

## Article

# Toward an Applied Cyber Security Solution in IoT-Based Smart Grids: An Intrusion Detection System Approach

Xiao Chun Yin <sup>1</sup>, Zeng Guang Liu <sup>2,\*</sup>, Lewis Nkenyereye <sup>3</sup> and Bruce Ndibanje <sup>4</sup>

<sup>1</sup> Facility Horticulture Laboratory of Universities in Shandong, WeiFang University of Science & Technology, Shouguang 262700, China; xiaochunyin@wfust.edu.cn

<sup>2</sup> College of Computer Science and Engineering, ShanDong University of Science and Technology, Qingdao 266590, China

<sup>3</sup> Department of Computer and Information Security, Sejong University, Seoul 05006, Korea; nkenyele@sejong.ac.kr

<sup>4</sup> Research and Development Center, Cyber Threat Intelligence Lab, YangJae Innovation Hub, 114 Taebong-Ro, Seocho-Gu, Seoul 06754-601, Korea; ndibabruce@gmail.com

\* Correspondence: st.lzg@163.com

Received: 8 October 2019; Accepted: 7 November 2019; Published: 14 November 2019



**Abstract:** We present an innovative approach for a Cybersecurity Solution based on the Intrusion Detection System to detect malicious activity targeting the Distributed Network Protocol (DNP3) layers in the Supervisory Control and Data Acquisition (SCADA) systems. As Information and Communication Technology is connected to the grid, it is subjected to both physical and cyber-attacks because of the interaction between industrial control systems and the outside Internet environment using IoT technology. Often, cyber-attacks lead to multiple risks that affect infrastructure and business continuity; furthermore, in some cases, human beings are also affected. Because of the traditional peculiarities of process systems, such as insecure real-time protocols, end-to-end general-purpose ICT security mechanisms are not able to fully secure communication in SCADA systems. In this paper, we present a novel method based on the DNP3 vulnerability assessment and attack model in different layers, with feature selection using Machine Learning from parsed DNP3 protocol with additional data including malware samples. Moreover, we developed a cyber-attack algorithm that included a classification and visualization process. Finally, the results of the experimental implementation show that our proposed Cybersecurity Solution based on IDS was able to detect attacks in real time in an IoT-based Smart Grid communication environment.

**Keywords:** sensor networks; smart grid; IoT; wireless network security; DNP3 Protocol; cybersecurity

## 1. Introduction

To provide new services and offer new features with excellent quality, modern critical infrastructure such as power plants, smart grids, and water plants nowadays use ICT technologies [1]. However, even if ICT technologies have made possible the provision of new services and new features, the types of connectivity have opened the door to a new wave of possible threats to critical installations. Extensive research has been performed in which the details of the vulnerability and security that affect SCADA systems have been analyzed [2–5].

In Smart Grids, protocols and architectures are designed for very particular functions in SCADA communication systems. The SCADA acronym stands for Supervisor Control and Data Acquisition, and it allows the supervision and control of plants either remotely or locally using hardware and software dedicated to that system. With this system, the system analyzes, collects and processes data in real time.

The DNP3 (Distributed Network Protocol) protocol is one of the most widely used network protocols in smart grid communication networks. This protocol is mostly used in academia and industry research projects, as it provides opportunity for customization as it is an open protocol. Based on this characteristics, any company can employ DNP3 developments that are compatible with their equipment. There are other protocols designed to control the operations of technical systems. In the case of the DNP3 protocol, it is based on the three layers model in the OSI 7 layer model as given in Figure 1.

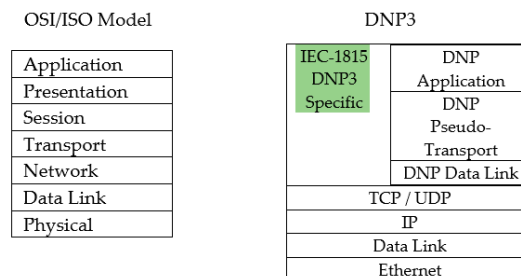


Figure 1. DNP3 protocol main concept with OSI/ISO mapping.

The DNP3 plays a vital role in the smart grid, and it is conventionally used in SCADA industrial processing, including both electricity and water distribution. For this reason, in this paper we focus on this protocol due to its importance in smart grids. The DNP3 protocol is based on a three-layered, enhanced performance architecture (EPA) reference model. The EPA defines the basic application functionality for the user layer, which is located between the OSI App Layer and App Program [6,7]. According to Figure 1, the DNP3 protocol mainly consists of three layers: the application layer, the data-link layer, and the pseudo-transport layer. The DNP3 protocol facilitates reliable communication between the SCADA nodes; for example, the last layer is in charge of the transmission of enormous quantities of data [8,9]. The proposed research emphasizes the DNP3 protocol by parsing its structure during the session, and we will develop an algorithm to analyze the vulnerabilities that will include modeling attacks on all layers, and we propose an intrusion detection system to detect those attacks.

Normally, a SCADA system is made up of three main layers or components such as Information Technology Network (IT Network), Operation Technology Network (OT Network) and Process Control System Network (or Field Layer), as given in Figure 2.

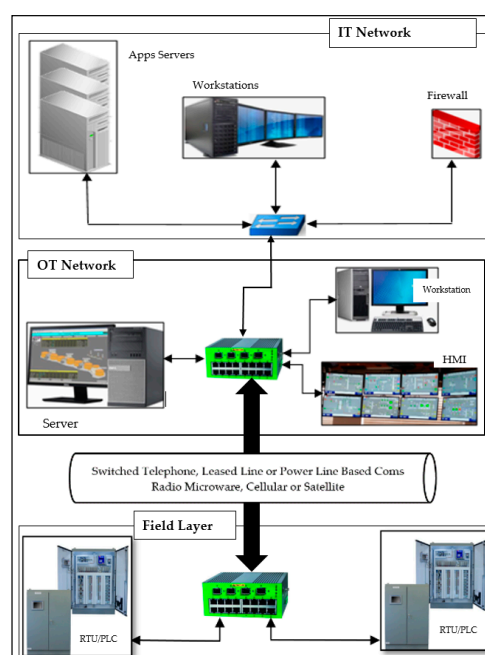


Figure 2. Basic SCADA concept components or layers.

In recent years, the OT Networks were operated as separate networks or stand-alone system without being connected to public communication and infrastructures. Nevertheless, as the Internet nowadays provides data accessibility and connected services, the businesses in Smart Grid have turned to exploit those services. This situation creates a complex architecture where non-secure systems are added to existing systems without strong security. Ultimately, they are both more exposed to attackers, as the separation that had previously protected these systems is decreased [10].

Unfortunately, with the presence of internet connections inside smart grids, the security risk is very high. After what happened in Ukraine, everyone knows that hackers can bring down the energy grid. Attacks often happen by using steps such as reconnaissance, which consists of gathering information about the targeted system; scanning, which is about finding any weakness or vulnerability in the system by looking for any open ports; and running a service through the port. Thirdly, the attacker exploits the system using the discovered vulnerability and then compromises it so that they can gain full control. Finally, the attacker tries to maintain access by which they will steal the data or damage the system whenever they want [11,12]. These types of attacks are not easy to detect using traditional antivirus software, which detects malware using pattern matching or heuristic methods for obfuscated malware. Cyber-attacks have evolved in business-driven situations, because the criminal actors behind these attacks know where the system weaknesses reside, and they employ appropriate malware, especially ransomware.

As previous experience shows, whenever there is a hack on a smart grid, people's life is in danger, and sometimes loss of life occurs.

To tackle malware threats on the SCADA system, especially on the smart grid, this article presents an intrusion detection system that isn't based on signatures, like traditional antivirus software. The proposed solution is mainly based on the DNP3 protocol, where the engine parses the packet format and then we train it to learn whether the sample from the frame protocol has been compromised or is good. The advantage of this protocol is that it is used in oil/gas and water utilities, as well as in wastewater. Furthermore, it is broadly utilized in electric facilities such as catapults [13]. The research contribution in this article includes the main steps as follows:

- An overall customized DNP3 protocol vulnerability exposure with reference to the original protocol.
- A new attack model using the DNP3 protocol that targets all layers; the attack follows three steps: (a) pre-attack step; (b) attack modeling defined on all layers; and (c) attack settings within DNP3 parameters and consequences.
- An algorithm that includes machine learning methods for data transformation and data process concept on SCADA/DNP3 protocol.
- A new cyber-attack algorithm targeting the SCADA/DNP3 system, and the visualization and classification process for an intrusion detection system (IDS).

In this paper, Section 2 presents a description of existing solutions in the area, while Section 3 presents the proposed solution along with the methods and data collection in an experimental environment. The results of experimental testing and the discussion are provided in Section 4, before concluding the work in Section 5.

## 2. Related Work

This section describes the works—including various solutions applied in SCADA/ICS/DNP3—in which IDS and other methods have been discussed. The Distributed Network Protocol (DNP3) is prevalent within critical infrastructure, especially in smart grids. Unfortunately, DNP3 has some vulnerabilities that have been exploited by hackers, and so SCADA systems would face serious problems [14]. However, before exploring solutions in SCADA systems, there is a great deal of research and many excellent results in IoT networks that are very promising. Yang et al. [15] proposed a security scheme in IoT-based healthcare systems. In this research, they proposed a self-adaptive access

control together with a privacy-preserving smart IoT-based healthcare big data storage system. Further security approaches have been developed for IoT systems, as described in [16–18].

With regard to machine learning methods, de Toledo et al. [19] developed a method that encrypts the traffic using the DNP3 protocol. This study used supervised algorithms to classify messages from the same protocol using datasets from the medium voltage of substations using simulation methods. The generated traffic followed two-direction communication using an encryption mode based on the IPsec and ESP (transport mode) with the exclusion UDP mode. This experiment is the most widely used, as it provides a way of privately limiting the cost of IP bandwidth within networks per byte sent [20]. Other techniques that have been proposed as solutions for protecting DNP3 traffic include statistical pattern recognition, classification-based real-time method with HTTP, FTP and SSH flow, TCP and TLS protocols [21,22].

Widely known layered security methods that provide protection in SCADA networks have been developed, but these methods have numerous limitations with respect to their dependency on the protocols. Among others, protocols such as SSH, SSL, IPsec, and TLS offer end-to-end security solutions, in addition to crypto-protocol encryption systems [23,24]. Further research oriented towards the security of the application layer [25] focusing on data integrity and authentication procedures was developed with the aim of providing solutions for known attacks such as modification, spoofing, and flooding [26]. Nevertheless, a certain number of limitations was revealed resulting from mechanisms defined in the DNP3 protocol—in particular, embedded security mechanisms [27]. A solution based on crypto-algorithms that includes known encryption methods like AES and RSA was developed to protect DNP3 protocol at the application layer [28]. In this research, the authors contributed three primary enhancements, including a new security scheme that was implemented together with the DNP3 protocol, a method for constructing the bytes within every layer, and use of the TCP/IP protocol for data exchange.

On the other hand, IDS based on different machine learning methods has been developed, whereby attacks can be detected based on highly accurate results of detected attacks. However, more improvements are necessary due to false alarms or false positive from the detection systems. This problem usually leads to the misclassification between good and bad data in the network [29]. In the same category of research, a group of five machine learning algorithms was tested for cybersecurity solutions to protect SCADA systems [30–33]. After the training process, the models were implemented in a real network environment to capture and analyze online data from network traffic. Both results from the testbed and live traffic revealed that the IDS based on machine learning algorithms was efficient for detecting attacks. Further research developed by Keliris et al. [34] showed that the Support Vector Machine (SVM) algorithm performs well for anomaly detection and classification. They used a supervised learning method to develop a process-aware defense tactic in the ICS accounting for behavior-based attacks. The work done in [35] suggests that a detection system using machine learning techniques in power systems would be feasible for detecting malicious states. Tomin et al. [35] claimed that such techniques, where applied in SCADA/ICS, offer a range of solutions with a satisfactory level of security. In the course of their research, they used an offline training process using a cross-validation method and they applied it to a semi-automated method for online testing purposes. Further research has been developed to provide security for Smart Grid DNP3, through the identification of malicious activities in ICS of IoT based on Deep Learning, IDS for SCADA systems, and Neural Network-based IDS for critical infrastructure. These have shown tremendous results in the development of models for the detection of attacks on power systems [36–41].

### 3. Proposed Solution: Method and Implementation Experiments

This section describes the proposed scheme for the SCADA/DNP3 protocol. The solution requires several steps, referred to as “modules”, and each of these plays a specific role in building a holistic cyber-security solution in an IoT-based Smart Grid environment.

### 3.1. System Model and Description

The proposed solution is based on the following modules: (a) data input system, (b) data analysis system, and (c) classification and detection system, as shown in Figure 3. However, before we could arrive at this holistic solution, we performed additional research on the DNP3 protocol. Firstly, we developed an attack model for each layer of the DNP3 protocol, as shown in Figure 4. These attacks had two main functions: (1) to collect data for the purpose of building a database to be used in the training and testing model, (2) to assess the vulnerabilities of the DNP3 protocol [42] that attackers are able to leverage in order to carry out cyber-attacks on IoT-based Smart Grids. Secondly, we developed an algorithm for analyzing a modified DNP3 protocol [43,44]. This algorithm uses the original DNP3 protocol as a reference for the purpose of comparison with the common vulnerabilities of the protocol stack.

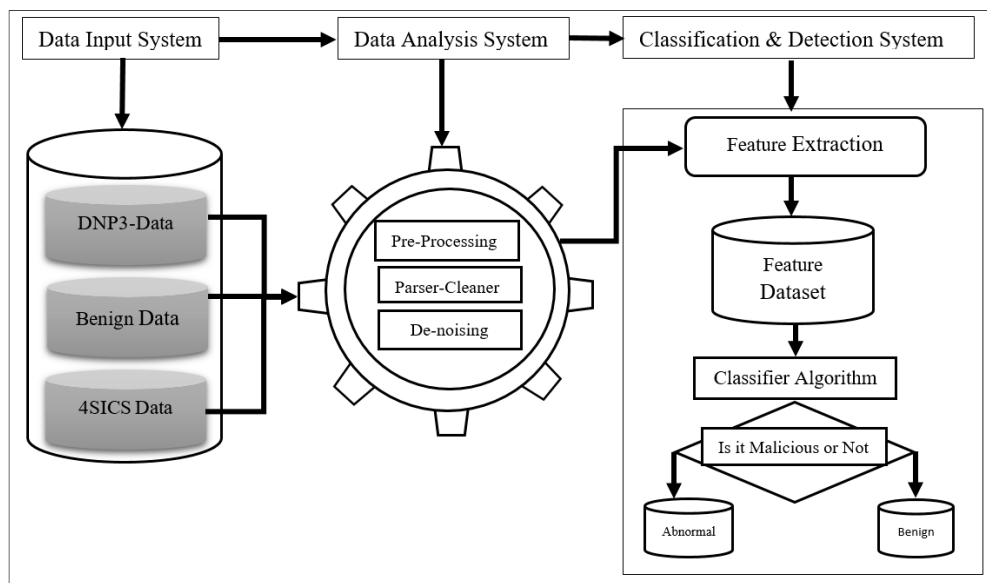


Figure 3. System model for the proposed solution.

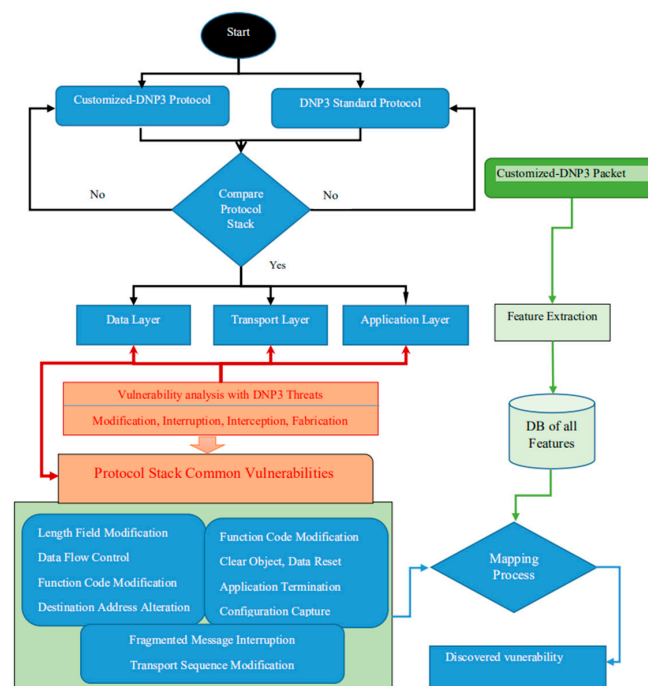


Figure 4. DNP3 overall vulnerability algorithm analysis.

We used four types of attack—modification, interception, interruption, and fabrication—targeting all layers in order to evaluate them. The collected vulnerabilities (based on the attacks on the two protocols) were used with a mapping function to modify the features of the DNP3 protocol. The results provide the vulnerabilities discovered for the customized protocol, as shown in Figure 5.

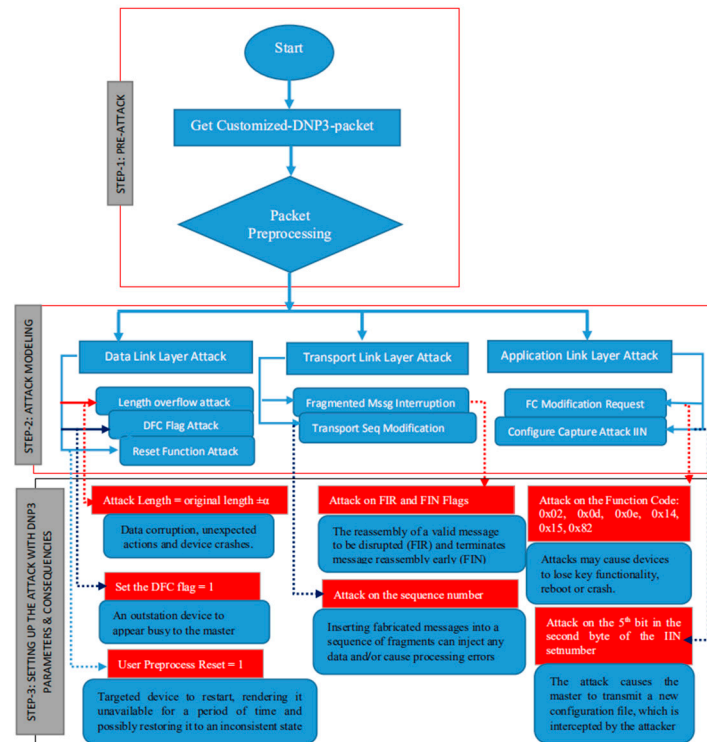


Figure 5. Attack modelling for the DNP3 protocol.

### 3.2. Data Input System: Data Generation

The dataset used in the experiment is from a variety of different sources, but the most important data, which is related to DNP3 packet parsing, was based on the assessment of vulnerabilities and attacks performed on the protocol. Therefore, we will only describe the data obtained from these experiments, as this is the focus of our research. The rest of the data was obtained from an open-source dataset used at the 4SICS industrial cybersecurity conference [3], which is an annual summit. The summit calls on experts in ICS/SCADA/DNP3 cybersecurity from the most critical infrastructures like smart grids, transportation, and so on. As far as our purposes are concerned, we only collected data—PCAP Files—related to the smart grid from the ICS Lab using RTUs, PLCs, and other industrial networks. Additionally, we included some known malware targeting ICS/SCADA systems. As described above, to generate the data, several steps are required, which can be summarized as two main steps: vulnerability assessment and attack modeling on the DNP3 protocol.

#### 3.2.1. DNP3 Protocol Vulnerability Assessment

Different methods have been proposed in order to analyze the weaknesses in the DNP3 protocol; one of these methods presents an assessment of specific attacks on function code within certain layers of the protocol stack [45–47]. In this paper, we used a customized DNP3 protocol to perform the vulnerability analysis, and this is compared with the original one, as shown in Figure 4. The novelty of our method is that we map common vulnerabilities onto the customized database features, with the results showing us the weakness of the protocol, meaning that we can ultimately launch different attacks in order to collect or generate the data to use in our experiments.



The proposed algorithm takes the two protocols as input and checks whether they satisfy the protocol stack requirements. If they do, they are parsed into the main layers; otherwise, they go back to the starting point. This process continues, using DNP3 threats such as modification, interruption, interception, and fabrication, where we define the common vulnerabilities of the protocol stack to be used for the mapping process. The mapping process is carried out based on a database of features from the DNP3 packet. Table 1 shows selected features from the layers of the DNP3 protocol, and a full account of the features is provided in Appendix A. A “Yes” in the column “Subject to Attack” means that they present a potential weakness that makes them vulnerable to DNP3 protocol threats.

**Table 1.** DNP3 partial features dataset.

Features	Subject to Attack
DNP3_START	
DNP3_LENGTH	Yes
DNP3_SOURCE	
DNP3_DESTINATION	
DNP3_CONTROL_DFC	Yes
DNP3_CONTROL_DIR	
DNP3_CONTROL_FCB	Yes
DNP3_CONTROL_FCV	Yes
DNP3_CONTROL_FUNC_CODE_PRI	Yes
DNP3_CONTROL_FUNC_CODE_SEC	
DNP3_CONTROL_PRM	
DNP3_CONTROL_reserved	
DNP3_CRC	
DNP3_Transport_FIN	Yes
DNP3_Transport_FIR	Yes
DNP3_Transport_SEQUENCE	Yes
DNP3_Application_request_Application_control_CON	
DNP3_Application_request_Application_control_UNSCON	
DNP3_Application_request_FUNC_CODE	Yes
DNP3_Application_response_Application_control_CON	
DNP3_Application_response_IIN_CLASS_3_EVENTS	
DNP3_Application_response_IIN_CONFIG_CORRUPT	Yes
DNP3_Application_response_IIN_DEVICE_RESTART	

### 3.2.2. Attack Modeling on the DNP3 Protocol

To launch attacks on the DNP3 protocol, we made an attack model that was specific to the vulnerabilities discovered. As shown in Figure 5, the model is based on three main steps:

- ✓ Step 1: Pre-attack. This is where the preliminary is carried out, including obtaining the DNP3 packets from the repository, and preprocessing the packet in order to obtain three layers for the next step.
- ✓ Step 2: Attack Modelling. In this step, we define the attacks on the basis of the vulnerabilities discovered in each of the following layers: Data Link Layer, Transportation Layer, and Application Link Layer. For the first layer, we defined three attacks (Length Overflow Attack, DFC Flag Attack and Reset Function Attack), for the second layer, we defined two attacks (Fragmented Message Interruption and Transport Sequence Modification), and in the last layer, we defined two attacks (FC Modification Request and Configuration Capture Attack IIN).
- ✓ Step 3: Setting up the attack with DNP3 parameters and Consequences. This step defines the parameters to be used during the attack (payload) and describes the consequences of each attack.

As given in the description of the consequences, each attack leads to bad behavior in the smart grid network. The aim is not to have these attacks, but rather to develop countermeasures in order to protect the network, devices, data, and human beings. Both the vulnerabilities and the attacks have several operational impacts that could cause damage to the system or take over the control system [48–52].

The data input system consists of malware and benign data, as already described in the introductory paragraph of this subsection. Table 2 gives a summary of the dataset used in this paper, where the name column describes the name of the malware or benign data, Qt is the amount of each type and the percentage of the distribution over the total. The overall distribution of malware is 55%, and that of benign data is 45%, which is acceptable for a classification and detection model. Bencsath et al. [53] described the most dangerous malware targeting industrial infrastructure in detail. Stuxnet was discovered in 2010, when it was reported to have destroyed numerous centrifuges in Natanz. The centrifuges had been designed for a uranium enrichment facility in Iran. The infection vector of Stuxnet was the USB, from which the worm was installed on and spread among interconnected computers. It is therefore very important to produce a cyber-security solution based on the IDS and ML techniques in order to protect such critical infrastructures against malware.

**Table 2.** Dataset of malware and benign data in this paper.

Name	Malware	Qt.	%	Benign	Qt.	%
<b>Triton</b>	Yes	1650	15.46	No	0	0.00
<b>Industroyer</b>	Yes	1521	14.25	No	0	0.00
<b>BlackEnergy</b>	Yes	650	6.09	No	0	0.00
<b>Stuxnet</b>	Yes	1120	10.50	No	0	0.00
<b>Duqu</b>	Yes	944	8.85	No	0	0.00
<b>Flame</b>	Yes	1062	9.95	No	0	0.00
<b>Gauss</b>	Yes	1267	11.87	No	0	0.00
<b>DNP3-Data(Original)</b>	No	0	0.00	Yes	4593	53.58
<b>DNP3-Packet (Experiment)</b>	Yes	2456	23.02	No	0	0.00
<b>4SICS</b>	No	0	0.00	Yes	3980	46.42
Total		10,670	55		8573	45

### 3.3. Data Analysis System

This module is located in the middle of the other modules, as it takes the input from the various repositories and then transforms the data into a format compatible with the functions of the next module. The data analysis is built up over many steps and requires advanced knowledge of Data Science, with several tools to be used in such work. In this paper, we describe a few steps taken from Figure 6. The data analysis consists of eight steps, from raw data input to the visualization step.

- Step 1: This is the initial action, where module one feeds raw data to the second module. As described above, 55% of the dataset is made up of malware and 45% is made up of benign data.
- Steps 2 to 4: After getting the raw data, the engine proceeds to DNP3 protocol extraction with the integration of various fields with pre-processing actions such as contextualization and mapping in order to prepare for loading to the DB. Before that, the engine carries out the data cleaning, removing unwanted fields, carrying out de-noising, and nullifying some fields that match with the DB used in our experiment.
- Steps 5 to 8: This is where the engine utilizes the DB constructed in Steps 2 through 4. At this stage, the important features are extracted based on their presence in the DB (presence refers to how frequently this feature occurred throughout the whole DB). Because the DB is a mixture of many types of data, the classification process first requires that the data be transformed from categorical and numerical data to a binary data format. Once we have one type of data, it is possible to apply the ML algorithms directly (green arrows) and then execute the classification process.



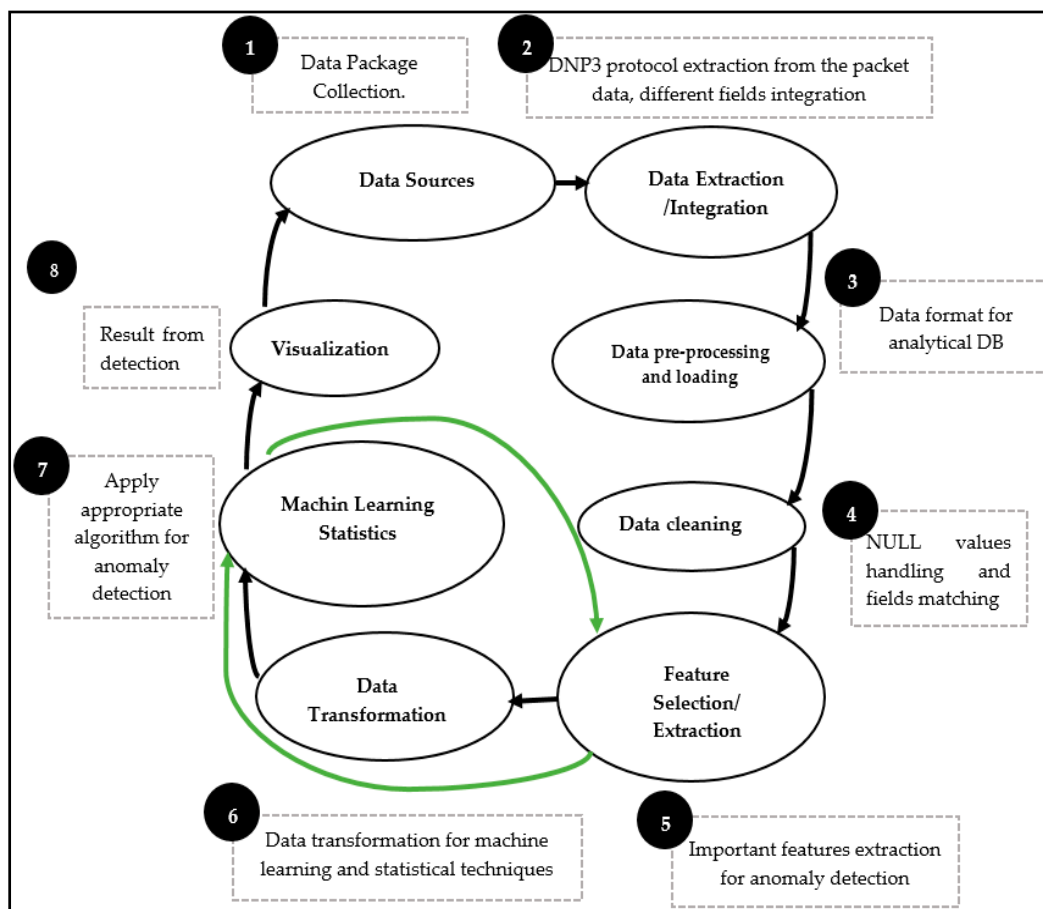


Figure 6. Data transformation processes.

Different algorithms were used for this process of data transformation in order to obtain the final DB and visualization results. Algorithm 1 is the pseudocode where all main steps are called to execute the ML algorithms. As given by the algorithm, we listed from Step 11 to 14 some of the algorithms used for the data transformation and visualization process. As input, we used a mixture of malware and benign data from module one, but for the paper objectives, we are going to focus only on the DNP3 packet analysis for more details. In the case of malware, we will describe Stuxnet and the features selected from the data analysis process.

Algorithm 1 has two main parts: the data input and processing part. The second part of the algorithm gives the main steps that implement the data transformation until the visualization step. The algorithm instructs to select all features from the raw data that include DNP3 protocol features, the 5 tuples (Source IP, Destination IP, Source Port, Destination Port, and Protocol), and eventually the features of the malware raw data. Next, it does a format check, which requires removing some unwanted characters that would cause errors in the database. In this case, null fields are not allowed, and categorical data and numerical data have to be mapped too. The cleaned data will then constitute the initial database, where we can make some queries to see the content.

The biggest part of the algorithm is where the call of each machine learning algorithm is running for different functions. Feature selection is the most important step in malware classification when using an ML algorithm. Bugra et al. [54] presented a method for malware classification where they applied DL (Deep Learning) methods. The authors performed the classification of malware based on a shallow deep learning network. To realize their experiment, they used a two-layer neural net to process the text, which consisted of turning text into a numerical form that is understandable by deep networks. This is called word2vec, developed by Tomas Mikolov [55,56] at Google and which is available from the Google code archive [57].

The work in [58] gives methods where ML has been used to classify malware and detection, in addition to implementations directions. The main goal of their work is to give a list of best classification methods such as feature selection, representation using Cuckoo Sandbox, k-Nearest-Neighbors (KNN), Decision Tree (DT), Support Vector Machines (SVM), Naive Bayes and Random Forest.

---

**Algorithm 1 Data Transformation & Visualization**


---

**Input:**  $RD = \{M \cup B\}$  //  $RD$  = Raw-Data,  $M$  = Malware dataset and  $B$  = Benign dataset  
**Output:**  $G$  and  $F$ -set //  $G$  = Graphs for visualization and  $F$  = Features  
**Process/Data Transformation:**

```

1: For all  $F \in RD$ ,
2:   Select  $F = \sum_{n=1}^F (RD)$  //  $F$  = DNP3 Features, 5-Tuple and Malware-Features
3:   for every each  $f$  category in  $F$ ,
4:      $f \in F$ , do:
5:       if any  $f$  does not satisfy DB format, do:
6:          $f = "-"$  // This is to remove unwanted format from features fields
7:       Otherwise,  $DB_{init} = \sum_{n=1}^F \{f_1 \cup f_2 \cup \dots \cup f_F\}$  // This is a first DB to be used
8:       End if
9:     End for
10:  For  $DB_{init}$ , call: // This is a function to call a set of ML algorithms
11:    Random Forest
12:    KNN
13:    Naïve Bayes
14:    Density Tree
15:    MVS
16:  End for
17:  Get :
18:    Important Features
19:    Final DB
20:    Visualization
21: End for

```

---

In this paper, we have used many algorithms, such as k-Nearest-Neighbors (KNN), Decision Tree (DT), Support Vector Machines (SVM), Naive Bayes and Random Forest. The results from our experiments and their descriptions are presented in Section 4.

### 3.4. Cyber-Attack Algorithm and IDS Solution

This subsection describes the cyber-attack algorithm that we created, in addition to the countermeasure (the IDS to detect the attack). This final step leads to the classification and detection processes from Figure 3. After the vulnerability assessment of the DNP3 protocol, the attack modeling, and data collection, we have now all we need to launch the attack and then perform the classification and detection solution. Algorithm 2 gives the steps to launch an attack on DNP3 protocol.

**Algorithm 2** Cyber-Attack on DL TL and AL

---

```

** START **
01: Input  $\leftarrow$  Raw data
02: Output  $\leftarrow$  Anomaly and Normal Traffic: {Classification and Detection}
** PRE-ATTACKS **
03: Procedure: INTERCEPTION (I)
04: Action: INJECTION (Inj) or MODIFICATION (Mod)
05: Packet  $\leftarrow$  {pre-process, get DNP3 packet ( $dnp_{pkt}$ )}
06: DNP3 protocol  $\leftarrow$  {DataLink (DL), TransportLink (TL), ApplicationLink (AL)}
07: Attack = {LOVA, DFC, FCA, FMI1, FMI2, TSM, FCM, CC_IIN}
*** SETTING-UP PARAMETERS & ATTACK LAUNCHING ***
08: LOVA  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(DL_{length} \leftarrow DL_{length \pm \alpha})\}$ 
09: DFC Flag  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(DL_{DFC=0} \leftarrow DL_{DFC=1})\}$ 
10: FCA  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(DL_{FC} \leftarrow DL_{FC=1})\}$ 
11: FMI1  $\Leftrightarrow I_{(dnp_{pkt})} \{Inj(TL_{FMI(FIN)} \leftarrow TL_{FMI(FIN=1)})\}$ 
12: FMI2  $\Leftrightarrow I_{(dnp_{pkt})} \{Inj(TL_{FMI(FIR)} \leftarrow TL_{FMI(FIR=0)})\}$ 
13: TSM  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(TL_{TSM(Seq.Number)} \leftarrow TL_{TSM(Seq.Number \pm \beta)})\}$ 
14: FCM1  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x02})\}$ 
15: FCM2  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x0d})\}$ 
16: FCM3  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x0e})\}$ 
17: FCM4  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x14})\}$ 
18: FCM5  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x15})\}$ 
19: FCM6  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{FC} \leftarrow AL_{FCM.req=0x82})\}$ 
20: CC_IIN  $\Leftrightarrow I_{(dnp_{pkt})} \{Mod(AL_{CC_IIN} \leftarrow AL_{CC_IIN=1 \rightarrow 5^{th} bit in 2^{nd} byte})\}$ 
21: If  $I_{(dnp_{pkt})}$  then, //Interception of the packet
22: Attacker  $I_{(dnp_{pkt})} \{(Mod \parallel Inj)_{DL \rightarrow TL \rightarrow AL}\}$  //Launch the attack on the layers
23: Get anomaly traffic
24: End if
25: End

```

---

All of these attacks on DNP3 protocols are assumed to occur during data transmission from one station to another. In practical cases, the system uses Master (client) and Slave (server) terminology. In this case, the server is defined as a station or device that holds and processes the information needed by an operator. To the other side, a client is a substation or device that requests information from the server. The DNP3 protocol provides the ability to facilitate data transmission between Master and Slave [59]. Figure 7 shows a schematic attack in which the attacker performs interception, modification and injection attacks on the DNP3 packet content as described in Algorithm 2.

The abovementioned algorithm was built up over three stages: start, pre-attack and setting parameters up & attack launching. The last stage includes seven major attacks that target all three layers on specific fields. The second stage instructs the algorithm to use the interception method that executes the injection and modification attacks in addition to the three layers and corresponding attacks. The first stage is related to data input and output data format information. The abbreviations used in the algorithm are described in Table 3.

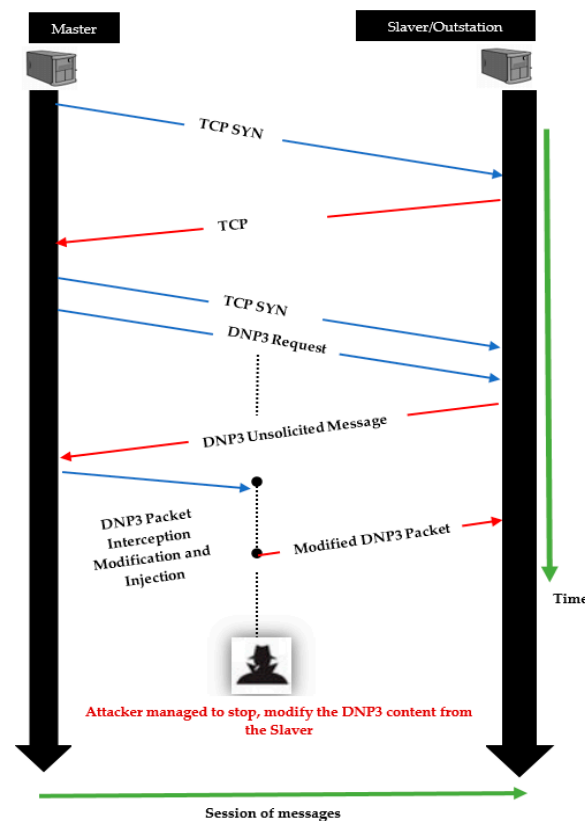


Figure 7. DNP3 cyber-attack types: interception, modification, and injection.

Table 3. Notations and description used in Algorithm 2.

Abbreviations	Description
DLL	Data Link Layer
TLL	Transport Link Layer
ALL	Application Link Layer
LOVA	Length Overflow Attack
DFCA	Data Flow Control Attack
FCA	Function Control Attack
FMIA	Fragmented Message Interruption Attack
T SMA	Transport Sequence Modification Attack
FCM	Function Code Modification
CCA_IIN	Configuration Capture Attack_Internal INdication

The main part (stage number 3) of the algorithm describes the attacks that target the layers as follows:

- From Steps 8 to 10: These are the sets of attacks aiming for the modification of the Data Link Layer parameters by intercepting the DNP3 packet. The actions that are carried out as part of the Length Overflow Attack (LOVA), Data Flow Control Flag (DFC Flag) and Function Code Attack (FCA) are executed using the interception procedure. During the attack, the length is modified by  $\pm \alpha$  to the original size, the DCF and FCA are modified with 0 or 1.
- From Steps 11 to 13: The targeted layer is the Transport Link Layer with the injection action of some fault parameters to the DNP3 packet. At this stage, the Fragmented Message Interruption Attack (FMI1 and FMI2) supports the fault parameters by injecting 1 or 0 to the First (FIR) and Final (FIN) Bit Number. The Transport Sequence Modification Attack (TSM) is also one of the TL attacks with sequence modification by  $\pm \alpha$  to the original order, but it is based on the modification procedure.

- From Steps 14 to 20: This is a range of attacks on the Application Link Layer with a large number of parameters. After the DNP3 is intercepted, the modification process is performed on the packet at the Application Layer. To do so, the Function Code Modification Attack (FCM1~6) is called, where the attacker sets up the parameters to be modified. The request to modify this function code at the application layer is based on the selected values (such as 0x02, 0x0d, . . . , 0x82 and modification of a byte of the internal indication, such as the 5th bit in the 2nd byte of the DNP3 packet at the Application Layer).

After the last step, the whole DNP3 packet (in the current session) is compromised, and it is time that the engine can classify between bad and good traffic. The results from the experiment are detailed in Section 4.

## 4. Experimental Results and Discussion

### 4.1. Malware Sample Feature Selection Results

The following section discusses the findings after data transformation for the classification process. For the purposes of our paper, we cannot include all of the figures and tables, but we have selected the most important results from among others. As described above in Section 3, the input data comprised about 10 malware and 80 of the benign dataset, which represents 56% malware and 45% benign data, respectively. For malware analysis and feature extraction, we selected the Stuxnet malware, and we parsed this sample using the Pepper tool, which is an open-source tool for malware static analysis on a portable executable [60]. We extracted the metadata, header, opt header, sections, and import features from the executable file, as shown in Figure 8.

```

user@kali:~/Downloads/Pepper$ sudo python pepper.py malware/Stuxnet/A0055521.sys
-----
PEPPER
-----
Th3Hurrican3

----- METADATA -----
File name: A0055521.sys
Upload time: 2019-07-30 22:47:05
File size: 20616 byte
File type: PE32 executable (native) Intel 80386, for MS Windows
MD5: f8153747bae8b4ae48837ee17172151e
SHA1: cb0793029c0c0bd059ff85de956619f7fdeb4fd
SHA256: 1635ec04f009ccc8331d01fdf31132a4bc8f6fd3830ac94739df95ee093c555c

----- HEADER -----
Signature: PE
Machine: MACHINE_TYPE5.I386
Number of sections: 6
Time Date stamp: 1230836005
Pointer to symbols: 0
Number of symbols: 0

----- OPT HEADER -----
Magic: PE32
Major linker version: 8
Minor linker version: 0
Size of code: 9472
Size of initialized data: 9600
Size of uninitialized data: 0

----- SECTIONS -----
.text
Virtual Address: 768
Virtual Size: 8237
Raw Size: 8320
Entropy: 6.50964599803
Readable: [u2713]
Writable: [X]
Executable: [u2713]
Suspicious: [X]

----- IMPORTS -----
ntoskrnl.exe
0x9104 ZwReadFile
0x9108 ZwClose
0x9112 ZwOpenFile

```

Figure 8. Pepper tool: Stuxnet PE malware reverse engineering and feature extraction.

The Stuxnet malware PE result shows that many system files are subject to compromises or attacks. Figure 9 shows the distribution of the Top 20 process names found after the reverse engineering of the malware using the Pepper tool. We selected only major information, with 50% and high score points as given in Table 4. The main reason is that after computing all of the features, it was necessary to statistically pick out only those with a high degree of presence in the original database. Presence refers to how frequently the feature occurs throughout the whole DB. For our experiment we set, 50% as the threshold. The table indicates that the malware target memory process has highest score, with four times, and it can be observed that the file names being compromised are related to the memory processes. The other process is related to the local security system authority service, which is a highly critical system file in Microsoft Windows (the lsass.exe). Most malware targets this file, because it is used to enforce security policies related to sensitive information such as password changes and login access verifications. The malware also targets another executable file with a task of high importance in the Windows Task Manager, and which contains machine code, and this is called vdmdbg. It has also a

high score in the below table. Appendix B provides all of the feature information from the Stuxnet Portable Executable (PE) file.

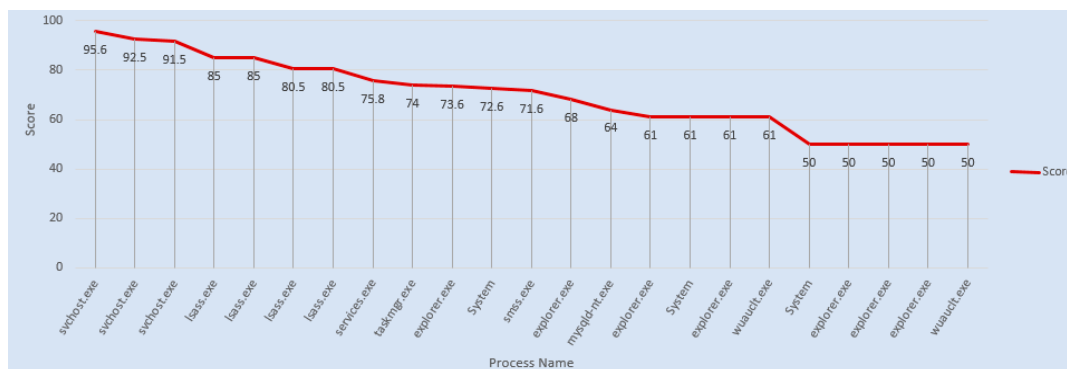


Figure 9. Top 20 targeted processes from the Stuxnet PE file.

Table 4. Stuxnet features and scores.

Number	File Name	Process Name	Score %
1	memory-mod-pe-0 × 20000000-0 × 10124000	service.exe	95.6
2	kerner32.dll.aslr.0013a1e	svchost.exe	92.5
3	kerner32.dll.aslr.0013b86	svchost.exe	91.5
4	Memory mod-pe-0 × 00090000-0 × 0010a000	lsass.exe	85
5	Memory mod-pe-0 × 00090000-0 × 0010a000	lsass.exe	85
6	lsass.exe	lsass.exe	80.5
7	lsass.exe	lsass.exe	80.5
8	memorymod-0 × 006b0000-0 × 006b1000	services.exe	75.8
9	vdmdbg.dll	taskmgr.exe	74
10	izarccm.dll	explorer.exe	73.6
11	ntoskn1.exe	System	72.6
12	ntdll.dll	smss.exe	71.6
13	olepro32.dll	explorer.exe	68
14	mysqld-nt.exe	mysqld-nt.exe	64
15	mlang.dll	explorer.exe	61
16	bhomanger.dll	System	61
17	hal.dll	explorer.exe	61
18	wuauclpl.cpl	wuauclt.exe	61
19	mrinet.sys	System	50
20	vmhgfs.dll	explorer.exe	50
21	odbc32.dll	explorer.exe	50
22	wuauclpl.cpl	explorer.exe	50
23	odbc32.dll	wuauclt.exe	50

#### 4.2. DNP3 Protocol Packet Sample Feature Selection Results

The DNP3 packets that include the attack types defined in Section 3 are collected using the Wireshark tool, which is a network packet analyzer that captures network packets and displays the packet contents with the maximum detail possible [61]. In order to generate the packet, we developed an exploit that was specifically designed to carry out a cyber-attack on the DNP3 protocol. This malicious software is real, and we advise the reader of this paper not to try this on a live product. The exploit, as given in Appendix C, carries the data (payload) that intercept the traffic and then injects some modified parameters, as described in Algorithm 2.



After the attacks, we collected the features from the DNP3 packet where the results revealed that the predicted attacks (as defined in Algorithm 2) achieved the goals. Table 5 describes our experimental results in detail, along with the impact on the SCADA/DNP3 devices. As can be seen, the impact depends on the attack type, the parameter modified in the original format, and the link layer that is attacked. As described above, it is prohibited to run the provided exploit in a real working environment, because the impact of the attack would be damaging. The rest of the features of DNP3 are given in Appendix A.

**Table 5.** Feature description of DNP3 protocol attack.

DNP3-Features	Description	Attack Type	Parameter	Impact	Link Layer
DNP3_LENGTH	Length of field	LOVA	Original length modification	Device crashes	DL
DNP3_CONTROL_DFC	The DFC tells other devices that the current device is busy	DFC	Flag = 1	Eternal busy	DL
DNP3_CONTROL_FUN_CODE_PRI	Primary Function code	User Process reset	Code = 1	Unwanted restart	DL
DNP3_Transport_FIN	Final bit	FMI	Flag modification	Early message termination	TL
DNP3_Transport_FR	First bit	FMI	Flag modification	Message processing error	TL
DNP3_Application_request_FUN_CODE	Function Code	FCA	$0 \times 02, 0 \times 0d, 0 \times 0e, 0 \times 14, 0 \times 15, 0 \times 82$	Crash or reboot	AL
DNP3-Application_response_IIN_CONFIG_CORRUPT	Configuration File System	CC_IIN	5th bit in 2 <sup>nd</sup> byte	Configuration file modified	AL

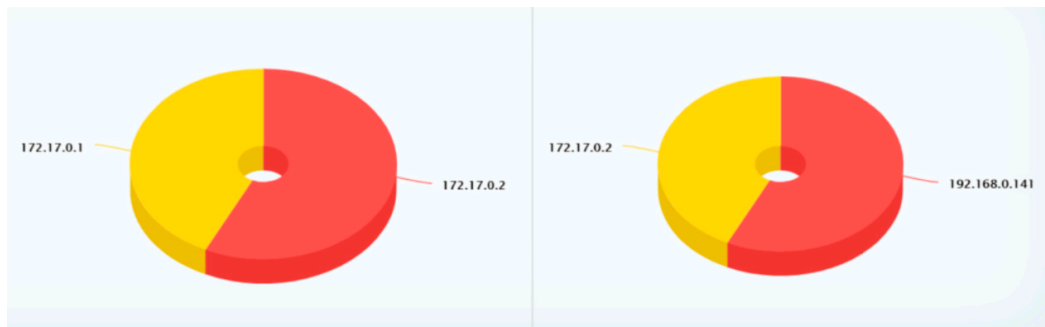
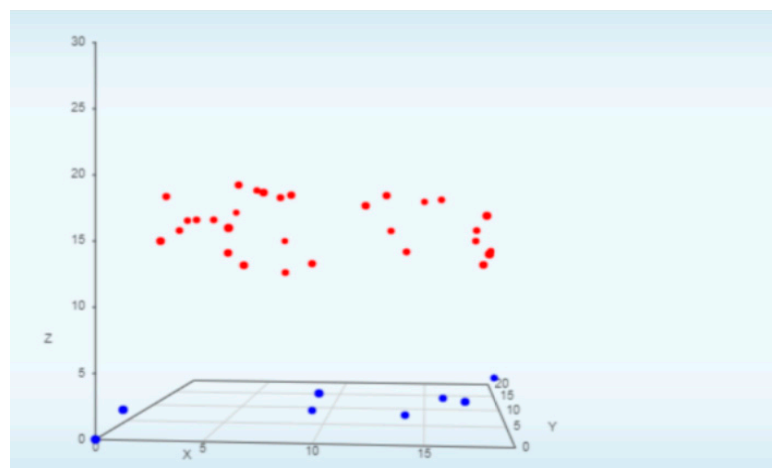
#### 4.3. Visualization and Classification

The discussion in this subsection is related to the results of the proposed methods based on the classification of malware, which is displayed in the form of a graph visualization. Table 6, with Figures 10 and 11, describes the classification results with the following explanation:

- **Login-Time:** The field indicates when the event happened. It is, therefore, easy to track down and find out the right moment for the attack on the system when there is a need for an investigation.
- **Source-IP:** Every traffic on the network includes the source IP address, which indicates the origin of the data, request, or other type of transaction. In our experiment, this is the IP address of the device that is sending information to the destination device.
- **Source-Port:** This is one of the user session parameters that tells the system where to reply to the response. It is always associated with the source-IP and the different applications and protocols used by the sender.
- **Destination-IP:** This is where to go. The receiving device in our experiment has a destination IP to which the packet is to be sent. This enables two-way communication in the configuration.
- **Destination-Port:** The same explanation as the source port, except that this is for the destination device.
- **Classification:** As stated before, the aim is to distinguish between benign and malware groups for elements in the dataset. Hence, after the process, the result results in an “anomaly”, as bad packets related to malware or any malicious activity are discovered during the analysis. We only provide those results that identify an anomaly.
- **Field:** With this information, we can see what type of feature, attack type or any other field has been targeted. In this case, the system gives “Transport FIR”, which indicates the DNP3 protocol feature.
- **Graph Visualization:** The 3D graph indicates the classification as either malware or benign data. The red dots indicate the malware sample in our experiment, while the blue ones indicate the benign dataset. Additionally, there other two graphs, which give an overview of the Top 5 source and destination IP addresses.

**Table 6.** Classification result description.

Source-IP	Source-Port	Destination-IP	Destination-Port	Classification	Field
172.17.0.1	59686	172.17.0.2	45000	Anomaly	Transport FIR
172.17.0.1	59686	172.17.0.2	45000	Anomaly	Transport FIR
172.17.0.2	41044	192.168.0.141	45000	Anomaly	Transport FIR
172.17.0.2	41044	192.168.0.141	45000	Anomaly	Transport FIR
172.17.0.2	41044	192.168.0.141	45000	Anomaly	Transport FIR
172.17.0.2	41044	192.168.0.141	45000	Anomaly	Transport FIR
172.17.0.2	41044	192.168.0.141	45000	Anomaly	Transport FIR

**Figure 10.** Top 5 Source IPs (right) and Top 5 Destination IPs (left).**Figure 11.** Classification with a 3D graph. Red dots are malware. Blue dots are benign.

## 5. Conclusions

This paper discussed cybersecurity solutions based on the Intrusion Detection System in the IoT-based Smart Grid. We described in detail the concept of a system based on the IoT for Smart Grids using the SCADA/DNP3 communication protocol. To achieve the proposed method, we developed and presented a series of algorithms for implementation along with experiments.

In this paper, we developed a new method for assessing DNP3 protocol vulnerability, which gave us an idea of where to perform the attack. This assessment was conducted on a modified DNP3 protocol with reference to the original protocol. Next, based on the discovered vulnerabilities, we developed the new attack model aiming at the Data Link Layer, Transport Link Layer and Application Link Layer of the DNP3 protocol. Moreover, we developed two algorithms that helped us perform data transformation using Machine Learning methods. The other algorithm includes all of the steps for the cyber-attack on the DNP3 protocol; this also includes the classification process. Finally, we presented the experimental results, showing that the proposed method was able to detect intrusions to the SCADA system based on an IoT Smart Grid and could classify them with detailed information about the compromised fields from the DNP3 packet.

**Author Contributions:** Conceptualization, Writing the Original Draft, Project Administration, and Funding Acquisition, X.C.X.; Methodology, Writing—Review & Editing, Formal Analysis and Validation, Z.G.L.; Investigation and Data Curation, L.N.; Visualization, Resources, and Supervision, B.N.

**Funding:** This research was supported by the Scientific Fund Project of Facility Horticulture Laboratory of Universities in Shandong of China (Grant number: 2018YY016) and the Doctoral Scientific Fund Project of Weifang University of Science & Technology of China (Grant number: 2017BS17), it was also supported by the Innovation Fund of Ministry of Education, Science and Technology Development Center of China (Grant number: 2018A02013).

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A DNP3 Features before Selection Process

**Table A1.** In this table ‘-’ means that the feature does not have value. We have 95 features from the DNP3 packets.

Original Format	Readable Format
dnp.data_chunk.CRC.status	-
dnp.hdr.CRC.status	-
dnp3.addr	-
dnp3.src	-
Number of Items:	-
Number of Items: 0	-
dnp3.al.fragment	al_fragment
dnp3.al.fragment.count	al_fragment_count
dnp3.al.fragment.reassembled.length	al_fragment_reassembled_length
dnp3.al.ctl	application_layer_control
dnp3.al.con	application_layer_control_confirm
dnp3.al.fin	application_layer_control_final
dnp3.al.fir	application_layer_control_first
dnp3.al.func	application_layer_control_function_code
dnp3.al.iin	application_layer_control_internal_indications
dnp3.al.iin.bmsg	application_layer_control_internal_indications_broadcast_msg_rx
dnp3.al.iin.cls1d	application_layer_control_internal_indications_class1_data_available
dnp3.al.iin.cls2d	application_layer_control_internal_indications_class2_data_available
dnp3.al.iin.cls3d	application_layer_control_internal_indications_class3_data_available
dnp3.al.iin.cc	application_layer_control_internal_indications_configuration_corrupt
dnp3.al.iin.rst	application_layer_control_internal_indications_device_restart
dnp3.al.iin.dt	application_layer_control_internal_indications_device_trouble
dnp3.al.iin.dol	application_layer_control_internal_indications_digital_outputs_in_local
dnp3.al.iin.ebo	application_layer_control_internal_indications_event_buffer_overflow
dnp3.al.iin.fcni	application_layer_control_internal_indications_function_code_not_implemented
dnp3.al.iin.oae	application_layer_control_internal_indications_operation_already_executing
dnp3.al.iin.pioor	application_layer_control_internal_indications_parameters_invalid_or_out_of_range
dnp3.al.iin.obju	application_layer_control_internal_indications_requested_objects_unknown
dnp3.al.iin.tsr	application_layer_control_internal_indications_time_sync_required
dnp3.al.seq	application_layer_control_sequence
dnp3.al.uns	application_layer_control_unsolicited
dnp3.al.obj	application_layer_object
dnp3.al.range.quantity	application_layer_object_items_range_quantity
dnp3.al.range.start	application_layer_object_items_range_start
dnp3.al.range.stop	application_layer_object_items_range_stop
Point Number	application_layer_object_point_number
dnp3.al.index	application_layer_object_point_number_index
dnp3.al.point_index	application_layer_object_point_number_index
dnp3.al.ana.int	application_layer_object_point_number_quality_analog_value
dnp3.al.aiq.b2	application_layer_object_point_number_quality_comm_fail
dnp3.al.biq.b2	application_layer_object_point_number_quality_comm_fail
dnp3.al.aiq.b4	application_layer_object_point_number_quality_local_force
dnp3.al.biq.b4	application_layer_object_point_number_quality_local_force
dnp3.al.aiq.b0	application_layer_object_point_number_quality_online
dnp3.al.biq.b0	application_layer_object_point_number_quality_online
dnp3.al.aiq.b5	application_layer_object_point_number_quality_over_range
dnp3.al.biq.b5	application_layer_object_point_number_quality_over_range
dnp3.al.aiq.b6	application_layer_object_point_number_quality_reference
dnp3.al.biq.b6	application_layer_object_point_number_quality_reference
dnp3.al.retimestamp	application_layer_object_point_number_quality_relative_timestamp
dnp3.al.aiq.b3	application_layer_object_point_number_quality_remote_force
dnp3.al.biq.b3	application_layer_object_point_number_quality_remote_force

Table A1. Cont.

Original_Format	Readable_Format
dnp3.al.aiq.b7	application_layer_object_point_number_quality_reserved
dnp3.al.biq.b7	application_layer_object_point_number_quality_reserved
dnp3.al.aiq.b1	application_layer_object_point_number_quality_restart
dnp3.al.biq.b1	application_layer_object_point_number_quality_restart
dnp3.al.time_delay	application_layer_object_point_number_quality_time_delay
dnp3.al.bit	application_layer_object_point_number_value
dnp3.al.objq.prefix	application_layer_object_prefix_code
dnp3.al.objq.range	application_layer_object_range_code
dnp3.al.timestamp	application_layer_timestamp
dnp3.ctl	control
dnp3.ctl.dir	control_direction
dnp3.ctl.fcb	control_frame_count_bit
dnp3.ctl.fcv	control_frame_count_valid
dnp3.ctl.prifunc	control_function_code
dnp3.ctl.prm	control_primary
dnp.data_chunk	data_chunk
dnp.data_chunk.CRC	data_chunk_crc
dnp.data_chunk.len	data_chunk_length
dnp3.hdr.CRC	data_link_header_crc
dnp3.dst	destination
_ws.expert.group	expert_info_group
dnp3.iin_abnormal	expert_info_iin_abnormal
_ws.malformed	expert_info_malformed_packet
_ws.expert.message	expert_info_message
_ws.expert.severity	expert_info_severity
dnp3.len	length
dnp3.start	start_bytes
dnp3.tr.ctl	transport_control
dnp3.tr.fin	transport_control_final
dnp3.tr.fir	transport_control_first
dnp3.tr.seq	transport_control_sequence
dnp3.al.2bit	‘/’
dnp3.al.count	‘/’
dnp3.al.ctrlstatus	‘/’
dnp3.al.off_time	‘/’
dnp3.al.on_time	‘/’
dnp3.al.range.abs	‘/’
dnp3.al.size	‘/’
dnp3.al.unknown_data_chunk	‘/’
dnp3.ctl.clr	‘/’
dnp3.ctl.op	‘/’
dnp3.ctl.trip	‘/’
dnp3.num_items_neg	‘/’

## Appendix B Stuxnet Portable Executable Features

**Table A2.** The complete list of Stuxnet PE reverse engineering malware with the names of the processes and the score.

Number	Name	Process Name	Score
1	memory-mod-pe-0 × 20000000-0 × 10124000	service.exe	95.6
2	kerner32.dll.aslr.0013a1e	svchost.exe	92.5
3	kerner32.dll.aslr.0013b86	svchost.exe	91.5
4	memorymod-pe-0 × 00090000-0 × 0010a000	lsass.exe	85
5	memorymod-pe-0 × 00090000-0 × 0010a000	lsass.exe	85
6	lsass.exe	lsass.exe	80.5
7	lsass.exe	lsass.exe	80.5
8	memorymod-0 × 006b0000-0 × 006b1000	services.exe	75.8
9	vdmdbg.dll	taskmgr.exe	74
10	izarccm.dll	explorer.exe	73.6
11	ntoskn.exe	System	72.6
12	ntdll.dll	smss.exe	71.6
13	olepro32.dll	explorer.exe	68
14	mysqld-nt.exe	mysqld-nt.exe	64
15	mlang.dll	explorer.exe	61

Table A2. Cont.

Number	Name	Process Name	Score
16	bhomanger.dll	System	61
17	hal.dll	explorer.exe	61
18	wuauclt.cpl	wuauclt.exe	61
19	mrxnet.sys	System	50
20	vmhgfs.dll	explorer.exe	50
21	odbc32.dll	explorer.exe	50
22	wuauclt.cpl	explorer.exe	50
23	odbc32.dll	wuauclt.exe	50
24	mrxcsl.sys	winlogon.exe	49
25	natlanman.dll	System	46
26	browseui.dll	explorer.exe	42
27	ksecdd.sys	explorer.exe	41
28	hidphone.tsp	System	34
29	cscdll.dll	svchost.exe	34
30	cscdll.dll	winlogon.exe	34
31	util.dll	explorer.exe	34
32	taskmgr.exe	taskmgr.exe	26
33	sfc_os.dll	taskmgr.exe	26
34	sfc_os.dll	spoolsv.exe	26
35	duser.dll	svchost.exe	26
36	sfc_os.dll	explorer.exe	26
37	ntdll.dll	wuauclt.exe	26
38	ntdll.dll	crss.exe	26
39	ntdll.dll	svchost.exe	26
40	ntdll.dll	VMwareUser.exe	26
41	mprapi.dll	lsass.exe	19
42	mprapi.dll	winlogon.exe	19
43	h323.tsp	svchost.exe	19
44	tapisrv.dll	svchost.exe	19
45	alg.exe	alg.exe	19
46	ntdll.dll	svchost.exe	19.6

## Appendix C

```

import socket
import datetime
import time

class bcolors:
    HEADER = '\033[95m'
    MARGENTA = '\033[35m'
    BLUE = '\033[34m'
    YELLOW = '\033[32m'
    GREEN = '\033[32m'
    RED = '\033[31m'
    CYAN = '\033[36m'
    OKBLUE = '\033[94m'
    OKGREEN = '\033[92m'
    WARNING = '\033[93m'
    FAIL = '\033[91m'
    ENDC = '\033[0m'
    BOLD = '\033[1m'
    UNDERLINE = '\033[4m'

now = datetime.datetime.now()
print bcolors.RED + bcolors.BOLD

ip = raw_input("[*] IP :")
port = input("[*] PORT :")

Socket=socket.socket(socket.AF_INET, socket.SOCK_STREAM)
Socket.connect((ip, port))

data = "\x05\x64\x05\x00\x1f\x00\x0b\x00\x24\x61"

print "[*] TARGET : ",ip," PORT : ",port
print "[*] PAYLOAD : ",data.encode('hex')

print "[*] SENDING PAYLOAD ..."
for i in range(0,20):
    Socket.sendall(data)
    # print "[*] SENDING PAYLOAD : ",data.encode('hex')
    # print "[*] SENDING PAYLOAD : ",data.encode('hex')
print "[*] DONE." + bcolors.ENDC

```

**Figure A1.** This exploit is a real hack, and we advise the reader not to use it in a real working environment such as SCADA/DNP3 devices. However, for research and academic purpose, you can set up a virtual box.

## References

1. Future Energy Grid, Migration to the Internet of Energy. Acatech STUDY. Available online: <http://en.acatech.de/> (accessed on 3 October 2019).
2. Rajendra, K.P.; Mohit, M. Cyber Security Threats–Smart Grid Infrastructure. In Proceedings of the National Power Systems Conference (NPSC), Bhubaneswar, India, 19–21 December 2016. [CrossRef]
3. Zhu, B.; Joseph, A.; Sastry, S. A Taxonomy of Cyber Attacks on SCADA Systems. In Proceedings of the 2011 International Conference on Internet of Things and 4th International Conference on Cyber, Physical and Social Computing, Washington, DC, USA, 19 October 2011; pp. 380–388.
4. Hong, S.; Lee, M. Challenges and Direction toward Secure Communication in the SCADA System. In Proceedings of the 8th Annual Communication Networks and Services Research Conference, Montreal, QC, Canada, 11–14 May 2010.
5. Badra, M.; Zeadally, S. Design and Performance Analysis of a Virtual Ring Architecture for Smart Grid Privacy. *IEEE Trans. Inf. Forensics Secure.* **2014**, *9*, 312–329. [CrossRef]
6. Jay, M. *Comparison of Protocols Used in Remote Monitoring: DNP3.0, IEC 870-5-101 & Modbus*. Credit Seminar Report; Electronics Systems Group, EE Dept. IIT Bombay: Mumbai, India, November 2003.
7. Collier, S. Ten Steps to a Smarter Grid. *IEEE Ind. Appl. Mag.* **2010**, *16*, 62–68. [CrossRef]
8. DNP Users Group. *DNP3 Application Layer Specification*; Version 2.00; DNP Organization: Washington, WA, USA, 2005; Volume 2.
9. Clarke, G.; Reynders, D.; Wright, E. *Practical Modern SCADA Protocols: DNP3, 60870.5 and Related Systems*; Elsevier: New York, NY, USA, 2004.
10. Tiago, C.; Luis, R.; Jorge, P.; Leandros, M.; Matthieu, A.; Leonid, L.; Jiang, J.; Paulo, S. A cybersecurity detection framework for supervisory control and data acquisition systems. *IEEE Trans. Ind. Inform.* **2016**, *12*, 2236–2246.
11. Byungho, M.; Vijay, V. Design and Analysis of Security Attacks against Critical Smart Grid Infrastructures. In Proceedings of the IEEE 2014 19th International Conference on Engineering of Complex Computer Systems, Washington, DC, USA, 4–7 August 2014. [CrossRef]
12. Peter, E.N.; Tanja, Z.; Joachim, F. Malware propagation in smart grid networks: Metrics, simulation and comparison of three malware types. *J. Comput. Virol. Hacking Tech.* **2019**, *15*, 109–125. [CrossRef]
13. Catapult. Available online: <https://www.catapultsoftware.com/products/electricity-distribution-management/dnp3.html> (accessed on 5 October 2019).
14. Torrisi, N.; Vukovic, O.; Dan, G.; Hagdahl, S. Peekaboo: A gray hole attack on encrypted SCADA communication using traffic analysis. In Proceedings of the 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm), Venice, Italy, 3–6 November 2014; pp. 902–907.
15. Yang, Y.; Zheng, X.; Guo, W.; Liu, X.; Chang, V. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. *Inf. Sci.* **2019**, *479*, 567–592. [CrossRef]
16. Liu, X.; Deng, R.H.; Choo, K.-K.R.; Yang, Y.; Pang, H. Privacy-Preserving Outsourced Calculation Toolkit in the Cloud. *IEEE Trans.* **2018**, *1*, 435–441. [CrossRef]
17. Chen, X.; Li, A.; Zeng, X.; Guo, W.; Huang, G. Runtime model-based approach to IoT application development. *Front. Comput. Sci.* **2015**, *9*, 540–553. [CrossRef]
18. Lin, C.; He, B.; Huang, X.; Choo, K.-K.R.; Vasilakos, A.V. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0. *J. Netw. Comput. Appl.* **2018**, *116*, 42–52. [CrossRef]
19. De Toledo, T.R.; Torrisi, N.M. Encrypted DNP3 Traffic Classification Using Supervised Machine Learning Algorithms. *Mach. Learn. Knowl. Extr.* **2019**, *1*, 384–399. [CrossRef]
20. Institute of Electrical and Electronics Engineers. *IEEE Standard for Electric Power Systems Communications 1815–2012*; Institute of Electrical and Electronics Engineers: Rio de Janeiro, Brazil, 2012.
21. Tan, X.; Su, X.; Qian, Q. The classification of SSH tunneled traffic using maximum likelihood classifier. In Proceedings of the 2011 International Conference on Electronics, Communications and Control (ICECC), Ningbo, China, 9–11 November 2011; pp. 2347–2350. [CrossRef]
22. Maiolini, G.; Baiocchi, A.; Iacovazzi, A.; Rizzi, A. Real-Time Identification of SSH Encrypted Application Flows by Using Cluster Analysis Techniques. In Proceedings of the Networking 2009: 8th International IFIP-TC 6 Networking Conference, Aachen, Germany, 11–15 May 2009.
23. Fratta, L.; Schulzrinne, H.; Takahashi, Y.; Spaniol, O. (Eds.) *NETWORKING*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 182–194.



24. Rezaei, S.; Liu, X. Deep Learning for Encrypted Traffic Classification: An Overview. *arXiv* **2018**, arXiv:1810.07906.
25. Kim, S.-J.; Cho, D.-E.; Yeo, S.-S. Secure Model against APT in m-Connected SCADA Network. *Int. J. Distrib. Sens. Netw.* **2014**, *2014*, 1–8. [[CrossRef](#)]
26. Nabil, S.; Mohamed, B. Security solution for semantic SCADA optimized by ECC mixed coordinates. In Proceedings of the 2012 International Conference on Information Technology and e-Services (ICITeS), Sousse, Tunisia, 24–26 March 2012. [[CrossRef](#)]
27. Shahzad, A.; Udagopola, K.P.; Lee, Y.-K.; Park, S.; Lee, M. The Sensors Connectivity within SCADA Automation Environment and New Trends for Security Development during Multicasting Routing Transmission. *Int. J. Distrib. Sens. Netw.* **2015**. [[CrossRef](#)]
28. Musa, S.; Aborujilah, A. Secure security model implementation for security services and related attacks base on end-to-end, application layer and data link layer security. In Proceedings of the 7th International Conference on Ubiquitous Information Management and Communication, Kota Kinabalu, Malaysia, 17–19 January 2013.
29. Jin, D.; Nicol, D.M.; Yan, G. An event buffer flooding attack in DNP3 controlled SCADA systems. In Proceedings of the Winter Simulation Conference (WSC'11), Phoenix, AZ, USA, 11–14 December 2011; pp. 2619–2631.
30. DNP Users Group. *DNP3 Specification, Secure Authentication*; Version 5 Overview; DNP Organization: Washington, DC, USA, 2013.
31. Shahzad, A.; Lee, M.; Kim, S.; Kim, K.; Choi, J.-Y.; Cho, Y.; Lee, K.-K. Design and Development of Layered Security: Future Enhancements and Directions in Transmission. *Sensors* **2016**, *16*, 37. [[CrossRef](#)] [[PubMed](#)]
32. Sharma, R.K.; Kalita, H.K.; Borah, P. Analysis of Machine Learning Techniques Based Intrusion Detection Systems. In Proceedings of the 3rd International Conference on Advanced Computing, Networking and Informatics, Bhubaneswar, India, 23–25 June 2015; pp. 485–493.
33. Teixeira, M.A.; Salman, T.; Zolanvari, M.; Jain, R.; Meskin, N.; Samaka, M. SCADA System Testbed for Cybersecurity Research Using Machine Learning Approach. *Future Internet* **2018**, *10*, 76. [[CrossRef](#)]
34. Keliris, A.; Salehghaffari, H.; Cairl, B. Machine learning-based defense against process-aware attacks on industrial control systems. In Proceedings of the IEEE International Test Conference (ITC), FortWorth, TX, USA, 15–17 November 2016.
35. Tomin, N.V.; Kurbatsky, V.G.; Sidorov, D.N.; Zhukov, A.V. Machine learning techniques for power system security assessment. In Proceedings of the IFAC Workshop on Control of Transmission and Distribution Smart Grids (CTDSG), Prague, Czech Republic, 11–13 October 2016.
36. Ihab, D.; Tarek, S. Attack Detection and Mitigation Techniques in Industrial Control System- Smart Grid DNP3. In Proceedings of the 2018 IEEE International Conference on Data Intelligence and Security, South Padre Island, TX, USA, 8–10 April 2018. [[CrossRef](#)]
37. Deng, R.; Xiao, G.; Lu, R.; Liang, H.; Vasilakos, A.V. False data injection on state estimation in power systems—Attacks, impacts, and defense: A survey. *IEEE Trans. Ind. Inform.* **2016**, *13*, 411–423. [[CrossRef](#)]
38. Darwish, I.; Igbe, O.; Saadawi, T. Experimental and theoretical modeling of DNP3 attacks in smart grids. In Proceedings of the 36th IEEE Sarnoff Symposium, Newark, NJ, USA, 20–22 September 2016.
39. Hodo, E.; Bellekens, X.; Hamilton, A.; Dubouilh, L.; Iorkyase, E.; Tachtatzis, C.; Atkin-son, R. Threat analysis of IoT networks using artificial neural network intrusion detection system. In Proceedings of the 2016 International Symposium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, Tunisia, 11–13 May 2016; pp. 1–6.
40. Muna, A.H.; Nour, M.; Elena, S. Identification of malicious activities in industrial internet of things based on deep learning models. *J. Inf. Secur. Appl.* **2018**, *41*, 1–11. [[CrossRef](#)]
41. Barnaby, S.; Luis, R.; Leandros, A.M.; Tiago, J.C.; Mohamed, A.F.; Paulo, S.; Helge, J. A Novel Intrusion Detection Mechanism for SCADA systems which Automatically Adapts to Network Topology Changes. *EAI Endorsed Trans. Ind. Netw. Intell. Syst.* **2017**, *4*, e4. [[CrossRef](#)]
42. Ihab, D.; Obinna, I.; Orhan, C.; Tarek, S.; Joseph, S. Smart Grid DNP3 Vulnerability Analysis and Experimentation. In Proceedings of the 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing, Washington, DC, USA, 3–5 November 2015. [[CrossRef](#)]
43. Raphael, A. Formal Security Analysis of the DNP3-Secure Authentication Protocol. Ph.D. Thesis, Philosophy Bachelor of Science, Queensland University of Technology Australia, Brisbane, Australia, 2016.
44. Munir, M.; Francesco, P.P.; Dumnda, W. DNP3Sec: Distributed Network Protocol Version 3 (DNP3) Security Framework. In *Advances in Computer, Information, and Systems Sciences, and Engineering*; Springer: Heidelberg, Germany, 2008; pp. 227–234. [[CrossRef](#)]

45. The Industrial Cybersecurity Conference 4SICS. Capture Files From 4SICS Geek Lounge. Available online: <https://www.netresec.com/?page=PCAP4SICS> (accessed on 7 July 2019).
46. Kyle, C.; Richard, S.; Leondros, M.; Helge, J. Vulnerability Analysis of Network Scanning on SCADA Systems. *Secur. Commun. Netw.* **2018**, *2018*, 3794603. [CrossRef]
47. Chetna, S.; Ashwin, N.; Mrinal, P. Function code-based vulnerability analysis of DNP3. In Proceedings of the 2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), Bangalore, India, 6–9 November 2016. [CrossRef]
48. Siddavatam, I.A.; Kazi, F. Security Assessment Framework for Cyber-Physical Systems: A Case-study of DNP3 Protocol. In Proceedings of the 2015 IEEE Bombay Section Symposium (IBSS), Bombay, India, 10–11 September 2015. [CrossRef]
49. Aaron, H.; Jason, S.; Sujeet, S. Security analysis of an advanced metering infrastructure. *Int. J. Crit. Infrastruct. Prot.* **2017**, *18*, 3–19. [CrossRef]
50. Dagoumas, A. Assessing the Impact of Cybersecurity Attacks on Power Systems. *Energies* **2019**, *12*, 725. [CrossRef]
51. Venkatachary, S.K.; Prasad, J.; Samikannu, R. Economic Impacts of Cyber Security in Energy Sector: A Review. *Int. J. Energy Econ. Policy* **2017**, *7*, 250–262.
52. Liu, R.; Vellaithurai, C.; Biswas, S.S.; Gamage, T.T.; Srivastava, A.K. Analyzing the Cyber-Physical Impact of Cyber Events on the Power Grid. *IEEE Trans. Smart Grid* **2015**, *6*, 2444–2453. [CrossRef]
53. Bencsáth, B.; Pék, G.; Buttyán, L.; Félegyházi, M. The Cousins of Stuxnet: Duqu, Flame, and Gauss. *Future Internet* **2012**, *4*, 971–1003. [CrossRef]
54. Bugra, C.; Erdogan, D. Malware Classification Using Deep Learning Methods. In Proceedings of the ACM SE '18: Southeast Conference, Richmond, KY, USA, 29–31 March 2018. [CrossRef]
55. Tomas, M.; Kai, C.; Greg, S.C.; Jeffrey, D. Efficient estimation of word representations in vector space. In Proceedings of the International Conference on Learning Representations, Scottsdale, AZ, USA, 2–4 May 2013.
56. Tomas, M.; Ilya, S.; Kai, C.; Greg, C.; Jeffrey, D. Distributed representations of words and phrases and their compositionality. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–8 December 2013; pp. 3111–3119.
57. Word2vec. Available online: <https://code.google.com/archive/p/word2vec/> (accessed on 28 July 2019).
58. Kateryna, C. Machine Learning Methods for Malware Detection and Classification. Bachelor's Thesis, Information Technology, University of Applied Sciences, Berlin, Germany, March 2017.
59. A DNP3 Protocol Primer. Available online: <https://documents.pub/document/dnp3-doc-library-558450d1ec2c1.html> (accessed on 28 July 2019).
60. Pepper. Available online: <https://github.com/Th3Hurricane3/PEpper> (accessed on 28 September 2019).
61. Wireshark. Available online: <https://www.wireshark.org/download.html> (accessed on 1 October 2019).

