	Verified	No attacks.
		SecureSCADA,RTU5	Alive	Ok	Verified	No attacks.
		SecureSCADA,RTU6	Weakagree	Ok	Verified	No attacks.
		SecureSCADA,RTU7	Niagree	Ok	Verified	No attacks.
		SecureSCADA,RTU8	Nisynch	Ok	Verified	No attacks.

Done.

Fig. 12. Claims & Attack Analysis of proposed protocol for secure communication between MTU and RTU using Scyther Tool.

TABLE X
COMPARATIVE ANALYSIS OF THE VARIOUS HASH FUNCTIONS (COL: COLLISION ATTACK, PR: PREIMAGE RESISTANCE ATTACK, LE: LENGTH EXTENSION ATTACK)

Hash Function	COL	PR	LE	Partial	Practical	Theory
MD5 [51]	YES	YES	NO	NO	YES	YES
PANAMA [56]	YES	NO	NO	NO	YES	NO
RIPEMD [51]	YES	YES	NO	YES	YES	NO
SHA-1 [52]	YES	YES	NO	YES	YES	YES
SHA256 [53]	YES	YES	NO	YES	NO	NO
BLAKE2s [54]	YES	YES	NO	YES	NO	NO
SHA3 [54]	YES	YES	NO	YES	NO	NO
BLAKE2b [36]	YES	YES	NO	YES	NO	NO

2) *Theoretically Broken:* Attack demonstrates in theory by proof of concept which is able to break all the rounds of the hash function.
3) *Partially Broken:* No attack has demonstrated to break the entire function successfully. However, only a reduced version of the hash is broken and requires more work than the claimed security level.

Table X compares the types of attacks and the breaking mechanisms of various popular hash functions. As illustrated in Table X, Blake2 is comparatively better than other functions. Blake2 can be partially broken and fragile due to collision and preimage resistance attacks. To overcome this issue, we have incorporated two approaches, namely, PNG (Prime number generator) and HMAC. To prevent the system from collision attacks, we have introduced the parameters FSRP and CDT, which generate a unique key at each iteration. In this case, even if the attacker identifies a similar input which generates the same hash function as the transmitted data, it will not help in successfully launching a correlation attack. In our proposed solution, we use HMAC, which not only relies on the hash of the message but also uses CDT & FSRP. Hence, during the validation process, the authentication and message integrity are identified at the receiver end and can prevent the system from correlation and preimage resistance attacks. The following discussion gives the security proof of our proposed approach against correlation and preimage resistance attack.

Security Proof: With reference to the proposed framework, let us denote the original Message as C(M1) and the key parameters used to generate HMAC as CDT & FSRP.

$$HMAC_{Sender} = Hash((C(M1), CDT||FSRP)) \quad (15)$$

Let us assume, over the communication channel, the attacker identifies another message C(M2) and replaces C(M1) with C(M2) where, Hash(C(M1)) = Hash(C(M2)). The receiver computes HMAC based on received message C(M2) as follows.

$$HMAC_{Receiver} = Hash((C(M2), CDT||FSRP)) \quad (16)$$

$$HMAC_{Sender} \neq HMAC_{Receiver} \quad (17)$$

The difference in signature of the HMAC identifies if the integrity is compromised and in such a case M1 is discarded.

The above proof illustrates that the proposed security framework prevents the collision attack. Similarly, even though the intruder can identify message C(M1) which generates Hash(C(M1)), the message integrity or authentication cannot be broken owing to the key parameters CDT and FSRP. Thus, the pre-image attack is prevented.

C. Analysis of Avalanche Effect for Hash Function

Confusion and diffusion techniques have traditionally been used to evaluate the security of cryptographic primitives [56]. In the context of the hash function, confusion is defined using the relation between the secret key and a hash value for a given input message. Confusion is obtained naturally due to the inherited property of chaos [57]. Diffusion, also known as the avalanche effect, is a desirable property for cryptographically secure hash functions [57]. This is one of the factors to check the randomization capability of the given function. The ideal hash function should exhibit the evidence of the avalanche effect up to the significant level which supports the randomization and make difficult to predict by cryptanalysis [58]. Generally, the butterfly effect and large data blocks are used to generate the avalanche effect [59], in which a small change to an input value will make a significant change in the output hash value. Moreover, there is no correlation between current and previous hash outputs. In our proposed approach, we have used the Blake hash function, which demonstrates a higher-order avalanche effect in that there is a probability of 50% of data alteration in the hash output if a single bit is modified in the input [60]. The example in [32] demonstrates the avalanche effect of Blake and is proven to generate random hash output that doesn't rely on the previous hash value.

D. Randomness Analysis of Keys

Session Key Generation (Parameters): A session key is derived and communicated to both parties during initial authentication. This key is derived using three parameters, namely, random number (RNi), where i = 1,2,3, ... n, index of the function of the fraction of square root of a prime number (FSRP(index)), where index = 1,2,3, ... n, and CDT = current date and time in a microsecond. These parameters are generated at each session and exchanged securely using NTRUEncrypt (public-key cryptography). We have analyzed the following test cases concerning the values of these three parameters.

Case 1: MTU/RTU generates unique values for RN, Index of FSRP, and CDT at every session:

$SessionKey_1 : Hash(RNi, FSRP(index), CDT)$ returns unique value

$SessionKey_2 : Hash(RNi, FSRP(index - k), CDT)$ returns unique value, where k is any random number

Case 2: MTU/RTU generates the same value of RN & seed of FSRP for two or more consecutive sessions, however, CDT is always unique:

$SessionKey_1 : Hash(RNi, FSRP(index), CDT)$ returns unique value as CDT is always distinct

$SessionKey_2 : Hash(RNi, FSRP(index - k), CDT)$ returns unique value as CDT is always distinct, where k = 0.

Here we have used the Blake2 hash function which is proven to be a cryptographically secure function [32] and hence in both the above cases, our proposed approach always generates unique and random session keys.

Symmetric key Generation: The symmetric key is derived using two parameters, namely, session key (Ks) and fraction square root of a prime number. As mentioned earlier, the session key is derived using three randomly generated parameters and distributed over the secure communication channel using public-key cryptography. Moreover, the value of FSRP is generated randomly using a random prime number generator or prime counter. In this case, the seed of the prime number is distributed to both the communication ends, namely, control center, and field site components during session key exchange. These parameters are further computed by combining the concept of hash chaining and FSRP. This is how the proposed approach generates unique and random parameters for the symmetric key used in the Vernam cipher for every message.

Parameters: Here we have used two parameters to derive a symmetric key for the Vernam cipher, namely, FSRP(index), where index = 1,2,3, ..., n (FSRP is generated using a random prime generator or prime counter, the index value is distributed during session key exchange), and session key SKi = Hash(RNi, FSRP(index), CDT), where i, index = 1,2,3, ... n.

Case 1: MTU/RTU generates distinct values of Ks and FSRP for every message:

SKi = Hash(SKi-1, FSRP(index)) returns unique value

SKi + 1 : Hash(SKi, FSRP(index-n)) returns a unique value, where n is any random number, and SKi is updated with the previous session key.

Case 2: MTU/RTU generates the same value of FSRP for two or more consecutive messages, however, Ks is always unique:

SK_i = Hash(SKi, FSRP(index)) returns a unique value as SK is always unique for all messages

$SK_i + 1$ = Hash(SKi-1, FSRP(index-n)) returns unique value as SK is always unique, where n = 0.

In both the above cases, the key is unpredictable and random as the index value of FSRP is only known to MTU and RTU. Moreover, the value of the symmetric key is different even though the value of the FSRP is the same for two consecutive messages as the current key depends on two parameters, namely, SK and FSRP, and is generated using a cryptographically secure hash function.

E. Security Analysis

In this section, the proposed framework is analyzed by considering various security mechanisms, namely, authentication, confidentiality, integrity, availability, and scalability. Moreover, the evaluation is extended by targeting various attacks and corresponding prevention mechanisms.

1) Message Integrity
   - Multi-layered hybrid architecture using symmetric and asymmetric key cryptography offers integrity.
   - Vernam stream cipher provides resistance to cryptography attacks [39].

TABLE XI
COMPARATIVE ANALYSIS OF STORAGE COST OF KEYS (M = NUMBER OF SUB-MTU'S KEYS, R = NUMBER OF RTU'S KEYS)

Key Management Schemes	MTU	Sub-MTU	RTU
SKE [4]	m(1+r)	1+r	1
SKMA [18]	m(1+r)	1+r	1
ASKMA [19]	2m+mr	r+logm	2+logr
ASKMA+ [7]	m	1+r+logm	1+logr
Symmetric [21]	r+1	-	2
Symmetric [21]	r+1	-	2
Hybrid [11]	m+2	2r+1	1+logr
CKMI [22]	2+r+m	2+r	1
Proposed Algo	**m+1**	**r+2**	**2**

TABLE XII
COMPARATIVE ANALYSIS OF VARIOUS CIPHER SUITES

Cipher Suite	Avg_Time (ms)
AGA_ECDHE, RSA, AES-128, GCM, SHA256	4.14
AGA_ECDHE, ECDSA, AES-128, GCM, SHA256	3.94
RSA, AES-128, CBC, SHA1	3.81
RSA, AES-128, CBC, SHA256	3.83
RSA, AES-256, CBC, SHA1	3.82
RSA, AES-256, CBC, SHA256	3.85
Multi-layered (NTRU, Vernam Cipher, Blake2s)	**2.61**
Random Prime Generator (NTRU, Vernam Cipher, Blake2s)	**5.25**
Prime Counter (NTRU, Vernam Cipher, Blake2s)	**1.91**
Hash Chaining (NTRU, Vernam Cipher, Blake2s)	**1.68**

- Randomness of Key offers immunity to collision and preimage resistance attacks [61].
- Dynamic Salt offers resistance to rainbow table attack and dictionary attack [61].
- NTRU based public key cryptography offers resistance to quantum attacks, brute force, and meet-in-the-middle attacks. It also prevents the system against data harvest attacks [62].
- HMAC provides immunity against length extension attacks [63].

2) Authentication, Confidentiality
   - Public key of sender and private key of receiver of NTRU based public key cryptography provides sender's authentication and recipient's confidentiality.
   - HMAC offers message integrity and authentication.
3) High Availability—Faster execution
   - Once the session key distribution is established using hybrid method, further communication will take place using symmetric key cryptography that increases the computation speed.
   - Symmetric key generation using hash chaining and prime counter offers high execution speed.
   - Use of Vernam stream cipher uses modulo operation for encryption and decryption which requires only 4 cycles in hardware implementation [64].
   - NTRU is one of the fastest public key cryptographic systems compared to well-known methods such as RSA and ECC [41].
   - HMAC is derived using the same components used to generate the key. This reusability of elements reduces the computational time.
4) Scalability
   - Same symmetric key cryptography (Vernam cipher) is used for both encryption and decryption.
   - Authentication and confidentiality are established using public-private key pairs amongst communication parties.

F. Storage Cost

The periodic session key agreement is a crucial step in SCADA communication that offers key refreshment. However, field control devices have limited power and memory requirements. Hence, an effective key agreement scheme with fewer stored keys can significantly improve the efficiency of SCADA networks. Many key management and agreement schemes have been proposed to address the problem of key storage costs. We have compared the key storage cost of our proposed scheme with various published techniques, as presented in Table XI.

G. Execution Speed

Table XII depicts the comparative analysis of the proposed scheme with various state-of-the-art techniques by implementing various cipher suites using the wolfSSL library. AGA has proposed two cipher suites for secure SCADA communication including the bundle of ECDHE, AES, RSA, and SHA256 and ECDHE, AES, ECC and SHA256 for authentication, confidentiality, message integrity and digital signature [16].The cipher suite RSA, AES, CBC and SHA is used in TLS communication, whereas we have used the NTRU, Vernam Cipher and Blake2s for our proposed framework. The average execution time of our proposed cipher suite is comparatively better than other protocol standards.

VII. CONCLUSION

The protection of critical industrial infrastructure against cyber-attacks is crucial for ensuring public safety, security, and reliability. SCADA system are used to control and monitor such industrial control systems. A robust solution to strengthen the security of these systems against cyber-attacks is a crucial requirement in the design of SCADA systems. Through this work, we aim to cover the protection of the industrial control system landscape by offering a low cost and robust framework for SCADA networks, which protects them against various cyber-attacks. In this paper, we have proposed a session key agreement in addition to lightweight multi-layered encryption techniques. The framework combines both symmetric and asymmetric cryptography to achieve high computational speed by covering all the security mechanisms. This security model is proposed to enhance the security of various industrial sectors such as water and sewage plants, power stations, chemical plants, oil industries, product manufacturing units, and transportation systems. The successful deployment of this model will allow operators and technicians to monitor and control the plant devices remotely as it will protect the entire system from potential breaches.

REFERENCES

[1] D. Upadhyay, S. Sampalli, and B. Plourde, "Vulnerabilities' assessment and mitigation strategies for the small linux server, Onion Omega2," *Electronics*, vol. 9, no. 6, p. 967, 2020.

[2] D. Upadhyay and S. Sampalli, "SCADA (supervisory control and data acquisition) systems: Vulnerability assessment and security recommendations," *Comput. Security*, vol. 89, Feb. 2020, Art. no. 101666.

[3] Y. Cherdantseva *et al.*, "A review of cyber security risk assessment methods for SCADA systems," *Comput. Security*, vol. 56, pp. 1–27, Feb. 2016.

[4] A. Rezai, P. Keshavarzi, and Z. Moravej, "Key management issue in SCADA networks: A review," *Int. J. Eng. Sci. Technol.*, vol. 20, no. 1, pp. 354–363, 2017.

[5] F. M. Salem, E. Ibrahim, and O. Elghandour, "A lightweight authenticated key establishment scheme for secure smart grid communications," *Int. J. Safety Security Eng.*, vol. 10, no. 4, pp. 549–558, 2020.

[6] D. Upadhyay, J. Manero, M. Zaman, and S. Sampalli, "Gradient boosting feature selection with machine learning classifiers for intrusion detection on power grids," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 1104–1116, Mar. 2021, doi: 10.1109/TNSM.2020.3032618.

[7] D. Choi, S. Lee, D. Won, and S. Kim, "Efficient secure group communications for SCADA," *IEEE Trans. Power Del.*, vol. 25, no. 2, pp. 714–722, Apr. 2010.

[8] T. C. Pramod and N. R. Sunitha, "Polynomial based scheme for secure SCADA operations," *Procedia Technol.*, vol. 21, pp. 474–481, Nov. 2015.

[9] A. Rezai, P. Keshavarzi, and Z. Moravej, "Secure SCADA communication by using a modified key management scheme," *ISA Trans.*, vol. 52, no. 4, pp. 517–524, 2013.

[10] A. Rezai, P. Keshavarzi, and Z. Moravej, "Advance hybrid key management architecture for SCADA network security," *Security Commun. Netw.*, vol. 9, no. 17, pp. 4358–4368, 2016.

[11] D. Choi, H. Jeong, D. Won, and S. Kim, "Hybrid key management architecture for robust SCADA systems," *J. Inf. Sci. Eng.*, vol. 29, no. 2, pp. 281–298, 2013.

[12] R. Jiang, R. Lu, C. Lai, J. Luo, and X. Shen, "Robust group key management with revocation and collusion resistance for SCADA in smart grid," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2013, pp. 802–807.

[13] A. Rezai, P. Keshavarzi, and Z. Moravej, "A new key management scheme for SCADA networks," in *Proc. 2nd Int. Symp. Comput. Sci. Eng.*, 2011, pp. 373–378.

[14] S. Ghosh and S. Sampalli, "A survey of security in SCADA networks: Current issues and future challenges," *IEEE Access*, vol. 7, pp. 135812–135831, 2019.

[15] V. Manjunatha, A. Rao, and A. Khan, "Complex key generation with secured seed exchange for vernam cipher in security applications," *Mater. Today Proc.*, vol. 35, no. 3, pp. 497–500, 2021.

[16] R. Chandia, J. Gonzalez, T. Kilpatrick, M. Papa, and S. Shenoi, "Security strategies for SCADA networks," in *Proc. Int. Conf. Crit. Infrastruct. Protect.*, 2007, pp. 117–131.

[17] M. F. Moghadam, M. Nikooghadam, A. H. Mohajerzadeh, and B. Movali, "A lightweight key management protocol for secure communication in smart grids," *Electr. Power Syst. Res.*, vol. 178, Jan. 2020, Art. no. 106024.

[18] R. Dawson, C. Boyd, E. Dawson, and J. M. G. Nieto, "SKMA—A key management architecture for SCADA systems," in *Proc. 4th Aust. Symp. Grid Comput. e-Res. (AusGrid) 4th Aust. Inf. Security Workshop (Network Security) (AISW-NetSec)*, vol. 54, 2006, pp. 183–192.

[19] D. Choi, H. Kim, D. Won, and S. Kim, "Advanced key-management architecture for secure SCADA communications," *IEEE Trans. Power Del.*, vol. 24, no. 3, pp. 1154–1163, Jul. 2009.

[20] D. Wu and C. Zhou, "Fault-tolerant and scalable key management for smart grid," *IEEE Trans. Smart Grid*, vol. 2, no. 2, pp. 375–381, Jun. 2011.

[21] D. J. Kang, J. J. Lee, B. H. Kim, and D. Hur, "Proposal strategies of key management for data encryption in SCADA network of electric power systems," *Int. J. Electr. Power Energy Syst.*, vol. 33, no. 9, pp. 1521–1526, 2011.

[22] T. C. Pramod, G. S. Thejas, S. S. Iyengar, and N. Sunitha, "CKMI: Comprehensive key management infrastructure design for industrial automation and control systems," *Future Internet*, vol. 11, no. 6, p. 126, 2019.

[23] T. M. D. Hadley and K. A. Huston. *AGA-12, Part 2 Performance Test Results*. Accesed: Oct. 12, 2020. [Online]. Available: https://www.energy.gov/sites/prod/files/oeprod/DocumentsandMedia/9-AGA-12_Part_2_Performance.pdf

[24] D. Abbasinezhad-Mood, A. Ostad-Sharif, and M. Nikooghadam, "Novel anonymous key establishment protocol for isolated smart meters," *IEEE Trans. Ind. Electron.*, vol. 67, no. 4, pp. 2844–2851, Apr. 2020.

[25] N. Saxena, B. J. Choi, and R. Lu, "Authentication and authorization scheme for various user roles and devices in smart grid," *IEEE Trans. Inf. Forensics Security*, vol. 11, pp. 907–921, 2015.

[26] K. Mahmood, S. A. Chaudhry, H. Naqvi, T. Shon, and H. F. Ahmad, "A lightweight message authentication scheme for smart grid communications in power sector," *Comput. Electr. Eng.*, vol. 52, pp. 114–124, May 2016.

[27] M. Keshk, E. Sitnikova, N. Moustafa, J. Hu, and I. Khalil, "An integrated framework for privacy-preserving based anomaly detection for cyber-physical systems," *IEEE Trans. Sustain. Comput.*, vol. 6, no. 1, pp. 66–79, Jan.–Mar. 2021.

[28] J. Qian, C. Hua, X. Guan, T. Xin, and L. Zhang, "A trusted-id referenced key scheme for securing SCADA communication in iron and steel plants," *IEEE Access*, vol. 7, pp. 46947–46958, 2019.

[29] D. G. Brosas, A. M. Sison, and R. P. Medina, "Modified OTP based Vernam Cipher algorithm using multilevel encryption method," in *Proc. IEEE Eurasia Conf. IOT Commun. Eng. (ECICE)*, 2019, pp. 201–204.

[30] R. C. B. Hink, J. M. Beaver, M. A. Buckner, T. Morris, U. Adhikari, and S. Pan, "Machine learning for power system disturbance and cyber-attack discrimination," in *Proc. 7th Int. Symp. Resilient Control Syst. (ISRCS)*, Aug. 2014, pp. 1–8.

[31] R. Zazkis, "Representing numbers: Prime and irrational," *Int. J. Math. Educ. Sci. Technol.*, vol. 36, nos. 2–3, pp. 207–217, 2005.

[32] Wikipedia. *Blake (Hash Function)*. Accesed: May 12, 2021. [Online]. Available: https://en.wikipedia.org/wiki/BLAKE_(hash_function)

[33] WolfSSL. *Embedded TLS Library for Applications, Devices, IoT, and the Cloud*.Accessed: Aug. 12, 2020. [Online]. Available: https://www.wolfssl.com/download

[34] Libntru. *The NTRU Project*. Accessed: Aug. 12, 2020. [Online]. Available: https://tbuktu.github.io/ntru/

[35] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, and C. Winnerlein, "BLAKE2: Simpler, smaller, fast as MD5," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Security*, 2013, pp. 119–135.

[36] J. O'Connor and J.-P. Aumasson. *BLAKE2: Simpler, Smaller, Fast as MD5*. Accesed: Feb. 15, 2021. [Online]. Available: https://www.blake2.net/blake2.pdf

[37] J. O'Connor, S. Neves, and Z. Winnerlein. *Blake3—One Function, Fast Everywhere*. Accesed: Feb. 12, 2021. [Online]. Available: https://github.com/BLAKE3-team/BLAKE3-specs/raw/master/blake3.pdf

[38] J. O'Connor, S. Neves, and Z. Winnerlein. *Blake3 is an Extremely Fast, Parallel Cryptographic Hash*. Accesed: Feb. 15, 2021. [Online]. Available: https:/www.infoq.com/news/2020/01/blake3-fast-crypto-hash/

[39] H. Delfs, H. Knebl, and H. Knebl, *Introduction to Cryptography*, vol. 2. New York, NY, USA: Springer, 2002.

[40] A. A. Kamal and A. M. Youssef, "An FPGA Implementation of the NTRUEncrypt cryptosystem," in *Proc. Int. Conf. Microelectron.*, 2009, pp. 209–212.

[41] J. Hermans, F. Vercauteren, and B. Preneel, "Speed records for NTRU," in *Proc. Cryptogr. Track RSA Conf.*, 2010, pp. 73–88.

[42] J. N. Gaithuru and M. Bakhtiari, "Insight into the operation of NTRU and a comparative study of NTRU, RSA and ECC public key cryptosystems," in *Proc. 8th. Malaysian Softw. Eng. Conf. (MySEC)*, 2014, pp. 273–278.

[43] D. Stehlé and R. Steinfeld, "Making NTRU as secure as worst-case problems over ideal lattices," in *Proc. Annu. Int. Conf. Theory Appl. Cryptogr. Techn.*, 2011, pp. 27–47.

[44] C. Jacomme and S. Kremer, "An extensive formal analysis of multi-factor authentication protocols," *ACM Trans. Privacy Security*, vol. 24, no. 2, pp. 1–34, 2021.

[45] N. Mouha and A. Hailane, "The application of formal methods to real-world cryptographic algorithms, protocols, and systems," *Computer*, vol. 54, no. 1, pp. 29–38, Jan. 2021.

[46] S. Szymoniak, "Security protocols analysis including various time parameters," *Math. Biosci. Eng.*, vol. 18, no. 2, pp. 1136–1153, 2021.

[47] N. Dalal, J. Shah, K. Hisaria, and D. Jinwala, "A comparative analysis of tools for verification of security protocols," *Int. J. Commun. Netw. Syst. Sci.*, vol. 3, no. 10, p. 779, 2010.

[48] A. H. Shinde, A. Umbarkar, and N. Pillai, "Cryptographic protocols specification and verification tools—A survey," *ICTACT J. Commun. Technol.*, vol. 8, no. 2, pp. 1533–1539, 2017.

[49] C. J. Cremers, "The scyther tool: Verification, falsification, and analysis of security protocols," in *Proc. Int. Conf. Comput. Aided Verification*, 2008, pp. 414–418.
[50] X. Wang, D. Feng, X. Lai, and H. Yu, "Collisions for hash functions MD4, MD5, HAVAL-128 and RIPEMD," IACR, Lyon, France, Rep. 2004/199, 2004.
[51] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full SHA-1," in *Proc. Annu. Int. Cryptol. Conf.*, 2017, pp. 570–596.
[52] Y. Sasaki, L. Wang, and K. Aoki, "Preimage attacks on 41-step SHA-256 and 46-step SHA-512," IACR, Lyon, France, Rep. 2009/479, 2009.
[53] D. A. Osvik, "Fast embedded software hashing," IACR, Lyon, France, Rep. 2012/156, 2012.
[54] J. Vidali, P. Nose, and E. Pašalić, "Collisions for variants of the BLAKE hash function," *Inf. Process. Lett.*, vol. 110, nos. 14–15, pp. 585–590, 2010.
[55] J. Daemen and G. Van Assche, "Producing collisions for PANAMA, instantaneously," in *Proc. Int. Workshop Fast Softw. Encrypt.*, 2007, pp. 1–18.
[56] M. Coutinho, R. T. De Sousa, and F. Borges, "Continuous diffusion analysis," *IEEE Access*, vol. 8, pp. 123735–123745, 2020.
[57] N. Abdoun. (2019). *Design, Implementation and Analysis of Keyed Hash Functions Based on Chaotic Maps and Neural Networks*. [Online]. Available: https://hal.archives-ouvertes.fr/tel-02271074/document
[58] N. Abdoun, S. E. Assad, T. M. Hoang, O. Deforges, R. Assaf, and M. Khalil, "Designing two secure keyed hash functions based on sponge construction and the chaotic neural network," *Entropy*, vol. 22, no. 9, p. 1012, 2020. [Online]. Available: https://www.mdpi.com/1099-4300/22/9/1012
[59] S. Al-Kuwari, J. H. Davenport, and R. J. Bradford, "Cryptographic hash functions: Recent design trends and security notions," IACR, Lyon, France, Rep. 2011/565, 2011. [Online]. Available: https://eprint.iacr.org/2011/565
[60] H. Feistel, "Cryptography and computer privacy," *Sci. Amer.*, vol. 228, no. 5, pp. 15–23, 1973. [Online]. Available: http://www.jstor.org/stable/24923044
[61] M. Stevens, "Attacks on hash functions and applications," Ph. D. dissertation, Math. Inst., Fac. Sci., Leiden Univ., Leiden, The Netherlands, 2012.
[62] H. Wang, Z. Ma, and C. Ma, "An efficient quantum meet-in-the-middle attack against NTRU-2005," *Chin. Sci. Bull.*, vol. 58, nos. 28–29, pp. 3514–3518, 2013.
[63] S. N. Kumar, "Review on network security and cryptography," *Int. Trans. Electr. Comput. Eng. Syst.*, vol. 3, no. 1, pp. 1–11, 2015.
[64] S. Ghosh, M. LeMay, D. M. Durham, and M. R. Sastry, "Processor hardware and instructions for SHA3 cryptographic operations," U.S. Patent 16 709 837, Apr. 16 2020.

Darshana Upadhyay received the master's degree in computer science from Nirma University, Ahmedabad, India. She is currently pursuing the Ph.D. degree with the Faculty of Computer Science, Dalhousie University. She also served as a Lecturer with Nirma University, before moving to Canada to pursue her Ph.D. degree. She was awarded the Gold Medal for securing the first position during her graduate study. Her primary research includes algorithm conceptualization, hardware design in the field of embedded systems, vulnerability assessments, and intrusion detection techniques for IoT/SCADA based systems. She is the co-recipient of the Indo-Canadian Shastri Research Grant in the field of wireless security and intrusion detection systems. She has been invited to be one of the Women in International Security–Canada's 2020 Emerging Thought Leaders.

Marzia Zaman received the M.Sc. and Ph.D. degrees in electrical and computer engineering from the Memorial University of Newfoundland, Canada, in 1993 and 1996, respectively. She started her career with Nortel Networks, Ottawa, ON, Canada, in 1996, where she joined the Software Engineering Analysis Lab and later joined the Optera Packet Core Project as a Software Developer. She has many years of industry experience as a Researcher and a Software Designer with Accelight Networks, Excelocity, Sanstream Technology, and Cistel Technology. Since 2009, she has been working closely with the Centre for Energy and Power Electronics Research, Queen's University, Canada, and one of its industry collaborators, Cistel Technology, on multiple power engineering projects. Her research interests include renewable energy, wireless communication, IoT, cyber security, machine learning, and software engineering.

Rohit Joshi received the bachelor's degree in mechanical engineering from the Birla Institute of Technology, Mesra, India, and the master's degree in innovation and technology management from the University of New Brunswick Saint John, Canada. He has over 20 years of experience in the domains of information security, risk management, and networking across multiple geographies. He has worked with organizations, such as Cistel Technology, Inc., Mariner Partners, HCL Technologies, Ramco System, and Sify Technologies Limited handling a variety of roles and providing end-to-end, IT security management consulting and solutions to large clients across various industry verticals. At Sify Technologies Limited, he was associated with Safescrypt which was the first licensed certifying authority in India that was set up in association with Verisign. His research interest include wireless communication, IoT, and cyber security.

Srinivas Sampalli (Member, IEEE) received the Bachelor of Engineering degree from Bangalore University and the Ph.D. degree from the Indian Institute of Science, Bangalore, India. He is currently a Professor and a 3M National Teaching Fellow with the Faculty of Computer Science, Dalhousie University. He has led numerous industry-driven research projects on Internet of Things, wireless security, vulnerability analysis, intrusion detection and prevention, and applications of emerging wireless technologies in healthcare. He currently oversees and runs the Emerging Wireless Technologies (MYTech) Lab and has supervised over 150 graduate students in his career. His primary joy is in inspiring and motivating students with his enthusiastic teaching. He has received the Dalhousie Faculty of Science Teaching Excellence Award, the Dalhousie Alumni Association Teaching Award, the Association of Atlantic Universities' Distinguished Teacher Award, the Teaching Award Instituted in his name by the students within his Faculty, and the 3M National Teaching Fellowship, Canada's most prestigious teaching acknowledgement. Since September 2016, he holds the honorary position of the Vice President (Canada), of the International Federation of National Teaching Fellows, a consortium of national teaching award winners from around the world.