2023 IEEE 64th International Scientific Conference on Information Technology and Management Science of Riga Technical University (ITMS) | 979-8-3503-7029-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ITMS59786.2023.10317671

Man-in-the-Middle Attack Against a Network Control System: Practical Implementation and Detection

HamidReza Chavoshi
K. N. Toosi University of Technology
Tehran, Iran
hr.chavoshi@email.kntu.ac.ir

AmirHossein Salasi
K. N. Toosi University of Technology
Tehran, Iran
salasi77@email.kntu.ac.ir

Omid Payam
K. N. Toosi University of Technology
Tehran, Iran
omid_payam@email.kntu.ac.ir

Hamid Khaloozadeh
Industrial Control Center of Excellence, Systems and Control Engineering
K. N. Toosi University of Technology
Tehran, Iran
h_khaloozadeh@kntu.ac.ir

***Abstract*—A Man-in-the-Middle (MITM) attack is a cyber-attack in which the attacker covertly intercepts and passes messages between two parties who mistakenly think they are communicating directly. However, in reality, the attacker intercepts data transfers between a client and a server by deceiving both parties. While the attack occurs, the data is secretly manipulated by inserting false information. This article explores how to create and use MITM attacks in a liquid-level networked control system. The essential tools to execute the attack include Ettercap and Wireshark software applications. Ettercap is a tool for capturing packets, allowing real-time redirection and modification of data streams by writing the packets back onto the network. Wireshark is a flexible network protocol analyzer used to analyze data packets of the networked control system. After implementing the MITM attack on the cyber-physical system, system data was collected and labeled to detect MITM attacks by leveraging machine learning classification algorithms.**

Keywords—cyber-attack detection, ettercap, machine learning, man-in-the-middle attack, networked control system

I. INTRODUCTION

Given the growing dependence of modern society on computers, digital systems, and automation, there are groups seeking to exploit these technologies for their gains by engaging in criminal activities such as cyber-attacks. These attacks target the security of these systems, intending to achieve diverse objectives. As a result, the significance of ensuring cybersecurity cannot be ignored. Implementing cybersecurity measures across various sectors of society, particularly for safe-guarding sensitive information, critical industries, and vital infrastructure, is paramount. Each government is susceptible to potential crisis spanning Network Control Systems (NCS) and Industrial Control Systems (ICS) and even on a grander scale, including industrial control hubs, power plants, transportation grids, and beyond.

NCS is formed by a physical system with its behavior regulated by a digital controller. This controller interacts with the physical processes through a communication network, effectively connecting the digital realm with the physical domain. This fusion of digital controllers and physical processes using communication networks is intended to enhance the operational and managerial functionalities of these systems.[1]

NCS has fundamentally altered how we manage and control numerous industrial processes. The network infrastructure plays a crucial role in NCS, serving as the fundamental framework that enables effortless communication among sensors, actuators, and the controller. Transmission Control Protocol/Internet Protocol (TCP/IP) is essential here. TCP/IP is a suite of communication protocols that control how data is transmitted, routed, and received within networks. The TCP/IP protocol operates through a layered structure. The network interface layer receives and addresses data in its physical form. The Internet layer handles addressing, routing, and path determination. The transport layer ensures smooth data transmission, and the application layer allows users to interact with the network and exchange information effectively. This layered approach enables efficient communication within the network.[2] The data packet of TCP/IP within the Ethernet frame is shown in Fig. 1.[3]

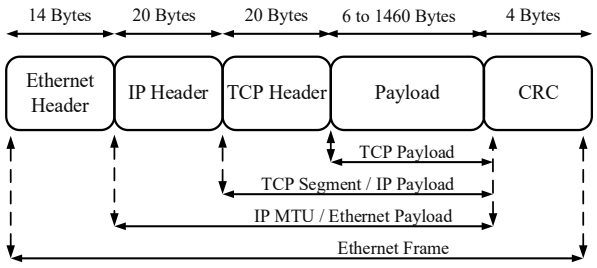

Fig. 1. Bytes lengths and headers of TCP/IP within the Ethernet frame.[3]

Although NCS and TCP/IP Provide many advantages, they pose some security risks. The reliance on networks makes

NCS vulnerable to cyber-attacks. The security considerations of TCP/IP networks generate challenges, which can be a source of threats. Some critical security disadvantages related to TCP/IP networks include Packet Sniffing and Eavesdropping, IP Spoofing, Denial of Service (DoS) attacks, and man-in-the-middle (MITM) attacks.[4]

In this article, we implemented the MITM attack on our cyber-physical system. MITM attack is a form of unauthorized intrusion where an attacker covertly intercepts and monitors unencrypted data exchanged between network devices and the targeted computers.[5] MITM attacks can be executed using various methods: ARP cache poisoning, DNS spoofing, SSL Hijacking, and Session hijacking, including side-jacking, evil twin, and sniffing.[6]

In this article, the ARP poisoning method is used. Address Resolution Protocol (ARP) and MITM attacks are closely intertwined, as ARP is a pivotal enabler for MITM attacks. ARP enables the discovery of the MAC address of a node through the known IP address, caching the information in the table of ARP. ARP poisoning seeks to compromise the ARP table of the victim by associating the IP address of the counterpart with the MAC address of the attacker, which allows the attacker to execute a MITM attack by sending malicious replies to ARP.[7]

Moreover, following an attack on the system, there arises a requirement for algorithms capable of comparing the regular behavior of the system, including the volume and nature of transmitted and received information, with its behavior during the attack. These algorithms should then alert the user to the possibility of an ongoing attack. A diverse range of machine learning algorithms can be found in this domain, including supervised, unsupervised, semi-supervised, and reinforcement learning.[8] The primary purpose of this article is to explore and implement MITM attacks, alongside using machine learning classification algorithms to detect these attacks and potential disruptions on a level liquid networked control system.

II. System Introduction

A. Cyber-Physical System Components

The employed system could be regarded as a cyber-physical system. A liquid-level control system is implemented with a PI controller. The physical section of our cyber-physical system is a liquid-level control plant. The system comprises a cylindrical tank with a height of 30 cm and a radius of 4 cm. It also includes a pump to transfer liquid to the tank with a variable input voltage ranging from 0.2 V to 3 V and an adjustable valve for draining the liquid.

A 5 kg load cell measures the liquid level inside the tank. This sensor converts the force exerted by the liquid due to its weight into a corresponding voltage signal. As the liquid level increases, the sensor voltage output also rises. The output ranges from 0 V to 10 V.

A physical interface panel facilitates communication between the control system and its data logger. The data logger employs a Wi-Fi module to establish communication with a computer. It has two digital-to-analog converters (DAC), four analog-to-digital converters (ADC), eight digital inputs, and eight digital outputs. Fig. 2 shows these parts of the system.

Fig. 2. Liquid-level control system, the interface panel and, data logger.

The liquid-level system is controlled using a networked control system (NCS), which falls under a broader category of control systems. NCSs involve closing the control loop through a serial communication network.[9] Our networked control system can engage in communication with other computers via a Wi-Fi network. This interaction is made possible through the TCP/IP protocol, which facilitates data exchange across an Ethernet network. It is like a network where Several devices are linked to a central router, and communication between them is facilitated using the TCP/IP protocol. In this setup, every device in the network is allocated a distinct IP address, which allows them to communicate with other devices within the network using the TCP/IP. The TCP/IP protocol provides a standardized set of rules and procedures for transmitting, routing, and receiving data across the network. Communication systems with a central router are prevalent in numerous networking systems, including home networks, corporate networks, and the Internet. TCP/IP is now the prevailing protocol for data communication in contemporary networks.

B. System Control Structure

The PI controller is chosen for simplicity, and its parameters, including proportional and integral gains, are determined. The digital PI controller can be represented in a general form, which is as follows:

$$C(z) = K_p + K_i \frac{T_s}{z-1} \tag{1}$$

It is necessary to have the system model to determine the controller coefficients. We use the model identified for this system in [10] with sample time Ts=0.2 second.

$$G_1(z^{-1}) = \frac{0.03217 z^{-5}}{1 - 0.9 z^{-1} - 0.09 z^{-2}} \tag{2}$$

This model is identified using the Least Square (LS) method. A Second-Order plus Time-Delay (SOPTD) model for the system has been considered. Generally, the liquid-level control system exhibits characteristics of a first-order system. However, our system has a limited and low output flow rate, causing it to behave more like an integrator. Consequently, an additional pole is incorporated into the system model to

enhance its alignment with the actual system dynamics. Furthermore, the model accounts for delays stemming from the inherent traits of the system, such as the delay from tubes connecting the pump to the tank and the time delay resulting from the computer Wi-Fi connection to the data logger. Finally, $K_p = 6.5$ and $K_i = 0.3$ were selected by fine-tuning the Ziegler-Nichols PI controller design method, so

$$C(z) = K_p + K_i \frac{Ts}{z-1} = 6.5 + 0.3\frac{0.2}{z-1} = \frac{6.5z - 6.44}{z-1} \quad (3)$$

The control loop includes transforming the analog output from the load cell into digital values using an analog-to-digital converter (ADC). These digital values are then transmitted to a computer through a Wi-Fi network. When the desired target value is applied to the system, the computed control signal is transmitted back to the data logger using the same Wi-Fi network. A digital-to-analog converter (DAC) is utilized to convert the digital control signal into analog values to activate the pump. The intended control algorithms are implemented through MATLAB/Simulink, and real-time execution is facilitated with a real-time synchronization block. The overview of the system and the part of the system where the MITM attack occurs is illustrated in Fig. 3.

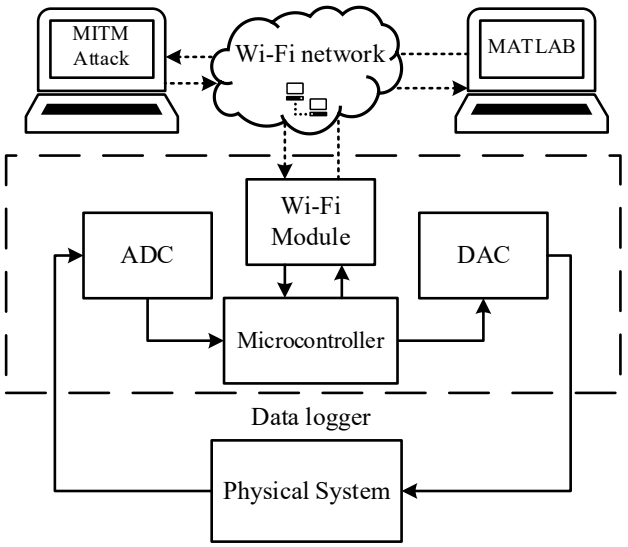

Fig. 3. Overview of the system and the part of the system where the MITM attack takes place.

III. Implementation of MITM Attack

A. Software Used

We have used Ettercap and Wireshark software to execute the MITM attack. Ettercap is a packet capture tool that redirects and modifies data streams in real-time by writing packets back onto the network. Ettercap is utilized for protocol analysis to analyze network traffic and identify applications responsible for generating the most traffic. Its main applications involve carrying out MITM attacks through ARP poisoning. Ettercap is mainly intended for Linux and Unix-like operating systems and is compatible with various Linux distributions. Here Ubuntu operating system has used. Ettercap attack methodology primarily relies on the ARP poisoning technique. The attacker associates their MAC address with the legitimate IP address by consistently sending fake ARP messages. As a result, any data the sender intends to send to the legitimate node gets rerouted to the attacker's node instead. Following successful network infiltration, the next step involves using the Ettercap filter tool to alter the targeted data. This task entails altering genuine information and manipulating the data bits that contain this information. These actions are enabled through the filter application in this stage.

Wireshark is utilized to observe and analyze packets transmitted and received within the network. Wireshark, a flexible network protocol analyzer, plays a crucial role in capturing and analyzing network traffic, supporting network administrators in resolving network-related problems. While Ettercap is geared towards actively intercepting and manipulating network traffic, Wireshark is predominantly employed for passive network traffic analysis. Together, they can monitor, analyze, and potentially manipulate network communication for various purposes, such as security testing, troubleshooting, and network optimization. The collaboration between Wireshark and Ettercap proves highly advantageous for network analysis and security evaluations. After all, by using Wireshark, the packets transmitted within the network are examined to locate the bytes that hold the targeted information. In this article, we used Wireshark to monitor the packets transmitted from the computer to the datalogger and vice versa. The purpose of implementing the MITM attack in this article is to disrupt the control signal and the output of the sensor.

B. Manipulating Control Signal

The controller signal is created in MATLAB software, and the client computer sends it to the data logger through the Wi-Fi network. Fig. 4 shows a photo of a packet sent from the computer to the data logger in Wireshark software. It can be seen that a packet consists of different parts, such as Ethernet, IP, and TCP headers. The value of the control signal is placed in the data part of the packet. The data part, sent from the client to the data logger, consists of 15 bytes; each is intended to send information related to a specific part of the data logger.

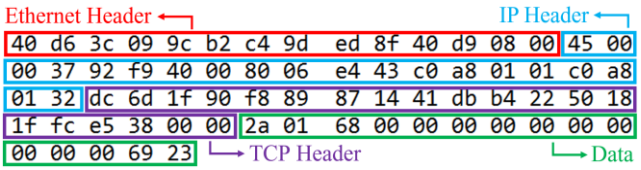

Fig. 4. Send packet from the client computer to the datalogger (control command).

In data bytes of Fig. 4, byte 1 equals the value of 42 or (2A hex) and indicates the beginning of the sent data packet. Also, byte 14 stores the result of the XOR of bytes 2 to 13 to check the correctness of the transmitted packet. Finally, byte 15 always contains the value 35 (23 hex), which indicates the end of the data part of the transmitted packet. The rest of the data bytes, which are the information of data logger ports, have been shown in Table I.

TABLE I. DATA BYTES OF THE DATA LOGGER OUTPUT PORTS IN THE CONTROL COMMAND DATA PACKET.

Data Logger Port	Data Byte
DAC1	2, 3
DAC2	4, 5
PWM1	6, 7, 8
PWM2	9, 10, 11
Digital Output (1-8)	12
Buzzer	13

The pump is attached to DAC1, so we need to alter the data of 2 and 3 data bytes to manipulate the control signal and prevent the correct control signal from applying to the pump. For changing the data via the filter option, we need to identify bytes 2 and 3 of the data packet and then alter the data of these bytes. First, we identified the intended packet with an if condition in the used filter, which identifies the send packets with their protocol and source port (ip.proto= TCP and tcp.src=8080). After the intended data packet is identified with the help of the "DATA.data" command, we change the desired bytes. (DATA.data + X == "\xYY", where X is the offset that specified the pointer location to the data packet and YY is the value that should be placed in the desired byte in hex format). We execute this filter when the system gets to its steady state to disrupt the system setpoint tracking.

C. Manipulating Sensor Output

The process of sending a packet from the data logger to the computer slightly differs from transmission from the computer to the data logger. In this route, there is a buffer that consists of some packets (each packet has 15 bytes) and sends the buffer to the client computer. The buffer sends from the data logger to the client computer through the Wi-Fi network. Fig. 5 shows a send buffer from the data logger to the computer in Wireshark software. It can be seen that a packet consists of different parts, such as Ethernet, IP, and TCP headers. The value of the sensor is placed in the data part of the packet. Whenever the client computer receives the buffer, it checks if there is a new data packet in the buffer, then it reads the last 15 bytes of the buffer related to the new packet.

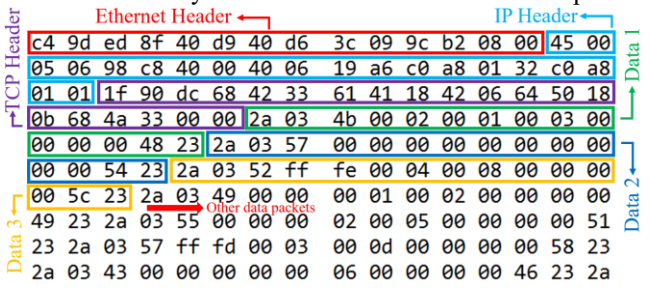

Fig. 5. Send packet from the data logger to the client computer (sensor output).

In data bytes of Fig. 5, byte 1 equals the value of 42 or (2A hex) and indicates the beginning of the sent data packet. Also, byte 14 stores the result of the XOR of bytes 2 to 13 to check the correctness of the transmitted packet. Finally, byte 15 always contains the value 35 (23 hex), which indicates the end of the data part of the transmitted packet. The bytes that are related to the ADC ports of the datalogger have been shown in Table II. The rest of the Bytes are related to the digital input and other information about the data logger.

TABLE II. DATA BYTES OF THE DATA LOGGER INPUT PORTS IN THE SENSOR DATA PACKET.

Data Logger Port	Data Byte
ADC1	2, 3
ADC2	4, 5
ADC3	6, 7
ADC4	8, 9

The Loadcell sensor is attached to ADC1, and for executing the attack, we need to alter the data of 2 and 3 data bytes to manipulate the sensor signal. The steps of executing the attack in this section are similar to the attack on the control signal. However, in this section, we must enter the destination port in the IF condition to select the packets sent from the data logger to the computer (tcp.dst=8080). In the same way, we execute the filter when the system gets to its steady state to disrupt the system setpoint tracking.

IV. ATTACK DETECTION BY MACHINE LEARNING ALGORITHMS

A. Dataset Construction

Four scenarios which are: normal conditions, MITM attack on the control signal, MITM attack on sensor output, and disturbance, have been considered to create the data set. Fig. 6 illustrates the system response and control signal in normal conditions. It can be seen that the system achieves good setpoint tracking, and the controller has acceptable performance.

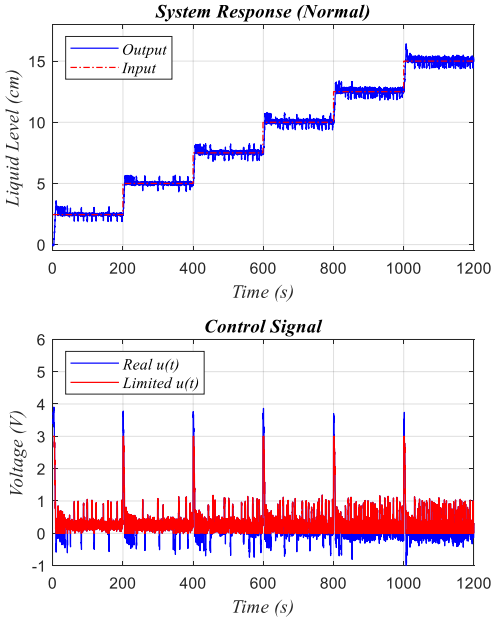

Fig. 6. System response and control signal in normal condition.

The system response and control signal when an MITM attack on the control signal has been performed is shown in Fig. 7. The attack on the control signal will occur whenever the system output reaches its steady state condition. The liquid level exceeds the desired level due to the execution of the attack. As long as the attack progresses, the controller

attempts to control the liquid level by generating a corrective control signal (negative control signal). However, this signal will not be applied to the pump.

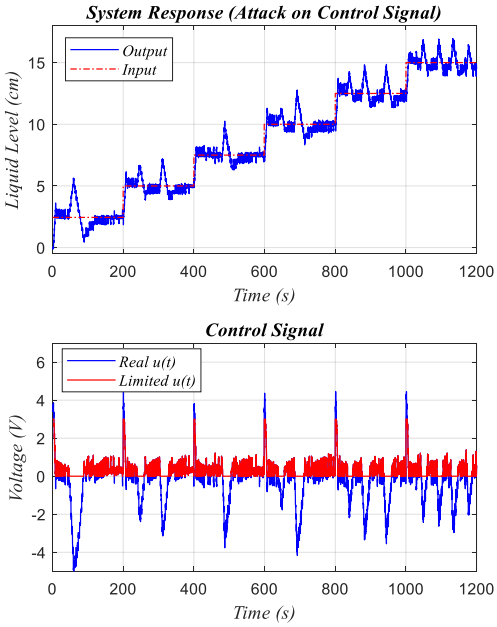

Fig. 7. System response and control signal when MITM attack on the control signal has been performed.

Fig. 8 shows the system response and control signal when an MITM attack on the sensor output has been performed. The attack on the sensor output will occur whenever the system output reaches its steady state condition. The output values of the sensor decrease or increase by 0.3 V or 0.4 V when an attack occurs, after which the controller reacts quickly and brings the output voltage back to the reference input level, but the liquid level of the physical system is lower or higher than the desired level.

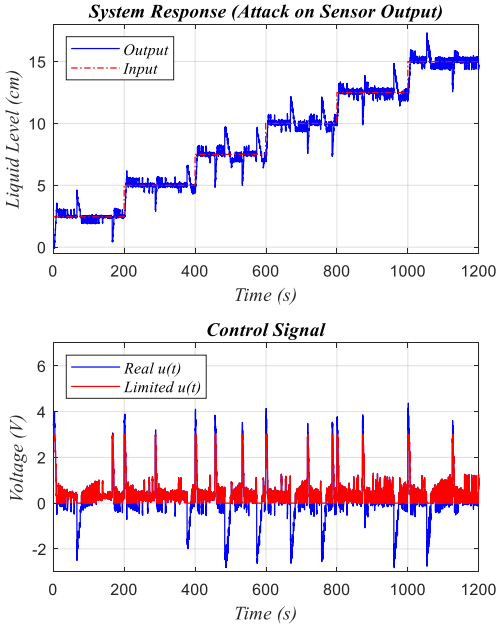

Fig. 8. System response and control signal when MITM attack on the sensor output has been performed.

The system response and control signal when the disturbance has been applied is shown in Fig. 9. From 50 to 350 seconds and 640 to 930 seconds; we added 5 V disturbance to the control signal. Because the PI controller is robust, it rejects this disturbance, and after passing through the transient state of entering the disturbance into the system, it performs well in setpoint tracking.

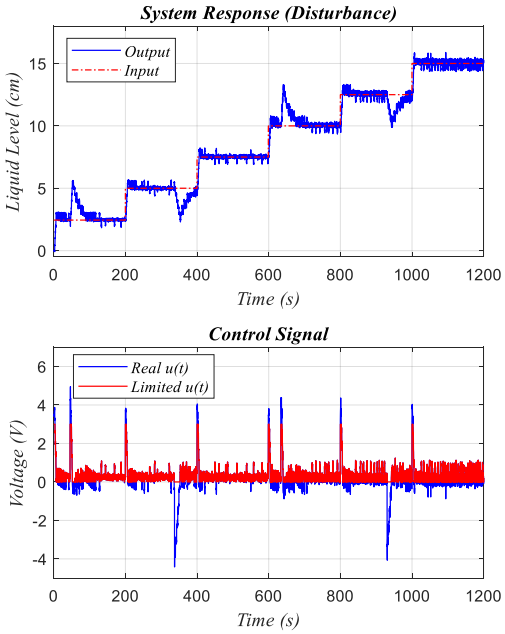

Fig. 9. System response and control signal when the disturbance has been applied.

According to the explanations, the behavioral difference between these four modes is explicit. Next, we will classify these four modes with the help of machine learning classification algorithms.

B. Implementation of Machine Learning Algorithms

In this section, we perform the classification using the four data categories explained in the previous section, which include four primary columns (reference, control signal, limited control signal, and sensor output). Each run contains 6010 samples. After labeling the data, we used the feature extraction method to achieve better and more accurate calcification. The underlying patterns of raw data can be identified with the help of Feature extraction, and this method could improve model accuracy.[11] In the feature extraction method, we extracted data features such as standard deviation, root mean square, maximum, peak to peak, mean, absolute mean, skewness, kurtosis, crest factor, and absolute Fast Fourier Transform (FFT). After that, data pre-processing tasks such as normalization and separation of test and train with a ratio of 80 to 20 were performed. Also, to prevent overfitting of the models, we implemented the K-fold method with K = 5. To classify the data, we used five standard machine learning methods. The accuracy of each method is shown in Fig. 10. As evident in Fig. 10. The Random Forest method has been able to classify the data more accurately.

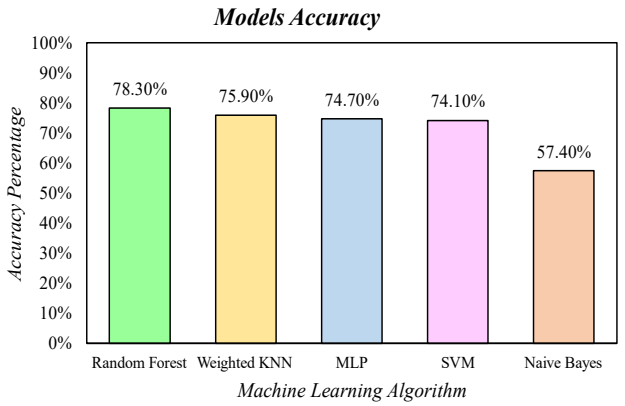

Fig. 10. Accuracy of 5 trained machine learning models.

The confusion matrix of the random forest method, which is the most accurate among the implemented methods, is shown in Fig. 11. As can be seen in Fig. 11, the method has high accuracy (the main diameter of the matrix), which shows that the classification is done correctly. Since the controller used in this experiment is PI and can reject disturbance. Also, the PI controller rejects the sensor offset when the sensor value is attacked. As a result, we have more errors in classifying normal, disturbance, and MITM attack on sensor classes. More parameters, such as TCP/IP network information or more advanced classification methods, should be used to reduce classification errors.

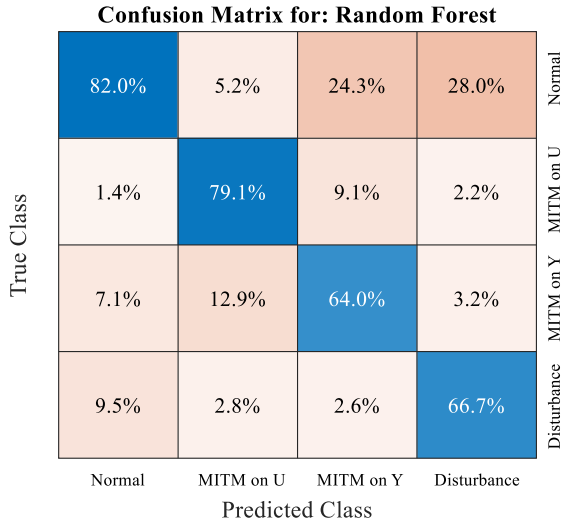

Fig. 11. Confusion matrix of the random forest method

V. Conclusion

This article discusses a weakness in the TCP/IP protocol and an approach to infiltrating a network that relies on this protocol. We designed and executed MITM attacks in the initial step, enabling us to intercept the information. We accessed and modified information within a Network Control System (NCS) that depends on the TCP/IP protocol for data exchange, subsequently influencing the control performance of the system. The tools required for this endeavor included the software Ettercap and Wireshark, and all the steps were executed within the Linux operating system environment. In the next step, we conducted attack detection and distinguished between the normal operation of the system, disturbance, and two distinct attack modes using machine learning algorithms. The results differed based on the employed machine learning classification algorithms. As described in the article, the random forest method algorithm demonstrated superior efficiency in classifying classes.

References

[1] A. O. de Sá, L. F. da Costa Carmo, and R. Machado, "A controller design for mitigation of passive system identification attacks in networked control systems," *Journal of Internet Services and Applications,* vol. 9, no. 1, pp. 1-19, 2018.

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

[3] G. Sanchez, "Man-in-the-middle attack against Modbus TCP illustrated with Wireshark," *SANS Inst., Tech. Rep.,* pp. 1-26, 2017.

[4] I. H. Sarker, A. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *Journal of Big data,* vol. 7, pp. 1-29, 2020.

[5] E. Ahmet, G. KALKANCI, D. Mehmet, S. CİHANGİR, and Z. UYSAL, "A Hidden Hazard: Man-In-The-Middle Attack in Networks," *Computer Science,* vol. 4, no. 2, pp. 96-116.

[6] Z. Cekerevac, Z. Dvorak, L. Prigoda, and P. Cekerevac, "Internet of things and the man-in-the-middle attacks–security and economic risks," *MEST Journal,* vol. 5, no. 2, pp. 15-25, 2017.

[7] S. Calvo, "Exploiting virtual network for CPS security analysis. Secure Water Treatment simulation with MiniCPS and GNS3," Politecnico di Torino, 2022.

[8] I. H. Sarker, "Machine learning: Algorithms, real-world applications and research directions," *SN computer science,* vol. 2, no. 3, p. 160, 2021.

[9] G. C. Walsh, O. Beldiman, and L. Bushnell, "Asymptotic behavior of networked control systems," in *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*, 1999, vol. 2: IEEE, pp. 1448-1453.

[10] H. Chavoshi, A. Salasi, O. Payam, and H. Khaloozadeh, "Experimental Comparison of STR and PI Controllers on a Nonlinear Liquid-Level Networked Control System," in *2023 15th International Conference on Electronics, Computers and Artificial Intelligence (ECAI)*, 2023: IEEE, pp. 1-8.

[11] R. Magar, L. Ghule, J. Li, Y. Zhao, and A. B. Farimani, "FaultNet: a deep convolutional neural network for bearing fault classification," *IEEE access,* vol. 9, pp. 25189-25199, 2021.