A Targeted Attack For Enhancing Resiliency of Intelligent Intrusion Detection Modules in Energy Cyber Physical Systems

M. El Hariri, E. Harmon, H. F. Habib, T. Youssef, *Student Members* and O. A. Mohammed, *Fellow IEEE*
Energy Systems Research Laboratory
Department of Electrical and Computer Engineering
Florida International University
Miami, Fl., USA
mohammed@fiu.edu

***Abstract*— Secure high-speed communication is required to ensure proper operation of complex power grid systems and prevent malicious tampering activities. In this paper, artificial neural networks with temporal dependency are introduced for false data identification and mitigation for broadcasted IEC 61850 SMV messages. The fast responses of such intelligent modules in intrusion detection make them suitable for time-critical applications, such as protection. However, care must be taken in selecting the appropriate intelligence model and decision criteria. As such, this paper presents a customizable malware script to sniff and manipulate SMV messages and demonstrates the ability of the malware to trigger false positives in the neural network's response. The malware developed is intended to be as a vaccine to harden the intrusion detection system against data manipulation attacks by enhancing the neural network's ability to learn and adapt to these attacks.**

Keywords*— Artificial intelligence, cyber security, fake data detection and mitigation, IEC 61850, merging units, sampled measured values

I. INTRODUCTION

The power industry is increasingly relying on robust communication infrastructures to transmit and analyze transmission and distribution measurements in adaptive protection and control schemes. As the reliance on these technologies increases, so does the threat posed by attackers. If not properly secured, these communication-enabled technologies will be vulnerable and pose a potential to cripple the reliability and economy of the grid. The main enabler of automated adaptive protection schemes is the IEC 61850 data modelling standard. One of the main security challenges faced by modern IEC 61850-based protection techniques is data manipulation attacks within the process bus. According to the IEC 61850 model, the process bus is the medium where current and voltage measurements and event triggered commands are communicated as Sampled Measured Values (SMV) and Generic Object Oriented Substation Event (GOOSE) messages, respectively, within a local area network. The core vulnerability is in the fact that these time-critical messages are broadcasted over the local area network (LAN) unencrypted. Therefore, in the event of a network breach or the presence of a malicious insider within the network, data manipulation of such messages is an easy task and thus the opening and closing of circuit breakers is possible via injecting fake current and voltage values [1][2].

In fact, the catastrophic impacts of data manipulation and false data injection attacks on the reliable operation of the power system have been widely researched in recent literature. Authors in [3] showed how false measurements feedback to automatic generation control could impact the physical system stability by causing sudden declines in the system frequency, which in its turn causes unwanted load shedding schemes. The work in [4] demonstrates two realistic false data attack scenarios in which attackers introduce arbitrary errors to state variables to achieve a false state estimation of the power system. In the study conducted in [5], the authors quantitatively analyze the damage caused by false data injection with regards to the power system operation and security.

On the other hand, there are several works in the literature that focus on defense strategies to minimize service loss through several defense mechanisms. In [6], the authors recognize the potential impacts of data injection on the process bus and proposed an agreement algorithm to detect, locate, and prevent malicious data from being accepted by the IEDs and protection devices. In [7], the authors present an overview of vulnerabilities in the IEC 61850 protocol suite and discuss a method of GOOSE message modification using a malware script to sniff, manipulate, and inject control messages into the process bus in detail. In [8], the authors presented an intrusion detection system that is capable of filtering malicious messages based on predefined rules and known malicious signatures. In [9], an intrusion detection system based on GOOSE and SMV rule violation indicators was presented. This system, similar to other rule based systems, will not be able to detect unknown attacks that are not defined in their rule base.

Even though the Intrusion Detection System (IDS) will filter out uncoordinated attacks, it is still not robust enough to secure IEC 61850 automation processes. These solutions are also network-based and are themselves vulnerable to data manipulation from a savvy attacker. Attackers with sufficient

This work was supported by grants from the US Department of Energy (DoE).
The authors are with the Energy Systems Research Laboratory, ECE Department, Florida International University, Miami, FL, USA (E-mail: mohammed@fiu.edu)

information can spoof different data fields to obfuscate themselves from this IDS.

To accomplish the goal of robust security, machine learning techniques have been introduced to leverage the intelligence of rule based IDS. Machine-learning systems use the approach of anomaly detection in which a model is defined and positioned as normal and if an outlier is detected that differs, it is considered to be an anomaly [10]. This need is recognized by the authors in [11] and to that end, they incorporated a protection algorithm using a trained coupled time-series neural network (NN) that predicts incoming current measurement based on the microgrid's recent operating history. Then, an intrusion is detected and announced if a real measured value deviates from the predicted one. The potential of this solution was tested on data collected from a simulated microgrid and the results in [11] are promising for rapid verification of data integrity. However, the accuracy demonstrated was cultivated in a controlled testing environment using high current values for the malicious data injection. In a case where the attack is designed for the target system, the accuracy of the IDS deviates from the results in [11]. This issue must be addressed in the NN to ensure reliability in real-world attack scenarios.

Accordingly, the work in this paper is an extension of the authors' previous work in [11] where a configurable malware is developed to be used as a tool to examine and quantify the reduction in accuracy experienced by the NN in targeted attack scenarios. The purpose of this tool is to be used as a vaccine to harden the IDS against smart attacks by fine tuning its decision criteria and model parameters.

The rest of the paper is organized as follows: Section II describes the power microgrid model and the predictive IDS. Section III explains the development of the details of the malware script. Section IV will demonstrate the reduction in the accuracy of the NN against a variety of attack scenarios and Section V concludes this work.

II. System Description

The developed microgrid model, shown in Fig. 1, consists of two AC generators with a power capacity of 4.5 kW each. The generators are feeding two AC loads at buses 3 and 5 and two DC loads through the unidirectional converters shown. The microgrid is connected to the main utility grid via CB21. The microgrid model was built on MATLAB SIMULINK. An IEC 61850-based protection scheme was developed for fault localization and clearance. Consider the case of transmission line TL3. Merging units (MU31 and MU32) publish the current measurements as SMV messages from the left and right ends of the transmission line, respectively. An intelligent electronic device (IED3) subscribes to these messages and calculates their difference. If the difference is greater than a pre-specified threshold, a fault on the line is detected and IED3 issues a trip signal to CB31 and CB32 to clear the fault. Therefore, an attacker can inject fake packets with high current values in order to trigger an unwanted trip signal.

Fig. 2 shows a flowchart of the proposed IDS. The main idea is to add a trained intelligent module to IED3 which holds a buffer of N previous current measurement samples. These samples represent the recent history of operation at the section where IED3 is located and are used to forecast the value of the current in the incoming message. Once a new message is received, IED3 will compare the real value to the forecasted one. If the difference in the error is less than 2%, the message is processed, otherwise an alarm is issued and the forecasted value is provided to assist the decision on whether to block or process the incoming packets by the protection control logic while the intrusion is identified and removed. The actual blocking decision is to be taken by a higher security layer which will be addressed in future work.

The microgrid was simulated under different operating conditions and varying fault scenarios to generate the database used to train the neural network. Two major cases were studied: the first is in the grid-connected mode of operation and the second is in the islanded mode of operation. For each of the

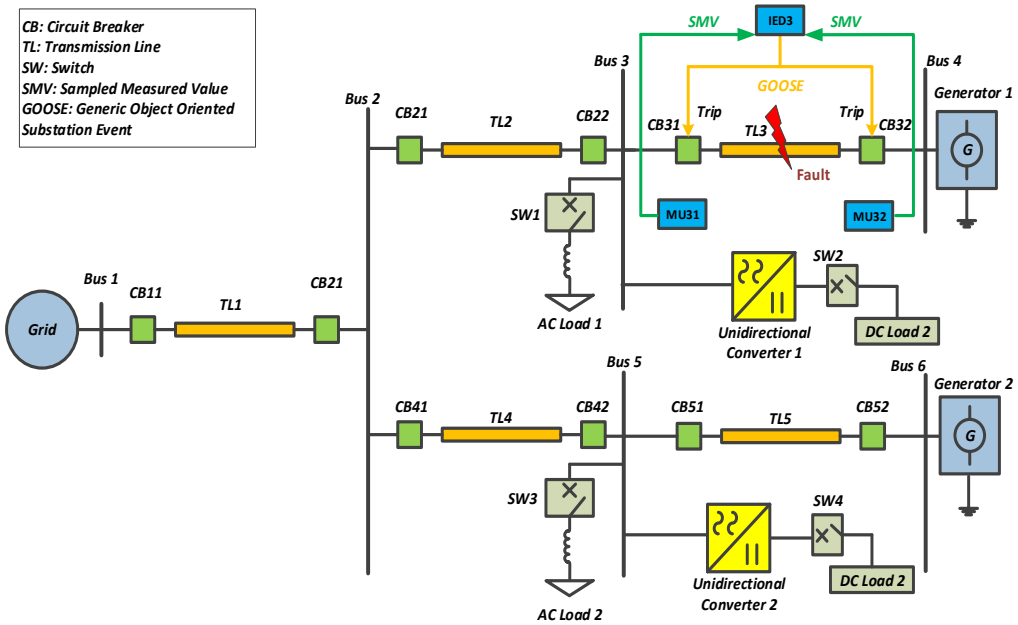


Fig. 1. Power Microgrid Model.

two cases mentioned, and on each transmission line, five types of faults were applied, namely, single-line-to-ground (A-G), line-to-line (B-C), double-line-to-ground (B-C-G), three-phase (A-B-C), and 3-phase-to-ground (A-B-C-G) faults. Additionally, each type of fault was applied on the beginning (10%), middle (50%), and end (90%) of each transmission line. For all cases mentioned above, the current measurements from both ends of TL3 were recorded. It is worth noting that the aforementioned contingency scenarios were utilized as a proof of concept of the developed IDS. The extension of this work would involve the utilization of the microgrid model to train the IDS neural network for more contingency scenarios, such as the sudden loss of a generating unit for example.

Fig. 3 shows the hardware setup built to test the IDS and implement the developed malware script over a real IEC 61850 process bus architecture. The recorded current values of both transmission lines from the simulated microgrid model

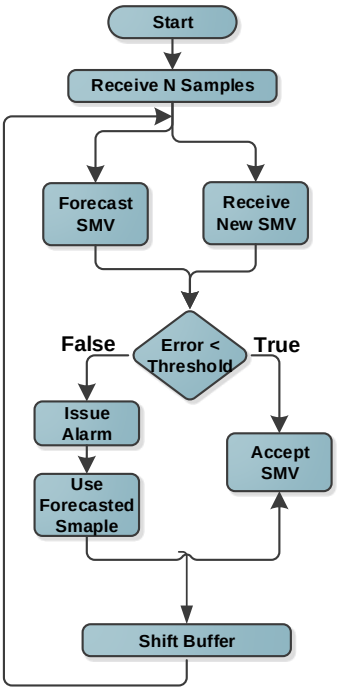


Fig. 2. IDS Algorithm Flowchart.

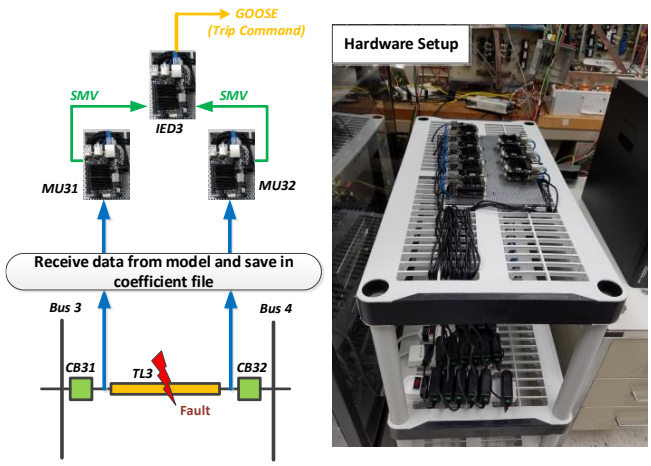


Fig. 3. Hardware Setup.

were recorded in a database of coefficient files and fed as inputs to the merging units. The measuring units then publish these measurements as IEC 61850 SMV packets. IED3 is programmed to subscribe to these messages and processes them through the IDS NN, which it hosts. The firmware for MU31, MU32, and IED3 along with the IDS were coded in C and downloaded on three different Odroid C2 devices running a Linux kernel.

III. Malware Development

A. SMV Message Structure

In order to better understand the malware development procedure shown in Fig. 4, the structure of the SMV message will be explained first. An SMV datagram follows a modified Abstract Syntax Notation One (ASN.1) Basic Encoding Rules (BER) Tag/Length pair encoding scheme [12]. The Tag field represents the type of information which is represented in the following SMV frame.

As shown in Fig. 5, the SMV datagram starts with the Destination MAC Address, which is a multicast address reserved for IEC 61850 applications always starting with 01-0C-CD and is followed by the source MAC address. An SMV message has an IEEE 802.1Q VLAN ID and a unique Ethernet type (88-BA). The APPID field is a 4 octet field which the subscribing IEDs use to identify messages they are subscribing to. The Length field represents the length of the overall SMV datagram and is followed by two reserved fields left out by the standard for future use.

The second layer of a SMV message is the Application Protocol Data Unit (APDU), which consists of one or more Application Service Data Units (ASDU). The number of ASDUs is in the *noASDU* field [13]. Each ASDU then contains the following subfields [14]:

- *svID*: unique identifier for each SMV message.
- *SmpCnt*: incrementing counter with each published SMV.
- *ConfRev*: counter for configuration changes.
- *SmpSynch*: A boolean value indicating synchronization with a clock signal.
- *SeqData*: List of data values related to the data set definition.

B. Malware Development

In order to properly inoculate the NN against smart attacks, a malware script for targeted attacks against the process bus was developed. The malware was written in Python in conjunction with network sniffing and packet crafting libraries from Scapy. As mentioned earlier, SMV messages are broadcasted over the LAN and are unencrypted. Once the malware is run, it starts to passively sniff packets from the network. Next, it filters those messages looking for the destination MAC address assigned for SMV messages, the Ethertype (88xBA), and the designated APPID identifier.

Once the SMV packet has been identified, it is converted into a list of hexadecimal pair strings that will make searching the packet more convenient. After the packet has been properly

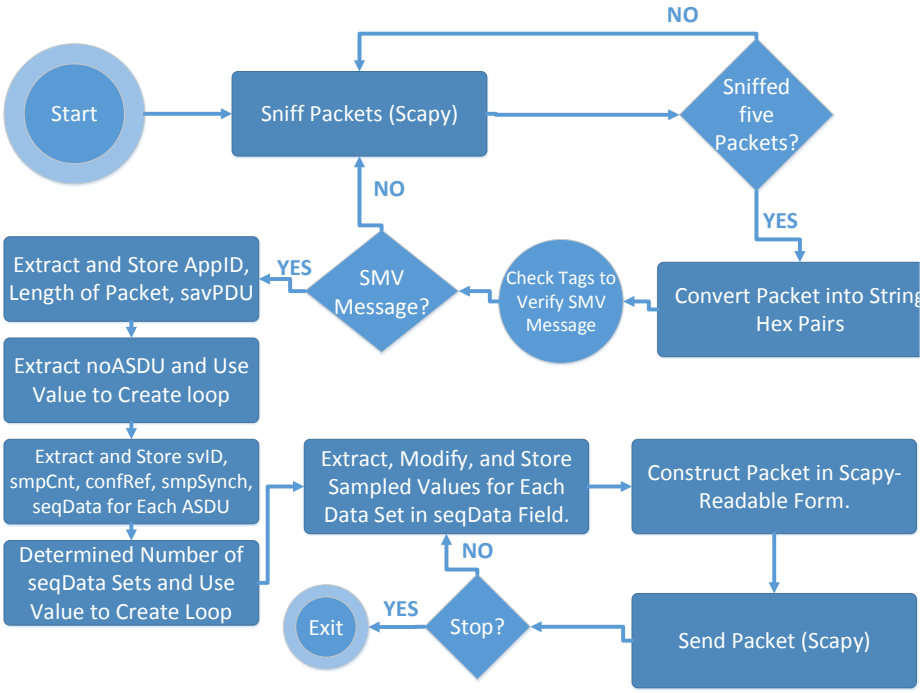


Fig. 4. Malware Development Process.

Destination Address		**Source MAC Address**		**Priority Tagging/ VLAN ID**	
Ethertype (88BA)		**APPID**		**Length**	
Reserved 1		**Reserved 2**		**APDU**	
Tag	**Length**	**noASDU**	**Tag**	**Length**	**svID**
Tag	**Length**	**SmpCnt**	**Tag**	**Length**	**ConfRev**
Tag	**Length**	**SmpSynch**	**Tag**	**Length**	**Sample 1**
Tag	**Length**	**Sample 2**	**Tag**	**Length**	**Sample 2**
••••• •••••			**Tag**	**Length**	**Sample N**

Fig. 5. SMV Datagram Structure.

decoded and stored as hex pairs, the malware can be utilized to spoof any desired field of the captured SMV packet. However, the developed malware will spoof all the fields (source MAC address, destination MAC address, APPID …) and change only the *seqData* field, which holds the value of the current measurement. This is intended to trick the IED into recognizing this packet as being sent from its original merging unit.

In order to manipulate the current measurements, a class called "ASDU" was created and used to store the data collected from the packet, and to conveniently build the new packet before it is injected into the process bus. A major obstacle in the design of this script is to write it in a way that will be able to process any number of ASDUs and *seqData* for the SMV packet. These values are not static and can be changed by the network administrators based on how often they want the collected measurements to be sent. In order to address this issue, we implement our *datafield* search function, which is called in an incremental manner to make sure the packet will be of the appropriate length and that it modifies all ASDUs and *seqData* fields detected by the script.

The first field searched for is the *noASDU* field as this will describe the number of ASDUs present in the packet. Once this value is determined, a loop is created that iterates through each ASDU and records the information into the ASDU class. An array of pointers to ASDU objects is used to easily navigate through the collected data as this can become quite tedious when an extremely large amount of ASDUs are introduced. For each ASDU, the *datafield* type, length, and value of each field is stored into its own array of hex pairs for reasons that become apparent when the script re-crafts the packet with the modified data. As each field is decoded, it is stored into the corresponding ASDU class member. Once the entire packet has been decoded, the final step is to modify the existing data and send the new packet.

It is important to note that the attacker must configure which fields they wish to modify and either manually change the value of the data, create an arithmetic function to manipulate the data, or read in predetermined values from a file. Depending on the method chosen, the script will begin to modify the data based on the attacker's configuration, and overwrite the original ASDU class members with these new values. Once the modification is complete, the packet is rebuilt in the correct order and the spoofed packet with false data is broadcasted into the LAN. The flowchart in Fig. 4 presents the malware algorithm for visualization of the process.

IV. Results and Discussion

Once this tool had been developed, spoofed messages were injected into the process bus to test the response of the IDS. The malware is assumed to be running on a computer device connected to the LAN of the microgrid under study. In order to test efficacy of the NN for intrusion detection, two data modification methods were used and the results for each were observed.

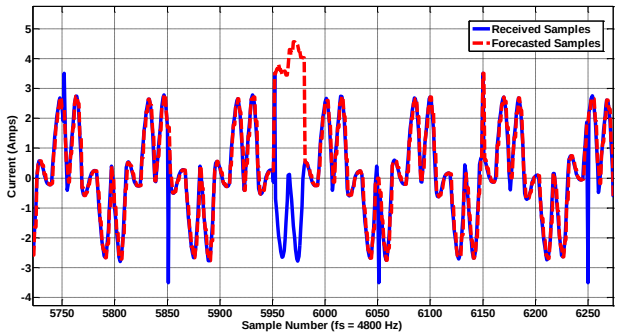


Fig. 6. Incorrect Forecasting Triggered by Malicious Alternating Sampled Values.

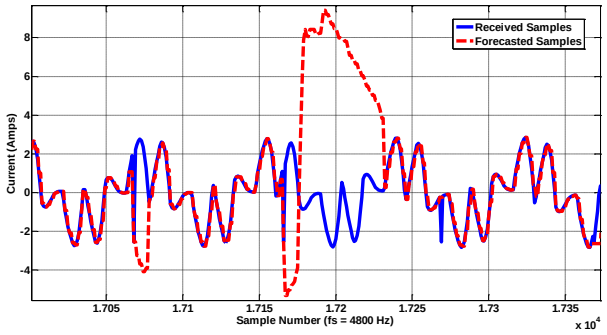
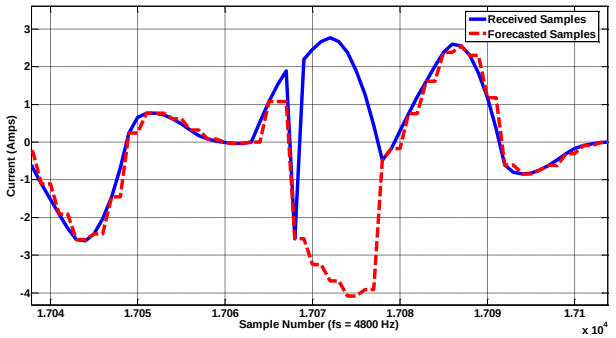
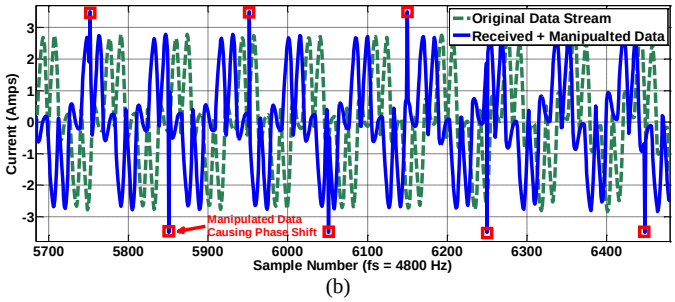


Fig. 7. Resuls for Injected Measurements Simulating a Real AC Waveform Measurements.



(a)



(b)

Fig. 8. Sample Shifting.

First, the values of the currents were recorded and analyzed for the first few AC cycles. It was noticed that the amplitude of the recorded current waveform was around 3 A. As appreciated from [11], injecting values far above 3 A triggered an intrusion alarm. However, as explained earlier, the NN was trained to recognize the current values for normal and fault conditions. Therefore, in the first attack, fake packets were injected with alternating 3.5 A and -3.5 A at a fixed rate to signal a fake beginning of a fault situation. When configuring the malware script, the two values will be written to an input file and then the malware changes the data fields of the sniffed packets by alternating between these values. The following algorithm was used to inject the fake packets and is written generically to be applicable for many systems:

```
f = fopen("FakeSV.txt", r+)
Generate Fake Values (inject +/- 3.5)
f.write(Fake Values)
SetSampleRate(pps)
while(!f.EndofFile){
    f.readline(n)
    sendpacket(rate, mod_pkt)
    n += 1 }
```

It is to be noted that the user of the malware tool is free to experiment with other algorithms or arithmetic functions for generating the fake values. The user can also experiment with the packet sending rate for a completely configurable data injection tool. The received and the forecasted samples from the IED were recorded in a log file and are plotted in Fig. 6. The blue waveform represents the SMV packets received by the IED, whereas the red waveform represents the forecasted values by the NN. The actual measurements (blue) were broadcasted at a rate of 4,800 Hz in accordance to the recommendations of the IEC 61850 standard, whereas the malware tool was transmitting its fake data at a rate of 48 Hz (1 fake packet for every 100 true packets). A stream of 9,400 real current samples was published and in the meantime, the malware published 80 manipulated packets. Out of the 80 attempts, 11 attempts (13.75%) were successful in attacking the NN and thus passing into the IEC protection logic false values reaching a maximum of around 4.5 A. This value is 1.5 times the rated value and was enough to issue a false positive and trigger a trip command.

The next data injection method is one that showed more promising results in successfully corrupting a network. This attack continuously replays values from a recoded log file that simulates the same waveform that the NN had been trained on for forecasting a fault. This method will increase the period of the injected values; however, the results show that the NN was not prepared to handle this kind of targeted attack. In Fig. 7, it is observed that the NN incorrectly forecasted the sampled values in two different locations within a window of 200 samples between sample 17,050 and sample 17,250. Any one of these incorrectly forecasted paths would trigger a false positive and cause service interruption. If the injected data results in a false negative for more than three AC cycles, permanent damage can be inflicted upon the power grid's equipment.

It is known that Layer 2 broadcast messages, such as SMV messages, cannot be blocked. An important observation from the results of these attacks is that when the IED received the fake packet, it was not able to receive the true message simultaneously. This is shown in packet number 17,166 in Fig. 8(a). As can be seen from Fig. 8(a), when the IED received the fake packet, the original data stream of legitimate SMV

Traffic	Captured
Packets	615021
Between first and last packet	63.692 sec
Avg. packets/sec	9656.212
Avg. packet size	95 bytes
Bytes	58441877
Avg. bytes/sec	917573.821
Avg. MBit/sec	7.341

Fig. 9. Malware Data Injeciton Statistics.

messages was shifted by 1 sample. This shift was accumulated as more fake SMV packets were injected. As can be seen in Fig. 8(b), the shift accumulation lead to the IED receiving SMV messages that are totally out of phase from the original data stream. Here again, the utilization of the developed malware exploited another vulnerability in the studied process bus which needs to be addressed.

It is also important to mention that the malware provides the user a means to control the rate of sending fake packets using the Scapy library function *pps* or packets per second. A test has been conducted to test the maximum speed at which the malware could broadcast fake SMV messages. Fig. 9 shows a summary of the statistics of the conducted experiment. For around one minute, the malware was sending at an average speed of 9,656 Hz, which is almost double than that set by IEC 61850 (4,800 Hz). These statistics are affected by the message length and the specifications of the machine hosting the malware. In this test, the malware was run on Linux machine with Intel i7 processor rated at 3.50 GHz with an average packet size of 95 bytes.

The previous results showed the effectiveness of the developed malware in testing the efficacy of the proposed predictive IDS. Using this developed malware as a training tool, not only can this NN be trained to detect several targeted attacks, but it can also be used to fine tune event thresholds to prevent service interruption during targeted attacks. Moreover, the developed malware tool can be used to benchmark other NN-related intrusion detection algorithms present in the literature. As mentioned earlier, the developed tool is configurable. Therefore, the user can control the type of attack by manually changing the value of the measurements, providing the tool with an arithmetic function to simulate certain scenarios or replay given values from a log file. Also, the user can adjust the rate of false data injection as desired. By these experimentations, the user can quantitatively analyze the performance of his or her IDS in terms of the ability of the tool to produce false positives and false negatives.

V. Conclusions

This paper developed the design and implementation of a targeted data injection attack that will simulate real AC waveforms in an attempt to interrupt power flow in a compromised power network. The targeted malware was developed in a configurable manner to allow the attacker to choose different methods of data injection. A predictive NN-IDS was then tested against the targeted malware to observe how it would handle the smart attack. The results of the experiment demonstrated that the NN is yet to be properly trained or fine-tuned enough to detect malicious measurements. Fake measurements could lead IEDs to issue trip signals that would result in service interruptions in a real system. The NN showed resistance to less sophisticated data injection methods; however, it is still vulnerable to malicious data injection methods. Also, the NN demonstrated very little resistance to the simulated AC waveform. In order to better prepare the NN, the developed malware will also be used as a training tool to identify attack signatures, and allow the user to tune the event thresholds that would result in controlling messages being sent to the IEDs. The malware is configurable, easy to understand, and is simple to use to train predictive intrusion detection systems.

References

[1] Tarek A. Youssef, Mohamad El Hariri, Nicole Bugay and O. A. Mohammed, "IEC 61850: Technology Standards and Cyber-Security Threats," *in the 16th IEEE International Conference on Environment and Electrical Engineering (EEEIC)*, Florence, Italy, 7-10 June, 2016.

[2] Mohamad El Hariri.; Tarek A. Youssef and Osama Mohammed, "On the Implementation of the IEC 61850 Standard: Will Different Manufacturer Devices Behave Similarly under Identical Conditions?" *Electronics* 2016, 5, 85.

[3] S. Sridhar and M. Govindarasu, "Model-Based Attack Detection and Mitigation for Automatic Generation Control," in IEEE Transactions on Smart Grid, vol. 5, no. 2, pp. 580-591, March 2014.

[4] Y. Liu, P. Ning, M. Reiter, False data injection attacks against state estimation in electric power grids, in: Proceedings of ACM Computer and Communication Security, 2009.

[5] Y. Yuan, Z. Li and K. Ren, "Modeling Load Redistribution Attacks in Power Systems," in IEEE Transactions on Smart Grid, vol. 2, no. 2, pp. 382-390, June 2011.

[6] R. Macwan et al., "Collaborative defense against data injection attack in IEC61850 based smart substations," 2016 IEEE Power and Energy Society General Meeting (PESGM), Boston, MA, 2016, pp. 1-5.

[7] M. T. A. Rashid, S. Yussof, Y. Yusoff and R. Ismail, "A review of security attacks on IEC61850 substation automation system network," *Proceedings of the 6th International Conference on Information Technology and Multimedia*, Putrajaya, 2014, pp. 5-10.

[8] U. K. Premaratne, J. Samarabandu, T. S. Sidhu, R. Beresh and J. C. Tan, "An Intrusion Detection System for IEC61850 Automated Substations," *in IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2376-2383, Oct. 2010.

[9] J. Hong, C. C. Liu and M. Govindarasu, "Integrated Anomaly Detection for Cyber Security of the Substations," *in IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1643-1653, July 2014.

[10] Nick Ismail, "How artificial intelligence is aiding the fight against cybercrime". Available Online: http://www.information-age.com/artificial-intelligence-aiding-fight-cybercrime-123461911/ (Accessed 04/07/2017).

[11] Mohamad El Hariri, Tarek A. Youssef, Hany F. Habib and Osama Mohammed, "Online False Data Detection and Lost Packet Forecasting System Using Time Series Neural Networks for IEC 61850 Sampled Measured Values" *In the proceeding of the IEEE Innovative Smart Grid Technologies (ISGT 2017)*, Washington DC, USA, April 23-26, 2017.

[12] Wen, J.; Hammond, C.; Udren, E.A. Wide-area Ethernet network configuration for system protection messaging. *In Proceedings of the 2012 65th Annual Conference for Protective Relay Engineers*, College Station, TX, USA, 2–5 April 2012; pp. 52–72.

[13] D. M. E. Ingram, P. Schaub, R. R. Taylor and D. A. Campbell, "Performance Analysis of IEC 61850 Sampled Value Process Bus Networks," *in IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1445-1454, Aug. 2013.

[14] IEC, Communication networks and systems in substation -- Specific communication service mapping. IEC 61850.8, 2008.