applied sciences

MDPI

Article

Mitigation against DDoS Attacks on an IoT-Based Production Line Using Machine Learning

Ladislav Huraj [1,*], Tibor Horak [2], Peter Strelec [2] and Pavol Tanuska [2]

[1] Department of Applied Informatics, University of Ss. Cyril and Methodius, 91701 Trnava, Slovakia
[2] Institute of Applied Informatics, Automation and Mechatronics, Faculty of Materials Science and Technology in Trnava, Slovak University of Technology in Bratislava, 91724 Trnava, Slovakia; tibor.horak@stuba.sk (T.H.); peter.strelec@stuba.sk (P.S.); pavol.tanuska@stuba.sk (P.T.)
* Correspondence: ladislav.huraj@ucm.sk

Abstract: Industry 4.0 collects, exchanges, and analyzes data during the production process to increase production efficiency. Internet of Things (IoT) devices are among the basic technologies used for this purpose. However, the integration of IoT technology into the industrial environment faces new security challenges that need to be addressed. This is also true for a production line. The production line is a basic element of industrial production and integrating IoT equipment allows one to streamline the production process and thus reduce costs. On the other hand, IoT integration opens the way for network cyberattacks. One possible cyberattack is the increasingly widely used distributed denial-of-service attack. This article presents a case study that demonstrates the devastating effects of a DDOS attack on a real IoT-based production line and the entire production process. The emphasis was mainly on the integration of IoT devices, which could potentially be misused to run DDoS. Next, the verification of the proposed solution is described, which proves that it is possible to use the sampled flow (sFlow) stream to detect and protect against DDoS attacks on the running production line during the production process.

Keywords: production line; IoT devices; DDoS attacks; machine learning

Citation: Huraj, L.; Horak, T.; Strelec, P.; Tanuska, P. Mitigation against DDoS Attacks on an IoT-Based Production Line Using Machine Learning. *Appl. Sci.* **2021**, *11*, 1847. https://doi.org/10.3390/app11041847

Academic Editor: Giorgio Di Natale

Received: 2 February 2021
Accepted: 17 February 2021
Published: 19 February 2021

1. Introduction

Cyberattacks occur in all areas of society and distributed denial-of-service (DDoS) attacks are among the most popular kinds and can overwhelm a network to the point of inoperability [1]. In a DDoS attack, the target machine or network is flooded with a number of redundant requests from a number of different sources. A DDoS attack can be a worry for various services related to digital infrastructure, such as banks, cloud services, Wikipedia, and political parties, but also households, hospitals, and industrial plants. Cybersecurity has become an increasing problem in industry as well.

Technological developments in recent years have made it possible to connect a variety of devices to computer networks, which brings various benefits to users. However, with the rise of the technologies involved, the number of cyberattacks is also increasing, using more sophisticated means to incorrectly access sensitive information and to extort money or the already mentioned interruption of services. One such technology is the Internet of Things (IoT) [2].

The concept of the Internet of Things includes various devices, sensors, objects, and intelligent nodes that are able to function autonomously and communicate with each other without human intervention. Such IoT devices are able to provide a number of useful services and, thanks to sensors and actuators, provide various data in real-time. In many cases, however, devices in the field of the Internet of Things, in particular, contain various software bugs brought in from the factory that make them vulnerable. Such vulnerabilities often allow attackers to perform various cyberattacks and compromise the security of the environment in which IoT devices are located [3].

IoT devices have also found application in "Industry 4.0," which seeks to increase efficiency in the industry through the collection, exchange, and analysis of information throughout the product life cycle. Intelligent production on the production line uses IoT as one of the important basic technologies and accelerates the modernization of traditional industrial production lines [4,5]. However, applying IoT devices to the production line can lead to various cyberattacks, including a DDoS attack.

Several defense mechanisms have been proposed in the past against DDoS attacks in IoT networks. They can be divided into two basic groups: traditional DDoS defenses and IoT-specific DDoS defenses. They differ in terms of location and complexity. While traditional DDoS defenses are applied to the target server and are essentially homogeneous, IoT-specific DDoS defenses are applied to IoT devices and are more complex, reflecting the heterogeneity of IoT devices. In both cases, detection techniques are used to detect abnormal activities in the network or host. The taxonomy of DDoS defense mechanisms in the IoT network is described in detail in [1].

One of the possible techniques used in defense against DDoS attacks is machine learning. Network and system security have long used machine learning to detect malware, anomalies, and intrusions [6].

Academic institutions and industry segments are trying to find the optimal solution to the problem of DDoS attacks. Finding a balance between academic design and industry practice to combat DDoS is a major challenge. Although both sectors are making great efforts to combat DDoS attacks, DDoS incidents occur every day, emphasizing that the problem is not resolved [7].

The production line is a specific production environment and requires a special approach to the problems of DDoS attacks and countermeasures when implementing cybersecurity, especially if it has integrated IoT devices.

1.1. Related Works

The need to use industrial IoT to monitor the production line is described in detail in [8]. It is emphasized that, in order to achieve an intelligent production, the production factors have to have the ability to self-identify, intelligently perceive, make independent decisions, and learn knowledge, which is supported by industrial IoT technology. However, further research is still needed in the area of IoT manufacturing, artificial intelligence algorithms, and machine learning.

Continuous efforts to increase the productivity of production lines affect security, and the implementation of intelligent manufacturing processes can also affect security in a negative way. The principles, overview, and threats of some cyberattacks on the production line, such as zero-day attacks, false data injection attacks, or denial of service attacks, are described in [9]. In 2017, Knudsen et al. [10] tested the production line for some vulnerabilities and the main shortcomings identified were the use of default passwords for various services and insufficient SSH encryption.

The research in [11] shows the link between security and safety of the production line. A cyberattack on a production line security is based on intercepting and modifying messages between control devices, and can lead to intentional misalignment of the production robot and so result in injury to production operators.

Tuptuk and Hailes [12] discussed significant challenges to the security of smart manufacturing systems; DDoS attacks are one of the serious threats to be considered when designing security. Prinsloo et al., in [9], draw attention to the fact that a DDoS attack in such an environment could have catastrophic consequences for production lines. This equipment is critical for efficient and safe operation, as a number of industrial sensors and devices of new-generation control systems have the ability to connect directly to computer networks and the Internet.

However, specific DDoS attacks and their effects on the production line, are not described thoroughly in the literature. It is assumed that DDoS attacks have a negative impact on the production line and thus on the production process, but there are no documented

real experiments. If tests of countermeasures against DDoS attacks are verified, they are usually only indirectly performed on simulation models of the production line or existing datasets, such as [10,13,14].

Some components of the production line state that they are able to operate even in the event of network anomalies. However, no further detail or conditions of anomalies are given. Real experiments performed on a running production line show that this is not always the case and seldom when the individual components need to communicate with each other. These facts cannot be revealed during the simulation of the production line and will only be revealed by testing on a real production line.

1.2. Contributions

The production line is a specific environment and the application of IoT equipment to the production line can increase the efficiency, but also the vulnerability of the entire production process. The aim of the present study was to show the results achieved during a DDoS attack on the real environment of the production line. The focus was primarily on the integration of IoT devices that could potentially be misused to run DDoS or distributed reflective DoS (DRDoS).

The aim of this case study was to show what vulnerability the introduction of IoT devices creates for the production line in terms of DDoS attacks. The challenge was to understand the threat of a DDoS attack for a production line and to look at this threat in a real production process. In addition, the aim was to show whether it is possible to protect the production line from such a massive attack by using machine learning and commonly available tools for network traffic analysis and evaluation. The results of the experiments performed indicate that the processing of streamed data and sFlow appear to be suitable tools for securing the production line against DDoS attacks.

The research results are widely applicable in the cybersecurity field in the industrial Internet of Things and can help increase the security of a production line.

2. Materials and Methods

The production line produces a large amount of network communication, which is why real-time analysis is difficult to implement or computationally intensive in such an environment.

An effort was made in order to analyze a network communication, to indicate anomalies or network collisions, and, at the same time, to prevent potential security attacks. Typical solutions, where firewall rules, as well as system switch settings, are statically created and deployed during the production line, involve the risk of inflexibility and complications of possible integration improvements or additions to new threads required for Industry 4.0.

This article explores the possibility of an adaptive security solution and innovative monitoring and evaluation of network communication. The required architecture should be sufficiently vertical and horizontally scalable and, at the same time, the monitored communication should not overwhelm or limit the communication that results from the production process.

2.1. IoT-Based Production Line

The production line is the basic cell in the manufacturing industry and usually depends on the stability of the production process of factories, companies, and industrial chains. IoT production is a new production mode that combines IoT technology with production technology to carry out the production and the service process, as well as the dynamic perception, intelligent processing, and optimal management of production resources and information resources throughout the product life cycle [15]. The IoT-based production line uses IoT technology to monitor objects and processes interactively in real-time. For this purpose, it uses sensors, actuators, RFID, image recognition, and other methods.

This case study handles a real production line (Figure 1). The production line is used by leading industrial producers of beverages and food and also by the chemical industry.

Figure 1. The real production line tested in the case study.

The production line consists of five zones. Every zone forms a part of the production process. Zones are controlled by Siemens PLC and there are integrated IoT devices. They have the task of performing operations from the simplest to the most complex. In the first zone, materials are collected and dosed. In the second zone, after mixing and dosing the materials, the production process continues by filling the material into bottles, which are closed by a cap on a rotating table. The bottles are sorted using an RFID reader and then shipped by a transport system. The transport system loads the filled bottles into crates. The high-speed camera checks the correct filling of the containers with bottles. The third zone is used for dosing semifinished products for the production of bottle closures. The fourth zone is used to transport crates between individual stations using a conveyor belt. The fifth zone is used for unloading bottles from overflows and their subsequent transport to the unpacking station. It is a recycling station where the robot is located, and the pump is used to empty the unnecessary liquid contents of the bottles. The individual zones communicate via an Ethernet network [16].

The production line, for efficient production, is equipped with an MES system based on the Wondervare platform. This system monitors and documents the production process. Based on this information, manufacturers can adjust the production conditions in order to improve the efficiency of the production process. Additionally, it includes the Historian Server subsystem, which serves as a high-speed database of historized information in real-time [17].

2.2. DDoS Attacks

The distributed architecture of IoT networks is their vulnerability. Any IoT device located in the environment is a possible point of failure and can be exploited for cyberattacks, including DDoS attacks [18].

Attacks on IoT devices are growing more quickly than the number of connected devices. Compromised devices usually become part of a botnet, which can then be exploited to trigger a DDoS or another type of attack. The main threat to the network and network services is harmful traffic from IoT devices that are part of botnets [19]. To perform a DDoS attack, the IoT device does not even have to be compromised and can become part of a DDoS attack.

In general, the role of a distributed denial of service attack is to perform a coordinated attack that affects the availability of the victim's services. Many distributed nodes are involved in carrying out an attack, and the target network or machine is flooded with a number of requests, which reduces the performance of the system or the entire network. Usually, a DDoS attack exploits protocols such as UDP, TCP, domain name system (DNS), or ICMP and various types of DDoS attacks such as SYN flood, ICMP flood, or HTTP Get Flood are based on this [20].

Distributed reflective denial of service (DRDoS) is one type of DDoS attack. To overwhelm a victim machine or network, DRDoS and DDoS attacks use a flood of connection requests that run from multiple IP addresses every second. Conversely, DRDoS attacks do not start attacks from compromised computers, but instead often use legitimate IoT devices to "reflect" or enhance traffic to the victim using modified packets with spoofed IP addresses. An attack is reflective when the attacker, to hide their identity, makes use of a potentially legitimate third party to resend attack traffic to the victim [21].

The subject of the tested network infrastructure is the addition of IoT devices to the production line; IoT devices have made the production line more vulnerable to DDoS attacks. Significant problems in the production process were caused by direct DDoS attacks as well as DRDoS attacks, which were conducted through IoT devices directly on the components of the production line and thus disrupted the production process.

2.3. A Real Network Infrastructure

As is shown in Figure 2, a real network infrastructure was used to perform the vulnerability testing. This infrastructure consists of a real production line. The production line was enriched with IoT devices that allow communication from the external Internet. Extending communication to an external network using IoT devices makes the infrastructure more vulnerable to DDoS attacks. The first IoT device tested was a thermostat. By using a mobile application, the thermostat enables efficient regulation of the temperature in the room where the production line is located. The second IoT device was the Fibaro IoT security system. This makes it possible to protect the production line from fire, floods, and the entry of unwanted people into the workplace by means of IoT sensors. The IoT Fibaro device also allows one to remotely inform the operator about a possible problem using a mobile application; based on this information, the operator can act in time to prevent damage. These IoT devices, together with the production line and the Historian server, are connected via switches. Together they form a common network infrastructure.

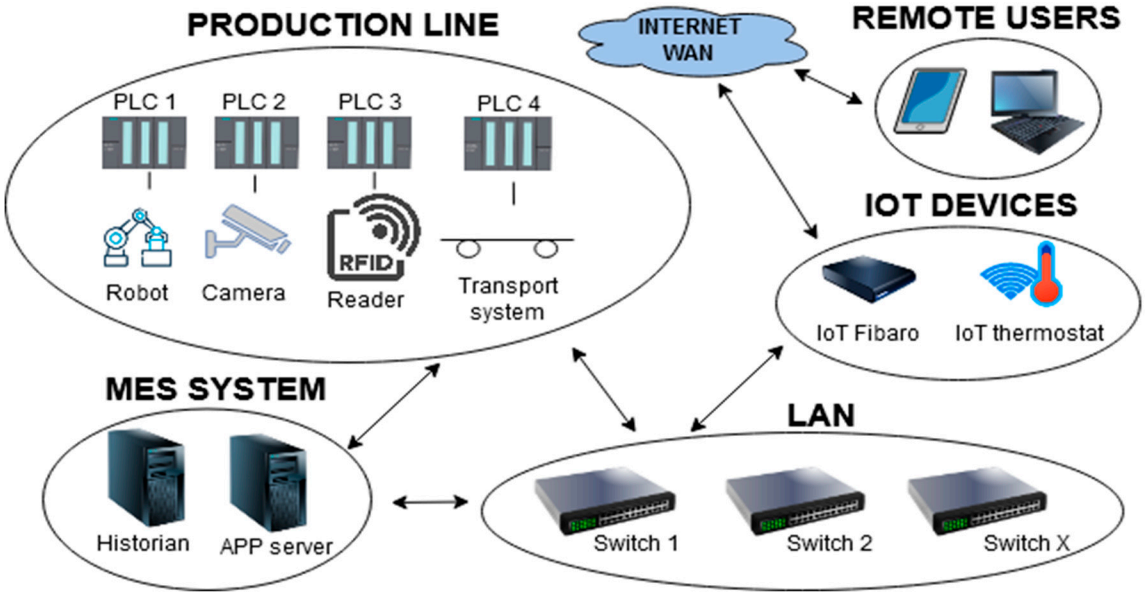

Figure 2. Network infrastructure tested in the case study.

2.4. Data Acquisition

A production line's generated data are acquired and used for the training and validation of a neural network. It is advantageous if the net flow is supported directly by a switch that forwards communication to a separate LAN in order to avoid overflow of the LAN industry. A basic test was performed for a summary of packets, the latter being recorded with a resolution of seconds. That is why it is not a full cap record.

A full data flow could cause a capacity problem for a large production line, and the implementation costs could easily exceed the expected benefits of the IoT device. Therefore, other data acquisition options for network communication were analyzed.

The flow variant of the sFlow protocol was selected: sampled flow. Sampled flow sFlow is different from net flow and does not work with a full set of data but only with samples. sFlow is considered a recognized industry standard as well. The protocol is supported and implemented in a wide range of L2 layer network switches and routers. The output of the protocol is information in the form of a sample. As the sample does not contain all the information, it may not be completely accurate. Although this fact can affect the accuracy, on the other hand, it provides a report on communications that is good enough to create a report on network traffic. The sFlow consists of two separate parts called the Agent and the Collector. The Agent sends the data to the Collector. After that, the data, using the sFlow protocol version 5, will be collected. The output form of the data is shown in Table 1.

The data are complex. Many of the data do not need to be described in more detail as they are well known. The key data type is Ethernet, which is important information that identifies the protocol that is used to communicate between devices. Several communication protocols were identified on the tested production line, as shown in Table 2.

IPv4 communication was represented mainly by the IoT devices or by the MES communication system providing production control and monitoring. Another protocol examined was IPv6, where it was discovered that the connected IoT device was trying to communicate on IPv6 even though IPv6 was not used in the production line. The rest consisted of ARP and LLDP. Although represented, they serve as ancillary communications and are not relevant to the analysis of network traffic.

Table 1. The output form of the data.

Column Name	Desc	Type
FLOWINDICATOR	Type of FLOW agent	string
AGENT_ADRESS	IP address of agent	IP4 address
INPUT_PORT	Input port	Number
OUTPUT_PORT	Output port	Number
SOURCE_MAC_ADD	Source MAC address	Hex value
DEST_MAC_ADD	Destination MAC address	Hex value
ETHERNET_TYPE	Ether type of Ethernet frame	Two Octet
IN_VLAN	Input virtual LAN	Number
OUT_VLAN	Output virtual LAN	Number
SRC_IP	Source IP address	IP4 address
DST_IP	Destination IP address	IP4 address
IP_PROTOCOL	Type of IP protocol	Number
IP_TOS	IP type of service	One Octet
IP_TTL	IP Time to live	Number
SRC_PRT_OR_ICMP_TYPE	Source Port IP, ICMP type	Number
DST_PRT_OR_ICMP_COD	Dest. Port IP, ICMP code	Number
TCP_FLAGS	TCP flags	Hex value
PACKET_SIZE	Packet Size	Number
IP_SIZE	IP Size	Number
SAMPLING_RATE	Sampling Rate of Agent	Number

Table 2. Used communication protocols.

Ethernet Type	Description
0×0800	Internet Protocol version 4 (IPv4)
0×0806	Address Resolution Protocol (ARP)
$0 \times 86dd$	Internet Protocol version 6 (IPv6)
0×8892	PROFINET Protocol
$0 \times 88cc$	Link Layer Discovery Protocol (LLDP)

The most significant part of the network traffic was the PROFINET Protocol. The PROFINET Protocol is used for almost the entire integration of the production line and accounted for 95% of the network traffic.

The sFlow collector provides data collection from agents; in this case, the agents were two switches supporting sFlow. The sFlow collector started as a server process and listened to data sent by agents on UDP port number 6343. Received sFlow datagrams were converted to structured data, where individual data points were separated by a comma, and so the data format was similar to a CSV file. The data prepared in this way were sent for processing to the neural network. The output is therefore a stream of data or even backed up to a CSV file for additional analysis and verification of accuracy.

2.5. Machine Learning

Apache Spark (https://spark.apache.org/ (accessed on 18 February 2021)) is a unified analytics engine for large-scale data processing. Spark provides several possibilities for the implementation of machine learning, specifically in the Spark's machine learning library (MLLib), which is also suitable for the tested solution, because it meets the assumption of scalability [22]. A native SCALA language or a Python language could be chosen. The native SCALA language is characterized by performance and wide support. Python, which, according to the available data, is slightly less powerful, in turn, allows for easy integration of the components used. MLLib supports both of these options. Python API written in python to support Apache Spark (PySpark) and the MLLib library were used to validate the solution. The integration architecture is shown in Figure 3.

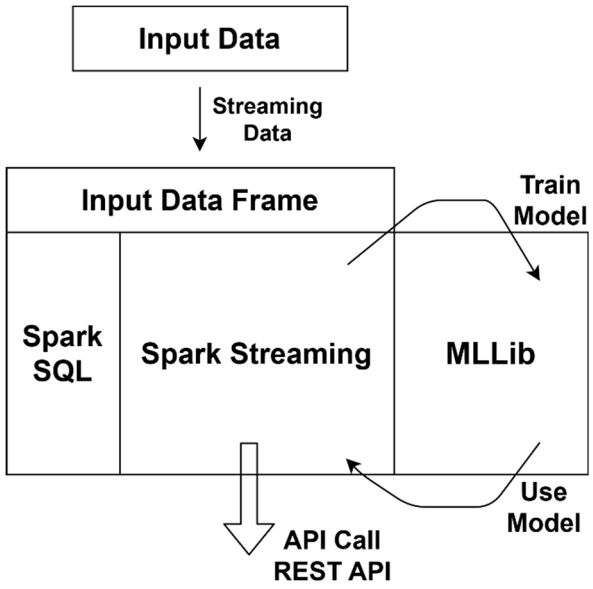

Figure 3. pySpark implemented architecture.

The input data have undergone a process of normalization. A part of the transformation took place in the sFlow collector. The sFlow collector transformed the packet into structured data. Another necessary step was to further adjust the data to values that could be used for learning, testing, and subsequent neural network verification. Data that use hex and vinegar data types have to be transformed into numeric values. An extract, transformation, and loading are shown in Figure 4.

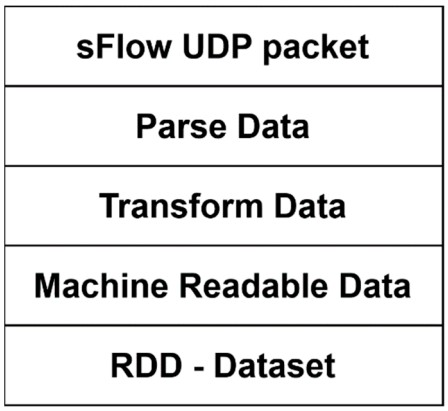

Figure 4. ETL (extract, transform, load) process architecture.

The final transformation and validation were performed using pyspark.sql, where it is possible to operate with whole columns, and can relatively easily perform the necessary conversions and validation of values. NewDF was defined for the entire structure, which specifies the type for each column. Columns where Hex, Octa, or String were used were defined as String. For these parameters, it was necessary to do another conversion, but only at the time when the data was loaded into the dataset. This section overlaps in Figure 4 in the Data Transformation and Machine Readable Data sections. The Spark dataset only needs to be processed using the methods offered by pyspark.

3. Results

3.1. Production Line under DDoS Attack

Three different types of DDoS attacks were performed during the production line testing. The production line, as well as the IoT equipment, were fully operational during the

attacks. The attacks were performed either from the internal LAN, in which the production line is located or from an external Internet network. The production line was tested for both direct and reflected DDoS attacks.

The highest possible packet sending speed was set during the attacks. The attacks were not of a specified length. The length of the attack was given by the ability of the target device to resist the attack, meaning that the DDoS attack lasted until the device stopped communicating on the network. The flooding of the individual components of the production line, directly or indirectly, via IoT devices was intended to disrupt the process that took place on the production line.

Scenario 1:

A direct UDP flood attack was the first DDoS attack performed. This attack aimed at the robot and an RFID reader on the production line. The direct UDP attack was conducted from the internal LAN in which the production line is located. During the attack, the attacker generated packets from random sources, which were then sent to individual components of the production line. As shown in Figure 5, after 14 s of attack, the RFID reader was disabled. After 17 s, the robot also stopped communicating in the network. Both devices were inactive after the attack and required a restart by the accompanying operator.

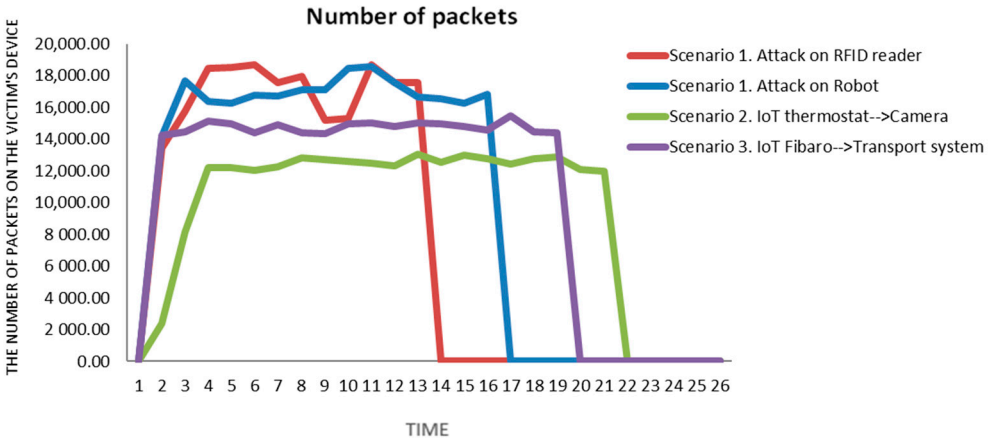

Figure 5. Time process of demonstrated attacks.

Scenario 2:

A reflected TCP SYN flood attack was the second DDoS attack performed. The IoT thermostat located on the production line was misused for the attack, where the attacker sent packets with a spoofed IP address and a nonexistent port to this IoT device. Then, the IoT thermostat reflected these packets to the camera on the production line. In this case, the DDoS attack on the production line was conducted from an external Internet network. Figure 5 shows that the camera on the production line was disabled by the attacker in 22 s of the intense attack. This attack required the intervention of the production operator, who had to restart the device to restart the operation.

Scenario 3:

The third DDoS attack tested was a reflected ICMP flood attack. An IoT device, a Fibaro control unit, was used as a reflector. As in the previous scenario, packets with a spoofed IP address and a nonexistent port were sent by the attacker to the IoT device. IoT Fibaro packets reflected on the transport system of the production line. Even during this experiment, the DDoS attack on the production line was conducted from an external Internet network. As can be seen in Figure 5, the entire attack lasted 20 s. During these 20 s, the attacker managed to deactivate the transport system on the production line, and then the operator's intervention was necessary. The operator was forced to restart the PLC controller of the transport system on the production line.

The impacts of the attacks on the production line were devastating. The line stopped working properly, and the individual components stopped working partially or even completely. DDoS attacks had an impact on the following parts of the production line:

(i.) A robot—The robot and its PLC showed only a slight slowdown in operation or inconsistent movement of the robot during the second and third attacks. However, during a targeted DDoS attack directly on the robot port using the UDP protocol (Scenario 1), the robot was taken out of service in the middle of the performed operation, meaning that the robot stopped and the robotic tools fell down.

(ii.) A camera—in order to produce products for production lines, it is necessary for the camera to read QR codes from the transport vehicle. This was not possible due to congestion of the camera. The following products remained on the conveyor belt, so there was no unloading, which, over time, led to the ability to report the unloading monitor about the delivery of goods.

(iii.) An RFID reader—failure of RFID readers during a DDoS attack interfered with the ability to test products beforehand and sort the types of products into incompatible packaging. The work cycle was also disrupted by other readers in the recycling station, where the state at the entrance and exit of the recycling station was incorrectly identified.

(iv.) A transport system—the conveyor belt stopped working because the PLC control unit was no longer able to control the propulsions due to the DDoS attack. The result was an inability to transport processed products and the production line reporting a failure.

(v.) An MES system (Historian)—although the system was constantly operating, it completely failed to collect data from production lines. Additionally, the Historian server recorded only empty data.

It should be noted that, for individual attacks, the human production operator incorrectly evaluated the errors caused by the DDoS attack as an error condition of the line or its component. The production line panel also reported a mechanical error. Since the analysis of the problem arising on the production line relied only on monitoring the production process, it was not sufficient, and it led to incorrect conclusions. The error rate of the production line was never evaluated as a consequence of the DDoS attack. Even the monitoring of packets in the network did not bring a correct evaluation because the reflected attacks were packets, which were reflected from the IoT equipment of the production line and thus generated by the legitimate equipment of the production line.

3.2. Neural Network

Neural network evaluates network traffic samples. The logistic regression is designed to predict binary results and uses binomial logistic regression [23]. It is represented by a one-layer neural network [24]. Classification logistic regression is suitable for data processing where binary output is expected. Spark's machine learning (SparkML) provides binomial logistic regression or multinomial logistic regression. It is possible to set up explicitly or it is possible to keep this parameter empty and SparkML library is able to choose one which fits better to the training dataset. The data transformation which was performed in pySpark is a dataset that is used to create a training set. Data are transformed into numerical values and all input parameters merge multiple columns into a vector.

The value of correlation can be between 1 and -1. When it is close to 1, it means there is a strong positive correlation. For the dataset used, there were strong positive correlations for INPUTPORT, OUTPUTPORT, and DST_MAC. When the coefficient is close to -1, it means there is a strong negative correlation like ETHERNET_TYPE or also maybe SRC_MAC. The remaining values, around zero, have no linear correlation, and are not suitable for further processing. An overview is given in Table 3, and the heatmap is shown in Figure 6.

Table 3. Correlation of input parameters.

Column	Correlation
AGENT_ADRESS	NaN [1]
INPUTPORT	0.6188414212601262
OUTPUTPORT	0.5909340090255149
SRC_MAC	−0.3779526827561006
DST_MAC	0.5892544767914198
ETHERNET_TYPE	−0.5886118381226132
IN_VLAN	NaN [1]
OUT_VLAN	NaN [1]
SRC_IP	0.08049091599517895
DST_IP	0.07355890187601573
IP_PROTOCOL	−0.07407919159621763
IP_TOS	NaN [1]
IP_TTL	−0.05768954593765455
SRC_PORT_OR_ICMP_TYPE	−0.0676195969509869
DST_PORT_ICMP_CODE	−0.042113943046892865
TCP_FLAGS	−0.06061824921449012
PACKET_SIZE	−0.0733419036600926
IP_SIZE	−0.06770010213058673
SAMPLING_RATE	NaN [1]

[1] NaN (not a number)—values are not important and can be removed from the training dataset.

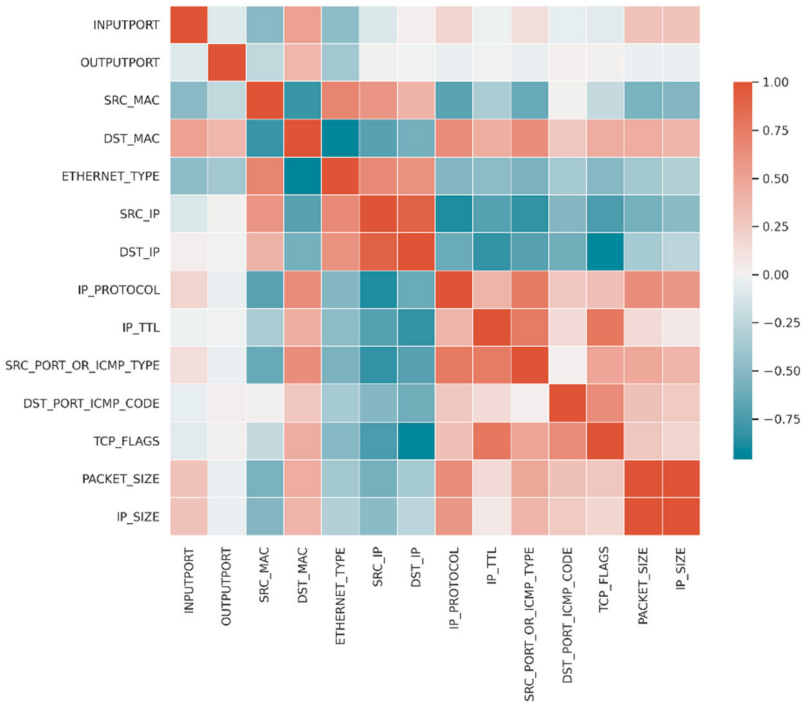

Figure 6. Correlation heatmap.

A decision to use the logistic regression algorithm was made in this case. The result indicates one of two states: Either the sample set conforms to the data stream and is marked as allowed, or it is marked as not allowed. The model was learned and a summary of the results is shown in Table 4. The goal is to evaluate the sFlow sample and predict whether its network traffic is safe or not.

Table 4. Model summary.

Summary	Status	Prediction
Count of training data	4970	4970
Mean	1.0643863179074446	1.0643863179074446
Std. dev.	0.24568714463363647	0.24568714463363647
Min.	1	1
Max.	2	2

A test was performed using machine learning library evaluation, specifically the Binary Classification Evaluator. The test dataset was verified, and a final coefficient of 1.0 was obtained, which indicates 100% success. Further validation was performed on the production line during the production process, and the model, which was taught, was further tested in practice.

The source for the sFlow dataset for the production line is given at the end of the article.

3.3. Firewall Provisioning

Firewall provisioning was implemented using a data flow technique with the Node RED tool (https://nodered.org/ (accessed on 18 February 2021)). Node RED software was used mainly for its easy implementation and the ability to create a debug dashboard, which was used to test the created solution. Provisioning consists of the implementation of the REST API with the help of the components Node RED http input. The protocol chosen for entering commands in order to block unwanted communication and ensure the provisioning of firewall rules of the switch was the SSH protocol. Specifically, according to the type of manufacturer, it is possible to implement the required execution of commands, as any command can be prepared and executed via SSH commands. If devices that do not support the SSH protocol are used, it is possible to replace this block as needed and select the appropriate protocol.

sFlow traffic, which is evaluated in Spark, is sent by the HTTP protocol by the PUT method. A file with JSON data structure is attached to the PUT method. The Node RED input block represents an http input called HTTP in Request. The data are converted to a Data Transformation block, which converts the JSON structure to a Javascript object. This structure represents the internal Node RED data type. The transformed data are routed to the data perser block. Consequently, a function that extracts the appropriate data to be prepared for execution is implemented. Network end elements are configured with configuration commands, and SSH communication is selected. This is supported by almost every manufacturer and can be used to communicate with devices that protect network traffic. This block can be replaced accordingly to the need and suitability of the protocol, depending on the type of existing infrastructure.

The SSH configuration parameters have to be set so that the connection to the end element is possible. It is usually advisable to generate a pair of SSH keys for authentication and use the private one in the Node RED SSH block. A private key is used for login, and the public key is uploaded to the list of authorized keys on the device that needs to be configured. This block can also be executed in parallel. The prepared command can be, for example, inserting the IP address into the list to be blocked. In this case, the entire port on the switches was blocked because the network traffic was evaluated as dangerous. A response is prepared in the Response function block. Here is the status of the response code. As an output, only the HTTP code is sent, but also an email message that records the action of blocking. The implementation of provisioning using Node RED is shown in Figure 7.

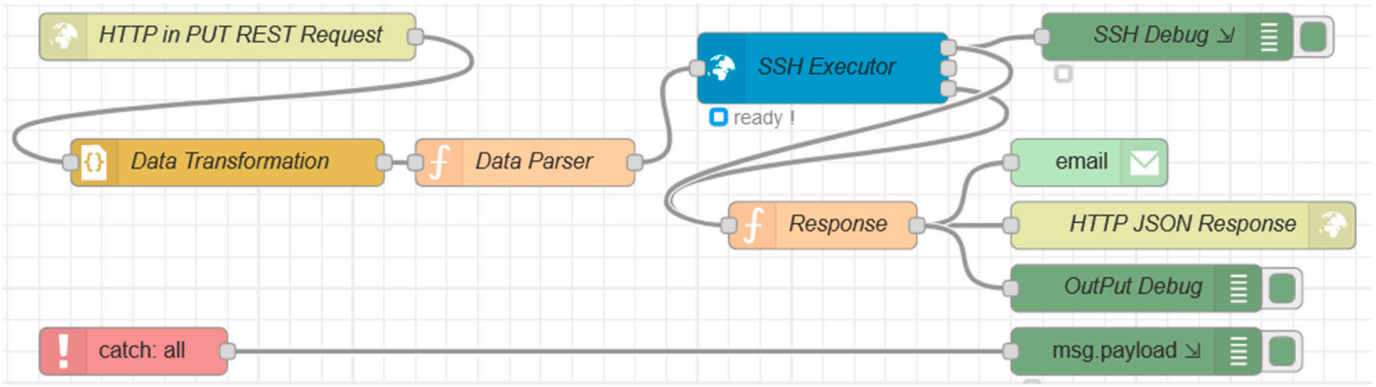

Figure 7. Node RED provisioning.

The reaction time was tested under the same scenarios as described in Section 3.1. Executed attacks were detected and successfully blocked (Figure 8). There are two situations noticeable. The first two peaks show the response time and attack blocking in Scenario 1 for RFID reader and the Robot. The attacks performed for Scenario 2 and Scenario 3 had a similar course; the attacks ran in parallel and a one-second delay in blocking the second attack was expected. The experiment showed that the switch was configured to log in with one session, and the execution of the second blocking command failed due to a login failure. Due to the fact that two attacks were detected and each detection was creating a blocking request, Node RED was forced to repeat login once again in the next second, causing a delay. The problem was solved by changing the parameter of SSH login limits on the network switch.

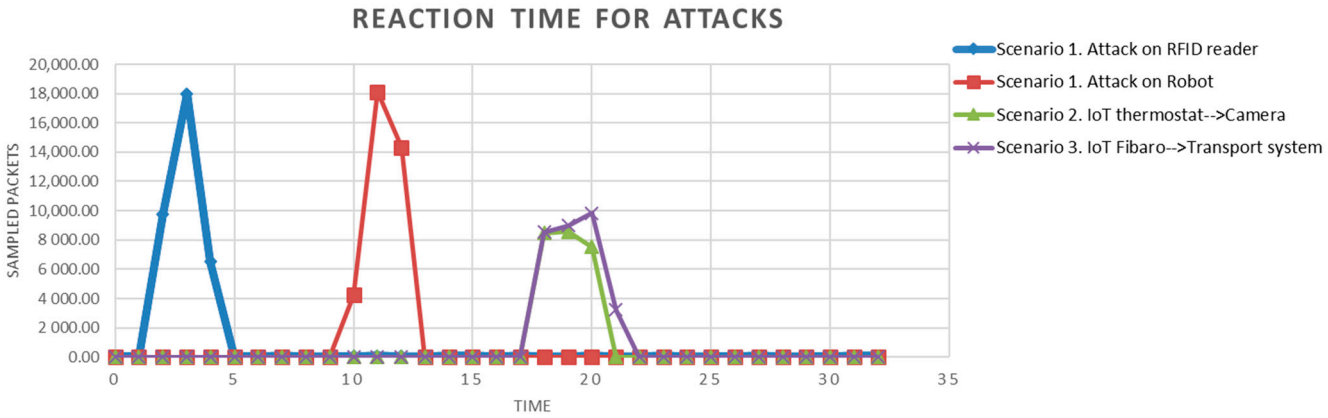

Figure 8. Reaction time for attacks.

4. Discussion

In order to track the traffic, it is possible to consider larger infrastructures. Testing was performed with two switches that generated a sampling rate. The sampling rate was sent over a different subnet to the sFlow Collector, which streamed the data for processing into the Spark application. Spark application generated a REST call. The connection of the verified production line is shown in Figure 9.

The production line is connected with two switches. Network switches are in the role of an sFlow agent. Network traffic created during the production process is sampled and delivered to the NetFlow collector. Network samples are transformed from UDP sFlow packet into the normalized format by an application written in Python. Data are formatted and stored into CSV structure and in parallel streamed to the prepared socket where Spark is expecting input data. Data is transformed and Spark is using a trained model and doing classification of samples. If traffic is detected with attributes as an attack, Spark is sending REST call to Node RED and it is sending requests to network switches to block

communication and sending a notification about it. There is a possibility to create a lot of actions and make a strategy of protection, which can be complex according to architecture and security policy.

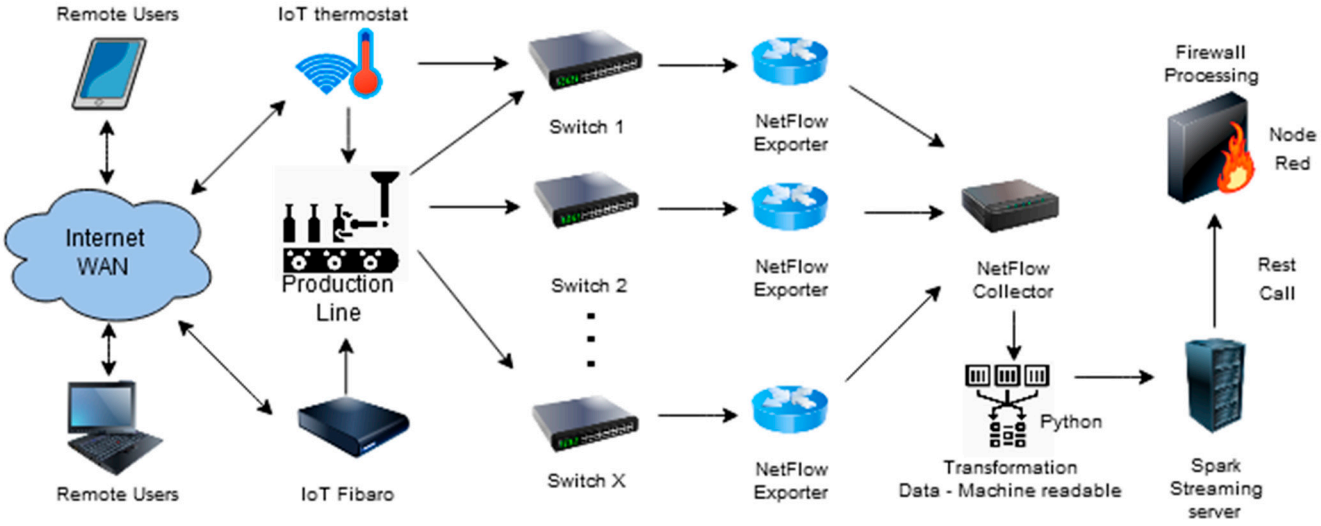

Figure 9. Suggested network infrastructure.

The process of evaluating network communication and predicting the correctness of packets has to ensure the blocking of dangerous network data flow until other devices are operational. The critical time, as measured in the experiment, is 14 s, as seen in Figure 5. A sample sFlow, data acquisition, transformation, neural analytics, and provisioning must therefore be completed before devices' communication capabilities fail and they need to be restarted. The sFlow itself is delayed as the sample needs to be prepared and then loaded. However, DDoS and DRDoS attacks generate so much data traffic that, in this case, sFlow appears to be a convenient data source.

The transformation of sFlow into readable data is fast and not performance-intensive. The data submission and evaluation by Spark are a critical part, and they are suitable as one of the tested parameters of the proposed solution. It is necessary to define the start and end times of the evaluation operation. Similarly, timestamp generation was applied during each possible operation in the provisioning count. During testing, the value was reached without much difficulty in less than 2 s from the start of the DDoS attack until it was blocked. This time was achieved with sFlow, which had a sampling rate set to 1 s. Machine learning was run in the local Spark installation on the local network. This was because Spark is intended primarily for big data, and a greater amount of data needed to be evaluated.

The solution for IoT devices is aimed at testing the possibility of evaluating DDoS and DRDoS attacks, which produce a large amount of data traffic, and to verify the possibilities of big data techniques in evaluating network communication from the point of view of security. This ensures easier expansibility and scalability of integration in larger production lines, and the solution can be applied outside the IoT equipment and can also provide a sufficiently adaptable solution. On the other hand, if the goal is to make the solution simple, the part that is implemented in Spark can be replaced by a modified implementation. The provisioning part was selected for its easy implementation and possible adaptation, which is required according to the comprehensive security design and the security rules of the company. The volume of data that can be processed far exceeds the amount of data that is generated from network traffic by the sFlow protocol.

The samples themselves do not carry information about the data but quantify ongoing network communication. On the one hand, this is an advantage, since the quantification of the network data flow takes place in the sampling process itself; at the same time, it can

be considered a disadvantage because the data flow is not precisely defined. However, the suitability of protection was verified by implementing data traffic samples for IoT and production lines. Based on analyses and measured values, it is possible to demonstrate the effectiveness of the proposed solution, which ensures that an adequate sFlow sample rate can be generated in a short time. With a greater nonstandard data stream, the samples are generated based on the size of the transmitted data and thus sent more often. The response of the solution to the detected DRDoS attack on the production line is shown in Figure 10.

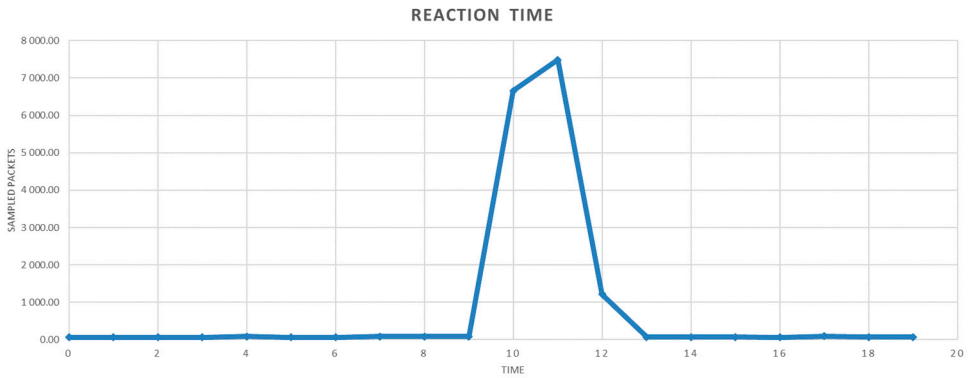

Figure 10. Time reaction of solution to DRDoS attack.

The 10th second indicates an increase in data traffic. The sampling rate is set to 1 s. As can be seen in Figure 10, the number of packets increases in the 11th second. The sample from the 10th second is thus delivered for processing in the 11th second, and the Spark stream evaluates the data stream. The response and blocking of the communication are observable in the 12th second when the flow of the previous packets decreases; in the 13th second, the usual data traffic is already visible. The communication was also tested when the file was copied using the FTP protocol in order to test the packet classification itself. The progress is shown in Figure 11.

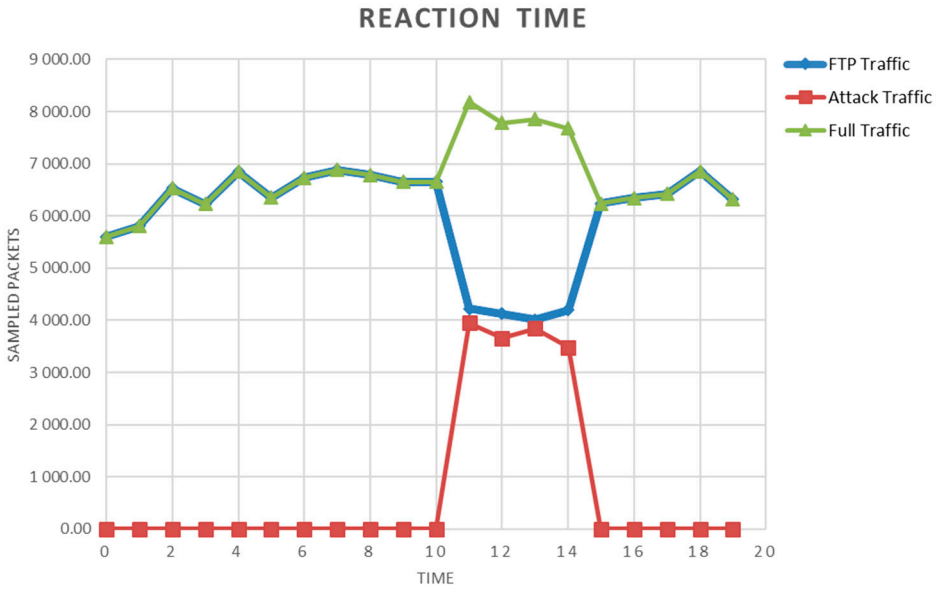

Figure 11. Reaction times of blocking attack with other traffic.

The traffic FTP was launched before the DRDoS attack. The attack was launched at the 12th second. The classification of packets in Spark had more samples for analysis and evaluation. The attack was launched by using the hping3 tool. As can be seen in Figure 11, the classification evaluated the attack as unsolicited communication, and a command was

sent to Node RED to block the given MAC address on the port. The solution was then verified by further tests. It turned out that the specific data flow of the production line could be evaluated relatively accurately to decide on the network communication. The success of the solution was also enhanced by the fact that IoT devices usually rely on TCP/IP communication, while production devices specifically use ProfiEthernet.

Both tested case scenarios were applied in the manufacturing process and the results fit the expectations. The proposed system was able to detect DDoS attack by the required time and the production line was able to continue without a time delay or a pause in the production process. There are opportunities to apply many more complex test cases, and future research will focus on this topic.

5. Conclusions

IoT devices that are integrated into the manufacturing process pose a potential threat in terms of possible DDoS attacks. The article describes a case study of the application of IoT devices to a real production line and shows the vulnerability of the production line to DDoS attacks after the introduction of IoT devices. In addition, verification of the proposed solution is described, proving that it is possible to use sampled flow (sFlow) to detect and protect against DDoS or DRDoS attack.

The potential of the proposed solution was demonstrated on a specific architecture and implementation. The model has been tested with up to 100% success, which provides a good basis for its further expansion in real situations. On the other hand, this model is just a prototype that needs to be incrementally enhanced with additional advanced network data. The proposed model is the basis for further exploration of the possibilities of IoT device applications using sFlow. This approach can significantly affect the deployment of Industry 4.0 in production lines and provide a mechanism for the securing and possible integration of security elements within BIG DATA and their subsequent processing, which will bring additional value to the production process and improve production. Processing streaming data appears to be advantageous, despite the delays brought about by sFlow and the need to properly configure network infrastructure elements. sFlow can also be considered a suitable source of data to secure the production line. An important parameter is the speed of sample creation, which can be set up to several tens of seconds. Depending on the expected response of the system, it is necessary to configure the infrastructure elements in such a way that they are able to process the samples quickly before sending them for evaluation. Since these are samples, there is no need to worry about excessive data flow.

This work serves as a basis for future research and the case study presented in the article will help with understanding and increasing the protection of the production line with integrated IoT devices against DDoS attacks.

A possible extension of the research is also the investigation of mitigation against DDoS attacks on the IoT-based production line in IPv6 networks. Especially IPv6-based networks like 6LoWPAN propose a framework to interconnect IoT sensors. The 6LoWPAN adaptation layer offers the necessary mechanism to adapt the network of wireless IoT devices to work with the IPv6 protocol, gathers information that allows the improvement of efficiency and cost reductions in manufacturing processes, and removes a lot of IPv6 overheads [24]. Future work will focus on analyzing the potential of such IPv6 networks during DDoS attacks on the production line.

Author Contributions: Conceptualization, L.H., T.H., and P.S.; methodology, L.H., T.H., P.S., and P.T.; validation, T.H. and P.S.; formal analysis, L.H., T.H., and P.S.; investigation, T.H., P.S., and L.H.; resources, P.T. and L.H.; writing—original draft preparation, L.H., T.H., P.S., and P.T.; writing—review and editing, T.H. and L.H.; visualization, T.H. and P.S.; supervision, L.H. and P.T.; funding acquisition, L.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Optimization of Network Security by Computational Intelligence under Grant VEGA 1/0145/18, and in part by the Algorithm of Collective Intelligence: Interdisciplinary Study of Swarming Behavior in Bats under Grant APVV-17-0116.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The data can be found here: http://kai.fpv.ucm.sk/huraj/dataset/SflowData.zip (accessed on 18 February 2021).

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

References

1. Vishwakarma, R.; Jain, A.K. A survey of DDoS attacking techniques and defence mechanisms in the IoT network. *Telecommun. Syst.* **2019**, *73*, 3–25. [CrossRef]
2. Di Natale, G.; Regazzoni, F.; Albanese, V.; Lhermet, F.; Loisel, Y.; Sensaoui, A.; Pagliarini, S. Latest Trends in Hardware Security and Privacy. In Proceedings of the 2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT), Frascati, Italy, 19 October–21 October 2020; pp. 1–4. [CrossRef]
3. Conti, M.; Dehghantanha, A.; Franke, K.; Watson, S. Internet of Things security and forensics: Challenges and opportunities. *Future Gener. Comput. Syst.* **2018**, *78*, 544–546. [CrossRef]
4. Ferrari, P.; Flammini, A.; Rinaldi, S.; Sisinni, E.; Maffei, D.; Malara, M. Impact of Quality of Service on Cloud Based Industrial IoT Applications with OPC UA. *Electronics* **2018**, *7*, 109. [CrossRef]
5. Huang, X. Intelligent remote monitoring and manufacturing system of production line based on industrial Internet of Things. *Comput. Commun.* **2020**, *150*, 421–428. [CrossRef]
6. Regazzoni, F.; Bhasin, S.; Pour, A.A.; Alshaer, I.; Aydin, F.; Aysu, A.; Beroulle, V.; Di Natale, G.; Franzon, P.; Hely, D.; et al. Machine Learning and Hardware security: Challenges and Opportunities-Invited Talk. In Proceedings of the 2020 IEEE/ACM International Conference on Computer Aided Design (ICCAD), San Diego, CA, USA, 2–5 November 2020.
7. Filho, L.; De, F.S.; Silveira, F.A.F.; Junior, A.D.B.; Vargas-Solar, G.; Silveira, F.L. Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning. *Secur. Commun. Netw.* **2019**. [CrossRef]
8. Chen, W. Intelligent manufacturing production line data monitoring system for industrial internet of things. *Comput. Commun.* **2020**, *151*, 31–41. [CrossRef]
9. Prinsloo, J.; Sinha, S.; von Solms, B. A Review of Industry 4.0 Manufacturing Process Security Risks. *Appl. Sci.* **2019**, *9*, 5105. [CrossRef]
10. Knudsen, A.H.; Pedersen, J.M.; Sørensen, M.A.M.; Villumsen, T.D. *Security in the Industrial Internet of Things, in Cybersecurity and Privacy: Bridging the Gap*; River Publishers: Gistrup, Denmark, 2017; pp. 119–134.
11. Perales Gómez, Á.L.; Fernández Maimó, L.; Huertas Celdrán, A.; García Clemente, F.J.; Gil Pérez, M.; Martínez Pérez, G. SafeMan: A unified framework to manage cybersecurity and safety in manufacturing industry. *Softw. Pract Exper.* **2020**, 1–21. [CrossRef]
12. Tuptuk, N.; Hailes, S. Security of smart manufacturing systems. *J. Manuf. Syst.* **2018**, *47*, 93–106. [CrossRef]
13. Sha, L.; Xiao, F.; Chen, W.; Sun, J. IIoT-SIDefender: Detecting and defense against the sensitive information leakage in industry IoT. *World Wide Web* **2018**, *21*, 59–88. [CrossRef]
14. Xu, C.; Zhu, G. Intelligent manufacturing Lie Group Machine Learning: Real-time and efficient inspection system based on fog computing. *J. Intell. Manuf.* **2020**, 1–13. [CrossRef]
15. Martinez, B.; Montón, M.; Vilajosana, I.; Vilajosana, X. Early scavenger dimensioning in wireless industrial monitoring applications. *IEEE Internet Things J.* **2015**, *3*, 170–178. [CrossRef]
16. Vaclavova, A.; Kebisek, M. Integration of production line with the Wonderware platform. In *Software Engineering and Algorithms in Intelligent Systems*; Springer: Cham, Switzerland, 2018; pp. 208–215. [CrossRef]
17. Vaclavova, A.; Kebisek, M. Design of Virtual Model of Production Line Using Wonderware ArchestrA. In Proceedings of the IEEE 22nd International Conference on Intelligent Engineering Systems (INES), Las Palmas de Gran Canaria, Spain, 21–23 June 2018; pp. 000425–000430. [CrossRef]
18. Panarello, A.; Tapas, N.; Merlino, G.; Longo, F.; Puliafito, A. Blockchain and IoT Integration: A Systematic Survey. *Sensors* **2018**, *18*, 2575. [CrossRef] [PubMed]
19. Mehmood, A.; Mukherjee, M.; Ahmed, S.H.; Song, H.; Malik, K.M. NBC-MAIDS: Naïve Bayesian classification technique in multi-agent system-enriched IDS for securing IoT against DDoS attacks. *J. Supercomput.* **2018**, 1–15. [CrossRef]
20. Šimon, M.; Huraj, L.; Čerňanský, M. Performance Evaluations of IPTables Firewall Solutions under DDoS attacks. *J. Appl. Math. Stat. Inform.* **2015**, *11*, 35–45. [CrossRef]
21. Šimon, M.; Huraj, L.; Horák, T. DDoS Reflection Attack Based on IoT: A Case Study. In *Cybernetics and Algorithms in Intelligent Systems. CSOC2018 2018. Advances in Intelligent Systems and Computing*; Silhavy, R., Ed.; Springer: Cham, Switzerland, 2019; Volume 765. [CrossRef]
22. Meng, X.; Bradley, J.; Yavuz, B.; Sparks, E.; Venkataraman, S.; Liu, D.; Freeman, J.; Tsai, D.B.; Amde, M.; Owen, S.; et al. Mllib: Machine learning in apache spark. *J. Mach. Learn. Res.* **2016**, *17*, 1235–1241.
23. He, H.; Li, S.; Hu, L.; Duarte, N.; Manta, O.; Yue, X.-G. Risk Factor Identification of Sustainable Guarantee Network Based on Logistic Regression Algorithm. *Sustainability* **2019**, *11*, 3525. [CrossRef]
24. Miguel, M.L.F.; Jamhour, E.; Pellenz, M.E.; Penna, M.C. SDN Architecture for 6LoWPAN Wireless Sensor Networks. *Sensors* **2018**, *18*, 3738. [CrossRef] [PubMed]