Journal of Network and Computer Applications 135 (2019) 32–46

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Review

Implementation of deep packet inspection in smart grids and industrial Internet of Things: Challenges and opportunities

Gonzalo De La Torre Parra [a], Paul Rad [b,a], Kim-Kwang Raymond Choo [b,a,*]

[a] Department of Electrical and Computer Engineering, The University of Texas at San Antonio, San Antonio, TX 78249, United States
[b] Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, United States

ARTICLE INFO

Keywords:
Smart grid
Industrial control system
Industrial internet of things
Deep Packet Inspection
Advanced metering infrastructure
Cybersecurity
Forensic-by-design
Forensic-driven security monitoring

ABSTRACT

Upgrading a power grid to a smart grid is a challenging task. For example, since power grids were originally developed to support unidirectional communications, the migration process requires architectural and cybersecurity upgrades due to the integration of devices using bidirectional communication. The integration of these devices opens numerous avenues for cyber attacks, although they also enable numerous capabilities in smart grids. To protect the smart grid from cyber threats, it is important for industry and academia to explore and implement practical cybersecurity models together, for example collaboratively designing and developing suitable smart grid testbeds to facilitate research. In this paper, we survey existing literature relating to the infrastructure and communications for the energy sector and smart grids. Specifically, we study existing recommendations and models from government agencies (e.g. NIST and DOE) and academia, and evaluate deep packet inspection (DPI) approaches as a security tool for smart grids. We also propose a conceptual SDN-based security monitoring framework based on SDN, Network Behavior Analysis (NBA), Deep Learning Models, and DPI attack corroboration, as well as a conceptual forensic-driven security monitoring framework where digital forensics and investigation capabilities are integrated to inform security monitoring.

1. Introduction

Cybersecurity plays an essential role in safeguarding a nation's critical infrastructure. Critical infrastructures are selected based on whether their "incapacitation or destruction would have a debilitating effect on security, national economic security, national public health or safety, or any combination thereof" ("Critical Infrastructure Sectors, 2019"). The US Department of Homeland Security currently has 16 critical infrastructures listed, including the Energy Sector.

Protecting current industrial systems with robust cybersecurity practices is essential for any organization. As history has shown us, the consequences of failing to do so can potentially be damaging. Two relatively high profile instances of malicious infiltration executed through cyber-attacks are the Stuxnet Worm and the Ukraine Power grid attack.

The Stuxnet Worm from the summer of 2010 used programmable logic controllers (PLC) to modify the accelerations of Iranian nuclear centrifuges. During this time, the PLCs reported normal prerecorded centrifuge data back to the nuclear control engineers. While the control engineers were viewing what was presented as accurate data, the constant changes in acceleration wore down the mechanical components of the centrifuges, causing their deterioration, and their eventual failure (Holloway, 2015). Specifically, the malware operated in two steps:

1. Exploiting Windows zero-day vulnerabilities by targeting Windows networks systems and replicating the worm across them.
2. Infiltrating the Siemens Step7 software.

Step7 software is generally used in nuclear enrichment facilities. Once the worm reached the target software, it was able to gain access to the programmable logic controllers. All of this started when the 500-kilobyte worm gained access to the system through a USB device. The end result was an estimated 30% reduction in enrichment efficiency (Broad et al., 2011).

The second high profile attack example is the Ukraine cyber-attack that took place on December 2015. The attack can be broken down into five components, namely ("The Ministry of Energy and Coal intends to form a group of representatives of all energy companies within the management of the Ministry to study the possibilities of preventing unauthorized interference in the operation of power grids, 2016"):

* Corresponding author. Department of Information Systems and Cyber Security, The University of Texas at San Antonio, San Antonio, TX 78249, United States.
E-mail addresses: gonzalo.delatorreparra@utsa.edu (G. De La Torre Parra), paul.rad@utsa.edu (P. Rad), raymond.choo@fulbrightmail.org (K.-K.R. Choo).

https://doi.org/10.1016/j.jnca.2019.02.022
Received 2 October 2018; Received in revised form 30 January 2019; Accepted 25 February 2019
Available online 1 March 2019

1. Social engineering used to infiltrate and infect the network (Black Energy Trojan was used)
2. Control of Application Service Data Unit (ASDU) and shut down substations
3. Failure of power supplies, modems, remote terminals units, and switches
4. Deletion of data on servers and workstations through KillDisk
5. Flooding call centers with calls to deny service to customers with loss of power

Additional cyber attacks targeting power grids and other utility companies in the US have been perpetrated using spear phishing and watering-hole attacks. The attacks that started on March 2, 2017 targeting All-Ways Excavating, a subcontractor of the federal government, provided the attackers access to the SCADA systems from these power grid and utility companies distributed across at least 24 states according to the Wall Street Journal (Smith and Barry, 2019).

These precedents made it quite clear that cyber attacks targeting power grids can in fact cause a vast amount of damage to a region or country and therefore cybersecurity should be considered as a high priority. With the introduction of smart grids and the Internet of Things (IoT), new security issues will arise for the energy sector. Conventional industrial control systems (ICS), such as Supervisory Control and Data Acquisition (SCADA), were not designed around the idea of being connected to the Internet. In order to consider the current role of the main components in power grids, the new components required for migrating to a smart grid or an industrial IoT (IIoT) setting, the security vulnerabilities brought by these new components (e.g. IIoT devices), and the available techniques to secure this highly distributed infrastructure.

In summary, this paper makes the following contributions:

- Provide an overview on the power delivery model, smart grid features, components in the power grid, network connectivity, smart grid interoperability panel, and communication protocols used to connected devices in the grid.
- Provide insight of the vulnerabilities associated to the integration of software defined networks into smart grids and preset an attack classification structure based on the grid's attack vectors.
- Present a survey on Deep Packet Inspection (DPI) which covers a large amount of conceptual works and proposals.
- We identify the most efficient DPI methods that could be utilized for detecting attacks perpetrated against smart grids.
- Based on the surveyed materials, we present a conceptual framework for cybersecurity data analytics composed of DPI and Deep Learning Modules integrated into the Smart Grid Infrastructure, and a conceptual forensic-driven security monitoring framework.

The remaining sections of this paper are organized as follows. Section 2 briefly introduces smart grids and the smart grid interoperability panel. Section 3 reviews existing DPI applications and literature. Additionally, this section covers challenges and potential solutions offered by existing approaches. Finally, in Section 4, we present our conclusion and remarks.

2. Smart grids

To understand the new functions that different technologies could bring to smart grids, in this section we present a background on the power delivery model, a summary of the communication protocols and standards used by different devices within Smart Grids, an overview of the Smart Grid Interoperability Panel (SGIP), the integration of Software Defined Networks in Smart Grids, Industrial IoT, and Deep Learning applied to this field.

2.1. Background

The traditional power delivery model consists of four phases, namely: generation, transmission, distribution, and customer. During the generation phase, power is being generated in power plants, wind farms, and solar farms, among other sources. The transmission phase deals with delivery of power over large distances. During transmission, it is sometimes necessary to increase/decrease (step-up/step-down) voltage with the use of transmission substations. The distribution phase is the last step in power delivery. It is the process of stepping down voltage, typically by transformer, from the transmission phase to be delivered to the customer. The customer phase is simply the consumption of the power. This power delivery flow is presented in Fig. 1.

There are a variety of components within the delivery model that assist in getting the correct amount of power to the desired location. Master Control Units (MCUs) serve as the human machine interface (HMI) for SCADA systems (Berry, 2011). These can be thought of as control rooms in power plants. The master control units are connected to Remote Terminal Units (RTUs) at substations in the field, as shown in Fig. 1. The RTUs provide the master control units with event-driven feedback and are designed to withstand harsh field condition. Since technological trends move quickly, they become susceptible to newer

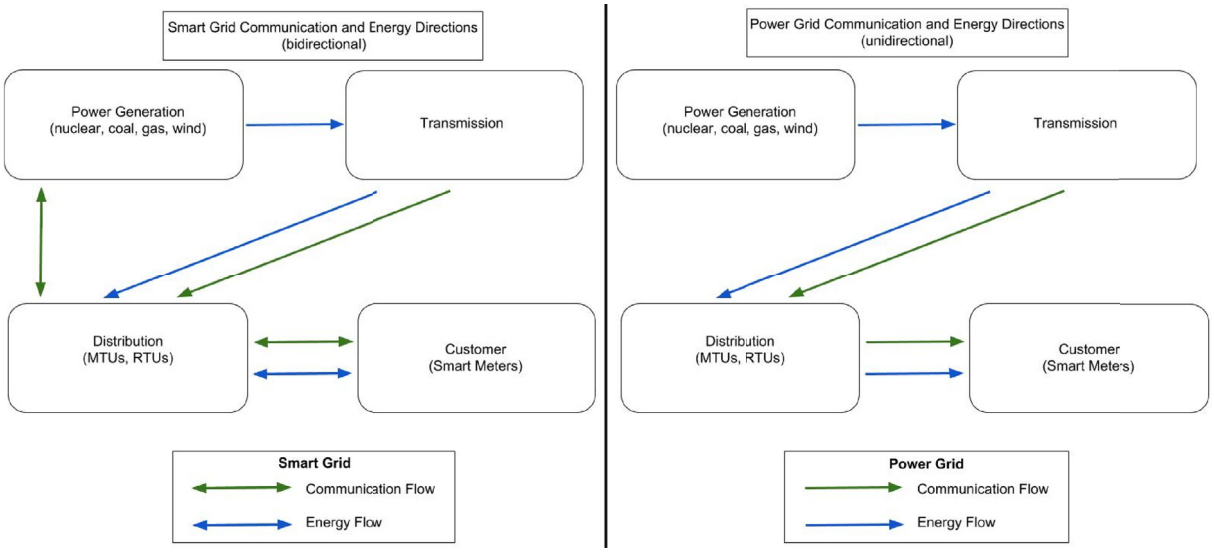

Fig. 1. Power delivery flow in smart grids and power grids (legacy).

cybersecurity threats more easily. MCU operators can issue high-level commands to the RTUs about their controls. After the RTUs update, the RTUs continue to function as controllers. RTUs can further communicate with Intelligent Electronic Devices (IEDs). IEDs are used for protection, metering, monitoring, and communication purposes (Darwish et al., 2015). Examples of intelligent electronic devices include circuit breakers, transformers, capacitor banks, Programmable Logic Controllers (PLCs), among others. This allows for automated device management at substations.

MCUs, RTUs, and IEDs communicate through Transmission Control Protocol/Internet Protocol (TCP/IP) platforms using Distributed Network Protocol (DNP3) (Zeceña and Molina, 2017). DNP3 utilizes the following layers from the Open System Interconnection (OSI) reference model: data link layer, transport layer, and application layer. The DNP3 message is encapsulated in the TCP, which is then encapsulated by internet protocol. DNP3 is the most widely used communication protocol within the energy sector of the United States (Darwish et al., 2015). It is currently an active IEEE standard as of 2019 and defined by the IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3) ("IEEE 1815–2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3), n.d."). Some other popular communication protocols are MODBUS, PROFIBUS.

2.2. Communication technologies and protocols used by smart devices

The grid's design is strictly focused on unidirectional communication. In contrast, smart devices require bidirectional communication to accomplish their functionality. An example of such devices is the smart meter which combines pre-existing automated meter reading systems and automatic meter management systems. The information collected by such devices is transmitted using home area networks (HANs), wide area networks (WANs), or private proprietary networks. This architecture is known as the Advanced Metering Infrastructure.

A smart meter is designed to obtain information from the customer and the utility grid. The smart meter contains a power module, control module, metering module, timing module, communication module, indicating module, encoding module, and timing module (Yang et al., 2014). Tampering with any of the listed modules could cause inaccurate pricing, loss of power, and privacy violations. To prevent these from happening, security measures need to be explored on communication protocols with smart meters.

To assist in building communication architecture of the smart grid, IEEE has developed the IEEE 2030–2011 standard. The standard provides streamlined definitions and order to the three subnetworks. The three subnetworks are the private network, wide area network (WAN), and core network. Private networks consist of home area networks (HAN), industrial area networks (IAN), and building area networks (BAN). Wide area networks consist of neighborhood area networks (NAN) and field area networks (FAN). Wide area networks are of interest as they provide communications for remote terminal units, advanced metering infrastructure, and phasor measurement units. Wide area network security is essential because remote terminal units contain programmable logic controllers and phasor measurement units (PMU). These are critical for monitoring the system's health. Core networks consist of local area networks (LAN), virtual private networks (VPN), voice over internet protocol (VoIP), and geographic information systems (GIS). There are two ways for data to be transmitted within these subnetworks: wireline communications and wireless communications. Tables 1 and 2, adapted from (Kabalci, 2016), present an overview of the communication technologies discussed previously. Table 3 provides a quick summary of standards and protocols associated with the communication technologies presented on Tables 1 and 2.

2.3. Smart grid interoperability panel

The Smart Grid Interoperability Panel (SGIP) was initiated by the National Institute of Standards and Technology (NIST) to assist in coordinating the standards development for the smart grid. SGIP allows NIST to solicit input from private and public stakeholders in the interest of developing a framework for smart grid standards. A variety of members from industry, academia, and government work together to find a reasonable solution for advancing the smart grid technologies. As the needs of each entity are vastly different, in this section we will focus on industry-oriented research. Specifically, this section will examine three white papers from SGIP ((TIA), 2008; Others, 2010; Wang and Wacks, 2017), and the Department of Energy (Stevens, 2014). The SGIP papers can be acquired through the sepapower.org weblinks included in the reference section.

The NIST created a voluntary framework to assist the energy sector and its consumers called the "Framework for Improving Critical Infrastructure Cybersecurity" (Cybersecurity Framework). The purpose of this document is to assist organizations in understanding their level of need for cybersecurity programs in their business. From here, the document helps set forth a path to create a strategic plan for improving the current cybersecurity systems. Another document of relevance is the US Department of Energy's Electric Subsector Cybersecurity Capability Maturity Model (ES-C2M2) (Stevens, 2014). ES-C2M2 focuses on evaluating the resiliency and longevity of a cybersecurity program. This evaluation allows the organization to gain insight about a program's shortcomings and address found issues. By using the Cybersecurity Framework and ES-C2M2 frameworks, an organization can develop a cybersecurity program that fits their needs. They provide a roadmap to develop the program as well as valuable information about working with

Table 1
Wireline communication technologies – adopted from (Kabalci, 2016).

Technology	Standards	Data Rate	Distance	Network
PLC	- NB-PLC: ISO/IEC 14908-3,14543- 3-5, CEA-600.31, IEC61334-3-1, IEC 61334-5 (FSK) - BB-PLC: TIA-1113 (HomePlug 1.0), IEEE 1901, ITU-T G.hn (G.9960/G.9961) - BB-PLC: HomePlug AV/Ext., PHY, HD-PLC	- NB-PLC: 1–10 kbps for low data rate PHYs, 10–500 kbps for high data-rate PHYs - BB-PLC: 1–10 Mbps (up to 200 on very short distance)	- NB-PLC: 150 km or more - BB-PLC: about 1.5 km	- NB-PLC: NAN, FAN, WAN, large scale - BB-PLC: HAN, BAN, IAN, Small scale AMI
Fiber Optic	- AON (IEEE 802.3ah) - BPON (ITU-T G.983) - GPON (ITU-T G.984) - EPON (IEEE 802.3ah)	- AON:100 Mbps up/down - BPON:155–622 Mbps - GPON: 155–2448 Mbps up, 1.244–2.448 Gbps down - EPON: 1 Gbps	- AON: up to 10 km - BPON: up to 20–60 km - EPON: up to 20 km	- WAN
DSL	- ITU G.991.1 (HDSL) - ITU G.992.1 (ADSL), ITU G.992.3 (ADSL2), ITU G.992.5 (ADSL2þ) - ITU G.993.1 (VDSL), ITU G.993.1 (VDSL2)	- ADSL: 8 Mbps down/1.3 Mbps up - ADSL2: 12 Mbps down/3.5 Mbps up - ADSL2+: 24 Mbps down/3.3 Mbps up - VDSL: 52–85 Mbps down/16–85 Mbps up - VDSL2: up to 200 Mbps down/up	- ADSL: up to 5 km - ADSL2: up to 7 km - ADSL2þ: up to 7 km - VDSL: up to 1.2 km - VDSL2: 300m–1.5 km	- AMI, NAN, FAN - HAN, BAN, IAN, NAN, FAN, AMI

Table 2
Wireless communication technologies – adopted from (Kabalci, 2016).

Technology	Standards	Data Rate	Distance	Network
Wi-Fi	- IEEE 802.11e - IEEE 802.11n - IEEE 802.11s - IEEE 802.11p (WAVE)	- IEEE 802.11e/s: up to 54 Mbps - IEEE 802.11n: upto 600 Mbps	- IEEE 802.11e/s/n: up to 300 m - IEEE 802.11p: up to 1 km	- HAN, BAN, IAN, NAN, FAN, AMI
WiMAX	- IEEE 802.16 (fixed and mobile broadband wireless access) - IEEE 802.16j (multi-hop relay) - IEEE 802.16 m (air interface)	- 802.16: 128 Mbps down/ 28 Mbps up - 802.16 m: 100 Mbps for mobile, 1 Gbps for fixed users	- IEEE 802.16: 0–10 km - IEEE 802.16 m: 0–5 (opt.), 5–30 acceptable, 30-100 km low	- NAN, FAN, WAN, AMI
GSM	- 2G TDM, IS95 - 2.5G HSCSD, GPRS - 3G UMTS (HSPA, HSPA+) - 3.5G HSPA, CDMA EVDO - 4G LTE, LTE-Advanced	- 2G: 14.4 kbps - 2.5G: 144 kbps - HSPA: 14.4 Mbps down/ 5.75 Mbps up - HSPA+: 84 Mbps down/ 22 Mbps up - LTE: 326 Mbps down/ 86 Mbps up - LTE-Advanced: 1 Gbps/ 500 Mbps	- HSPA+:0–5 km - LTE-Advanced: optimum 0–5 km, acceptable 5–30, 30–100 km (reduced performance)	- HAN, BAN, IAN, NAN, FAN, AMI
Satellite	- LEO: Iridium, Globalstar - GEO: Inmarsat, BGAN, Swift, MPDS - NGAN: up to 1 Mbps	- Iridium: 2.4–28 kbps - Inmarsat-B: 9.6 up to 128 kbps	- 100-6000 km	- WAN, AMI

Table 3
Communication standards and protocols.

Standard	Summary
IEEE Standard C37.118.1-2011	Phasor measurement unit standards (C37.118.1-2011 IEEE Standard for Synchrophasor Measurements for Power Systems., n.d.)
IEEE Std. 1588	Protocol enabling precise synchronization of clocks in measurement and control systems implemented with technologies such as network communication, local computing and distributed objects ("IEEE 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, n.d.")
IEEE Std. 2030–2011	Establishes the smart grid interoperability reference model (SGIRM) and provides a knowledge base addressing terminology, characteristics, functional performance and evaluation criteria, and the application of engineering principles for smart grid interoperability of the electric power system with end-use applications and loads ("IEEE 2030–2011 - IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads, n.d.")
IEEE Std. 1547.4	Design, operation, and integration of distributed resource (DR) island systems with electric power systems (EPS) (Institute of Electrical and Electronics Engineers. and IEEE-SA Standards Board., 2011a)
IEEE Std. 1547.6	Recommendations and guidance for distributed resources (DR) interconnected on the distribution secondary networks, including both spot networks and grid networks (Institute of Electrical and Electronics Engineers. and IEEE-SA Standards Board., 2011b)
ISO/IEC 15067-3	Home electronic system (HES) application model – Part 3: Model of a demand-response energy management system for HES ("ISO/IEC 15067–3:2012 - Information technology – Home Electronic System (HES) application model – Part 3: Model of a demand-response energy management system for HES, n.d.")
IEEE Std 1815–2012	DNP3 protocol structure, functions, and interoperable application options (subset levels) are specified ("IEEE 1815–2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3), n.d.")
IEEE Std 802.22	Protocol wireless regional area network (WRAN) (Ucar et al., 2016)
IEEE 802.15.4–2015	Protocol wireless personal area network (WPAN) (Stevenson et al., 2009)
TIA-1113	Medium-Speed (up to 14 Mbps) Power Line Communications (PLC) Modems using Windowed OFDM ((TIA), 2008)
IEEE 1901 FFT-OFDM	Standard for high-speed communication devices via electric power lines, so called broadband over power line (BPL) devices (Others, 2010)
IEEE 1901 Wavelet-OFDM	Standard for high-speed communication devices via electric power lines, so called broadband over power line (BPL) devices (Others, 2010)

various entities in an organization ensure that the cybersecurity program is in alignment with the needs of the entire organization. This "Utility Lessons Learned" document provides a quick overview of both documents while highlighting the essential components.

In ("NIST Cybersecurity Framework Implementation Case Study, 2017"), the goal is to facilitate an organization in utilizing the frameworks. Due to the amount of resources that were put into the frameworks, there is a strong interest in seeing them implemented. SGIP provides the vessel for implementing sustainable cybersecurity risk-management programs with the support of executive leaders ("NIST Cybersecurity Framework Implementation Case Study, 2017") targeting the electric power industry.

The case study acts as a walkthrough for implementing the Cybersecurity Framework. It breaks down into the following steps: (1) scope and prioritize - identify and prioritize business functions and systems, (2) orient - identify in scope systems, (3) create a current profile - identify the current state, (4) conduct risk management, (5) create a target profile - identify the desire state, (6) identify gaps, (7) brief management through dashboard, (8) create action plans to mitigate gaps, and (9) maintain cybersecurity framework program.

In ("Smart Grid System Security with Broadcast Communications, 2017") the authors focus on the broadcast communication functions of two types of devices: direct load control and indirect load control. It provides specific security guidelines and regulations associated with smart grid communications. There are three major infrastructures that require broadcast security measures: the utility, broadcast, and the customer. These can be considered as potential points of entry for malicious content.

2.4. Integration of Software Defined Networks in Smart Grids

Approaches developed since 2015, including those of Dong et al. (2015), present thorough insight of the opportunities and risks that SDNs can bring to smart grids. Opportunities include enhancing system resiliency, enriching functionality over smart grid communications, and improving Quality of Service. On the other hand, suggested risks include

software vulnerabilities and reduced operational quality over smart grids. The authors validate their proposals by producing a testbed using Mininet and PowerWorld to produce a ground truth and Bro IDS for attack detection. The greatest advantage of SDNs is their capability to separate the control plane from the data plane, distinguishing themselves from legacy switches, which serve only as forwarding tasks. The switch and the controller can conform to a control plane protocol such as OpenFlow, enabling operators to re-define networks in run time.

In addition, SDNs can be easily integrated with DPI systems, IDS, IPS, behavioral analytics and deep learning models to detect attacks, classify attacks by family, and re-structure the network based on the detected attack types. Smart grids integrating SDNs can be segmented as presented in Fig. 2.

We can further classify attacks on Smart Grids incorporating SDNs as presented in Fig. 3.

3. Deep packet inspection (DPI)

Earlier implementations of DPI generally use exact string matching to perform fast string matching for signatures. The approach of using exact string matching is now obsolete as signatures became more complicated; thus, making exact string matching ineffective. Modern network intrusion detection systems, such as SNORT, BRO, and L7-filter, as well as many network security systems, use regular expression for rule representation and matching. Currently, most DPI applications are based on automaton-based pattern matching which utilize finite state machines capable of recognizing the language expressed by regular expression and use their capabilities for signature matching (Xu et al., 2016). Finite state machines (FSM) can be divided into nondeterministic finite automata (NFA) and deterministic finite automata (DFA). The main difference between them is observed in the number of transitions and states based on a received character. In this sense, DFA only has one transition going to a single state, while NFA can have multiple transitions going to different states. Although this is a meaningful difference, an equivalent DFA can be constructed based on a given NFA and similarly a NFA can be constructed based on a DFA. The major problem when producing a conversion of NFA to DFA is that the number of number of states will exponentially increase given that NFA can have multiple transitions to multiple states; this problem is known as state explosion. Table 4 presents the processing cost and memory consumption when compiling m regular expressions into an integrated DFA (m being the number of regular expressions and n being the average length of regular expression) while Table 5 presents the processing costs and memory consumption when compiling m regular expressions into m FAs.

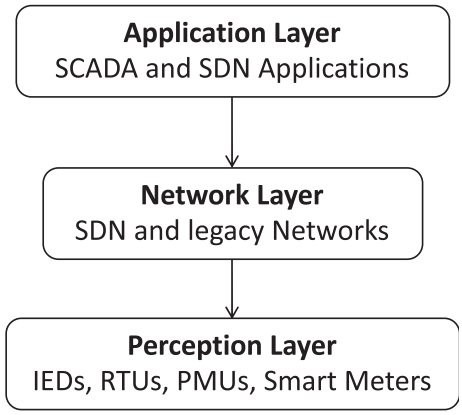

Fig. 2. Smart Grids incorporating SDN division by layers.

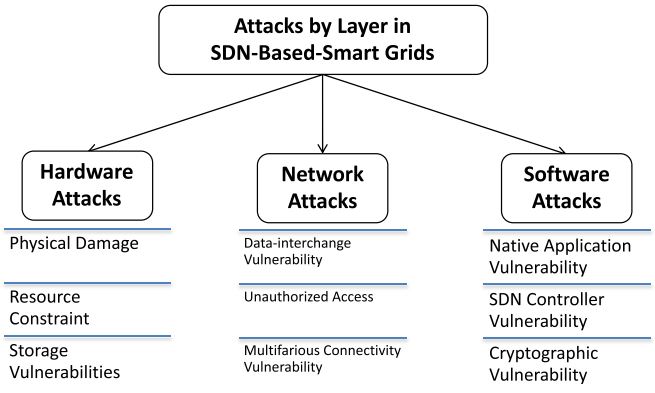

Fig. 3. Smart Grids incorporating SDN attack classification.

Table 4
DFA vs. NFA compiling m reg. expressions into an integrated DFA.

	Processing Cost	Memory Consumption
DFA	$O(1)$	$O(2^{mn})$
NFA	$O(n^2m)$	$O(mn)$

Table 5
DFA vs. NFA compiling m regular expressions into m FAs.

	Processing Cost	Memory Consumption
DFA	$O(m)$	$O(m2^n)$
NFA	$O(n^2m)$	$O(mn)$

Based on the information presented in the tables above, in order to comply with the demands of wire-speed pattern matching along with the increased number of patterns to be matched by the increased number of network applications, the ideal would be to perform pattern matching as fast as DFA can process while keeping NFA's memory consumption. Additional information about this methodology and proposals covering NFAs and DFAs are covered in sub-section 3.3. While there are a number of proposals seeking to achieve these objectives, it is important to mention that devices such as GPUs, FPGAs, TCAM, and Multi-core processors are used for pattern matching acceleration.

3.1. Application areas

DPI application areas have been classified by different authors such as Xu et al. (2016) and Mueller (2011), into families presented in Sections 3.1.1 to 3.1.5.

3.1.1. Network security

The data created by the diverse range of devices requires resilient and robust networks capable of transmitting the massive amount of information generated by these devices. In addition, such networks need to facilitate the use of monitoring and security systems capable of analyzing real-time data, as well as detection of network traffic anomalies, external attacks, and insider threats. The threat is real. For example, according to Symantec (2018), in 2017 there was an 8500% increase of coin-miners detected at client devices (e.g. mobile devices), a 600% increase in overall IoT attacks which can potentially be used for mining purposes, and a 200% increase in supply chain attacks such as the high profile attack against Ukraine using Petya/NotPetya.

Another recent trend is ransomware attacks (as previously noted), where data on infected systems are encrypted (Azmoodeh et al., 2018), and messages requesting for ransom to be paid will then be displayed on such infected systems. A similar attack against the energy sector was reported at Saudi Arabia in 2012. In another attack, disk-wiping malware W32.Disttrack and a different variant named W32.Disttrack.B were used against targets between November 2016 and January 2017. The politically motivated attacks were allegedly performed by different groups “Greenbug” and “Timberworm” by exploiting operating system features, legitimate tools, and cloud services to compromise networks. In the 2018

report of Cisco (Cisco Systems, 2018), for example, network-based ransomware observed in 2017 reportedly do not require human involvement, and it was also noted that attackers are now concealing their attacks using encrypted channels. These encrypted channels were in cases facilitated by legitimate services such as Google, Github, and Dropbox. Such an approach permits attackers more time to operate and maximize the damage to their targets. Finally, the report from Cisco also noted the increase of malicious activity targeting IoT devices. Unpatched IoT devices are an attractive attack vector that can be used to infiltrate networks or create automated botnets that can be used to create advanced DDoS attacks against other systems.

In (Pandalabs, 2018), PandaLabs summarized critical data breaches such as the controversial Cambridge Analytical scandal. In addition to this high profile incident, there are other significant data breaches such as those involving Aadhaar (1.1 billion records), Marriott (500 million affected users), Exactis (340 million records), Under Armor (150 million records), Panera Bread (37 millio records), US voters (35 million records); these incidents exemplify the potential impact and consequences due to a successful attack.

Network security was, probably, the early motivator for DPI seeking to combine the malware detection capabilities with packet capture and analysis (Mueller, 2011), (Prokhorenko et al., 2016). The desire to achieve the stated combination lead to the early development of IDS aimed to detect known attacks while IPS was used to respond to the detected threats. Many solutions utilize DPI for detecting attacks on incoming traffic, one example is presented by Wendel and Roessler ("Data loss prevention, 2003"). Their patent focuses on data loss prevention that employs DPI to detect hidden information in normal communication and preventing this from leaving the organization's network.

In addition, it is important to consider the strong dependency on cloud computing for a variety of network resources, Cloud forensics was developed as new field within network forensics, but there are a number of challenges (e.g. technical, organizational, and legal) associated with its practice (Hooper et al., 2013) (Quick et al., 2014) (Choo and Dehghantanha, 2017) (Teing et al., 2017). Some of the challenges within these areas are unique to cloud forensics such as data replication, location transparency, and multi-tenancy (Choo et al., 2016). Also, in recent years, researchers have proposed the need to integrate forensic best practices into the design of cloud and related systems such as cyber-physical cloud systems, vehicular fog computing, and Internet of Drones (Ab Rahman et al., 2017).

3.1.2. Bandwidth management

Asghari et al. (2013) proposed the use of DPI for bandwidth management, which is of great interest for internet service providers due to the increase of network media and file-sharing popularity. DPI enables bandwidth management by classifying traffic based on recognized applications or protocols. Furthermore, it enables mechanisms such as scheduling algorithms, traffic shaping, and congestion avoidance for each traffic class. Similarly, DPI can be utilized to allow or block certain applications by recognizing the utilized protocols for the purpose of prioritizing and avoiding different types of traffic. One of the most important traffic types targeted for DPI bandwidth management systems is P2P since large file sharing can jeopardize latency-sensitive applications such as video conferencing, streaming video content, or voice over IP.

An example of latency-sensitive applications growth, according to Cisco Visual Networking Index: Forecast and Methodology, 2016–2021 (Cisco, 2017), internet video to TV grew 50% in comparison to the traffic seen in 2015, and it is expected that consumer video-on-demand (VoD) traffic will be equivalent 7.2 billion DVDs per month by 2021 while Content Delivery Network (CDN) traffic is expected to carry 71% of all internet traffic by 2021. On the other hand, Global compound annual growth rate (CAGR) for P2P file transfer is expected to decrease 6% from 2016 to 2021. It is important to note that during the same time period P2P file transfer is expected to increase 13% in North America, meaning that different strategies must be taken by internet service providers to manage bandwidth using DPI. These forecasts become of relevant importance considering the study presented by Perez et al. (2007), who take into consideration the P2P business potential in CDN for transferring large files such as live streaming and VoD.

3.1.3. User profiling/ad-injection

Internet Service providers have the capability to inspect all traffic passing through their networks using DPI, making user profiling possible. User profiling (Peng et al., 2016) is of great interest to a variety of business and stakeholder groups, since it opens the opportunity for targeted advertising for each specific user and has potential in user authentication. Ad injection is based on the user's profile knowledge, which can be collected by tracking and analyzing visited websites and recent purchases. Based on this knowledge, online marketing and advertising can then be tailored for the targeted user. Organizations have been known to utilize their tools (e.g. search engine tools) and capabilities to collect such information and then facilitate third-party companies to promote their products through ads presented on the user's main page or search engine. An interesting case study is presented by Cheng (Cheng, n.d.), where $8 million US Dollars were reportedly earned in 2013 by the Yontoo browser. In this case study, it was revealed that Facebook sessions of about 4.5 million users were modified using ad injection. Thomas et al. (2015) explained that at least 5.5% of the users connecting to Google services have an ad injector and that about 50,870 of Chrome Extensions where malicious. Their study also revealed that there is great discomfort among the users due to how it affected user's browsing experience, privacy and security. Chrome received about 100,000 complaints in July 2014, 20% of which were concerning ad injection.

3.1.4. Copyright enforcement

The entertainment industry has a great interest for internet service providers to utilize DPI for detecting copyright infringement activities. Due to the monitoring capabilities of DPI, it can be utilized for inspecting traffic containing copyright materials. However, such a process cannot be inspected through hash comparison or bit-mash methods due to the different formats a media file can contained (Mueller et al., 2012). There are a number of related products, such as Audible Magic, which utilized fingerprinting technology with the utilization of DPI for copyright enforcement (Ikezoye and Schrempp, 2000) (Schmelzer and Pellom, 2002). Audible Magic users utilize the software to generate a unique signature from the given material and this signature is further registered in a database. When a specific copyright content passes through a network, DPI is utilized to calculate the signature and attempts to match it with the registered fingerprint. Although copyright enforcement solutions are of great use for parties holding copyright material, internet service providers may not benefit from such operation. There have also been a series of lawsuits against internet service providers, as explained by Bendrath and Mueller (2011) and Bendrath (2009).

3.1.5. Government surveillance and censorship

A top priority for governments around the world is to preserve security within their territory, and their ability to monitor and pinpoint conversations and network behavior from specific targets is critical for accomplishing such operations. Full-scale surveillance and monitoring capabilities are often demanded by governments to Internet service providers in order to analyze traffic passing through their networks and keep government organizations informed. Such activities can be pinpointed to e-mails, IP addresses, phone numbers, or user accounts. Regardless of how such activities may benefit a particular government, these often surpass foreseen monitoring limits; as an example, the National Security Agency's deployment of PRISM (reenwald and MacAskill, 2013) around the world has the capability of monitoring any particular individual.

A number of researchers and international bodies have criticized the extension of such activity. For example, Bendrath (2009) reported on the

legal issues and conflicts from the U.S. government's actions on individual privacy. In addition to providing such capability, DPI can also be utilized by government entities to enforce Internet censorship, which might include political criticism. Due to the difficulty of provisioning content-based censorship, the most common practice is URL blocking as stated by Mueller (2011).

3.2. Methods and deployment

In this section, we will review existing DPI methods including pattern matching, statistical analysis, and protocol decoding, as presented in Fig. 4. In addition, common deployments of DPI will be covered in this section.

As previously discussed, DPI can be utilized to inspect packet headers and identify applications. It can also be used to inspect the payload to recognize content (through signature matching) to classify traffic, provide NIDS capabilities, and so on. Existing DPI methods can be broadly categorized into Port-based identification, narrow pattern matching (string matching and regular expression matching), statistical analysis, and protocol decoding (Xu et al., 2016), (Finsterbusch et al., 2014), (Lin et al., 2006), (Lin et al., 2008).

For protocol identification, a traditional method such as a port-based approach can be utilized. This is achieved by identifying used ports in the TCP/UDP headers as presented in IANA ("Service Name and Transport Protocol Port Number Registry, 2016") in order to relate known ports with applications that use them. As presented by Qi et al. (2009), this method is widely used for access control lists and firewall rules due to its high efficiency and encryption immunity.

Based on the studies presented by Moore and Papagiannaki (2005) and Madhukar and Williamson (2006), it was determined that such an approach is capable of recognizing 30%–70% of traffic generated by certain protocols due to the way applications such as P2P use random ports for connectivity. Dusi et al. (2011), used this method for identifying applications that always used pre-assigned ports.

Piskac et al. (Piskac and Novotny, 2011) used extended flow statistical analysis to observe time characteristics in the data flow to classify traffic. These statistics looked at inter-arrival packet time, packet direction, and packet sizes. The data flow was vectorized and those vectors were compared using root-mean-square distance, Euclidian distance, and the angle between vectors. 90% accuracy was achieved using the extended flow statistical analysis methodology. One of the advantages of this is it is faster than DPI and it can inspect encrypted data. The disadvantage is it can only be used in one specific situation for one protocol detection.

Vector Space Modeling (VSM) is used in natural language processing (NLP) to classify documents based on certain information requests or queries in the work of Chung et al. (2009). This is accomplished through determining the significance of a term by weighing them. Term weighing components used in natural language processing are not suitable for traffic classification. Cosine similarity is more suitable to classify network traffic. This is done by comparing the cosine similarity packet payloads once they are converted into vectors. The outcome defines how similar the payload vector flows are to each other. Such approach is capable of delivering an accuracy of 96%.

Rocha et al. (2011) performed multi-scale analysis on samples of IP data-streams to gather multi-scale estimators for all upload/download streams. Estimators are processed by mapping a dimension to each time scale using Discrete Wavelet Transforms. This allows the multivariate Gaussian distributions to be used to detect if a protocol from a known stream correlates with that of an application protocol from an unknown stream. The advantage of this methodology is the potential for detection of illicit streams in near real time. As an example, for NMAP and SnapShot, illicit traffic streams are 100% and 97.23% respectively.

Nguyen and Armitage (2008) proposed a method, which combines statistical traffic properties along with machine learning techniques to classify traffic from 2004 to 2007. The machine learning techniques were able to classify Internet traffic applications with the highest rates being 99% accurate.

Protocol Decoding performs protocol recognition using the protocol characteristics at the headers and its behavior, which can be represented using a state machine. Due to this, protocol decoding depends on the re-establishment of application sessions and packets captured during these sessions. Protocol decoding is presented by Finsterbusch et al. (2014) as a class of traffic classification while Xu et al. (2016) considers it a light-weight pattern matching method. The advantage of protocol decoding is its high accuracy with a low false negative rate as presented by Risso et al. (2008); nevertheless, it requires thorough understanding of the protocol and it is computationally expensive.

Heuristic-based matching algorithms are presented in a number of studies, such as those of Wu and Manber (1994), Boyer and Moore (1977), Yao (1979), Galil (1978), Horspool (1980), Navarro (Navarro and Raffinot, 2002), and Liu et al. (2004). Such an approach focuses on exact string matching by skipping the characters within the payload as much as possible using heuristics in the form sliding windows covering a set of characters and jumping to the next window based on the matching results in order to speed up the matching process.

In contrast with heuristic-based matching algorithms, a hashing-based method computes a value for each string in the payload and compares this value with pre-computed string patterns. When a match is found, the substring is extracted and is once again compared with the current substring byte by byte. The approaches of Muth and Manber (1996) and Karp and Rabin (1987) are based on such a methodology.

Automaton-based methods have been used for string matching or regular expression matching. The algorithm presented by Aho and Corasick (1975) is a classical algorithm in this category and while the time complexity is independent of the pattern size, performance will drop with large pattern sets due to the constructed large transition tables as denoted by Lin et al. (2006). Improved algorithms based on this classical

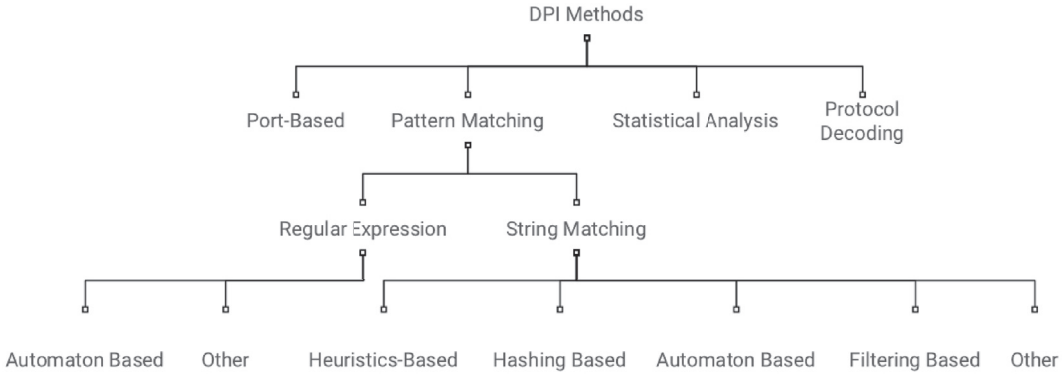

Fig. 4. DPI methods.

algorithm have been presented by Norton (2002), Tuck et al. (2004), and Tan and Sherwood (Tan et al., 2005). Their studies aim to reduce the memory space required for automaton storage.

The methods presented depend on a series of software and hardware deployments for their correct implementation. Starting with packet capture, there are several methods to perform such tasks such as using a cable splitter, port mirroring as presented by Callado et al. (2009), and virtual interface mirroring, among others. Antonello et al. (Antonello et al., 2012a, b) presented in their study the capability of capturing packets at speeds ranging the Mbps with commodity hardware; nevertheless, it is still difficult to capture packets at Gbps speeds due to the high CPU load to process the packets. Additionally, the authors evaluated and compared the performance of popular APIs such as Libpcap, ("The libpcap project, n.d."), Pf_Ring ("Pf_Ring:High Speed Packet Capture Filtering and Analysis, n.d."), and libe1000 ("User-space E1000 driver library, n.d."), all of which bypass default packet handling mechanisms to save CPU resources. Other studies presented by Bonelli et al. (2012), Rizzo (2012), and Rizzo et al. (2012) propose solutions to deliver improved performance using customized packet technologies Kim et al. (2014). In addition to these studies, Intel released a set of libraries and drivers for a fast paced processing named data plane development kit (DPDK) ("DPDK: Data Plane Development Kit, n.d."). Schneider et al. (2007) present an analysis focused on the challenges for packet capturing from the hardware perspective implemented in Ethernet environments on which he identified that inherent memory and system bus throughput as the main constraints for packet capture. Deri et al. (Deri, 2004), Alcock et al. (2012), and Hofstede et al. (2014) present in their papers a deeper understanding for packet capture.

Flow-based inspection focuses on an analysis of a flow, which consists of a set of packets with the same five-tupple (protocol number, source ip, destination ip, source port, and destination port). Cisco NetFlow is one of the most popular standards of flow information; similarly, such flow information can be reconstructed from packet captures using open source TCP reassembly programs such as Softflowd for Ubuntu or Pfflowd for OpenBSD with the aid of Tshark, Libnids, and TCP Flow. This inspection method requires reconstructing all packets before reaching the receiver. Due to this, more computational resources are required in order to maintain associations between packets from the same flow. In the case of TCP, this process is called TCP reassembly through which the fragmented or disordered packets are fabricated into integrated blocks. Experiments produced by Egorov and Savchuk (2002) and Chen et al. (2014) showed that TCP reassembly generates most of the workload. Additional information on this topic is covered by Wagener et al. (2008), Zhang and Ju (2003), and Dharmapurikar and Paxson (2005). Regardless of the computationally expensive process, flow-based inspection imposes a tradeoff between performance and inspection accuracy.

Trabelsi et al. (2016) propose a hybrid mechanism to enhance the packet filtering capabilities of intrusion detection systems and deep packet inspection performance on the packet's payload. The IDS mechanism consists of four algorithms: BSPL, splay filter builder, optimized early packet rejection, and module stability test. The Deep Packet Inspection module consists of an algorithm enhancing the IDS's multi-pattern matching algorithm by adjusting the DPI rules order based on traffic statistics for the purpose of enhancing the process of packet inspection. Together, the IDS-DPI modules compute the search of predefined patterns within the fields in the application layer of the payload data. The experiment was performed on a 1.6 GHz i7 and 4 GB Ram Windows 7 system. The dataset consist of the CAIDA dataset ("CAIDA Data, n.d.") and it was divided into different subsets using different window sizes in order to measure performance. The optimized DPI module provisioned an average time gain of 20%.

3.3. Prior DPI surveys

In this section, we summarize existing literature surveys focused on different aspects, such as Internet traffic classification, string matching, and finite state automata.

Finsterbusch et al. (2014) presented a DPI survey focused on the analysis of the performance, technical requirements and classification accuracy for Internet Traffic (based on the used protocols and services), which is achieved through the content analysis of captured packets. Furthermore, they tested, evaluated, and compared popular open-source DPI modules. Fig. 5 presents the Internet classification approaches presented by the author.

In this work, the following open-source classification approaches are evaluated: OpenDPI, nDPI, libprotoident, IPP2P, HiPPIE, and L7-filter. The dataset used contained 799,029 packets (captured using TCPdump and Wireshark) distributed over 2,104 flows. The protocols used in the dataset include BitTorrent, DNS, eDonkey, HTTP, IMAP(S), Oscar, POP(S), RTP, SIP, and SMTP(S). The presented results include classification accuracy, memory utilization, and CPU usage.

Nguyen and Armitage (2008) present a survey focused on machine-learning based traffic classification presented in different works ranging from 2004 to 2007 presenting an accuracy of up to 99%. On the other hand, Callado et al. (2009) present a survey on the techniques and problems for IP traffic classification and focus on application detection. The presented classification approach is divided into two families: packet-based and flow-based. The authors conclude that payload-based

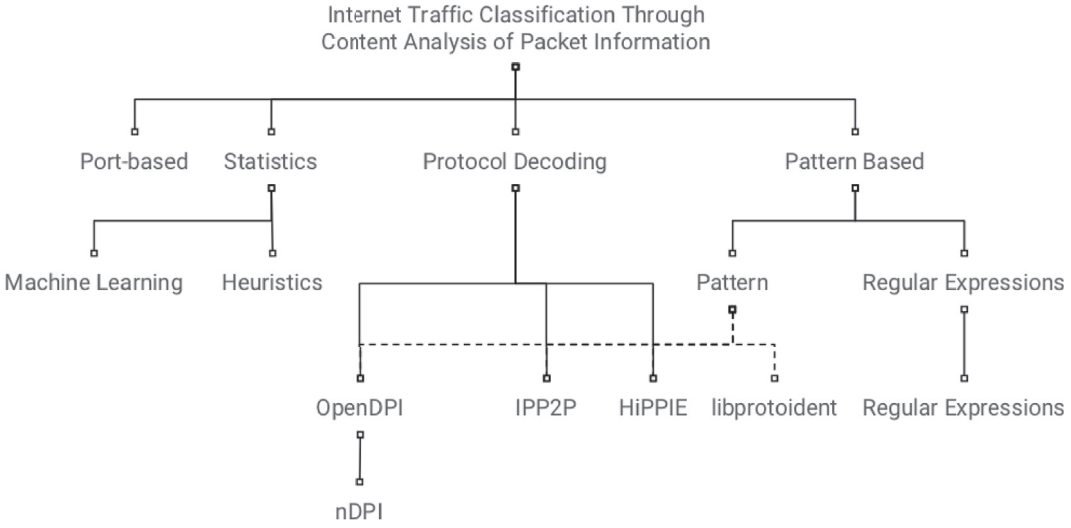

Fig. 5. DPI for Internet Classification of Finsterbusch's et al. (Finsterbusch et al., 2014).

schemes shall not be used in real-time applications due to their time consumption. On the other hand, interference-based methods deliver high efficiency and precision according to other authors; nevertheless, the survey does not indicate any scheme preference.

The surveys presented by Lin et al. (2006), (Lin et al., 2008) focus on string matching based classification on used data structures and on the implementation issues of string matching accordingly. The first survey classifies string matching into four categories based on the algorithm's data structure. Antonello et al. (Antonello et al., 2012a, b) presents a survey focused on efficient DPI Automaton-based systems for commodity platforms and classifies DPI into string matching, regular expressions, and finite automata categories. In addition, the authors explore the challenges for developing an efficient DPI system and provide some guidelines for the development of such system. AbuHmed et al. (2008) explores the implementation techniques and challenges of DPI for NIDS while Rathod et al. (2014) focuses on finite automata algorithms for pattern matching.

Xu et al. (2016) present a survey on Finite State Automata based Deep Packet Inspection focused on providing solutions to state explosion problems. Finite State Automata use Regular Expressions, which in comparison with exact string matching; these have the capability to represent a range of exact strings by using metacharacters. Due to the flexibility provided by Regular Expression Matching, it has been utilized for protocol identification and network intrusion detection systems; furthermore, it can be utilized for finding patterns representing application-level protocols, viruses, spam, and malware. Regular expressions can also be used to represent regular languages such that a regular expression can be used to determine whether a particular string belongs to a specific regular language.

Finite state automata (also known as well as finite state machine) is a mathematical model that it is used to represent a regular language; therefore, for every finite state automata, there exists a regular expression capable of representing the same regular language. In contrast, for every regular expression there exist a finite number of finite state automata capable of accepting the same regular expression. Two traditional finite state automata models are the nondeterministic finite automata (NFA) and the deterministic finite automata (DFA).

A DFA consist of a 5-tuple: a state, a set of input symbols, a transition function, start state, and a set of final states. For a given string consisting of multiple characters c1, c2,… cM, the initial state reads the first character c1 and the active state is defined by the transition function which takes the character and the previous active state (in this case the initial state) as an input. The process is repeated until all characters within the string are processed. Given that all the characters in combination with active states lead to sequence of states, given that the last active state is a member of the set of final states, we can determine that the DFA accepts the string; otherwise, the string will be rejected. Similarly, NFA consist on the same 5-tupple with the contrasting difference that NFA's transition function takes a state and a character as input but it returns a subset of the finite states leading to multiple active states in parallel. Despite of such differences, NFA and DFA are equivalent in regular language recognition.

The common steps to compile regular expressions are the following: regular expressions compiled to NFA, converting the compiled NFA to an equivalent DFA, and finally performing DFA minimization. McNaughton and Yamada (1960), Thompson (Thompson and Ken, 1968), and Hopcroft (Hopcroft et al., 2001) propose algorithms for generating NFA from regular expressions. Then we can build an NFA for each regular expression and combine the generated NFAs into an integrated NFA; finally, we proceed to covert the integrated NFA to an equivalent DFA by using construction algorithm as presented by Aho et al. (2007). A different approach for converting NFA to DFA is proposed by Becchi and Crowley in (Becchi and Crowley, 2008a). This approach optimizes the traditional NFA in order to produce smaller state size and active state set size.

Xu et al. (2016) identified the following goals DPI should be able to attain to support the increasing demand of link speed and pattern scale: Provide high processing rates and performance, scale to support a large amount of patterns, dynamically update, and provide additional functions such as enabling the user to customize a subset of patterns. Furthermore, the authors identify the gap between DFA memory requirements and the space of current fast memories as the major challenge for DPI using DFA. These gabs are based on the fact that researchers apply DFA for matching regular expressions in architectures with large memory; nevertheless, in practice DFA's organization is set into two-dimensional matrices, which are stored in fast memory modules. Another major problem is the continuously increasing size of DFA due to the rapid growth of application patterns, which incurs a greater need of memory. Solutions presenting compressed DFA structures are irregular, disregard memory bandwidth constraints, and require additional memory access to process a given symbol. Additionally, state explosion, which occurs when constructing a DFA with enough states capable of capturing all desired patterns, is another impeding factor for regular expression matching.

State explosion has been addressed with novel FSMs called scalable FAs, which mitigate state explosion by using novel compact structures. The downside is that they offer poor matching efficiency. Studies presented by Yu et al. (2006) and Becchi and Crowley (2007a) have shown that state explosion is generated due to the ambiguity of metacharacters in regular expressions since a large amount of states are required to capture all possible input sequences. Furthermore, Yu et al. (2006) summarizes some of the models that can lead to fast state inflation which are classified into inflation produced when compiling a single regular expression and inflation when compiling multiple regular expressions. On the other hand, studies presented by Yang and Prasanna (2011) and Liu et al. (2014) explore the state explosion from an NFA perspective. In their study, they observed that the NFA state inflation taking place during its conversion to DFA was caused when two NFA states are inclusive which will lead to new DFA states on conversion. This is caused since the DFA state represents a combination of NFA states.

Another important aspect of DFA implementation is the use of algorithms designed for state transition table compression, which is the standard representation for FSM. State Transition Tables are represented using two-dimensional matrices where the rows are the DFA states and the columns the input characters. These DFA based compression algorithms can be classified into thee main families as presented in Fig. 6.

Table 6presents an overview of each algorithm family along with some of the proposals focused on each of them.

In Table 7, we present a comparative summary of classical transition compression algorithms as shown by Xu et al. (2016)

Beyond traditional and compressed DFAs, a number of other studies have focused on supporting large scale pattern matching and simultaneously providing solutions for state explosion. Xu et al. (2016), for example, classified these FAs into the categories presented in Fig. 7.

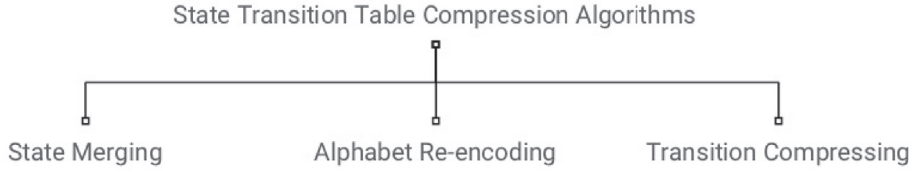

Fig. 6. DFA based compression algorithms.

Table 6
Proposals by DFA algorithm.

DFA Algorithm	Literature	Overview
State Merging	Becchi and Cadambi (2007)	Reduces the number of rows in transition table by merging states with same destination
Alphabet Re-encoding	(Becchi and Crowley, 2008b) (Brodie et al., 2006) (Kong et al., 2008) (Becchi and Crowley, 2013)	Reduces the number columns in the transition table by merging equivalent input characters and re-encoding the input alphabet
Transition Compressing	(Tuck et al., 2004) (Becchi and Cadambi, 2007) (Antonello et al., 2012a, b) (Kumar et al., 2006a) (Kumar et al., 2006b) (Becchi and Crowley, 2007b) (Ficara et al., 2008) (Ficara et al., 2011) (Becchi and Crowley, 2013) (Patel et al., 2014) (Liu and Torng, 2014) (Antonello et al., 2015) (Qi et al., 2011)	Focuses on reducing massive redundant transitions.

Table 7
Classical Transition Compression Algorithms, adapted from [30].

Algorithms	Compression Ratio	Time Complexity	Space Complexity	Memory Bandwidth
D^2FA	95%	$O(n^2 \log(n))$	$O(n^2)$	No guarantee
$A-DFA$	>90%	$O(n^2)$	$O(n)$	≤ 2
σFA	>90%	$O(n \times \lvert\Sigma\rvert^2)$	$O(n \times \lvert\Sigma\rvert)$	≈ 1

Table 8presents a performance overview for each one of the scalable FA families as well as a brief performance overview based on the presented research proposals.

4. Conclusion and future work

Based on the different technologies we surveyed in the presented paper, where we included a study on smart grids, SDN integrated in Smart Grids, and DPI, we recommend considering a more comprehensive approach. For example, we should consider other alternatives to a DPI-only based solution for securing smart grids due to the mutability of attacks and the resource intensive nature, in order to achieve enhanced performance and accuracy. One such alternative is a SDN-based solution, although the latter is not foolproof. There are a number of challenges associated with SDN-enabled smart grids, such as single-point-of-failure due to a centralized controller, compromised network switches, grid devices such as RTUs, relays, or SCADA slaves, and compromised SDN controllers or SDN controller applications, protecting hypervisors enabling virtual network slices or virtual SDNs.

One potential future research agenda is to explore a smart grid design that is controlled mainly by IIoT devices and other underpinning components, which will manage grid control applications. The IIoT devices will collect data and send control commands to other connected devices and systems using the grid communication network enabled by legacy and SDN networks using in-band control traffic. The control devices collecting information from sensors, such as PMUs, have the capability to issue commands to actuators. Using SDN technologies, the communication network used by IIoT devices such as control devices, sensors, and actuators can be then controlled. Furthermore, SDN applications can be utilized to reconfigure the network in real-time to manage threats or improve QoS.

4.1. A conceptual SDN-based security monitoring framework

SDN technologies can be utilized for controlling the communication going from the control center to the power grid sensors using the capabilities of network nodes and communication protocols such as OpenFMB. Common SDN architectures communicate SDN controller nodes to SDN Network nodes using management communication lines. The Network Nodes can be utilized to control the communication flow of the OpenFMB bus protocol, which is used by Substation Nodes as well as Grid Nodes. The latter have direct communication with a variety of sensors, which include smart meters, line sensors, capacity banks, and solar PV inverters. Fig. 8 presents the presented SDN schema.

Additionally, the control center can run behavioral analysis and deep learning models to monitor communications transmitted between the control center and the communication network. If a traffic anomaly is detected, this can be notified to the SDN controller and trigger a reconfiguration to protect the network and prevent the spread of an attack. In addition, all data analyzed by behavioral analysis modules and deep learning models can be utilized for reinforcement learning (by properly corroborating an attack and its family using DPI), providing updated models with additional information making the smart grid an evolving and resilient entity against attacks. Fig. 9 presents the conceptual framework.

The proposed conceptual framework seeks to enable smart grid devices across the network, provide dynamic and real-time control over the network using SDN technology, while simultaneously detecting and protecting the network against a variety of cyber attacks using behavioral

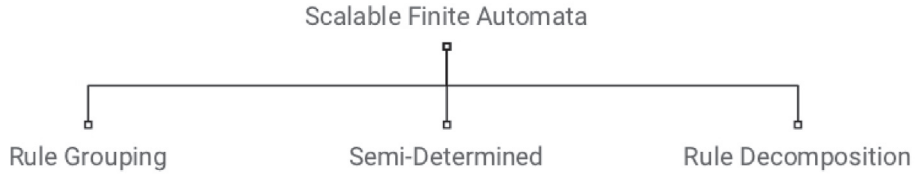

Fig. 7. Scalable Finite Automata Families, adapted from [30].

Table 8
FA proposals by family and performance overview.

Scalable FAs	Research	Performance Overview
Rule Grouping	(Van Lunteren, 2006) (Yu et al., 2006) (Liu et al., 2014) (Rohrer et al., 2009)	Requires a large construction time, high to moderate storage space, provides moderate matching speed and limited scaling capabilities
Semi-Determined	(Becchi and Crowley, 2007a) (Yang and Prasanna, 2011) (Kumar et al., 2007) (Smith et al., 2008b)	Requires short to medium construction time, low to moderate storage space, provides a moderate matching speed and flexible scalability
Rule Decomposition	(Kumar et al., 2007) (Smith et al., 2008b) (Smith et al., 2008a) (Wang et al., 2013) (Yu et al., 2014) (Becchi and Crowley, 2008c) (Wang et al., 2014)	Requires short construction time, small storage space, provides low matching speed due to variable calculation and large instruction fetching, and is capable of delivering flexible scalability

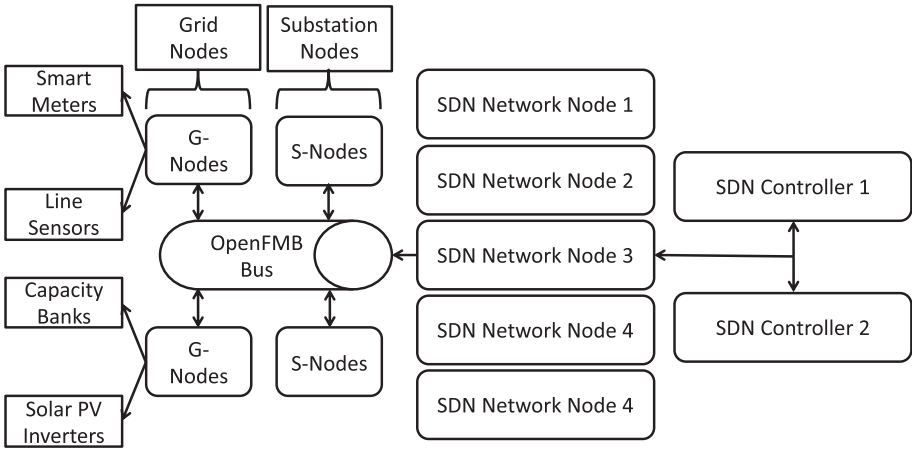

Fig. 8. SDN and OpenFMB based control.

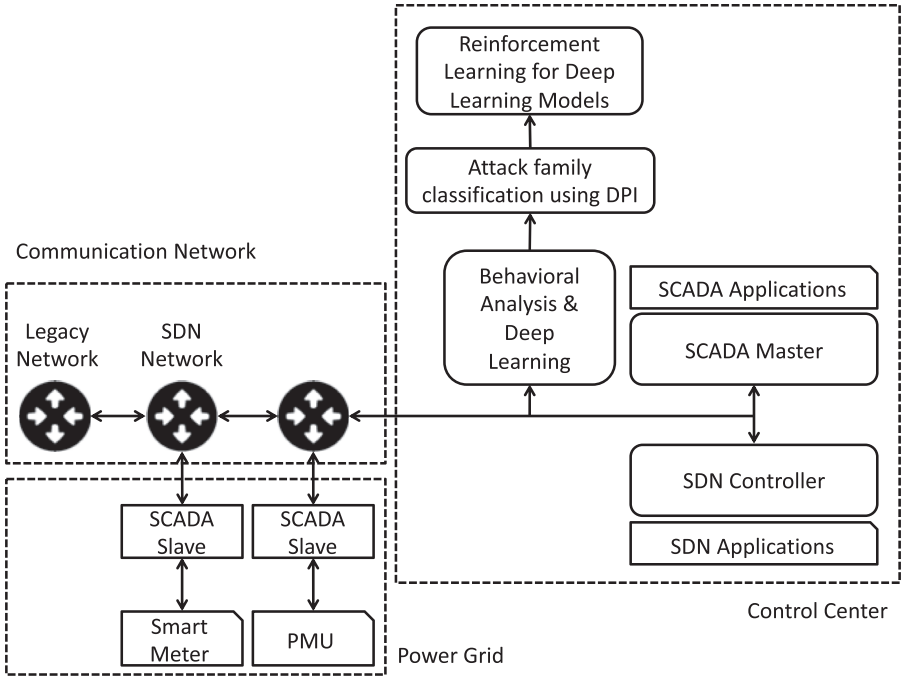

Fig. 9. Conceptual SDN-based security monitoring framework.

analysis and deep learning models. Such models need to be tailored to the needs of a particular entity and while a simulated environment can provide us with interesting results, these will not compare to the ones produced by real data extracted from production devices, sensors, and networks. The acquisition of such data is imperative for the development of deep learning models capable of assessing and performing anomaly detection over network traffic, analysis on SDN controller requests, and attack classification when an anomaly is detected.

4.2. A conceptual forensic-driven security monitoring framework

In addition to security monitoring, it would be prudent to also incorporate incident response/handling and forensic investigation capabilities in the event that a security incident occurs or is detected. This will allow smart grid operators to identify and isolate the root cause(s) of a security incident (e.g. compromise or data breach) and how such incidents could be prevented in the future. While residual data from systems and IIoT devices is crucial for forensic analysis, but such data may not always be available. For example, due to a lack of extraction capabilities, short data retention times or data has been overwritten, and more practically the costs associated in acquiring such data. This, therefore, necessitates the design and implementation of forensic-ready systems and infrastructure, for example in the form of a digital forensic black-box (see component 3 in Fig. 10) (Ab Rahman et al., 2017).

For example, based on the forensic-by-design concept, forensic and investigation-related requirements can be integrated into relevant phases of the smart grid development lifecycle and/or retrofitted into existing smart grid systems, with the objective of developing forensic-ready systems. It is also important that protection (e.g. the conceptual SDN-based security monitoring framework in Fig. 9) and digital forensics and investigation capabilities are integrated in a holistic manner, rather than designed and implemented in isolation. For example, data associated with protection can inform detection and investigation activities, and vice versa.

Fig. 10 depicts the proposed forensic-driven security monitoring framework. In such an approach, the secure multi-tenant storage

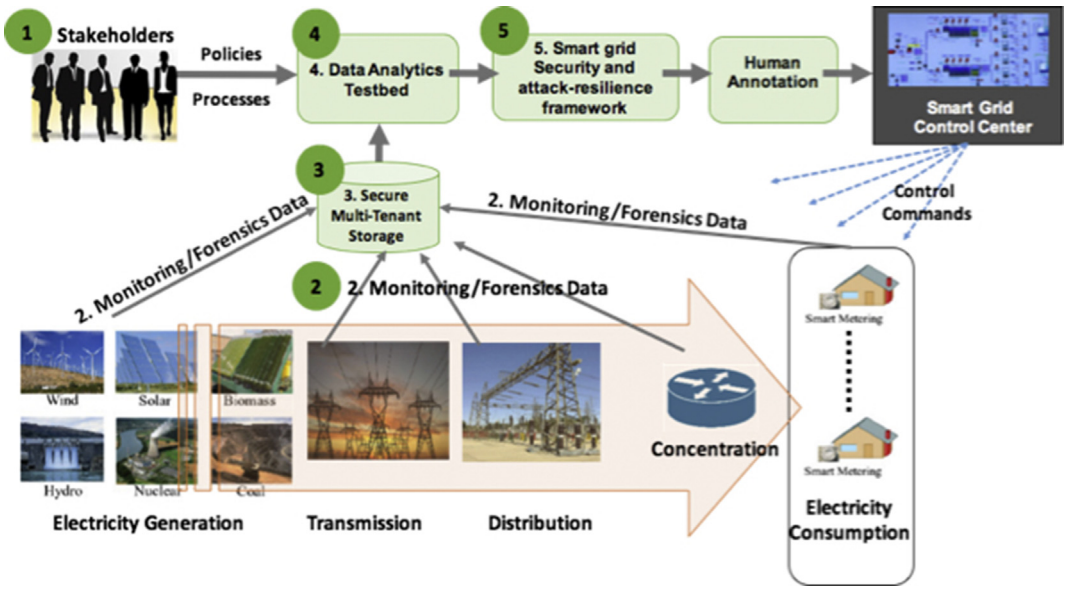

Fig. 10. Conceptual forensic-driven security monitoring framework.

(component 3) can act as the digital forensic black-box (analogous to a flight recorder on commercial flights). Once the multi-tenant storage/digital forensic black-box environment has been setup and secured appropriately, artefacts of forensic interest/relevance can then be collected and stored, to facilitate future investigations.

Acknowledgments

This project and the preparation of this publication were funded in part by monies provided by CPS Energy through an agreement with The University of Texas at San Antonio. The last author is also supported by the Cloud Technology Endowed Professorship.

References

Ab Rahman, N.H., Cahyani, N.D.W., Choo, K.-K.R., 2017. Cloud incident handling and forensic-by-design: cloud storage as a case study. Concurrency Comput. Pract. Ex. 29, e3868. https://doi.org/10.1002/cpe.3868.

AbuHmed, T., Mohaisen, A., Nyang, D., 2008. A survey on deep packet inspection for intrusion detection systems. Inf. Secur. 24, 10. https://doi.org/10.1109/ICNISC.2015.17.

Aho, A.V., Corasick, M.J., 1975. Efficient string matching: an aid to bibliographic search. Commun. ACM 18, 333–340. https://doi.org/10.1145/360825.360855.

Aho, A.V., Lam, M.S., Sethi, R., Ullman, J.D., 2007. Compilers: Principles, Techniques and Tools (Reading MA).

Alcock, S., Lorier, P., Nelson, R., 2012. Libtrace. ACM SIGCOMM comput. Commun. Rev. 42, 42. https://doi.org/10.1145/2185376.2185382.

Antonello, R., Fernandes, S., Kamienski, C., Sadok, D., Kelner, J., Gódor, I., Szabó, G., Westholm, T., 2012. Deep packet inspection tools and techniques in commodity platforms: challenges and trends. J. Netw. Comput. Appl. 35, 1863–1878. https://doi.org/10.1016/J.JNCA.2012.07.010.

Antonello, R., Fernandes, S., Sadok, D., Kelner, J., Szabo, G., 2012. Deterministic Finite Automaton for scalable traffic identification: the power of compressing by range. In: 2012 IEEE Network Operations and Management Symposium. IEEE, pp. 155–162. https://doi.org/10.1109/NOMS.2012.6211894.

Antonello, R., Fernandes, S., Sadok, D., Kelner, J., Szabó, G., 2015. Design and optimizations for efficient regular expression matching in DPI systems. Comput. Commun. 61, 103–120. https://doi.org/10.1016/J.COMCOM.2014.12.011.

Asghari, H., van Eeten, M., Bauer, J., Mueller, M., 2013. Deep Packet Inspection: Effects of Regulation on its Deployment by Internet Providers. https://doi.org/10.2139/SSRN.2242463.

Azmoodeh, A., Dehghantanha, A., Conti, M., Choo, K.-K.R., 2018. Detecting crypto-ransomware in IoT networks based on energy consumption footprint. J. Ambient Intell. Humaniz. Comput. 9 (4), 1141–1152. https://doi.org/10.1007/s12652-017-0558-5.

Becchi, M., Cadambi, S., 2007. Memory-efficient regular expression search using state merging. In: IEEE INFOCOM 2007 - 26th IEEE International Conference on Computer Communications. IEEE, pp. 1064–1072. https://doi.org/10.1109/INFCOM.2007.128.

Becchi, M., Crowley, P., 2007a. A hybrid finite automaton for practical deep packet inspection. In: Proceedings of the 2007 ACM CoNEXT Conference on - CoNEXT '07. ACM Press, New York, New York, USA, p. 1. https://doi.org/10.1145/1364654.1364656.

Becchi, M., Crowley, P., 2007b. An improved algorithm to accelerate regular expression evaluation. In: Proceedings of the 3rd ACM/IEEE Symposium on Architecture for Networking and Communications Systems - ANCS '07. ACM Press, Orlando, Florida, United States, p. 145. https://doi.org/10.1145/1323548.1323573.

Becchi, M., Crowley, P., 2008a. Efficient regular expression evaluation. In: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems - ANCS '08. ACM Press, New York, New York, USA, p. 50. https://doi.org/10.1145/1477942.1477950.

Becchi, M., Crowley, P., 2008b. Efficient regular expression evaluation: theory to practice. In: Proceedings of the 4th ACM/IEEE Symposium on Architectures for Networking and Communications Systems - ANCS '08. ACM Press, San Jose, California, USA, p. 50. https://doi.org/10.1145/1477942.1477950.

Becchi, M., Crowley, P., 2008c. Extending finite automata to efficiently match Perl-compatible regular expressions. In: Proceedings of the 2008 ACM CoNEXT Conference on - CONEXT '08. ACM Press, Madrid, Spain, pp. 1–12. https://doi.org/10.1145/1544012.1544037.

Becchi, M., Crowley, P., 2013. A-dfa: a time- and space-efficient DFA compression algorithm for fast regular expression evaluation. ACM Trans. Archit. Code Optim. 10, 1–26. https://doi.org/10.1145/2445572.2445576.

Bendrath, R., 2009. Global technology trends and national regulation: explaining variation in the governance of deep packet inspection. Int. Stud. Annu. Conv. New York City 15–18. Febr. 2009 32.

Bendrath, R., Mueller, M., 2011. The end of the net as we know it? Deep packet inspection and internet governance. New Media Soc. 13, 1142–1160. https://doi.org/10.1177/1461444811398031.

Berry, B., 2011. SCADA Tutorial: A Fast Introduction to SCADA Fundamentals and Implementation [WWW Document]. DPS Telecom. URL. http://www.dpstele.com/scada/what-is.php (accessed 1.17.18).

Bonelli, N., Di Pietro, A., Giordano, S., Procissi, G., 2012. On Multi–Gigabit Packet Capturing with Multi–Core Commodity Hardware. Springer, Berlin, Heidelberg, pp. 64–73. https://doi.org/10.1007/978-3-642-28537-0_7.

Boyer, R.S., Moore, J.S., 1977. A fast string searching algorithm. Commun. ACM 20, 762–772. https://doi.org/10.1145/359842.359859.

Broad, William J., Markoff, J., Sanger, D.E., 2011. Stuxnet Worm Used against Iran Was Tested in Israel. New York Times.

Brodie, B.C., Taylor, D.E., Cytron, R.K., Brodie, B.C., Taylor, D.E., Cytron, R.K., 2006. A scalable architecture for high-throughput regular-expression pattern matching. ACM SIGARCH Comput. Archit. News 34, 191–202. https://doi.org/10.1145/1150019.1136500.

C37.118.1-2011 IEEE Standard for Synchrophasor Measurements for Power Systems., (n.d.).

CAIDA Data [WWW Document], n.d. URL http://www.caida.org/data/(accessed 1.15.18).

Callado, A., Kamienski, C., Szabo, G., Gero, B., Kelner, J., Fernandes, S., Sadok, D., 2009. A survey on internet traffic identification. IEEE Commun. Surv. Tutorials 11, 37–52. https://doi.org/10.1109/SURV.2009.090304.

Chen, S., Lu, R., Shen, X.S., 2014. SRC: a multicore NPU-based TCP stream reassembly card for deep packet inspection. Secur. Commun. Network. 7, 265–278. https://doi.org/10.1002/sec.727.

Cheng, J., n.d. Ad-Injecting Trojan Targets Mac Users on Safari Firefox and Chrome.

Choo, K.-K.R., Dehghantanha, A., 2017. Contemporary digital forensics investigations of cloud and mobile applications. Contemp. Digit. Forensic Investig. Cloud Mob. Appl. 1–6. https://doi.org/10.1016/B978-0-12-805303-4.00001-0.

Choo, K.-K.R., Herman, M., Iorga, M., Martini, B., 2016. Cloud forensics: state-of-the-art and future directions. Digit. Invest. 2016, 77–78. https://doi.org/10.1016/j.diin.2016.08.003.

Chung, J.Y., Park, B., Won, Y.J., Strassner, J., Hong, J.W., 2009. Traffic Classification Based on Flow Similarity, pp. 65–77. https://doi.org/10.1007/978-3-642-04968-2_6.

Cisco, 2017. Cisco Visual Networking Index: Global Mobile Data Traffic Forecast , 2016 – 2021. CISCO, pp. 1–7. White Pap. Feb. 2017. https://doi.org/10.1109/SURV.2008.080403.

Cisco Systems, 2018. Cisco 2018 Annual Cybersecurity Report, p. 675. Cisco [Online]. https://doi.org/10.1002/ejoc.201200111.

Critical Infrastructure Sectors, 2019. Homel. Secur [WWW Document]. URL. https://www.dhs.gov/critical-infrastructure-sectors (accessed 1.17.19).

Darwish, I., Igbe, O., Celebi, O., Saadawi, T., Soryal, J., 2015. Smart grid DNP3 vulnerability analysis and experimentation. In: International Conference on Cyber Security and Cloud Computing. IEEE, New York, United States, pp. 141–147.

Data Loss Prevention, 2003.

Deri, L., 2004. Improving passive packet capture: beyond device polling. In: Proc. SANE, vol. 2004, p. 85.

Dharmapurikar, S., Paxson, V., 2005. Robust TCP stream reassembly in the presence of adversaries. In: USENIX Security Symposium. Baltimore, MD, United States, pp. 65–80.

Dong, X., Lin, H., Tan, R., Iyer, R.K., Kalbarczyk, Z., 2015. Software-defined networking for smart grid resilience: opportunities and challenges. In: Proceedings of the 1st ACM Workshop on Cyber-Physical System Security - CPSS '15. ACM Press, New York, New York, USA, pp. 61–68. https://doi.org/10.1145/2732198.2732203.

DPDK: Data Plane Development Kit [WWW Document], n.d. URL http://dpdk.org/ (accessed 1.16.18).

Dusi, M., Gringoli, F., Salgarelli, L., 2011. Quantifying the accuracy of the ground truth associated with Internet traffic traces. Comput. Network. 55, 1158–1167. https://doi.org/10.1016/J.COMNET.2010.11.006.

Egorov, S., Savchuk, G., 2002. SNORTRAN: an optimizing compiler for snort rules. Fidel. Secur. Syst.

Ficara, D., Giordano, S., Procissi, G., Vitucci, F., Antichi, G., Di Pietro, A., 2008. An improved DFA for fast regular expression matching. ACM SIGCOMM Comput. Commun. Rev. 38, 29. https://doi.org/10.1145/1452335.1452339.

Ficara, D., Di Pietro, A., Giordano, S., Procissi, G., Vitucci, F., Antichi, G., 2011. Differential encoding of DFAs for fast regular expression matching. IEEE/ACM Trans. Netw. 19, 683–694. https://doi.org/10.1109/TNET.2010.2089639.

Finsterbusch, M., Richter, C., Rocha, E., Muller, J.-A., Hanssgen, K., 2014. A survey of payload-based traffic classification approaches. IEEE Commun. Surv. Tutorials 16, 1135–1156. https://doi.org/10.1109/SURV.2013.100613.00161.

Galil, Z., 1978. On Improving the Worst Case Running Time of the Boyer-Moore String Matching Algorithm. Springer, Berlin, Heidelberg, pp. 241–250. https://doi.org/10.1007/3-540-08860-1_18.

Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A., 2014. Flow monitoring explained: from packet capture to data analysis with NetFlow and IPFIX. IEEE Commun. Surv. Tutorials 16, 2037–2064. https://doi.org/10.1109/COMST.2014.2321898.

Holloway, M., 2015. Stuxnet Worm Attack on Iranian Nuclear Facilities. Introd. to Nucl. Energy, Stanford Univ, pp. 14–15.

Hooper, C., Martini, B., Choo, K.-K.R., 2013. Cloud computing and its implications for cybercrime investigations in Australia. Comput. Law Secur. Rep. 29, 152–163. https://doi.org/10.1016/J.CLSR.2013.01.006.

Hopcroft, J.E., Motwani, R., Ullman, J.D., 2001. In: Introduction to Automata Theory, Languages, and Computation, second ed., vol. 32. ACM SIGACT News, p. 60 https://doi.org/10.1145/568438.568455.

Horspool, R.N., 1980. Practical fast searching in strings. Software Pract. Ex. 10, 501–506. https://doi.org/10.1002/spe.4380100608.

IEEE 1588-2008 - IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems [WWW Document], n.d. URL https://standards.ieee.org/findstds/standard/1588-2008.html (accessed 1.17.18).

IEEE 1815-2012 - IEEE Standard for Electric Power Systems Communications-Distributed Network Protocol (DNP3) [WWW Document], n.d. URL https://standards.ieee.org/findstds/standard/1815-2012.html (accessed 1.17.18).

IEEE 2030-2011 - IEEE Guide for Smart Grid Interoperability of Energy Technology and Information Technology Operation with the Electric Power System (EPS), End-Use Applications, and Loads [WWW Document], n.d. URL https://standards.ieee.org/findstds/standard/2030-2011.html (accessed 1.17.18).

Ikezoye, V.E., Schrempp, J.B., 2000. Method and Apparatus for Identifying Media Content Presented on a Media Playing Device.

Institute of Electrical and Electronics Engineers, IEEE-SA Standards Board, 2011a. IEEE Guide for Design, Operation, and Integration of Distributed Resource Island Systems with Electric Power Systems. Institute of Electrical and Electronics Engineers.

Institute of Electrical and Electronics Engineers, IEEE-SA Standards Board, 2011b. IEEE Recommended Practice for Interconnecting Distributed Resources with Electric Power Systems Distribution Secondary Networks. Institute of Electrical and Electronics Engineers.

ISO/IEC 15067-3:2012 - Information technology – Home Electronic System (HES) application model – Part 3: Model of a demand-response energy management system for HES [WWW Document], n.d. URL https://www.iso.org/standard/55596.html (accessed 1.17.18).

Kabalci, Y., 2016. A survey on smart metering and smart grid communication. Renew. Sustain. Energy Rev. 57, 302–318. https://doi.org/10.1016/J.RSER.2015.12.114.

Karp, R.M., Rabin, M.O., 1987. Efficient randomized pattern-matching algorithms. IBM J. Res. Dev. 31, 249–260. https://doi.org/10.1147/rd.312.0249.

Kim, N., Choi, G., Choi, J., 2014. A scalable carrier-grade DPI system Architecture using synchronization of flow information. IEEE J. Sel. Area. Commun. 32, 1834–1848. https://doi.org/10.1109/JSAC.2014.2358836.

Kong, S., Smith, R., Estan, C., 2008. Efficient signature matching with multiple alphabet compression tables. In: Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks - SecureComm '08. ACM Press, Istanbul, Turkey, p. 1. https://doi.org/10.1145/1460877.1460879.

Kumar, S., Dharmapurikar, S., Yu, F., Crowley, P., Turner, J., Kumar, S., Dharmapurikar, S., Yu, F., Crowley, P., Turner, J., 2006a. Algorithms to accelerate multiple regular expressions matching for deep packet inspection. In: ACM SIGCOMM Computer Communication Review. ACM, Pisa, Italy, p. 339. https://doi.org/10.1145/1151659.1159952.

Kumar, S., Turner, J., Williams, J., 2006b. Advanced algorithms for fast and scalable deep packet inspection. In: Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems - ANCS '06. ACM Press, San Jose, California, United States, p. 81. https://doi.org/10.1145/1185347.1185359.

Kumar, S., Chandrasekaran, B., Turner, J., Varghese, G., 2007. Curing regular expressions matching algorithms from insomnia, amnesia, and acalculia. In: Proceedings of the 3rd ACM/IEEE Symposium on Architecture for Networking and Communications Systems - ANCS '07. ACM Press, Orlando, Florida, United States, p. 155. https://doi.org/10.1145/1323548.1323574.

Lin, P., Li, Z., Lin, Y., Lai, Y., Lin, F., 2006. Profiling and accelerating string matching algorithms in three network content security applications. IEEE Commun. Surv. Tutorials 8, 24–37. https://doi.org/10.1109/COMST.2006.315851.

Lin, P.-C., Lin, Y.-D., Lai, Y.-C., Lee, T.-H., 2008. Using string matching for deep packet inspection. Computer 41, 23–28. https://doi.org/10.1109/MC.2008.138.

Liu, A.X., Torng, E., 2014. An overlay automata approach to regular expression matching. In: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications. IEEE, Toronto, ON, Canada, pp. 952–960. https://doi.org/10.1109/INFOCOM.2014.6848024.

Liu, R.-T., Huang, N.-F., Chen, C.-H., Kao, C.-N., 2004. A fast string-matching algorithm for network processor-based intrusion detection system. ACM Trans. Embed. Comput. Syst. 3, 614–633. https://doi.org/10.1145/1015047.1015055.

Liu, T., Liu, A.X., Shi, J., Sun, Y., Guo, L., 2014. Towards fast and optimal grouping of regular expressions via DFA size estimation. IEEE J. Sel. Area. Commun. 32, 1797–1809. https://doi.org/10.1109/JSAC.2014.2358839.

Madhukar, A., Williamson, C., 2006. A longitudinal study of P2P traffic classification. In: 14th IEEE International Symposium on Modeling, Analysis, and Simulation. IEEE, pp. 179–188. https://doi.org/10.1109/MASCOTS.2006.6.

McNaughton, R., Yamada, H., 1960. Regular expressions and state graphs for automata. IEEE Trans. Electron. Comput. EC 9, 39–47. https://doi.org/10.1109/TEC.1960.5221603.

Moore, A.W., Papagiannaki, K., 2005. Toward the Accurate Identification of Network Applications, pp. 41–54. https://doi.org/10.1007/978-3-540-31966-5_4.

Mueller, M., 2011. DPI technology from the standpoint of Internet governance studies: an introduction. Sch. Inf. Stud. Syracuse Univ 1–11.

Mueller, M., Kuehn, A., Santoso, S.M., 2012. Policing the network: using DPI for copyright enforcement. Surveill. Soc. 9, 348–364.

Muth, R., Manber, U., 1996. Approximate Multiple String Search. Springer, Berlin, Heidelberg, pp. 75–86. https://doi.org/10.1007/3-540-61258-0_7.

Navarro, G., Raffinot, M., 2002. Flexible Pattern Matching in Strings: Practical Online Search Algorithms for Texts and Biological Sequences. Computer (Long. Beach. Calif). https://doi.org/10.1109/MC.2002.1033033.

Nguyen, T.T.T., Armitage, G., 2008. A survey of techniques for internet traffic classification using machine learning. IEEE Commun. Surv. Tutorials 10, 56–76. https://doi.org/10.1109/SURV.2008.080406.

NIST Cybersecurity Framework Implementation Case Study, 2017. SGIP's cybersecurity comm. Framew. Implement.

Norton, M., 2002. Optimizing Pattern Matching for Intrusion Detection. System, pp. 1–11.

Others, I.S.A., 2010. IEEE 1901-2010 - IEEE Standard for Broadband over Power Line Networks: Medium Access Control and Physical Layer Specifications. IEEE Std 1901 1–191586.

Pandalabs, 2018. Pandalabs Annual Report 2018 1–32.

Patel, J., Liu, A.X., Torng, E., 2014. Bypassing space explosion in high-speed regular expression matching. IEEE/ACM Trans. Netw. 22, 1701–1714. https://doi.org/10.1109/TNET.2014.2309014.

Peng, J., Choo, K.-K.R., Ashman, H., 2016. User profiling in intrusion detection: a review. J. Netw. Comput. Appl. 72, 14–27. https://doi.org/10.1016/J.JNCA.2016.06.012.

Perez, P., Nta, P., Ramirez, D., 2007. Advanced DPI: intelligent security and insight add up to opportunity. Alcatel-Lucent 1–16.

Pf_Ring:High Speed Packet Capture Filtering and Analysis [WWW Document], n.d. URL https://www.ntop.org/products/packet-capture/pf_ring/(accessed 1.16.18).

Piskac, P., Novotny, J., 2011. Using of time characteristics in data flow for traffic classification. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 173–176. https://doi.org/10.1007/978-3-642-21484-4_21.

Prokhorenko, V., Choo, K.-K.R., Ashman, H., 2016. Web application protection techniques: a taxonomy. J. Netw. Comput. Appl. 60, 95–112. https://doi.org/10.1016/J.JNCA.2015.11.017.

Qi, Y., Xu, L., Yang, B., Xue, Y., Li, J., 2009. Packet classification algorithms: from theory to practice. In: IEEE INFOCOM 2009 - the 28th Conference on Computer Communications. IEEE, pp. 648–656. https://doi.org/10.1109/INFCOM.2009.5061972.

Qi, Y., Wang, K., Fong, J., Xue, Y., Li, J., Jiang, W., Prasanna, V., 2011. FEACAN: front-end acceleration for content-aware network processing. In: 2011 Proceedings IEEE INFOCOM. IEEE, Shanghai, China, pp. 2114–2122. https://doi.org/10.1109/INFCOM.2011.5935021.

Quick, D., Martini, B., Choo, K.-K.R., 2014. Cloud Storage Forensics.

Rathod, P.M., Marathe, N., Vidhate, A.V., 2014. A survey on Finite Automata based pattern matching techniques for network Intrusion Detection System (NIDS). In: Advances in Electronics, Computers and Communications. IEEE, Bangalore, India, pp. 1–5.

reenwald, G., MacAskill, E., 2013. NSA Prism program taps into user data of Apple, Google and others. Guardian 7, 1–43.

Risso, F., Baldi, M., Morandi, O., Baldini, A., Monclus, P., 2008. Lightweight, payload-based traffic classification: an experimental evaluation. In: 2008 IEEE International Conference on Communications. IEEE, pp. 5869–5875. https://doi.org/10.1109/ICC.2008.1097.

Rizzo, L., 2012. NetMap: A Novel Framework for Fast Packet I/O. Atc ' 12 101–112. https://doi.org/10.1145/2043164.2018500.

Rizzo, L., Deri, L., Cardigliano, A., 2012. 10 Gbit/s Line Rate Packet Processing Using Commodity Hardware: Survey and New Proposals.

Rocha, E., Salvador, P., Nogueira, A., 2011. Detection of illicit network activities based on multivariate Gaussian fitting of multi-scale traffic characteristics. In: 2011 IEEE International Conference on Communications (ICC). IEEE, pp. 1–6. https://doi.org/10.1109/icc.2011.5962651.

Rohrer, J., Atasu, K., van Lunteren, J., Hagleitner, C., 2009. Memory-efficient distribution of regular expressions for fast deep packet inspection. In: Proceedings of the 7th IEEE/ACM International Conference on Hardware/Software Codesign and System Synthesis - CODES+ISSS '09. ACM Press, Grenoble, France, p. 147. https://doi.org/10.1145/1629435.1629456.

Schmelzer, R.A., Pellom, B.L., 2002. Copyright Detection and Protection System and Method.

Schneider, F., Wallerich, J., Feldmann, a, 2007. Packet capture in 10-gigabit ethernet environments using contemporary commodity hardware. Passiv. Act. Netw. 207–217. https://doi.org/10.1007/978-3-540-71617-4_21.

Service Name and Transport Protocol Port Number Registry [WWW Document], n.d. URL https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml (accessed 1.16.18).

Smart Grid System Security with Broadcast Communications, 2017. SGIP's Home/Building/Industry-to-Grid Domain Expert Work (Gr).

Smith, R., Barry, R., 2019. America's electric grid has a Vulnerable Back Door—and Russia Walked through it. Wall St. J. https://www.wsj.com/articles/americas-electric-grid-has-a-vulnerable-back-doorand-russia-walked-through-it-11547137112.

Smith, R., Estan, C., Jha, S., 2008a. XFA: faster signature matching with extended automata. In: 2008 IEEE Symposium on Security and Privacy (Sp 2008). IEEE, Oakland, California, United States, pp. 187–201. https://doi.org/10.1109/SP.2008.14.

Smith, R., Estan, C., Jha, S., Kong, S., Smith, R., Estan, C., Jha, S., Kong, S., 2008b. Deflating the big bang fast and scalable deep packet inspection with extended finite automata. In: Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication - SIGCOMM '08. ACM Press, Seattle, Washington, United States, p. 207. https://doi.org/10.1145/1402958.1402983.

Stevens, J., 2014. Electricity Subsector Cybersecurity Capability Maturity Model (ES-C2M2)(Case Study).

Stevenson, C., Chouinard, G., Zhongding, Lei, Wendong, Hu, Shellhammer, S., Caldwell, W., 2009. IEEE 802.22: the first cognitive radio wireless regional area network standard. IEEE Commun. Mag. 47, 130–138. https://doi.org/10.1109/MCOM.2009.4752688.

Symantec, 2018. ISTR Internet Security Threat Report, vol. 23. https://doi.org/10.1007/s10207-014-0262-9.

Tan, L., Sherwood, T., Tan, L., Sherwood, T., 2005. A high throughput string matching architecture for intrusion detection and prevention. ACM SIGARCH Comput. Archit. News 33, 112–122. https://doi.org/10.1145/1080695.1069981.

Teing, Y.-Y., Dehghantanha, A., Choo, K.-K.R., 2017. CloudMe forensics: a case of big data forensic investigation. Concurrency Comput. Pract. Ex. e4277. https://doi.org/10.1002/cpe.4277.

The libpcap project [WWW Document], n.d. URL https://sourceforge.net/projects/libpcap/(accessed 1.16.18).

The Ministry of Energy and Coal intends to form a group of representatives of all energy companies within the management of the Ministry to study the possibilities of preventing unauthorized interference in the operation of power grids, 2016. Minist. Energy Coal Ind. Ukr [WWW Document]. URL. http://mpe.kmu.gov.ua/minugol/control/uk/publish/article?art_id=245086886&cat_id=35109 (accessed 1.26.18).

Thomas, K., Bursztein, E., Grier, C., Ho, G., Jagpal, N., Kapravelos, A., Mccoy, D., Nappa, A., Paxson, V., Pearce, P., Provos, N., Rajab, M.A., 2015. Ad injection at scale: assessing deceptive advertisement modifications. In: 2015 IEEE Symposium on Security and Privacy. IEEE, pp. 151–167. https://doi.org/10.1109/SP.2015.17.

Thompson, K., Ken, 1968. Programming Techniques: regular expression search algorithm. Commun. ACM 11, 419–422. https://doi.org/10.1145/363347.363387.

T.I.A, 2008. TIA-1113: medium-speed (up to 14 Mbps) power line communications (PLC) modems using windowed OFDM. Telecommun. Ind. Assoc. 1–182.

Trabelsi, Z., Zeidan, S., Masud, M.M., 2016. Network packet filtering and deep packet inspection hybrid mechanism for IDS early packet matching. In: 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA). IEEE, pp. 808–815. https://doi.org/10.1109/AINA.2016.178.

Tuck, N., Sherwood, T., Calder, B., Varghese, G., 2004. Deterministic memory-efficient string matching algorithms for intrusion detection. In: IEEE INFOCOM. IEEE, pp. 2628–2639 n.d. https://doi.org/10.1109/INFCOM.2004.1354682.

Ucar, S., Ergen, S.C., Ozkasap, O., 2016. Security vulnerabilities of IEEE 802.11p and visible light communication based platoon. In: 2016 IEEE Vehicular Networking Conference (VNC). IEEE, Columbus, OH, USA. https://doi.org/10.1109/VNC.2016.7835972.

User-space E1000 driver library [WWW Document], n.d. URL https://sourceforge.net/projects/libe1000/(accessed 1.16.18).

Van Lunteren, J., 2006. High-performance pattern-matching for intrusion detection. In: Proceedings IEEE INFOCOM 2006. 25TH IEEE International Conference on Computer Communications. IEEE, Barcelona, Spain, pp. 1–13. https://doi.org/10.1109/INFOCOM.2006.204.

Wagener, G., Dulaunoy, A., Engel, T., 2008. Towards an estimation of the accuracy of TCP reassembly in network forensics. In: Future Generation Communication and Networking. IEEE, Hainan Island, China, pp. 273–278.

Wang, J., Wacks, K., 2017. Smart Grid System Security with Broadcast Communications. SGIP, pp. 1–17.

Wang, K., Li, J., Wang, K., Li, J., 2013. Towards fast regular expression matching in practice. In: Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM - SIGCOMM '13. ACM Press, Hong Kong, China, p. 531. https://doi.org/10.1145/2486001.2491705.

Wang, K., Fu, Z., Hu, X., Li, J., 2014. Practical regular expression matching free of scalability and performance barriers. Comput. Commun. 54, 97–119. https://doi.org/10.1016/J.COMCOM.2014.08.005.

Wu, S., Manber, U., 1994. A fast algorithm for multi-pattern searching. Science 80, 1–11. https://doi.org/10.1145/177424.177579.

Xu, C., Chen, S., Su, J., Yiu, S.M., Hui, L.C.K., 2016. A survey on regular expression matching for deep packet inspection: applications, algorithms, and hardware platforms. IEEE Commun. Surv. Tutorials 18, 2991–3029. https://doi.org/10.1109/COMST.2016.2566669.

Yang, Y.-H.E., Prasanna, V.K., 2011. Space-time tradeoff in regular expression matching with semi-deterministic finite automata. In: 2011 Proceedings IEEE INFOCOM. IEEE, Shanghai, China, pp. 1853–1861. https://doi.org/10.1109/INFCOM.2011.5934986.

Yang, Z., Chen, Y.X., Li, Y.F., Zio, E., Kang, R., 2014. Smart electricity meter reliability prediction based on accelerated degradation testing and modeling. Int. J. Electr. Power Energy Syst. 56, 209–219. https://doi.org/10.1016/J.IJEPES.2013.11.023.

Yao, A.C.-C., 1979. The complexity of pattern matching for a random string. SIAM J. Comput. 8, 368–387. https://doi.org/10.1137/0208029.

Yu, F., Chen, Z., Diao, Y., Lakshman, T.V., Katz, R.H., 2006. Fast and memory-efficient regular expression matching for deep packet inspection. In: Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems - ANCS '06. ACM Press, New York, New York, USA, p. 93. https://doi.org/10.1145/1185347.1185360.

Yu, X., Lin, B., Becchi, M., 2014. Revisiting state blow-up: automatically building augmented-FA while preserving functional equivalence. IEEE J. Sel. Area. Commun. 32, 1822–1833. https://doi.org/10.1109/JSAC.2014.2358840.

Zeceña, J.C.C., Molina, V.L.O., 2017. Hydra-A DNP3 multiplexing platform for SCADA system switchover. In: International Conference on Electronics, Electrical Engineering and Computing. IEEE, Cuzco, Peru, pp. 1–4.

Zhang, M., Ju, J., 2003. Space-economical reassembly for intrusion detection system. In: International Conference on Information and Communications Security. Springer, Huhehaote, China, pp. 393–404.

Gonzalo De La Torre Parra received the B.S. in Electrical Engineering from Texas A&M University-Kingsville, USA, in 2009 and the M.S. in Electrical Engineering (Telecommunications Concentration) from The University of Texas at San Antonio, USA, in 2015. He is currently pursuing the Ph.D. in Electrical Engineering at The University of Texas at San Antonio. His current research is focused on the development of Deep Learning Algorithms on Network Security. From 2015 to 2017, he has been a developer of Chameleon Cloud, an NSF sponsored cloud infrastructure, has held the Co-Chair position of OpenStack's Cloud Application Hack Work Group, and manages research projects focused on cloud, networks, and security at the Open Cloud Institute. Mr. De La Torre Parra's awards and honors include the 2nd Place Award on SHPE's Extreme Engineering National STEM Competition (2013), the 2nd Place Award on HENAAC's XIV National STEM Competition (2013), the San Antonio Mexican Foundation for Education (SAMFE) award (2013), CONACYT's Master's Degree Fellowship (2013), NSF-Open Cloud Institute Fellowship (2016), and CONACYT's Ph.D. Degree Fellowship (2017).

Paul Rad is cofounder and assistant director of Open Cloud Institute (OCI), and Associate Professor at The University of Texas at San Antonio, USA. His research interests include artificial intelligence and machine learning, cyber analytics, and cloud computing with applications to cyber-physical systems and IoT, machine sensing, and decentralized decision making and trust. He received Ph.D. degree in Electrical and Computer Engineering on Cyber Analytics from the University of Texas at San Antonio. He holds fourteen US patents on Cyber Infrastructure, Cloud Computing, and Big Data Analytics. Rad has advised over 200 + companies on cloud computing and data analytics with over 50 keynote presentations.

Kim-Kwang Raymond Choo received the Ph.D. in Information Security in 2006 from Queensland University of Technology, Australia. He currently holds the Cloud Technology Endowed Professorship at The University of Texas at San Antonio (UTSA). In 2016, he was

named the Cybersecurity Educator of the Year - APAC (Cybersecurity Excellence Awards are produced in cooperation with the Information Security Community on LinkedIn), and in 2015 he and his team won the Digital Forensics Research Challenge organized by Germany's University of Erlangen-Nuremberg. He is the recipient of the 2018 UTSA College of Business Col. Jean Piccione and Lt. Col. Philip Piccione Endowed Research Award for Tenured Faculty, Outstanding Associate Editor of 2018 for IEEE Access, IEEE TrustCom 2018 Best Paper Award, ESORICS 2015 Best Research Paper Award, 2014 Highly Commended Award by the Australia New Zealand Policing Advisory Agency, Fulbright Scholarship in 2009, 2008 Australia Day Achievement Medallion, and British Computer Society's Wilkes Award in 2008. He is also a Fellow of the Australian Computer Society, an IEEE Senior Member, and Co-Chair of IEEE Multimedia Communications Technical Committee (MMTC)'s Digital Rights Management for Multimedia Interest Group.