Honeypot Coupled Machine Learning Model for Botnet Detection and Classification in IoT Smart Factory – An Investigation

Seungjin Lee[1,*], *Azween* Abdullah , *N.Z*. Jhanjhi, and *S.H*. Kok

[1]School of Computer Science and Engineering (SCE), Taylor's University, Malaysia.

Abstract. In the United States, the manufacturing ecosystem is rebuilt and developed through innovation with the promotion of AMP 2.0. For this reason, the industry has spurred the development of 5G, Artificial Intelligence (AI), and Machine Learning (ML) technologies which is being applied on the smart factories to integrate production process management, product service and distribution, collaboration, and customized production requirements. These smart factories need to effectively solve security problems with a high detection rate for a smooth operation. However, number of security related cases occurring in the smart factories has been increasing due to botnet Distributed Denial of Service (DDoS) attacks that threaten the network security operated on the Internet of Things (IoT) platform. Against botnet attacks, security network of the smart factory must improve its defensive capability. Among many security solutions, botnet detection using honeypot has been shown to be effective in early studies. In order to solve the problem of closely monitoring and acquiring botnet attack behaviour, honeypot is a method to detect botnet attackers by intentionally creating resources within the network. As a result, the traced content is recorded in a log file. In addition, these log files are classified quickly with high accuracy with a support of machine learning operation. Hence, productivity is increase, while stability of the smart factory is reinforced. In this study, a botnet detection model was proposed by combining honeypot with machine learning, specifically designed for smart factories. The investigation was carried out in a hardware configuration virtually mimicking a smart factory environment.

1 Introduction

Industry is changing rapidly its way of production with customized products to clients. Manufacturing systems is gradually changed from digitalisation and automation to a new application of artificial intelligence and machine learning which results in a new type of factory, known as "smart factory". In order to maintain a smooth manufacturing line of the smart factory, a huge volume of Internet of Things (IoT) devices are required, and it was forecasted that the demand of these IoT devices will increase dramatically to more than 50 million in 2020 as can be seen in Fig. 1 [1].

* Corresponding author: leephael8707@gmail.com

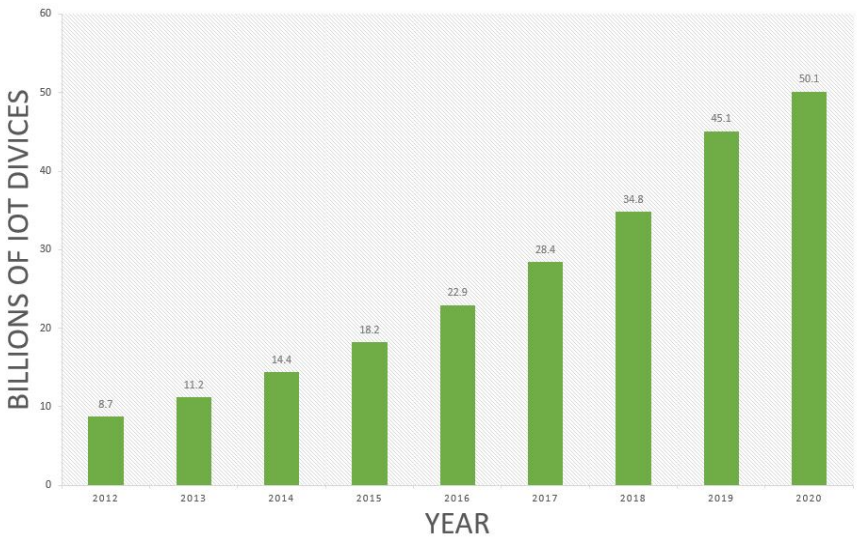

Fig. 1. Demand for IoT devices [2].

For smart factories to operate and maintain its autonomy, they should be trained to analyse by themselves and to accurately carry out quality process management, production management as well as product design. This is done by utilising important information such as process know-how, requirements of analysis data, product design drawings, business secrets and research. Consequently, smart factories are able to maximise the use of real-time production management, simultaneously minimise defect rate, and errors in the manufacturing lines.

Due of the instability of IoT devices, the number of attacks on the Internet infrastructure has soared, making it an ideal target for IoT botnet – a type of cyber attacks. An example of botnet is the attack on the Dyn DNS infrastructure, which mobilized 100,000 IoT devices (mainly CCTV cameras) in October 2016. Another botnet attack on the IoT devices in 2017 using a new Mirai source code posed a serious threat to many manufacturing industries. Therefore, it is urgent to identify and mitigate the impacts of the IoT botnet by developing new technologies [3].

Among many security solutions to the botnet attacks, a detection method known as honeypot offers an advantage of giving real-time detection which plays a very critical role in the operation of smart factory. This is because the honeypot approach offers the following specificities: botnet attacks being captured in a log file and log analysis can provide details on exploitation and attack patterns. Hence it is easy to observe botnets with previously unseen tools or special attack patterns such as Mirai botnet and their interaction with others. Only malicious traffic entering sever simplifies processing. So the information collected is selected and more valuable. Because only attack traffic is detected (no legitimate traffic), it is less likely to be false positive than other security solutions. Private companies can find improvements in detection with minimal resource usage without any additional budgets. It has the advantage of being easy to understand and easy to configure and install programs as an interface for users. Unlike IDS, no known attack signature is required [4].

In the field of cybersecurity for smart factories, after identifying the challenges and limitations of the current methods in the research problem statement, this study presents an improved method to detect botnet as a suggestion. Literature review provides a critical review of the latest research on the topic of developing security solutions for smart factories.

Afterwards, a proposed model describes the method for an efficient smart factory detection model by combining honeypot and machine learning. At last, a preliminary result is revealed together with a conclusion and recommendation for the further work.

2 Problem statement

Industrial manufacturers need to maximize the efficiency of production and factory operation. It is important to identify and solve problems that arise in the manufacturing process, especially security-related issues in order to ensure a smooth implementation of the operating system. Since smart factory is operated using various IoT devices, botnet threats can cause serious damages to its production. The increasingly serious pattern of botnet attacks makes it mandatory for producers to detect botnets. The most important part is to increase the accuracy and time to detect security problems such as botnets. The problem of data transmission between botnet, sensor, CCTV, PLC equipment and primary database server is that data is leaked from smart factory network, which affects data update and also smart factory operation. Anomaly detection monitoring model was earlier proposed to detect network traffic in smart factory. However, a significant amount of data is needed to achieve accurate results. Calculations are expensive, often take time to learn and require a complex training process. In addition, research on the current Intrusion Detection System (IDS) models has some limitations as they are not suitable for smart manufacturing environments. In fact, IoT devices not only provide various information services to users, but also are constrained in terms of resources. These restrictions prevent extensive procedures from being carried out. Therefore, real-time botnet detection is proposed to investigate in this study by developing a model of combining honeypot with machine learning technology for smart factories.

3 Literature Review

Various botnet detection methods and their basis are illustrated in a taxonomy of Fig. 2. In addition of this, a smart factory is layered into perception, communication, network, data and application with the function of each as shown in the next taxonomy of the same figure. In this study, botnet detection using honeypot method/approach is chosen to address the perception layer of the smart factory, which involves usage of many IoT devices and its network with the main PC to process data.

Smart factories raise a great interest in both of the manufacturing companies and academic researchers whose interest are in the 4.0 industrial revolution [5]. Although smart factories are already constructed and operated in the industry, implementation standards are yet to be established [6]. The manufacturing system could be rated in scale based on different perspectives, such as function [7]. A smart factory is conceptualized as adaptive and flexible manufacturing which consists of three main aspects, i.e., interconnection, collaboration and execution [8]. Furthermore, architecture of the IoT smart factory is segregated into four layers arranged hierarchically starting at physical resource layer, networking layer, application layer and interface layer, as shown in Fig. 2 [9].

With the aim of transforming a modern factory into a smart factory, technology relating to all four layers require an in-depth research [10]. The key element of a smart manufacturing system, such as smart factories is intelligence. It is based on the network technology and manufacturing data. Additionally, system maintenance and manufacturing requirements should be incorporated into the implementation of the smart factories. Due to such

complexity in the design and operation of the smart factories, many technical problems arise and need to be solved [11].

At the physical resource/sensing layer, an acquisition of the real-time information is obtained by physical equipment and transmission of heterogeneous information at high-speed through communication devices. As a result, rapid reconfiguration and adaptability have to be ensured at the workplace by increasing the intelligence of these basic devices/equipment to meet the requirements for the Internet of Things platform [12]. In the entire operating cycle of smart factories, including the physical resources, efficiency of the smart manufacturing to produce customized products creates a new demand for adjusting manufacturing equipment, production lines and data acquisition. Due to the limitation in flexibility and configurability, current manufacturing equipment in the workplace is highly specific and relatively narrow in scope, making it less adaptive to changes in the manufacturing environment.

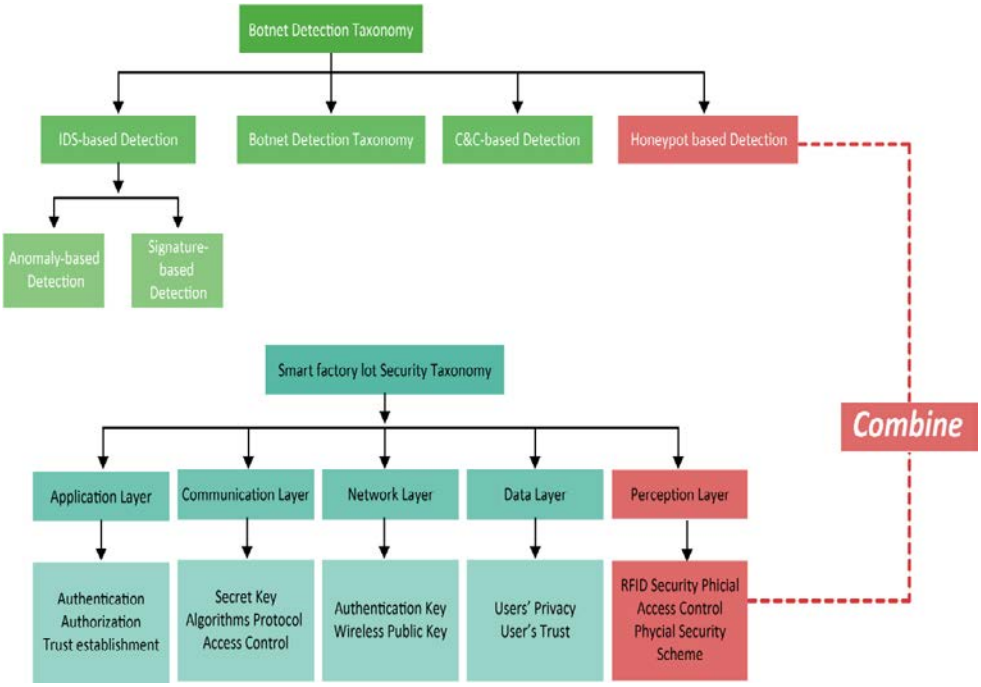

Fig. 2. Taxonomies for botnet detection and security of IoT smart factories.

Cyber-physical integration of cognitive robots in manufacturing was proposed earlier [13]. Specifically, a humanoid robot is used to integrate with the smart manufacturing plant in coordination with the execution system. These robots can cognitively adjust their own manufacturing behaviours to recognize information uncertainty, changes in schedule management and independently deal with complex problems of manufacturing. There is a relationship between the level of intelligence of smart factories and the modular maturing units. Therefore, increasing the intelligence of the modular manufacturing units such as robotic units is very important to make them work synergistically to accomplish common tasks by mutual recognition and co-working mechanisms. The heterogeneity of interactions should also be considered. It is important to create an optimal or lane combination method because the functions of other modular manufacturing devices can be duplicated for a particular product. Each manufacturing unit not only meets the manufacturing requirements

of the product, but it can also systematically improve the smart factory efficiency on its own. However, a deadlock can occur in smart manufacturing as some different products enter the production system in a disordered manner [14]. Currently, a solution to the deadlock in a flexible manufacturing system increasingly becomes a research interest [13].

In the work of Nguyen et al. (2020), the mechanism of spreading Mirai botnet was identified and its effects on the IoT devices were investigated. A combination of more than 60 basic user credentials was employed in Mirai to access the shell of any devices which are open to the public. Once, a smart device becomes a part of the botnet, other connected devices which are subjected to vulnerability as it will scan for other IPv4 address spaces. It would be subsequently identified and damaged. Despite becoming a botnet and receiving the order of the enemy to perform malice, the botnet infected devices are still able to carry out the default activities set by the manufacturer. Such attacks caused by Mirai botnet has laid a foundation for the rising of botnet targeting IoT devices such as botnet list and botnet amnesia. For example, the attack that happened to Telnet and SSH services caused by most botnets resulted from gaining unauthorized access to IoT devices. Another example is the unauthentic access to nearly 400,000 IoT devices via two services reported in the Cybersecurity Survey on IoT [15]. Therefore, due to the availability of numerous network devices that are vulnerable to protection, botnets remain one of the main concerns of cyberspace. To become a useful part of the botnet, vulnerable network devices go through the step sequence.

Bots communicate through command and control, bots and botmasters exchange messages to regularly perform certain tasks. Botnet relies heavily on C&C servers and provides low-latency communication [16]. The botsniffer, HTTP-based botnets can also be detected and abnormal detection technologies can be applied to stop all bots. In addition, servers as channels are detected in similar types of behaviour i.e., flexible in the substation of C&C server addresses. It also provides the information needed to detect hybrid botnet structures [17].

Mirai botnet detection using binary code is a classic method developed by Lee Jun-soo. First, the binary code of the malicious code is analysed and the structure of the binary file is determined before it is used [18]. Specifically, the binary code is described as a portable file structure that runs in a Windows environment. PE format should be implemented based on the nature of the detection, i.e., compatibility in various operating systems (OS) to facilitate detection. So it was named the 'Easy Movement' format. This is because this format is a file format for executable files used in Windows, such as Dynamic-Link Library (DLL), Object Code (object code) and FON-type font files. Similar to PE, there are executable files and connection formats (ELF) and Mac OS X formats, which are x86-based UNIX and UNIX systems [19].

Some detection methods are for post-botnet penetration. In particular, post-intrusion measures are much less effective in terms of detection rates. Contrary to previous research, honeypot protects IoT equipment inside the smart factory by installing a trap in advance, not after the botnet intrusion. Indeed, real-time detection is of paramount importance, especially in the smart manufacturing environment. Various methods of detecting botnet are compared with the honeypot method for the pros and cons, as shown in Table 1.

Signature-based detection is a method of analysing, scaling and detecting botnets based on their knowledge. A typical one is Snort. The signature-based detection method has an advantage of high detection and low false detection for botnets and malware, as earlier found. However, the signature-based detection is unable to detect new botnet attacks, since they does not have a signature. For example, in Lishi, a bot is judged by the IRC name of the bot. The IRC name of the bot was thought to be much different from the nickname of the end-user. However, making IRC nicknames similar to end-users is difficult to detect and impossible to detect without an IRC-based botnet [20].

Table 1. Comparison of honeypot with other detection methods.

Methods	**Honeypot**	**Binary detection**	**Anomaly detection**	**C&C detection**
Configuration	One host	Binary	Heuristic Rules	Server
Advantages	Efficient platform	Easy implementation	High capacity	Expand HTTP-based botnets.
Disadvantages	Information processing is slow.	Information processing is slow.	Unable to monitor	Not simple structure

Anomaly-based detection is a way to suspect and detect strange things that behave differently than normal users in the network traffic, such as intensive traffic or abnormal port use. Yeung's study presented a method for detecting botnets by analysing a data flow of the transport layer. The data suspected from the bot is extracted separately from the data flow and scores are calculated and determined by the bot if the score exceeds the threshold. This method is detectable, even if botnet communication is encrypted and has a low error rate. It is also highly scalable and can help to measure the size of a botnet. Again, only IRC-based botnets can be detected [21].

Notably, honeypot is a system that is installed with a purpose to detect abnormal access and it also serves to track down attackers and gather information. To deceive an attacker, a trap is created, as if the attacker had infiltrated into a normal system. And then a bot is caught and analysed. Based on the analysis results, software disguised as a bot is created and traffic exchanged with the software is analysed to find a botmaster or botnet. One advantage is that botnets can be detected at a high detection rate without the existing knowledge [22].

Table 2 summarizes some particular studies in using the honeypot approach for detecting botnet in smart factories. Honeypot and honeynet can respond to attacks in real-time and attract attackers to deceptive assets rather than real assets.

Table 2. A comparative summary of studies in botnet detection using honeypot and/or machine learning approaches for smart factories.

Approaches	**Strengths**	**Weaknesses**	**Research gap**	**Ref.**
Machine learning for smart factory	Able to labor cost	Complex system	Data can be leaked from the manufacturing process.	[23]
IoT Botnet	Monitoring Web-based real-time	Capacity	New optimisation requires expansion of utilization	[24]
Machine learning	91.66% graph-based detection accuracy	Difficult to apply detectors	Bot mark is required to increase the accuracy	[25]
IoT Honeypot	Fast processing	Unnecessary	Necessary to activate network protocol	[5]
Honeypot machine learning	Real-time monitoring	Depending on system.	Problems with cloud server application	[26]

A study in IoT botnet detection suggested that it is easy to monitor, if the IoT devices are infected through web services [24]. Some restrictions have pointed out that monitoring algorithms for IoT devices is simple to implement and scalable for smart factories using IoT equipment [24]. The capacity for the IoT devices has certain limits. This approach was first designed with a hypothesis that botnet contacts IoT devices were used to invent a detection model based on the binary.

Another approach is botnet detection by machine learning which shows a high accuracy of 99.94%. The combination of flow-based with graph-based detection and machine warning has a high accuracy of detecting a botnet attack as an advantage. A disadvantage of the machine learning approach is that it is harder to detect quickly in the randomized number of packets. Thus, feasibility of applying this approach for smart manufacturing needs more research in real-time and it is time-consuming [25].

For smart factories, botnet detection using honeypot integrated with IoT, known as IoT honeypot) was studied [5]. There is a stochastic basis compared to the machine learning approach with superstitious running. Although the IoT honeypot has stopped scalability by simply applying it to sandboxes IoT, it aims to apply for common expansion in more situations and environments [5]. In the botnet detection system using honeypot machine learning, the learning logging for detection and tracking are highly accurate. The system using most standard equipment, is suitable for performance smart factories. Hence, it is likely to be adopted in the future [26]. The machine learning approach which is applied for smart factory using IDS, can reduce costs, considerable limitations such as low detection rates, highly complex and unsustainable systems were found [23]. These studies in IDS, IoT botnet and honeypot machine learning showed some application results for smart factories. Such solutions are possible to trace through logging at low cost and are most cost-saving for the IoT devices. In summary, for binary, anomaly and C&C detection methods, reaction to real-time is slower than honeypot and honeynet methods [18, 27, 28].

Although binary detection is simple in structure, the detection processing is too slow for smart manufacturing environments that seek real-time detection. In terms of cost-effectiveness, honeypot has an advantage in being capable of responding to attack in real-time at relatively low cost for construction and management. Thus, it is suitable for smart manufacturing environment [29, 30]. However, processing botnet information by the honeypot is slow for analysis. It results in a decrease in accuracy and processing speed [31, 32]. Notably, using machine learning techniques to process botnet information has yet to be investigated so far in any studies combining with honeypot into one botnet detection model, especially designed for smart factory environment. Thus, this gap in the literature needs to be fulfilled with a more in-depth research which is being proposed in this present study.

4 Proposed Model

4.1. Configuration of hardware for a virtual smart factory environment

In the configuration setup, a virtual smart factory environment was created using some IoT devices (camera, RFID, temperature sensor) and two Raspberry Pi devices (Pi1 and Pi2).

The first Rasberry Pi (Pi1) was installed with Open CVS, and assigned as the actual main IoT data collection server. Its function was to transmit the collected IoT data to the main PC. The second Raspberry Pi (Pi2) was installed with honeypot acting as a virtual server (VM) in T-pod platform. Raspberry Pi2 was aimed to intentionally collect detection information. As Pi2 is a VM, botnet would be deceptively attracted to Pi2 by using a botnet dataset. The reason T-pod platform was chosen was because of its suitability for such a virtual smart factory environment in which real time detection was very crucial to observe and display

through dashboard. For ease of tracking, Raspberry Pi1 and Pi2 were at the same time installed on the log server to keep track of IoT product line information and botnet attack patterns during the simulation testing.

Mechanism of operation of the honeypot-coupled machine learning model in a smart factory is illustrated in Fig. 3. When an attacker combined multiple IDs and passwords to log into an IoT device in the physical resource layer and attempted to attack with a botnet through an open port, the honeypot installed in Raspberry Pi2 intentionally broke into its own firewall and entered the attacker as a person who can reach the attacker. The main intention was to obtain information about the attacker and the malicious botnet code by recording each activity between the attacked device and the intruder in the form of a log file. These log files captured information that allowed administrators of the smart factory to identify characteristics, transformations, target device types, C&C server IP addresses, and port numbers of the new malware families or botnets. Log file data must be converted into an appropriate tabular format.

Detail information of botnet attacks contained in the log file (after being converted into tables) was then extracted from Raspberry Pi2 and used as dataset. The machine learning (ML) tool uses the dataset as input for training. An algorithm written for this ML training was to classify botnets. In this training, memory-efficient classification was desirable to predict useful information, since less training data is used. So that, IoT devices would not be overwhelming. Based on the classification results of the ML, appropriate measures were taken. ML system would dynamically repeats whenever the course exceeds the allowed training data size.

4.2. Simulation of hardware design with Raspberry pi - transfer log files

Each product in the smart factory has an RFID tag that contained information. Raspberry Pi collected product information by reading the camera, temperature sensor, and RFID tags. The collected information was stored on Raspberry Pi1 and Pi2 with the calling device. This meant the collected data was programmed to be sent to the IoT service server in the network through the terminal (camera, temperature, RFID – IoT device). The information stored in the log files of Pi1 and Pi2 was sent to the server on the main PC. This hardware-based process simulation is shown in Fig. 4.

4.2.1. Raspberry Pi2 setting - T-pot platform for the honeypot

After setting up Raspberry Pi1 to transmit open CV data, Raspberry Pi2 was set up as a virtual machine (VM) in the T-pot honeypot platform. Verification and testing of the honeypot was done to balance at runtime. To perform this task, a studio status command line was used to write scripts and to verify the transfer of log files.

The script in Fig. 5 shows the load of the platform, the state of each honeypot, and the uptime. Furthermore, data collected by the honeypot was visually displayed networks attacked by malicious users and botnets via Kibana dashboard (Fig. 6). The Kibana dashboard was easy to monitor and comprehensive to analyse the type, location and malicious threat actors of botnet attacks. This had many potential uses for data system and Metrif collection in a smart manufacturing environment that require real-time monitoring.

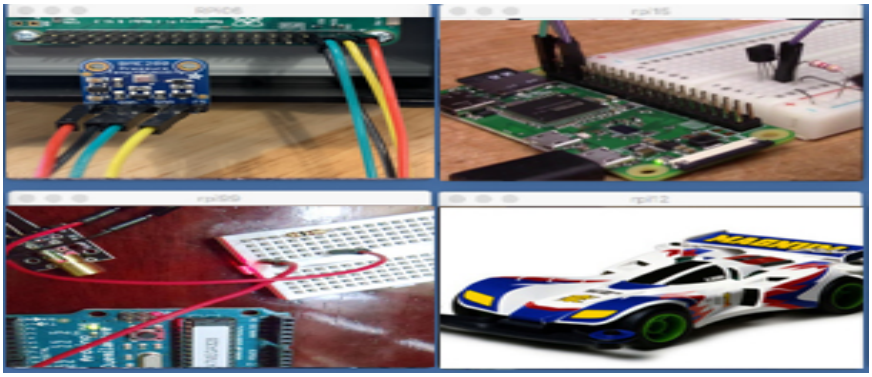

Fig. 3. Simulation of hardware design.

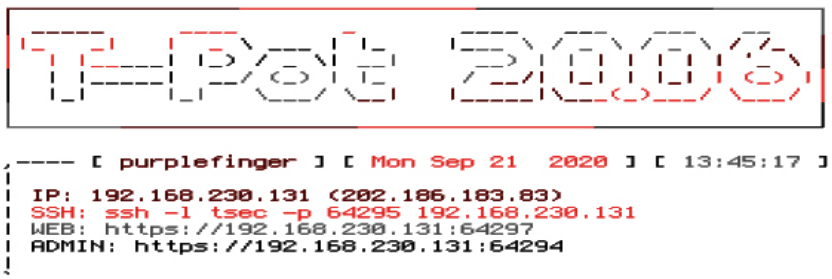

Fig. 4. Test script for Raspberry Pi2.

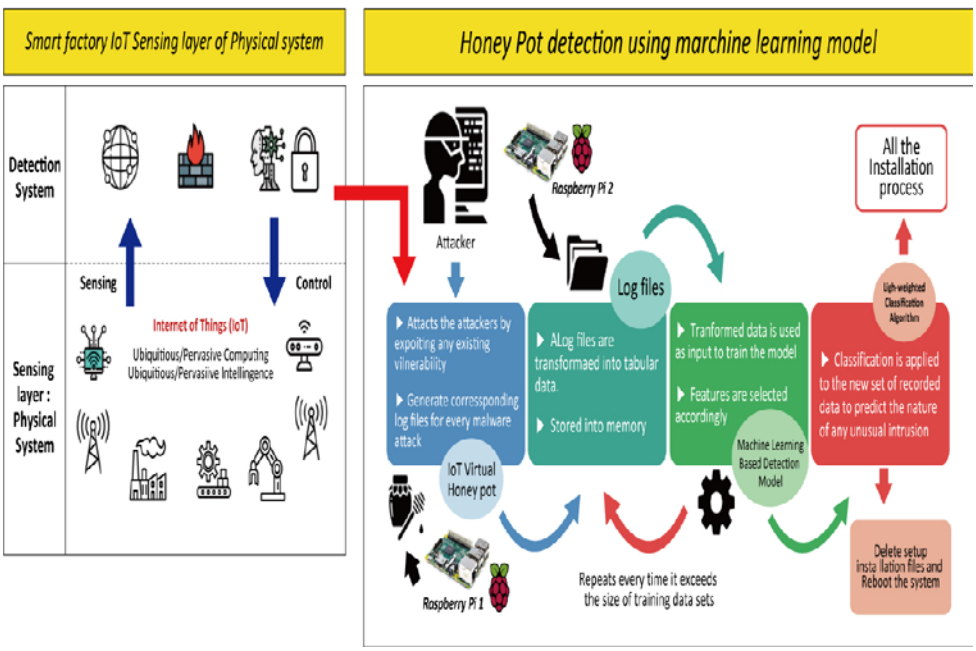

Fig. 5. Operating mechanism of honeypot coupled machine learning model in smart factory.

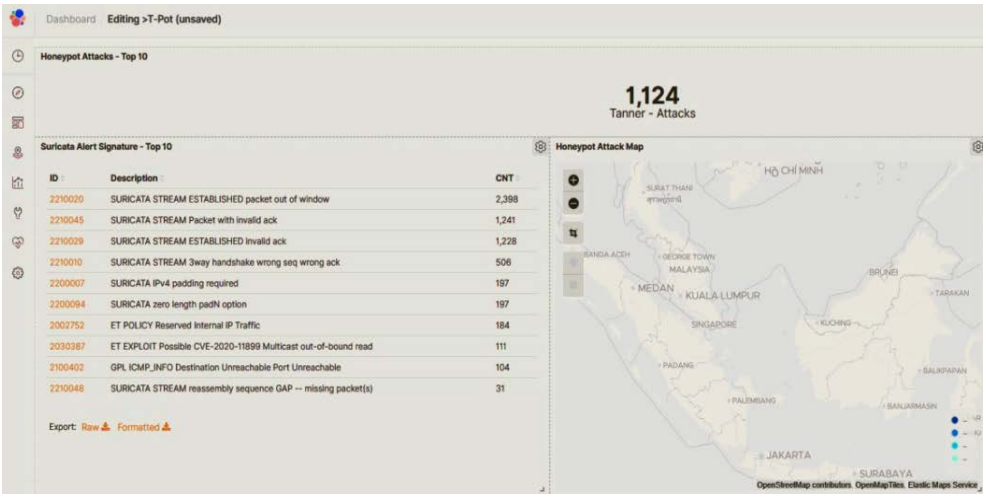

Fig. 6. Kibana dashboard.

4.3. Honeypot coupled machine learning model for botnet detection and classification

The classification process of data contained in the log file generated by the honeypot is described in Fig. 7. Basically, this process mainly consisted of three stages. The first stage was honeypot simulation. The raspberry Pi2 server verified the botnet credentials, finished loading, and started detecting honeypots on the T-Pod platform. In stage 1, a username and password must be provided to verify botnet credentials of the user. Accurate information provided by authenticated users would trigger the log data collect (stage 2), which underwent a series of two-step honeypot detection and three-step automated process in sequence.

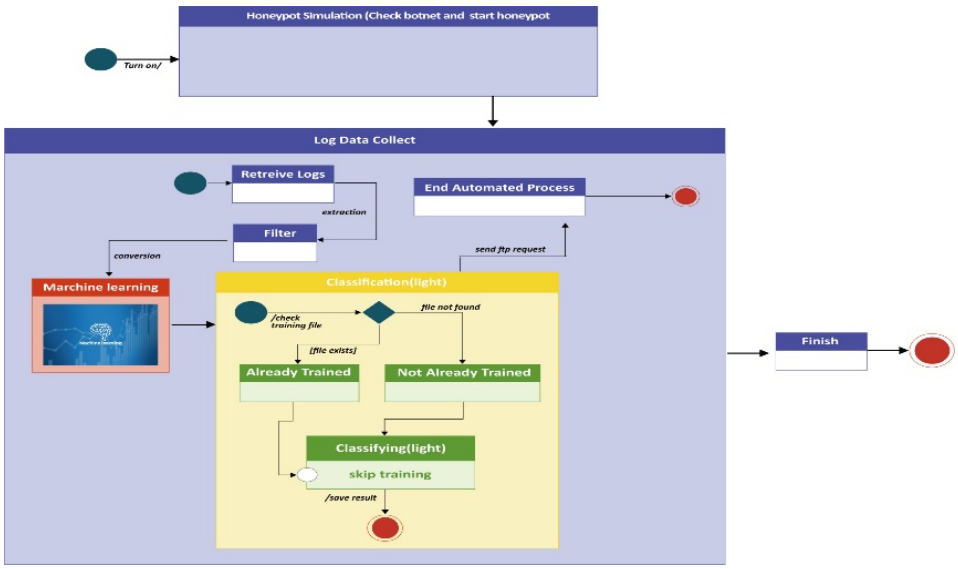

Fig. 7. Design of classification algorithm for the honeypot machine learning model.

Inside this stage, user data was classified into either botnet or non-botnet types via processing through serial steps, i.e. Retrieve Logs, Extract Record, Filter, Text Processing, Upload to End Automated Process. In between the text processing step and the uploading steps, machine learning was integrated to speed up the data classification. For the botnet-type data, a log data file belonging to botnet attacks was created. Then, botnet-type data in the log file was transferred to the final stage of the aggregate collect.

The integration of machine learning was applied as an effective tool to support the log data collection step in term of classification. In the idle phase, botnet credentials and information of authenticated users with different access levels were collected in the honeypot detection phase. The idle phase sent a signal for terminating/deactivating data processing.

In the part of data classification, two machine learning techniques were used i.e. R-studio and Weka. Both of the ML techniques functioned as data mining for classification analysis in the smart factory. Data input for machine learning was datasets which contained IoT bots. Such datasets was selected based on the most 10 common features of IoT botnet attacks to the smart factory. These 10 common features reflects in the network architecture and network type used in the sensing/physical layer of the smart factory ,smart home network and smart city network [33].

Since the model testing was still going on for the training purpose, details in the network architecture, network parameter and feature selection used for training will be shown in later publication with the result of data analysis as a support.

5 Results and Discussion

Smart Factory (SF) security is a rapidly growing and expanding manufacturing sector. Resulting from the gap in literature and limitation of the current research, a combination of honeypot and machine learning into one botnet detection model was proposed to investigate for its feasibility. One of the most important factors in this study was the classification speed of the machine learning using the log file dataset. The proposed model was simulated in a virtual smart factory being created based on a hardware configuration with IoT devices and two Raspberry Pi. A comparison with some recent study approaches was discussed from a performance perspective with some suggestion for further study.

Two machine learning classification techniques (R-studio and Weka) were employed to efficiently distinguish innocuous traffic from botnet. The results of both techniques were compared. The weka technique was written with a number of default classifiers using other randomly selected seed values. The final prediction was a simple average of all predictions made by the underlying classifier. The idea of using R-studio was taken from a recent study [34].

Through this, it combines the results of the classifier used with various seed values to improve the results and reduce the likelihood of errors. The R-studio technique formed a decision tree-based classifier that maintains the highest accuracy for educational data and improves generalization accuracy as complexity increases. The classifier consists of several trees that are systematically constructed by selecting a subset of the components of the shape vector (i.e. a tree consisting of randomly selected subspace).

A previous experimental research using WEKA written in Java to test the aforementioned machine learning, is a collection of machine learning algorithms accessible by Java code or GUI [35]. The dataset for the experiment was prepared as follows. Random parts of the data from all sources were combined together. This choice was carried out randomly to maintain authenticity and unbiased results even when working on smaller datasets. Large datasets required small datasets for analysis because they required processing power not available in personal computer systems. Combining different datasets also provides data diversity when

compiling different types of botnet traffic flows from different types of environments. Several factors were used to determine the efficiency of the study approach and determine the results of the experiment. The algorithm was evaluated based on accuracy, which is the percentage of the correct results predicted by the classifier. This percentage represented the percentage of results flagged as botnet traffic but actually marked as normal traffic. Calculation of accuracy and fall positive rate are as followed.

$$Accuracy\ ACC = \frac{TP}{TP+FP} \quad (1)$$

$$False\ positive\ rate\ FPR = \frac{FP}{FP+TN} \quad (2)$$

whereas, TP (True Positive): number of botnet packets labelled as botnet, FP (False Positive): number of normal packets labelled as botnet, TN (True Negative): number of normal packets labelled as normal traffic and, FN (False Negative): number of botnet packets labelled as normal traffic. Calculation of false positive rate is as followed.

Table 3. R-Studio and Weka results in classification.

ML Techniques	Accuracy	FPR	p-value
Weka	0.953	0.219	n/a
R-studio	0.96	0.2667	0.05

Table 3 shows the results in term of p-value, FPR and accuracy of the R-studio and Weka techniques. Accuracy and FPR of the Weka technique are 0.953 and 0.219 respectively, which is considered as a good result. FPR of the Weka is lower than that of the R-studio at 0.2667. Notably, the p-value of R-studio is only 0.05, which is understood that 95% of the botnet classification results from the use of R-studio technique in the honeypot combined machine learning model is statistically significant.

6 Conclusion

The purpose of the present study was to present a model for botnet detection and rapid botnet classification in smart factories by combining honeypot and machine learning together. With the dataset collected in the log file, this proposed model is expected to efficiently minimise failure of botnet detection and information leakage in smart factory. In addition, using two ML classification techniques (R-studio and Weka), results were obtained for high accuracy, p-value, false positive of botnet tracking. Setup of the hardware configuration was very useful in this study to simulate the operation of smart factory and conduct the investigation of the honeypot combined machine learning model.

7 Future Work

In future work, time taken build the model will be measured appropriately. Accuracy, false positive ratio and p-value are good measures to evaluate the results. It is suggested that more working parameters should be considered to include in order to increase the satisfaction and expectation of the capacity/scale of smart factory. Additionally, optimisation of this model is suggested to test in a real factory environment for a real-time evaluation.

References

1. M. S. Smith, Inf. Manag. J. **49**, 36 (2015)
2. E. Casalinuovo, Trop. Comment. **4**, 1 (2019)
3. K. S. H. Ramos, M. A. S. Monge, and J. M. Vidal, Sensors (Switzerland) **20**, 1 (2020)
4. C. Kelly, N. Pitropakis, S. McKeown, and C. Lambrinoudakis, 1 (2020)
5. W. Jiafu, T. Shenglong, S. Zhaogang, L. Di, W. Shiyong, I. Muhammad, and V. V. Athanasios, IEEE Sens. J. **16**, 7373 (2016)
6. N. E. Vaskenly and M. Dhanya, Int. J. Pure Appl. Math. **118**, 505 (2018)
7. F. Galati and B. Bigliardi, Comput. Ind. **109**, 100 (2019)
8. Z. Zhang, Y. Zhang, J. Lu, X. Xu, F. Gao, and G. Xiao, (2018)
9. B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, IEEE Access **6**, 6505 (2017)
10. L. Seungjin, A. Abdullah, and N. Z. Jhanjhi, Int. J. Adv. Comput. Sci. Appl. **11**, 418 (2020)
11. M. Ghobakhloo, J. Manuf. Technol. Manag. **29**, 910 (2018)
12. Y. Liu, Y. Peng, B. Wang, S. Yao, and Z. Liu, IEEE/CAA J. Autom. Sin. **4**, 27 (2017)
13. A. Valente, S. Baraldo, and E. Carpanzano, CIRP Ann. - Manuf. Technol. **66**, 17 (2017)
14. Nguyen, 1 (2017)
15. H. Nguyen, Q. Ngo, D. Nguyen, and V. Le, ICT Express (2020)
16. R. S. and A. Thakral, *A Review of Various Mechanisms for Botnets Detection* (2018)
17. S. Mulik and A. Patil, **6**, 108 (2019)
18. A. Aziz, IEEE Trans. Aerosp. Electron. Syst. **47**, 2208 (2011)
19. F. Gerstmayer, J. Hausladen, M. Kramer, and M. Horauer, 2017 12th IEEE Int. Symp. Ind. Embed. Syst. SIES 2017 - Proc. (2017)
20. A. Mathematics, **116**, 73 (2017)
21. D. S. Terzi, R. Terzi, and S. Sagiroglu, 2nd Int. Conf. Comput. Sci. Eng. UBMK 2017 592 (2017)
22. J. Zhen and Z. Liu, Proc. - 2012 IEEE Symp. Robot. Appl. ISRA 2012 627 (2012)
23. S. T. Park, G. Li, and J. C. Hong, J. Ambient Intell. Humaniz. Comput. **11**, 1405 (2020)
24. S. K. Choi, C. H. Yang, and J. Kwak, KSII Trans. Internet Inf. Syst. **12**, 906 (2018)
25. W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, Inf. Sci. (Ny). **511**, 284 (2020)
26. R. Vishwakarma, 2019 3rd Int. Conf. Trends Electron. Informatics 1019 (2019)
27. P. Duessel, C. Gehl, U. Flegel, S. Dietrich, and M. Meier, Int. J. Inf. Secur. **16**, 475 (2017)
28. G. Fedynyshyn, M. C. Chuah, and G. Tan, Lect. Notes Comput. Sci. (Including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) **6906 LNCS**, 228 (2011)
29. N. C. Rowe, Auton. Cyber Decept. 35 (2019)
30. A. Noaman, A. Abdel-Hamid, and K. Eskaf, 2019 Int. Conf. Innov. Intell. Informatics, Comput. Technol. 3ICT 2019 1 (2019)

31. S. Morishita, T. Hoizumi, W. Ueno, R. Tanabe, C. Ganan, M. J. G. Van Eeten, K. Yoshioka, and T. Matsumoto, 2019 IFIP/IEEE Symp. Integr. Netw. Serv. Manag. IM 2019 134 (2019)
32. C. Dalamagkas, P. Sarigiannidis, D. Ioannidis, E. Iturbe, O. Nikolis, F. Ramos, E. Rios, A. Sarigiannidis, and D. Tzovaras, Proc. 2019 IEEE Conf. Netw. Softwarization Unleashing Power Netw. Softwarization, NetSoft 2019 93 (2019)
33. M. Humayun, N. Z. Jhanjhi, M. Z. Alamri, and A. Khan, Employing Recent Technol. Improv. Digit. Gov. IGI Glob. 87 (2019)
34. S. H. Kok, A. Abdullah, and N. Z. Jhanjhi, J. King Saud Univ.Comput. Inf. Sci. (2020)
35. L. Mathur, M. Raheja, and P. Ahlawat, Procedia Comput. Sci. **132**, 1668 (2018)