Classification of Botnet Attacks in IoT Smart Factory Using Honeypot Combined with Machine Learning

Seungjin Lee[1], Azween Abdullah[1], NZ Jhanjhi.[1], Kok Sim Hoong[1]

[1]School of Computer Science and Engineering, Taylor's University, No. 1 Jalan Taylor's, 47500, Subang Jaya, Selangor, Malaysia

Corresponding Author:
Seungjin Lee[1]
School of Computer Science and Engineering, Taylor's University, No. 1 Jalan Taylor's, 47500, Subang Jaya, Selangor, Malaysia
Email address: leephael8707@gmail.com

Abstract

Industrial revolution 4.0 with the breakthrough technological advances in 5G, and artificial intelligence has innovatively transformed the manufacturing industry from digitalization and automation to the new era of smart factories. A smart factory can do not only more than just produce products in a digital and automatic system, but also is able to optimize the production on its own by integrating production with process management, service distribution, and customized product requirement. A big challenge to the smart factory is to ensure that its network security can counteract with any cyber attacks such as botnet and DDos (Distributed Denial of Service), They are recognized to cause serious interruption in production, and consequently economic losses for company producers. Among many security solutions, botnet detection using honeypot has shown to be effective in some investigation studies. It is a method of detecting botnet attackers by intentionally creating a resource within the network with the purpose of closely monitoring and acquiring botnet attacking behaviors. For the first time, a proposed model of botnet detection was experimented by combing honeypot with machine learning to classify botnet attacks. A mimicking smart factory environment was created on IoT device hardware configuration. Experimental results showed that the model performance gave a high accuracy of above 96%, with very fast time taken of just 0.1 milliseconds and false positive rate at 0.24127 using random forest algorithm with Weka machine learning program. Hence, the honeypot combined machine learning model in this study was proved to be highly feasible to apply in the security network of smart factory to detect botnet attacks.

Introduction

39   The Industrial Revolution 4.0 has brought a great innovation to the conventional manufacturing
40   into the new era of smart factories [1]. The conventional factories involve automation or
41   digitalization within each production process. This however make it very difficult to manage the
42   entire production chain from general to specific levels. More innovatively, smart factory can
43   effectively manage many processes in the production chain thanks to the use of many IoT
44   (Internet of Things) devices. They are installed and interconnected with each other in every
45   machine or equipment along the production chain. Hence a smart factory is advantageous in
46   producing a variety of products according to customer's desire at better quality and higher
47   productivity. And, IoT devices/equipment play a very important role in the operation and
48   management of smart factories. A demand for the IoT equipment in smart factories has been
49   increasing since 2012 as shown in Figure 1. Especially, in the last five years (2015-2020), the
50   use of IoT devices has increased tremendously from 18.2 billion to 50 billion for application in
51   the smart factories [2].
52
53   Additionally, as smart factories are combined with ICT (Information and Communications
54   Technology), all the facilities and devices are connected at the central wireless communication.
55   This allows data to be freely linked between the processes and provides a more systematic,
56   integrated and optimal production environment. Efficiency in time management for production
57   can be greatly enhanced with a minimal production cost. Therefore, products produced by smart
58   factories become more competitive in the market.
59
60   Although IoT smart factories have been built and operated in the industry, standards of
61   implementation for smart factories have yet to be established [4]. Basically, a smart factory
62   consists of three aspects, i.e. interconnection, collaboration and execution, which all attribute to
63   the manufacturing conceptualization of being adaptive and flexible [5]. This concept is reflected
64   in the architecture of the smart factory operating on IoT system as shown in Figure 2 [10]. With
65   four layers arranged hierarchically, it starts at the physical resource layer, followed by the
66   networking layer and the application layer, and ends at the terminal layer. A manufacturing
67   system in the smart factory can be assessed from different layers [6]. With the aim of
68   transforming conventional factories into smart factories, in-depth research needs to look into all
69   layers.
70
71   From the security perspective, research should focus more on the physical resource/sensing
72   layer. As it is directly related to the vast usage of theIoT devices in order to reinforce the security
73   network for smart factories. Finding any security-related issues is one of the priorities required
74   for a smooth system operation by means of resolving any failover problems arising from the
75   entire manufacturing chain [7]. Especially, the IoT devices are such as RFID (radio-frequency
76   identification), CCTVs, PLC (programmable logic controller) equipment, sensor and main
77   database servers are installed or located at the physical resource layer. Data transmission

78   between these IoT devices can be easily affected in case that data leakage in smart factory
79   network occurs. In the worst case, data updates can be abused by unauthorized users [8].
80
81   To mitigate the impact of data leakage and data abuse, real-time detection of cyber attacks to
82   smart factory obviously becomes an extremely important factor to take into consideration of
83   developing and improving security network of the smart factory [9].
84
85   Network security in the smart factory is highly at risk of being under cyber attacks due to the
86   interconnection of a huge number of IoT equipment. According to a recent report, instability is
87   recognized as one of the biggest limitations out of 250 vulnerable features found in the IoT
88   devices [3]. As a result, cyber attack to smart factories can easily spread to not only quality
89   process control and production control, but also product design which can be analyzed or copied
90   by unauthorized or illegal accesses. In the worse scenario, highly confidential information such
91   as process know-how, requirements of data analysis, product design drawings, R&D results are
92   shared outside of the smart factory. Such threats of information leakage can cause serious
93   damages and economic losses to both manufacturing and business sectors. This kind of cyber
94   attack can be done by the act of exploiting security vulnerabilities in the ICT system via remote
95   control or surveillance of systems in the IoT smart factory.
96
97   One of the most serious cyber attacks to smart factory is botnet. An example of the botnet attack
98   is a temporary unavailability on some commercial websites such as Amazon, Netflix, Twitter,
99   CNN and PayPal. A notable case ever recorded is the attack on the Dyn DNS infrastructure,
100   which mobilized 100,000 IoT devices (mainly CCTV cameras) in October 2016. Another
101   example is the new Mirai-source-code being launched in 2017. These Mirai-induced IoT botnets
102   have occurred frequently in the recent years with a very serious consequence. Therefore, it is
103   very important and urgent to identify and mitigate IoT botnets through the development of new
104   technologies for network security [11]. As the number of attacks has soared due to unstable IoT
105   devices in the Internet infrastructure, smart factories undoubtedly are highly possible to become
106   an ideal victim of the IoT botnet attack.
107
108   Among many detection methods, honeypot has been investigated to apply for detecting botnet
109   attack in various studies in the recent years[12] . However, a huge volume of attacking data
110   collected by honeypot is highly complex and non-classified. This causes to lower the efficiency
111   of botnet detection by the honey method in term of time taken and accuracy. In order to improve
112   the efficiency, it is crucial to focus on classifying botnet attacking information and obtaining
113   botnet attacking behaviors (intrusion type in other words). In this particular area of dealing with
114   big data, artificial intelligence or machine learning has recently been applied effectively to speed
115   up data processing, and make prediction as well as detection [13]. Hence, it becomes very
116   potential to apply machine learning for botnet classification, which is notably yet to be
117   investigated in previous studies, especially in the smart factory environment.

118

119 Therefore, this study was aimed to investigate the feasibility of combining honeypot with
120 machine learning in developing a botnet detection model for IoT smart factories. In this work, a
121 configuration of hardware representing a physical layout of a smart factory was built and
122 installed with software of honeypot combined machine learning. The whole setup was then
123 programmed to run for simulation to detect and classify botnet attacks into intrusion types.

124

125 Problem statement
126 The problem statement can be elaborated with following points,

- 127 • A huge amount of attacking data collected by honeypot is highly complex.
- 128 • Without data classification, efficiency of the honeypot model is low, since the current time
- 129   taken is long but at low accuracy to detect botnet.
- 130 • A very limited number of studies focused on botnet detection for the smart factory.
- 131 • Strategy of using honeypot with machine learning has been suggested very recently with
- 132   only study framework and lack of model verification for supporting.

133 Study approach

- 134 • Apply machine learning as a supporting tool for classifying botnet attacks captured in log
- 135   files generated by honeypot. Selection of random forest algorithm for machine learning to
- 136   improve the classification process. Model testing on a hardware configuration mimicking
- 137   a real smart factory environment.

138

139 **Related work**
140 Various botnet detection methods and their rationale are described in the Taxonomy in Figure 3.
141 Besides, a smart factory is layered into the perception (physical), communication, network, data
142 and applications with the function of each, as shown in the following taxonomy of the same
143 figure.

144

145 Comparison of various detection models are presented in Table 1. Real-time detection is a very
146 important factor which smart factories seek for [14].

147

- 148 • Honeypot can respond to attacks in real time and attract attackers to deceptive assets
- 149   instead of actual assets [15]. Whereas for binary, anomaly detection methods, the
- 150   response to real-time is however slower than that of the honeypot method [16] [17].

151

- 152 • Binary detection is simple in the structure of using 1 and 0. But its detection processing
- 153   speed is too low to be compatible to smart manufacturing environment which seeks real-
- 154   time detection [14]

155

156   • Although the C&C (command and control) method targets for HTTP-based botnet
157   detection and expansion, its structure is not simple to implement and the detection result
158   is high false-positive [18].
159   • In terms of cost effectiveness, honeypot has an advantage since it requires a relative low
160   cost of construction and management [19].
161
162 With an increasing interest in the potential application of machine learning, it offers a new
163 solution for detecting abnormalities in the malicious Internet traffic. In fact, the Internet traffic
164 which allows communication between IoT devices is distinguished from the Internet
165 connectivity which runs on a variety of web server Many Internet-connected devices are such as
166 smartphones, computer, laptops using a variety of web servers. Moreover, for the IoT devices,
167 patterns of the network traffic are repeated in the regularity of network ping with small packets
168 for logging.
169
170 Applying machine learning in botnet detection for smart factories can become useful to enhance
171 performance of the honeypot model in term of speeding up the processing time or detection time
172 [21]. Interestingly, it is notified that very limited studies making attempts to mount both
173 honeypot and machine learning on IoT device networks to target attack on the IoT traffic.
174
175 Table 2 summarizes a few studies in botnet detection using the approaches of honeypot and/or
176 machine learning.
177   • IoT botnet detection is an approach used to design a detection model based on the binary
178   when botnet attacks IoT device as a hypothesis [22]. Although monitoring algorithms for
179   the infected IoT device are simple and easy through web services, capacity of the IoT
180   devices has certain limits as a restriction in the IoT botnet detection.
181
182   • Another approach to detect botnet is using machine learning which gave a high accuracy
183   in detection at 91.66% [23]. However one disadvantage of using machine learning
184   approach is that fast detection is hard to achieve in the randomized number of packets.
185   Consequently, the feasibility of applying this approach for smart manufacturing needs
186   more research looking into real-time factor and accuracy.
187
188   • Botnet detection using honeypot integrated with IoT, named as IoT honeypot was
189   studied in the environment of smart factories [24]. In comparison with the machine
190   learning approach by Wang et al. (2020), the IoT honeypot approach is able to gather
191   information at high speed with less resource consumption [5]. Although the IoT
192   honeypot approach has been shown to be scalable by applying it to sandboxes IoT to
193   support high protocols, more expansion in various situations and environments is needed
194   with features to activate the architecture of the IoT devices [5].
195

196   • One study suggested to apply machine learning to detect botnet in the smart factory
197   environment [25]. On the one hand, the machine learning approach can reduce the cost
198   as an advantage. On the other hand, low detection rate and high complexity and
199   uncertainty are recognized as big limitations. Thus, it might not suitable for smart
200   factories, unless a construction of machine learning with Kenta-aware intrusion tower
201   system is built, which bears an additional cost.
202

203   • Advancing from the IoT honeypot approach and the machine learning approach,
204   honeypot combined with machine learning named as honeypot machine learning uses
205   learning logging for detection and tracking at high accuracy [26]. In accordance with
206   most standard equipment at various functions, the honeypot machine learning approach
207   is suitable for the performance of smart factory with a minimal resource required.
208   Hence, it is likely to be adopted in the future [26].
209

210 Among those approaches being discussed, IoT botnet and honeypot machine learning approaches
211 shows some effective results in detecting botnets. These two approaches are possible to trace
212 through logging at low cost and are most cost-saving for the IoT devices. Notably, feasibility of
213 applying the honeypot machine learning approach in the smart factory especially has yet to
214 investigate in any studies so far as being highlighted in the recent review work [13]. Therefore,
215 more research in botnet detection should look insight into this particular area for application in
216 smart factory.
217

Materials & Methods

219 **Proposed model**
220 This section is organized to focus on three main aspects. The first and second aspects mention
221 configuration and simulation of hardware in a virtual smart factory environment. The last aspect
222 presents algorithms of the honeypot detection model in combination with machine learning
223 programming.
224

225 **Configuration of hardware for a virtual smart factory environment**
226 In the configuration setup, some IoT devices (camera, RFID, temperature sensors) and two
227 raspberry pie devices (Pi1 and Pi2) were used to create a virtual smart factory environment.
228

229 The first Raspberry Pi (Pi1) was assigned as the actual main IoT data collection server by
230 installing Open CVS. It was responsible for transmitting collected IoT data to the main PC. T-pot
231 platform was chosen because it was suitable for virtual experiments using raspberry and allowed
232 to monitor real-time botnet detection through dashboards. The second raspberry pi (Pi2) was
233 installed with a virtual server (VM) (T-pot platform) So that collection of detection information
234 in such the environment was deliberately established. One assumption was that botnet attacked
235 the raspberry pi 2 (honeypot server) using 10 feature botnet datasets. Raspberry Pi1 and Pi2 were

236 installed on a log server to keep the information on IoT product line in the factory and records of
237 botnet attack pattern time zone, making it easy to track.
238
239 Operating mechanism of the honeypot combined machine learning model in smart factory is
240 illustrated in Figure 4. When an attacker attempted to inject a malicious code through an open
241 port. This step was done by logging into an IoT device at the physical resource/perception layer
242 by combining multiple IDs and passwords. The honeypot intentionally broke into his protective
243 wall and came in as a person who could reach the attacker. The main intention was to obtain
244 information about attackers and malicious code botnets by recording each activity between the
245 device and the intruder in the form of a log file. These log files captured information that allowed
246 administrators to identify characteristics, transformations, target device types, C&C server IP
247 addresses, port numbers of the new malware suites or botnets. Log file data was then converted
248 to an appropriate table format that can be used as a dataset.
249
250 A machine learning (ML) tool was then trained with the dataset which contained 10 network
251 parameters. These parameters were based on the most 10 common features of IoT botnet attacks
252 to smart factory reported in the previous work studying in network architecture and network type
253 used in the physical layer of smart factory, smart home network and smart city [27] [28] [29].
254 Algorithm written for this ML tool classified botnet data using R-studio and Weka. Memory-
255 efficient classification was desirable to predict useful information by using less training data to
256 prevent IoT devices from becoming overwhelming. Afterwards, appropriate measures were taken
257 according to the results of the classification. Whenever the course exceeded the allowed size of
258 training data, it would dynamically repeat.
259
260 **Simulation of hardware design with raspberry Pi transfer log files**
261 Each product in the smart factory was attached with a RFID (Radio-frequency identification) tag
262 containing information. Camera reads the RFID tags to collect product information. The
263 collected information was then stored in the raspberry Pi1 and Pi2 as the calling terminals. In
264 other words, it was programmed to transmit the collected data to the IoT service server of the
265 network through a terminal (camera, temperature, and RFID – IoT devices). The information
266 kept in Pi1 and Pi2 as a log file was then transferred to the server of the main PC. This process
267 simulation is illustrated in Figure 5.
268
269 **Raspberry Pi1 setting - Data transporting Open CV**
270 Transporting and receiving of data in this virtual environment werer similar to those taking place
271 in the real smart factory. Data transporting Open CV was used as a collection of Python classes
272 that transferred Open CV images and data from the raspberry Pi1 to the main computer via Data
273 transporting Open CV messages.
274

275   For example, on the main computer screen, video and picture streams were shown
276   simultaneously by sending signal data of raspberry Pi1 as shown in Figure 6. Algorithm was
277   required for the main PC and raspberry Pi1 for such data transfer as shown in Table 3.
278

Raspberry Pi2 setting - T-pot honeypot platform

279   **Raspberry Pi2 setting - T-pot honeypot platform**
280   After Raspberry Pi1 setting with data transporting Open CV, Raspberry Pi2 was set with T-pot
281   honeypot platform followed by virtual machine (VM). Verification and testing were done on all
282   the honeypot to be balanced at runtime. To do this task, a studio state command line was used to
283   write script and verify the transmission of log file.
284

285   The script in Figure 7 shows the load on the platform, the status of each honeypot, and the
286   uptime. Furthermore, the data collected by the honeypot was visually displayed using the Kibana
287   dashboard showing network attacked by malicious users and botnet. The Kibana dashboard
288   shown in Figure 8 is convenient and comprehensive for analysis of the type, location, and
289   malicious threat actors of botnet attacks. It infiltrated the Raspberry Pi server inside the VM.
290   This had many potential uses for data systems and metrif collection in smart manufacturing
291   environment that required real-time monitoring.
292

293   **Honeypot and machine learning classification process design and algorithms**
294   Diagram in Figure 9 describes the process design for botnet detection by honeypot combined
295   with machine learning classification model. The entire process consisted of two stages. In the
296   first stage (honeypot simulation), it took place at the raspberry Pi2 server which finished its
297   loading by checking botnet credentials and then started the honeypot detection in the T-pot
298   platform.
299

300   To verify botnet credentials in this step, a user name and password must be provided. Accurate
301   information given by authorized users would be proceeded to start honeypot detection. The
302   verification step took approximately 60 seconds.
303

304   Moving to stage 2 – log data collect, data which was earlier obtained from the authorized users in
305   stage 1, was being processed through a series of steps, i.e. retrieve logs, extract records, filter,
306   text processing, upload, end automated process. During the text processing step before
307   uploading, this was where machine learning was integrated with the honeypot for the purpose of
308   classifying botnet attacks as illustrated in Figure 9.
309

310   As a result of classification, botnet attacking behaviors or botnet intrusion types were obtained
311   and therefore used for machine learning training to detect botnets. After machine learning
312   classification, processed data was converted to output result file and uploaded to end the
313   automatic process. Algorithm for botnet classification by machine learning is shown in Table 4.
314

315   Instruction to verify the codes and dataset:

316   1. Read the package setting for CARET, DPLYR, and READR in the library (this package
317   can balance the dataset for four categories of botnet attacks).

318   2. Make setting in the computer for dataset with frame set and honeypot log file.

319   3. Start the test based on the 10 best feature data and wait until the true setting to come out.

320   4. Proceed to filter, extract records, review logs and test processing.

321   5. Apply (flgs_number, srate, drate, rate, max, state_number, mean, min, stddev, seq) to
322   each classified category.

323   6. Obtain log files as samples for machine learning classifying into four types of botnets
324   (DDos, Dos, Reconnaissance and Theft).

325   7. Call in the algorithm (random forest or SVM) to start classification.

326   8. Predict botnets based on the classification result in term of accuracy, time taken, false
327   positive rate, and p-value.

328

329   **Results**
330   In this section, results were collected mainly from the machine learning classification of botnet
331   attacks. The data collection for the experiment was randomized if the reference was used. The
332   experiment was carried out using Raspberry Pi and a personal computer. Features of the selected
333   dataset were first described in Dataset section. To perform the classification on the dataset,
334   honeypot combined machine learning model simulation was run by two techniques i.e. Weka,
335   and Rstudio. The collected results were evaluated by comparison between the two, and with
336   other study.
337   **Dataset**
338   In order to use a machine learning method to identify botnet as the target of IoT-based network
339   and physical layer in smart factories, we experimented with data set 10 features of this
340   paper[30]., which is the most suitable data set for this study.
341
342   Dataset used for classification was selected based on ten features which were closely related to
343   the botnet intrusion types. Details of the 10 features are presented in Table 5, which was
344   extracted from a direct comparison of entropy and correlation scores in the previous study [31]
345   [30]. Specifically, transmission formation was calculated as correlation indices which were
346   evaluated for their statistical measurement values.
347
348   The calculation of indices was performed using the following equation.

$$x_i = (x_i - x_{min}) * \frac{(b - a)}{(x_{max} - x_{min})} + a$$

349

350

351 Model simulation was evaluated using some of the evaluation metrics of machine learning as
352 shown in Table 6 [30]. Based on the results, the model can be evaluated whether it is highly
353 efficient in optimization and able to reduce the error of drate. The values of max and min were to
354 represent the values of training and response. Examples of correlation are in Figure 10 [30]..
355

356 **Classification by Weka-machine learning**
357 Classification result of the Weka-machine learning technique for four types of botnet attacks
358 namely DDoS, DoS, reconnaissance, and theft is shown in Figure 11. For 76 instances, the
359 average percentage of correct prediction (accuracy) for all four botnet attacks achieved
360 96.00526%. The kappa statistic showed the model stability was 0.9466 with a mean absolute
361 error of just 0.0478. In term of accuracy and precision for each type of attacks, reconnaissance
362 has the highest values.
363

364 Furthermore, after collecting pcap files from the virtual settings, statistical measures using
365 correlation coefficients and entropy techniques were adopted to extract flow data using Argus
366 tools in order to evaluate datasets based on the 10 best features. A new function was created
367 based on the transaction flow of network connections to find out normal and intrusive events. A
368 machine learning model has been trained and validated in different versions of datasets to assess
369 the value of datasets compared to other datasets as shown in Figure 12.
370

371 **Classification by Rstudio-machine learning**
372 Figure 13 shows the results of classification using Rstudio machine learning technique. Two
373 methods were used, i.e. support-vector machine (SVM) and random forest (RF). RF was
374 calculated using the decision tree (DT) to predict mean values. RF was selected because it
375 showed the effective detection in discrete datasets such as botnet [32]. Evaluation of the
376 classification result were based on nine parameters namely, sensitivity, specificity, pos pred
377 value, neg prered value, prevalence, detection rate, detection prevalence, balance accuracy,
378 average.
379

380 In the RF method, a high accuracy was achieved at 0.96. The 96% CIs were 0.8875 and 0.9917.
381 The RF provides Kappa at 0.964. However, for SVM method, the accuracy was obtained at
382 0.7733 which is much lower than that of the RF. The 96% CIs were 0.6666, 0.8621. As can be
383 seen in Figure 13, classification of DDoS attack type is highly efficient using Rstudio with
384 respect to all 9 evaluating parameters. The detection rate and detection prevalence have low
385 probabilities of 0.24 and 0.25 respectively. The overall detection using R-studio showed a
386 statistically significant result since p-value is less than 5%.
387

388 # Discussion
389 The two machine learning programs (Weka and Rstudio) following the random forest algorithm
390 showed good result of classification and comparable with another study as shown in Figure 14.

391   In the study of Mathur *et al.*, botnet were detected via mining of the network traffic flow with
392   random committee method [33]. The resultant accuracy of the random committee was achieved
393   at 95.3%, which was 1.3% lower than those obtained in this study at 96.66667% for random
394   forest-Weka and 96% for random forest-Rstudio. In term of time taken, both random forest-
395   Weka and random committee were able to detect botnets within a very short time of 0.00001
396   seconds or 0.1 milliseconds. Whereas it took a very much longer time of 19.37 second for the
397   random forest-Rstudio to detect botnets because of its program package. In addition, the p-value
398   representing the significance of the detection models, p value to the random forest-Rstudio was
399   less than 5% (2.2 x 10-16), which showed that the detection model is statistically significant. As
400   mentioned earlier that real time detection is the key factor importance to the network security of
401   the smart factory, random forest method is therefore considered to be highly suitable for the
402   environment of smart factory operating 24/7 in time.
403
404   The result of this study can be said to be just relatively comparable with the work of Mathur *et*
405   *al.* (2018), since both of the studies were based on network traffic flow and targeting at the
406   botnet attack [33]. However, for the feasibility to apply in the smart factory environment, this
407   study shows to be more feasible because of two reasons. First, the experiments were conducted
408   on simulation of hardware specifically configured to mimic the real smart factory environment
409   using IoT devices as mentioned in the proposed model section, which is lacked in the work of
410   Mathur et al. (2018). Secondly, the experimental results obtained in this study addressed directly
411   to the three deciding factors (time taken - 0.1 millisecond, accuracy – above 96% and FPR –
412   0.24127, refer to Table 7) which are very useful for evaluating any tested methods being applied
413   in the smart factory.
414
415   Whereas the latest publication work in 2020 reported that their study was based on smart factory
416   ambient/environment to detect context-aware intrusion using machine learning [34]. But there
417   was no mention in that study to target any specific types of cyber attacks or virus, but only for
418   anomaly signs. Another limitation of the study is that its result just stated a very general
419   possibility of process improvement of 29% from 1.29% [34]. Without showing the three
420   deciding factors (time taken, accuracy and FPR), it can be hardly possible to evaluate the
421   feasibility to apply for the smart factory. In addition to this point, the part of using machine
422   learning used for training to obtain intrusions did not mention to include the best features of
423   smart factory in the datasets. If including it would be very helpful to increase the feasibility or
424   applicability of any detection models at the physical layers with interconnection of many IoT
425   devices, as this present study were conducted accordingly. Furthermore, the Kibana platform
426   supporting for a visualization of the system/model performance could provide a user-friendly
427   interface for the administrators in the smart factory to analyze from a variety of perspectives
428   more than just a visible display.
429

430   This study had provided a basic background for developing a security network just for the smart
431   factory environment with a mimicking IoT device hardware configuration and random
432   algorithms for the experimental work. The results can be used as reference points or benchmarks
433   for more comparison with other future studies relating to smart factory. In fact, the number of
434   studies focusing on the security network for smart factory to target specifically botnet attacks
435   using honeypot is currently scarce throughout the literature, future work can based on this smart
436   factory hardware configuration design for the experimental testing for models or systems. Also,
437   it is suggested to further this study by conducting experiments in a real smart factory. By doing
438   that, a better result can be obtained for analysis when many factors of smart factories are taken
439   into consideration. Instead ,a controlled virtual smart factory environment was created in this
440   study. The expected results will be more valuable for improving the productivity of smart
441   factories.
442

Conclusions

444   In this work, the model of combining honeypot with machine learning was proved to be feasible
445   in detecting botnet in the smart factory. Since the botnet can be easily spread into IoT smart
446   factory environment with a high risk, hardware-based simulation and classification using random
447   forest algorithm for Weka machine learning program showed a very good result. 96.66667 % for
448   accuracy, 0.1 milliseconds were achieved for the proposed model to detect botnet and the FPR
449   was low at just 0.24127. Comparing this result to other studies showed that the proposed model
450   (honeypot combined machine learning to detect and classify botnet attack) in the smart factory
451   was evaluated to be better because of three outstanding advantages. First, IoT devices were used
452   in the hardware simulation configured to mimic the real smart factory environment. Second, the
453   result of model testing has showed that with a short time taken - 0.1 millisecond,high accuracy –
454   above 96% and low FPR – 0.24127 by the random forest Weka machine learning as the deciding
455   factors. Lastly, machine learning have been used the dataset which included the best 10 features
456   of the smart factory for training to obtain intrusions.
457

Acknowledgements

459   The author acknowledge the support of Taylor's University, School of Computer Science and
460   Engineer in carrying out this experimental research work. This research work has been
461   underwent for English proofreading prior to submitting to this journal.

462

References

464

465   [1]   E. Oztemel and S. Gursev, "Literature review of Industry 4.0 and related technologies," J.
466   Intell. Manuf., vol. 31, no. 1, pp. 127–182, 2020.
467   [2]   M. S. Smith, "Protecting Privacy in an IoT-Connected World.," Inf. Manag. J., vol. 49, no.
468   6, pp. 36–39, 2015.

469 [3]   E. Casalinuovo, "Thematic Investment Opportunity – Internet of Things," Trop. Comment.,
470   vol. 4, no. 1, pp. 1–4, 2019.
471 [4]   D. Guo, R. Y. Zhong, S. Ling, Y. Rong, and G. Q. Huang, "A roadmap for Assembly 4.0:
472   self-configuration of fixed-position assembly islands under Graduation Intelligent
473   Manufacturing System," Int. J. Prod. Res., vol. 58, no. 15, pp. 4631–4646, 2020.
474 [5]   Jiafu, Wan,Shenglong, Tang,Zhaogang, Shu,Di, Li,Shiyong, Wang, Muhammad,
475   Imran,Athanasios, V. Vasilakos., "Software-Defined Industrial Internet of Things in the
476   Context of Industry 4.0," IEEE Sens. J., vol. 16, no. 20, pp. 7373–7380, 2016.
477 [6]   X. Li, D. Li, J. Wan, C. Liu, and M. Imran, "Adaptive transmission optimization in SDN-
478   based industrial internet of things with edge computing," IEEE Internet Things J., vol. 5,
479   no. 3, pp. 1351–1360, 2018.
480 [7]   S. Mittal, M. A. Khan, D. Romero, and T. Wuest, "Smart manufacturing: Characteristics,
481   technologies and enabling factors," Proc. Inst. Mech. Eng. Part B J. Eng. Manuf., vol. 233,
482   no. 5, pp. 1342–1361, 2019.
483 [8]   K. S. H. Ramos, M. A. S. Monge, and J. M. Vidal, "Benchmark-based reference model for
484   evaluating botnet detection tools driven by traffic-flow analytics," Sensors (Switzerland),
485   vol. 20, no. 16, pp. 1–31, 2020.
486 [9]   Brett, Stone-Gross, Marco, Cova, Lorenzo, Cavallaro, Bob, Gilbert, Martin, Szydlowski,
487   Richard, Kemmerer, Christopher, Kruegel, Giovanni, Vigna, "Your botnet is my botnet:
488   Analysis of a botnet takeover," in Proceedings of the ACM Conference on Computer and
489   Communications Security, 2009, pp. 635–647.
490 [10] B. Chen, J. Wan, L. Shu, P. Li, M. Mukherjee, and B. Yin, "Smart Factory of Industry 4.0:
491   Key Technologies, Application Case, and Challenges," IEEE Access, vol. 6, no. c, pp.
492   6505–6519, 2017.
493 [11] M. Ozcelik, N. Chalabianloo, and G. Gur, "Software-Defined Edge Defense Against IoT-
494   Based DDoS," IEEE CIT 2017 - 17th IEEE Int. Conf. Comput. Inf. Technol., pp. 308–313,
495   2017.
496 [12] F. Ja'fari, S. Mostafavi, K. Mizanian, and E. Jafari, "An intelligent botnet blocking approach
497   in software defined networks using honeypots," J. Ambient Intell. Humaniz. Comput., no.
498   0123456789, 2020.
499 [13] L. Seungjin, A. Abdullah, and N. Z. Jhanjhi, "A review on honeypot-based botnet detection
500   models for smart factory," Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 6, pp. 418–435,
501   2020.
502 [14] G. Katz, P. Piantanida, and M. Debbah, "Distributed Binary Detection with Lossy Data
503   Compression," IEEE Trans. Inf. Theory, vol. 63, no. 8, pp. 5207–5227, 2017.
504 [15] P. Duessel, C. Gehl, U. Flegel, S. Dietrich, and M. Meier, "Detecting zero-day attacks using
505   context-aware anomaly detection at the application-layer," Int. J. Inf. Secur., vol. 16, no. 5,
506   pp. 475–490, 2017.
507 [16] F. Gerstmayer, J. Hausladen, M. Kramer, and M. Horauer, "Binary protection framework
508   for embedded systems," 2017 12th IEEE Int. Symp. Ind. Embed. Syst. SIES 2017 - Proc.,
509   2017.

510   [17]   F. Fenzl, R. Rieke, Y. Chevalier, A. Dominik, and I. Kotenko, "Continuous fields: Enhanced
511   in-vehicle anomaly detection using machine learning models," Simul. Model. Pract.
512   Theory, vol. 105, no. June, p. 102143, 2020.
513   [18]   G. Fedynyshyn, M. C. Chuah, and G. Tan, "Detection and classification of different botnet
514   C&C channels," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect.
515   Notes Bioinformatics), vol. 6906 LNCS, pp. 228–242, 2011.
516   [19]   A. Aziz, "A soft-decision fusion approach for multiple-sensor distributed binary detection
517   systems," IEEE Trans. Aerosp. Electron. Syst., vol. 47, no. 3, pp. 2208–2216, 2011.
518   [20]   W. Zhang, B. Zhang, Y. Zhou, H. He, and Z. Ding, "An IoT Honeynet based on Multi-port
519   Honeypots for Capturing IoT attacks," IEEE Internet Things J., vol. PP, no. c, p. 1, 2019.
520   [21]   M. Lim, A. Abdullah, N. Jhanjhi, M. Khurram Khan, and M. Supramaniam, "Link
521   prediction in time-evolving criminal network with deep reinforcement learning technique,"
522   IEEE Access, vol. 7, pp. 184797–184807, 2019.
523   [22]   S. K. Choi, C. H. Yang, and J. Kwak, "System hardening and security monitoring for IoT
524   devices to mitigate IoT security vulnerabilities and threats," KSII Trans. Internet Inf. Syst.,
525   vol. 12, no. 2, pp. 906–918, 2018.
526   [23]   W. Wang, Y. Shang, Y. He, Y. Li, and J. Liu, "BotMark: Automated botnet detection with
527   hybrid analysis of flow-based and graph-based traffic behaviors," Inf. Sci. (Ny)., vol. 511,
528   pp. 284–296, 2020.
529   [24]   S. Dowling, M. Schukat, and H. Melvin, "A ZigBee honeypot to assess IoT cyberattack
530   behaviour," 2017 28th Irish Signals Syst. Conf. IEEE, pp. 1–6, 2017.
531   [25]   S. T. Park, G. Li, and J. C. Hong, "A study on smart factory-based ambient intelligence
532   context-aware intrusion detection system using machine learning," Journal of Ambient
533   Intelligence and Humanized Computing, vol. 0, no. 0, Springer Berlin Heidelberg, p. 0,
534   2018.
535   [26]   R. Vishwakarma, "A Honeypot with Machine Learning based Detection Framework for
536   defending IoT based Botnet DDoS Attacks," 2019 3rd Int. Conf. Trends Electron.
537   Informatics, no. Icoei, pp. 1019–1024, 2019.
538   [27]   Y. Fan et al., "SNPL: One scheme of securing nodes in iot perception layer," Sensors
539   (Switzerland), vol. 20, no. 4, pp. 1–21, 2020.
540   [28]   Z. A. Almusaylim and N. Zaman, "A review on smart home present state and challenges:
541   linked to context-awareness internet of things (IoT)," Wirel. Networks, vol. 25, no. 6, pp.
542   3193–3204, 2019.
543   [29]   M. Humayun, N. Z. Jhanjhi, M. Z. Alamri, and A. Khan, "Smart Cities and Digital
544   Governance," Employing Recent Technol. Improv. Digit. Gov. IGI Glob., pp. 87–106,
545   2019.
546   [30]   N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of
547   realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT
548   dataset," Futur. Gener. Comput. Syst., vol. 100, pp. 779–796, 2019.

549 [31] Y. Zheng and C. Keong, A feature subset selection method based on high-dimensional
550   mutual information, vol. 13, no. 4. 2011.

551 [32] S. H. Kok, A. Abdullah, and N. Z. Jhanjhi, "Early detection of crypto-ransomware using
552   pre-encryption detection algorithm," J. King Saud Univ. - Comput. Inf. Sci., 2020.

553 [33] L. Mathur, M. Raheja, and P. Ahlawat, "Botnet Detection via mining of network traffic
554   flow," Procedia Comput. Sci., vol. 132, pp. 1668–1677, 2018.

555 [34] S. T. Park, G. Li, and J. C. Hong, "A study on smart factory-based ambient intelligence
556   context-aware intrusion detection system using machine learning," J. Ambient Intell.
557   Humaniz. Comput., vol. 11, no. 4, pp. 1405–1412, 2020.

558

Comparison of honeypot with other detection methods

Table 1. Comparison of honeypot with other detection methods.

	Honeypot	**Binary detection**	**Anomaly detection**	**C&C detection**
Configuration	High-interaction virtual server	Binary	P2P	Command & control server
Advantages	Monitor the interaction of the grid with infected devices User friendly UI system	Easily applicable to multi-connection systems	Systems have the capability to detect zero-day attacks as well	Able to detect and expand HTTP-based botnets.
Disadvantages	The analysis of information on an attack is slow and passive.	Spend a lot of time training	Not simple structure high false-positive	Not simple structure High false-positive
	[20] [15]	[16]	[17]	[18]

A comparison of studies in botnet detection using honeypot and/or machine learning approaches for smart factories.

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

Approaches	Strengths	Weaknesses	Research gap	Ref.
Machine learning for smart factory environment	Cost reduction	Detection rate is very low and the accuracy is low. The system is complicated.	Building machine learning and Kenta-aware intrusion tower systems for information that will be leaked from manufacturing processes	[25]
IoT Botnet	Monitoring Web-based real-time IoT equipment. Easy and simple interface	Limited capacity	New optimization requires expansion of utilization	[22]
Machine learning	91.66% graph-based detection accuracy	Difficult to apply flow-based detectors	A graph-based bot mark is required to increase the accuracy of botnet detection	[23]
IoT Honeypot	Speed of gathering information is fast. Less resource consumption.	Unnecessary data piles up	It is necessary to activate network protocol by expanding IoT equipment and sandbox.	[5]
Honeypot machine learning	Real-time monitoring with the combination of honeypot and machine learning	It is greatly affected by the system environment.	Problems with device data capacity cloud server application.	[26]

4
5
6

Algorithm of data transfer in raspberry Pi 1.

1   Table 3. Algorithm of data transfer in raspberry Pi 1.

Algorithm: Raspberry Pi image data transfer
1. *Input: List of data transfer*
2.   *Output: PI image data (task mapped to VM)*
3.   *Begin*
4.   *True: CV, VM. Show streamed images and data*
5.   *If*
6.   *two tasks get data from RFID, image, signal then*
7.   *Pick task with earliest*
8.   *Else*
9.   *Transfer data*
10.   *while VM*
11.   *Compute Utilization*
12.   *Sent image , signal each*
13.   *Repeat If data is available & task allocated then*
14.   *Migrate task to less utilized data*
15.   *Else*
16.   *Start scheduling*
17.   *Until all send images as stream*
18.   *Image = Pi Cam read*
19. *End*

Algorithm for data classification.

1 Table 1. Algorithm for data classification.

Algorithm: botnet-based classification ML and honeypot detection for improving the security of smart factory IoT
1. *Input*: List of Classified 10 feature-valued training data set (t1,t2,t3,...,tn) & VM
2. *Output*: 10 feature tasks DT
3. *Begin*
4. *Initialization*: DT; 1.Sort tasks according to DT ascendingly
5. *If*
6. DT belongs to same VM
7. TA = testing attribute
8. {combine DT =T;
9. *Else*
10. Priorities based on (Data set)
11. *For each VM* i= to n
12. *{calculate Information_gain}*
13. *Repeat If change is available factor & TA then*
14. *For* (Each DF in the splitting of TA)
15. *If* (T' is empty)
16. *Else*
17. Start get sample again
18. *Until* all 10 features allocated to a VM
19. *End*

10 Features of the selected dataset for classification.

1   Table 1. 10 Features of the selected dataset for classification.

No.	Name of features	Description
1	srate	Foundation-to-target t time packets for each second
2	drate	Target-to-foundation packets for each second
3	rate	Over-all packets for each second in transaction
4	max	Maximum period of collected archives Source
5	state_number	Numerical illustration of characteristic state
6	mean	Average period of collecting records
7	min	Minimum time of collecting records
8	stddev	Standard deviation of aggregated records Total
9	flgs_number	Numerical representation of feature flags
10	seq	Argus sequence number

Machine learning evaluation metrics.

1   Table 6. Machine learning evaluation metrics.

Accuracy	$ACC = \dfrac{TP}{TP + FP}$
Precision	$PPV = \dfrac{TP}{TP + FP}$
Recall	$TPR = \dfrac{TP}{TP + FN}$
Fall-out	$FPT = \dfrac{FP}{FP + TN}$

Note:

TP (True Positive): number of botnet containers represent as botnet

FP (False Positive): number of regular containers symbolized as botnet

TN (True Negative): number of normal containers represent as standard traffic

FN (False Negative): number of botnet containers symbolized as standard traffic

Result comparison between random forest with random committee algorithms

1 Table 7. Result comparison between random forest with random committee algorithms.

	Accuracy	Time taken (detection time)	Fall positive rate (FPR)	p-value	Ref.
Random forest-Weka	96.66667 %	0.0001 s	0.24127	-	This study
Random forest-Rstudio	96 %	19.37 s	0.219	0.05	
Random committee	95.3 %	0.0001 s	0.219	-	[33]

Figure 1

Demand for IoT equipment.
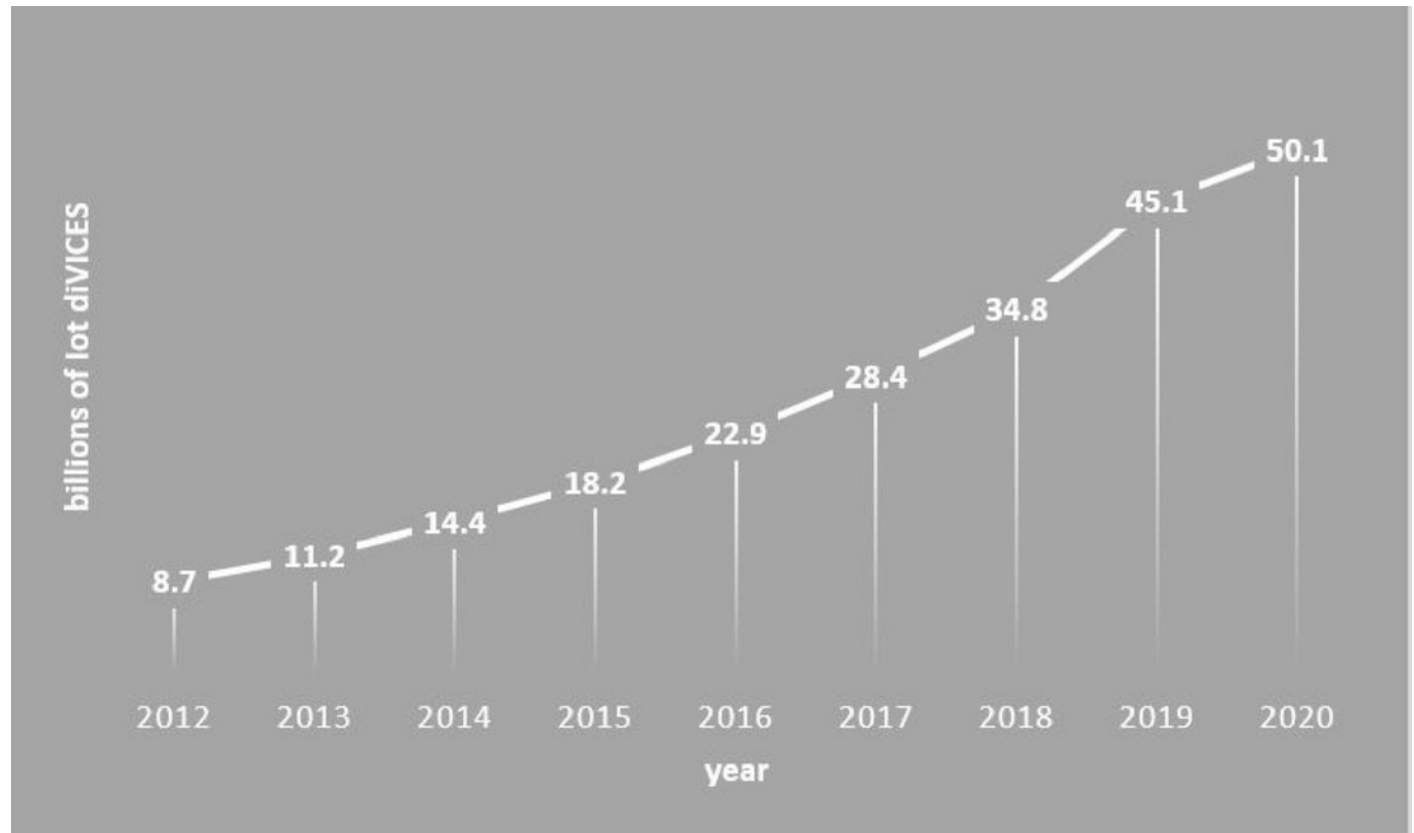
Figure 2

Function requirements of smart factory.
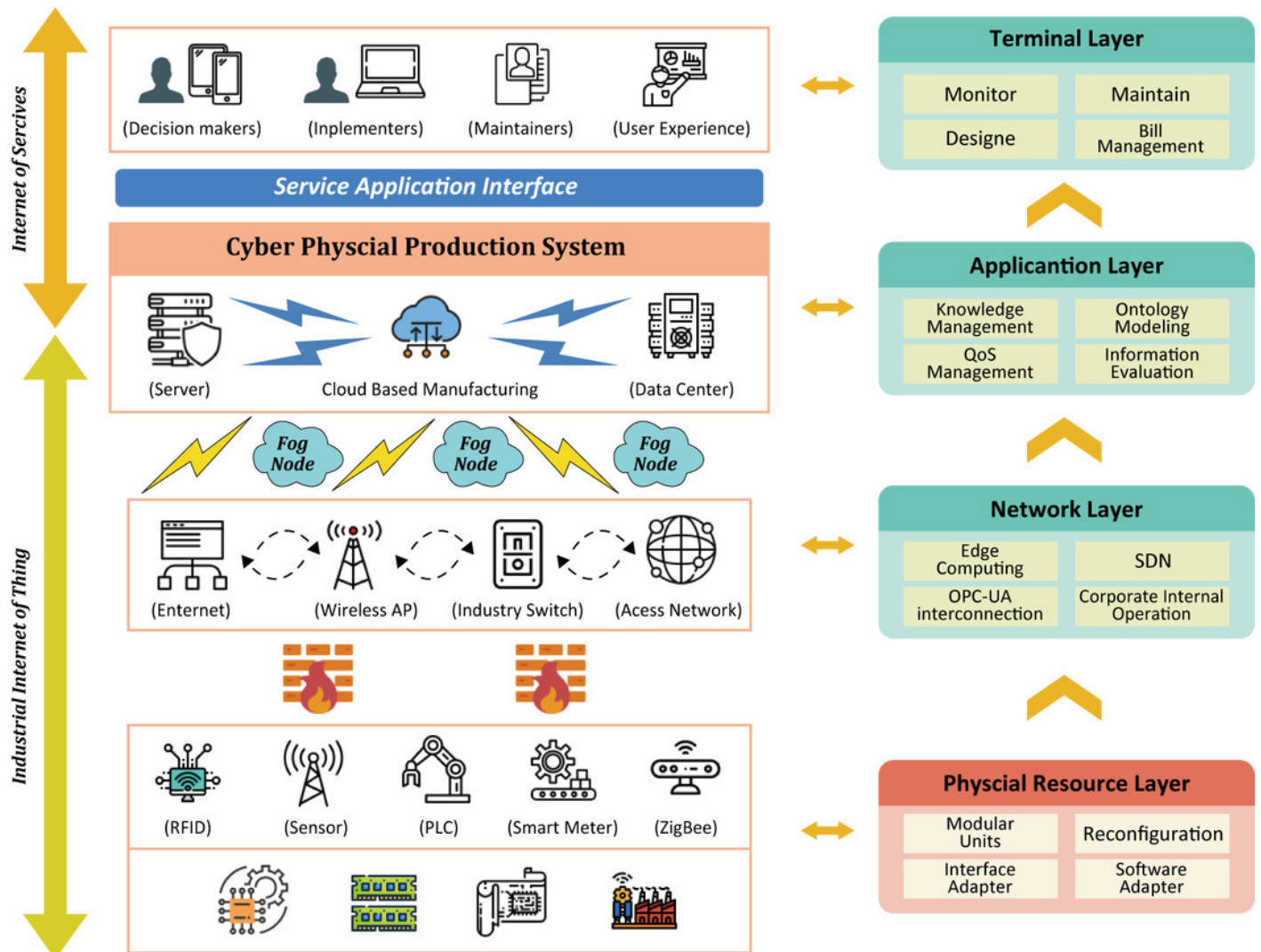
Figure 3

Taxonomies for botnet detection and security layers of IoT smart factories.
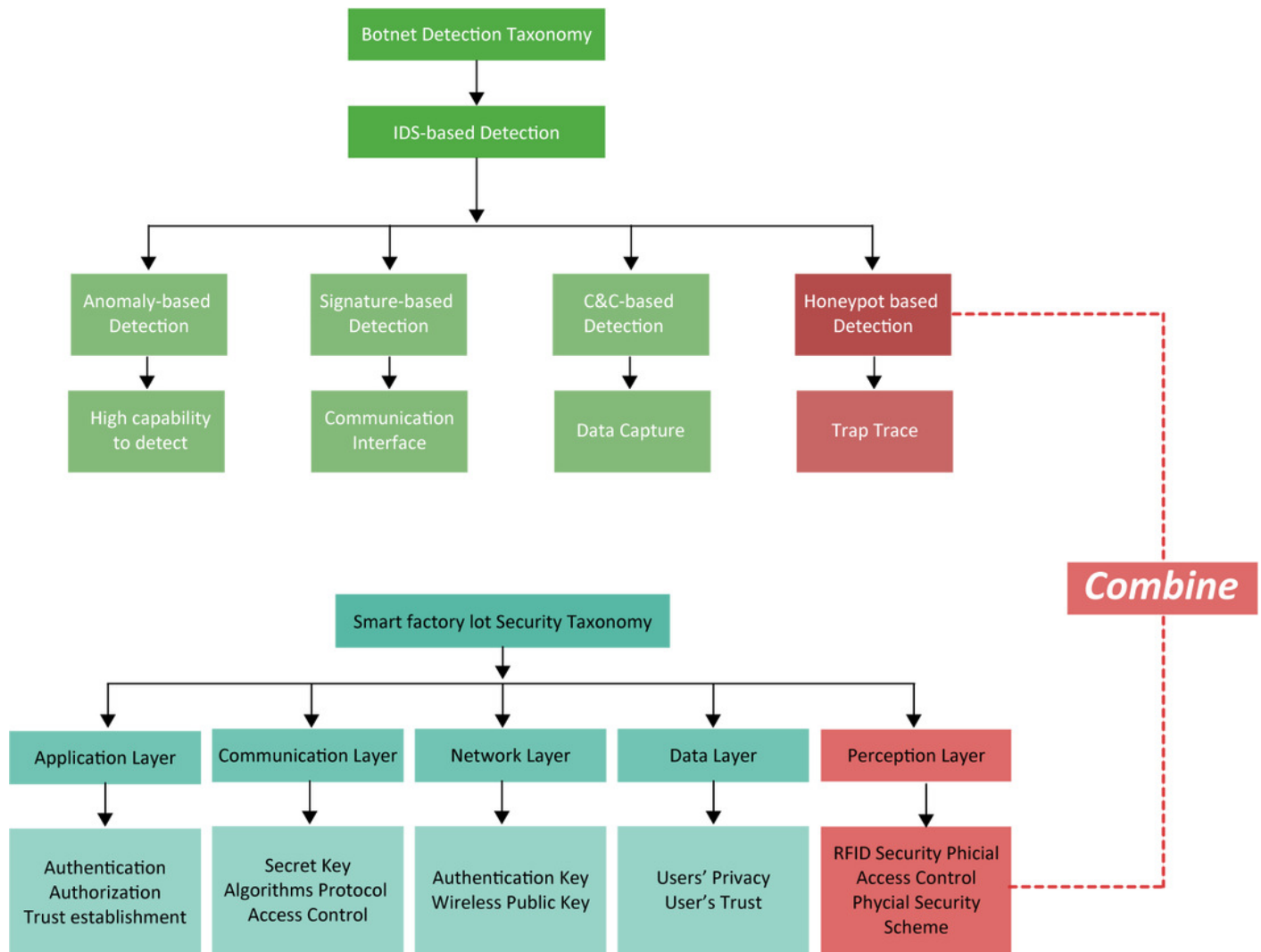
Figure 4

Operating mechanisms of botnet attacks to the physical sensing layer of a smart factory and design of the honeypot combined machine learning model.
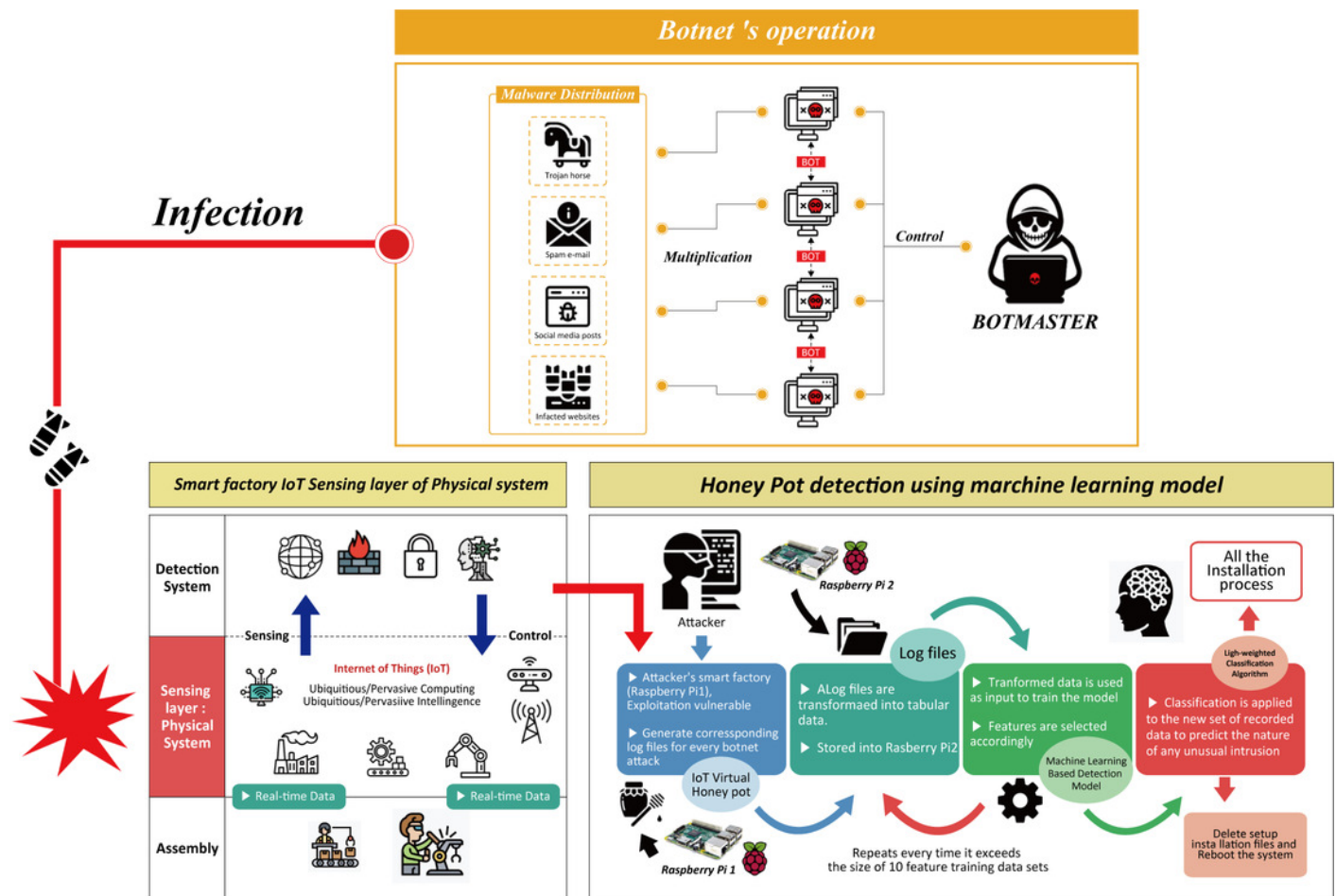
Figure 5

Simulation of hardware design
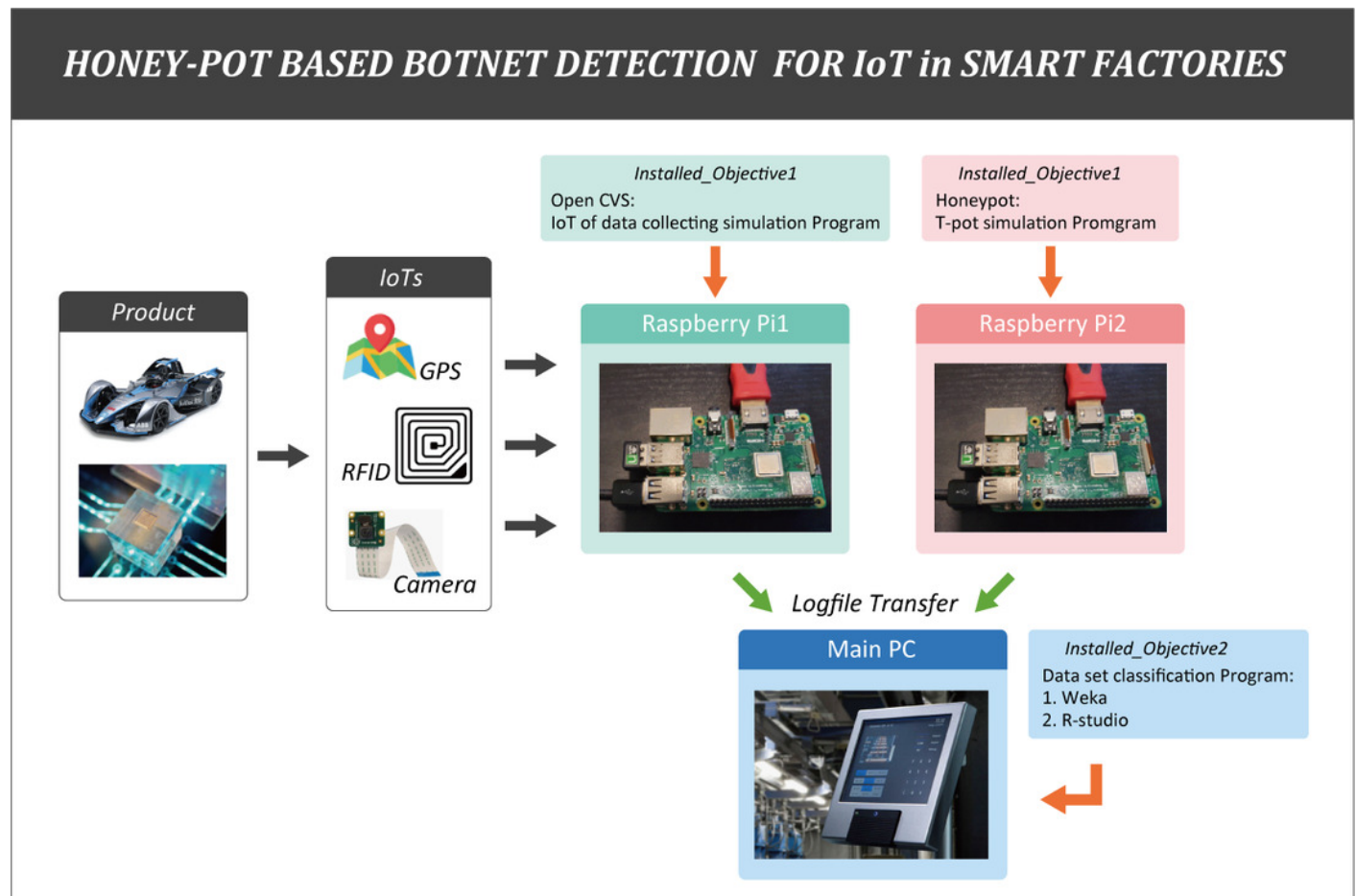
Figure 6

Pi Image data image.
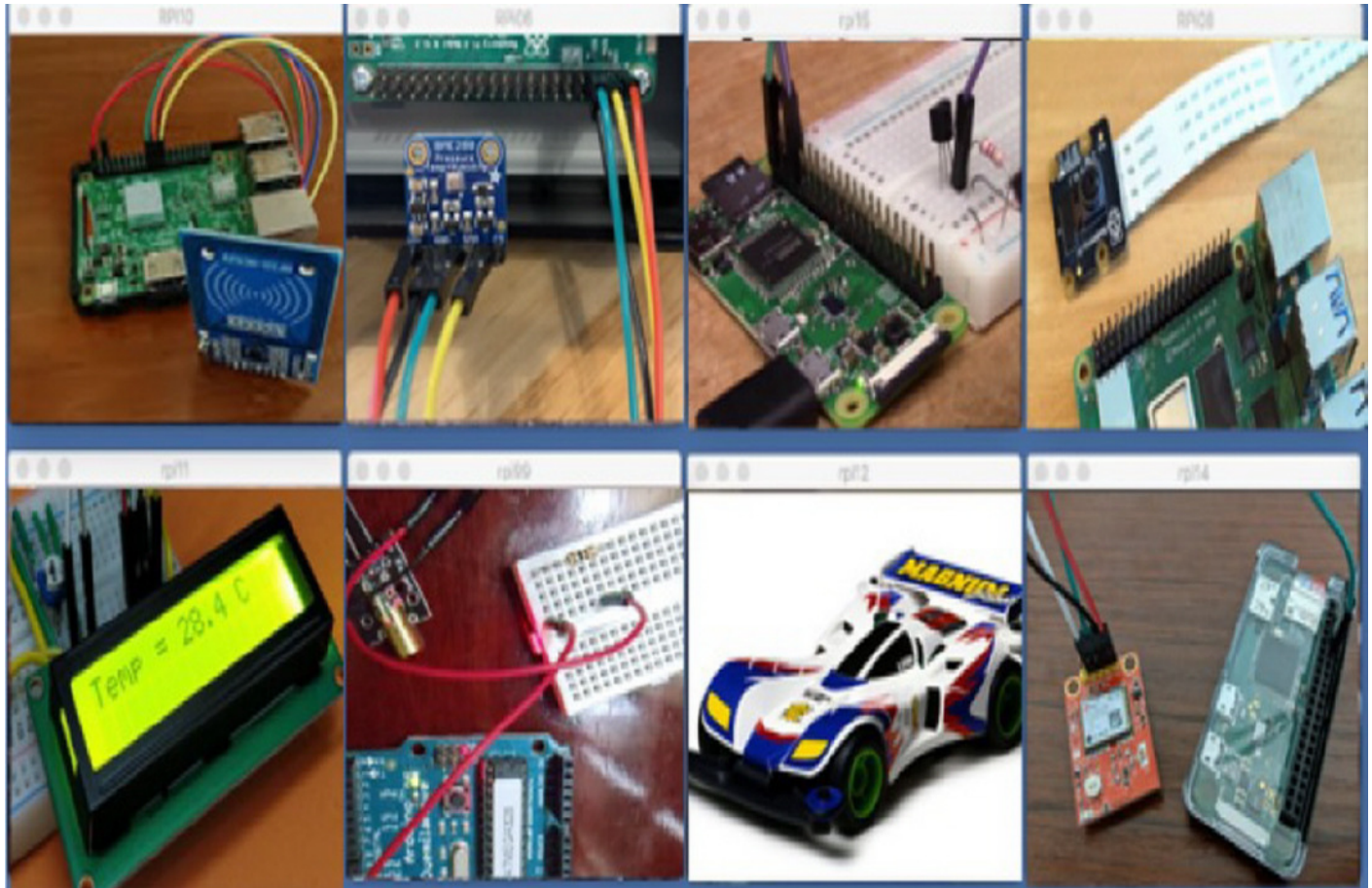
Figure 7

T-pot test script for raspberry Pi2.

Figure 8

Kibana dashboard.

Figure 9

Process design of honeypot combined machine learning model to detect botnet attacks in smart factory.
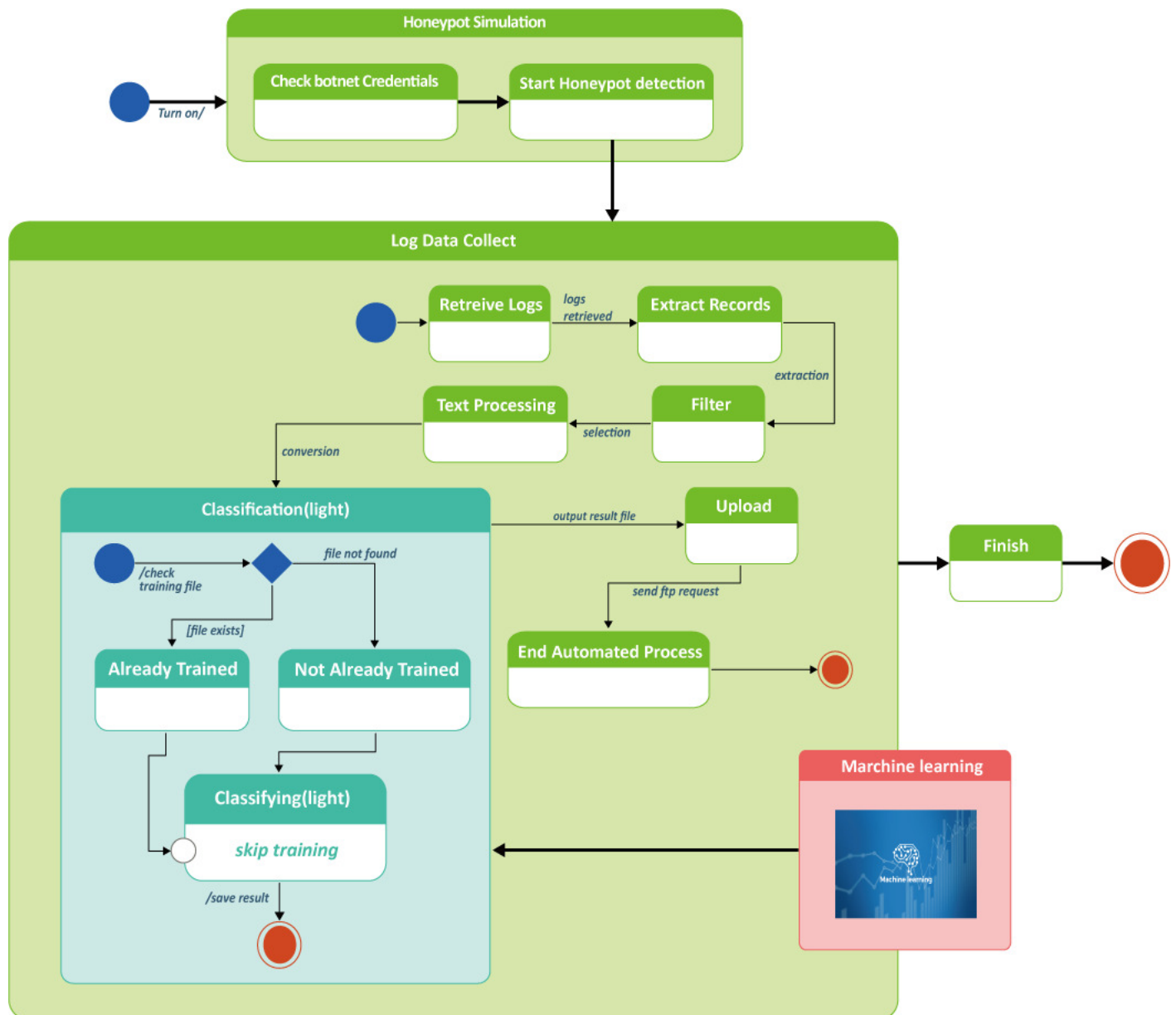
Figure 10

Correlation of true vs. false positive rates for the 10 best features.
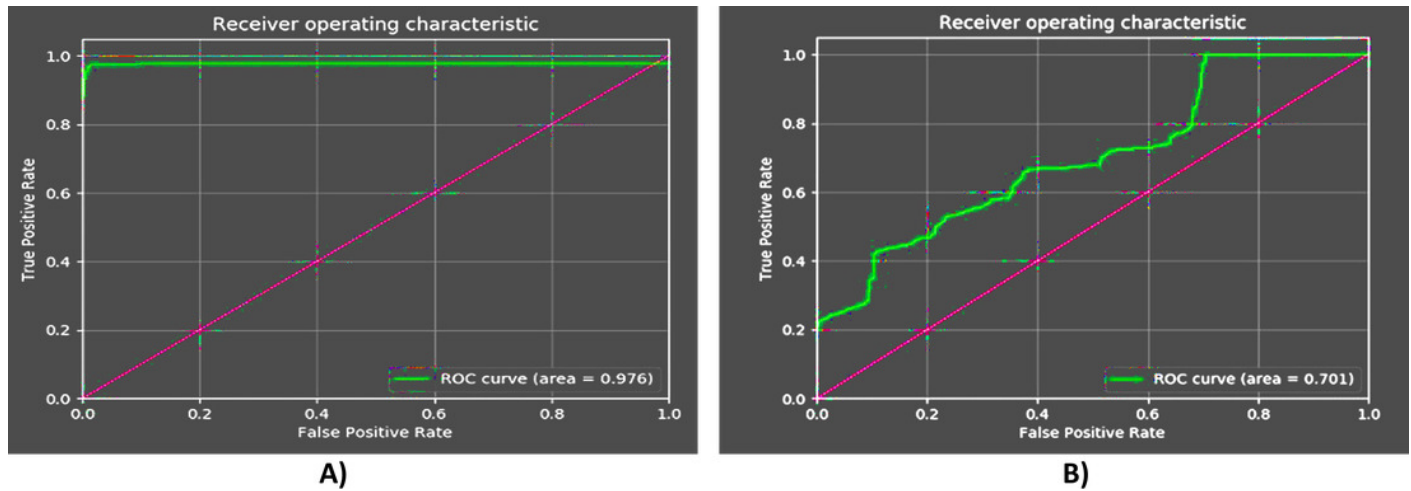
Figure 11

Detailed Accuracy by classification.
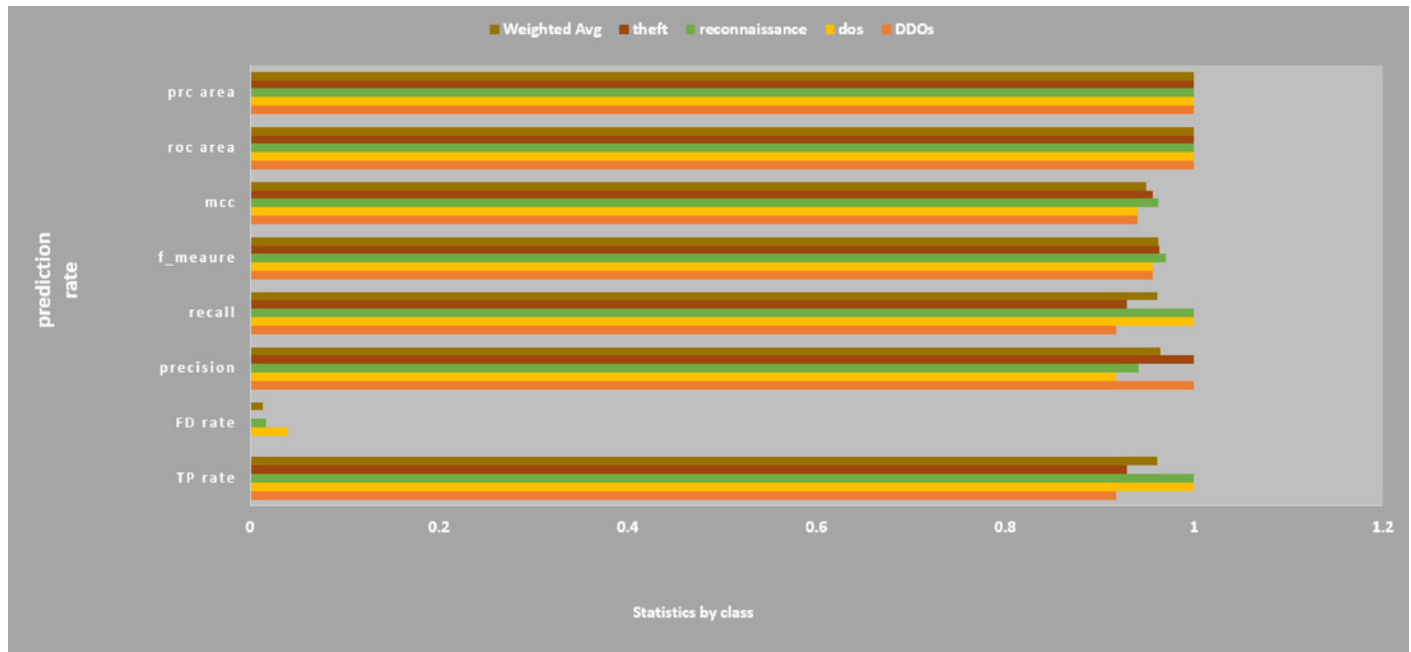
Figure 12

Correlation of 10 features for Weka-machine learning.
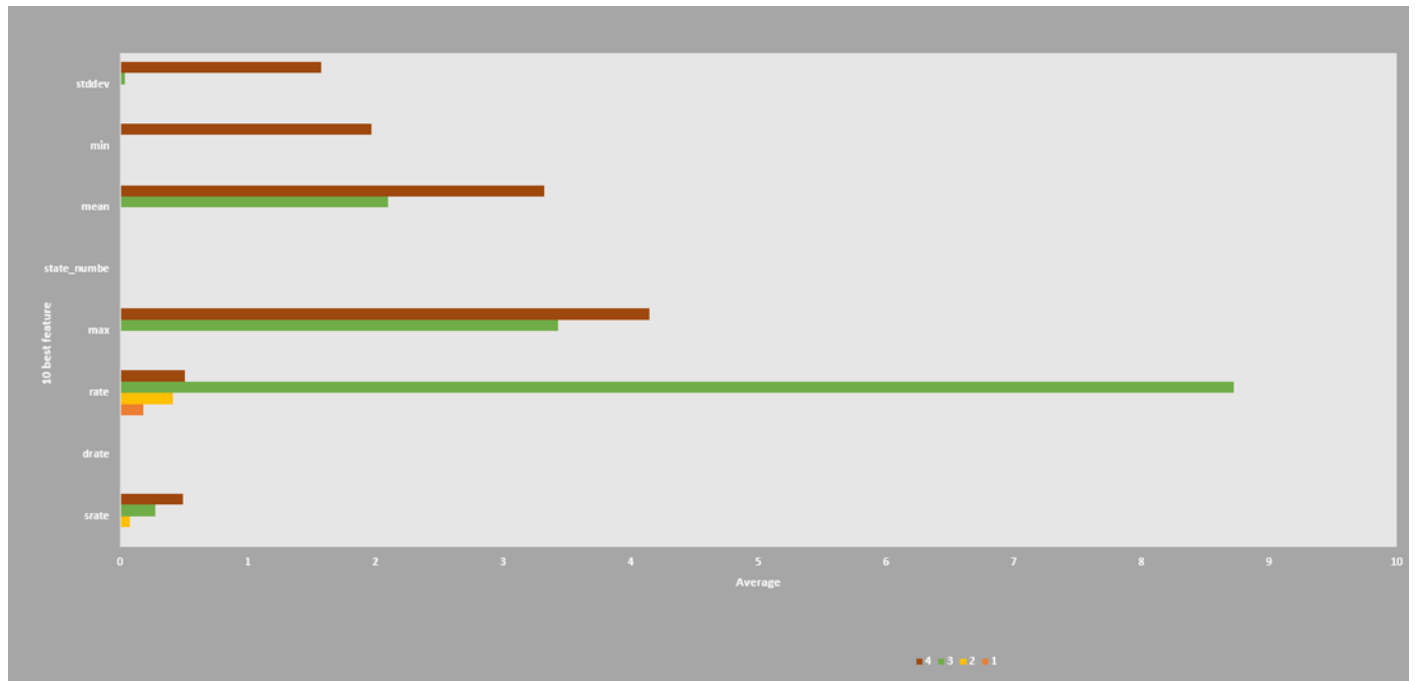
Figure 13

R-studio results.
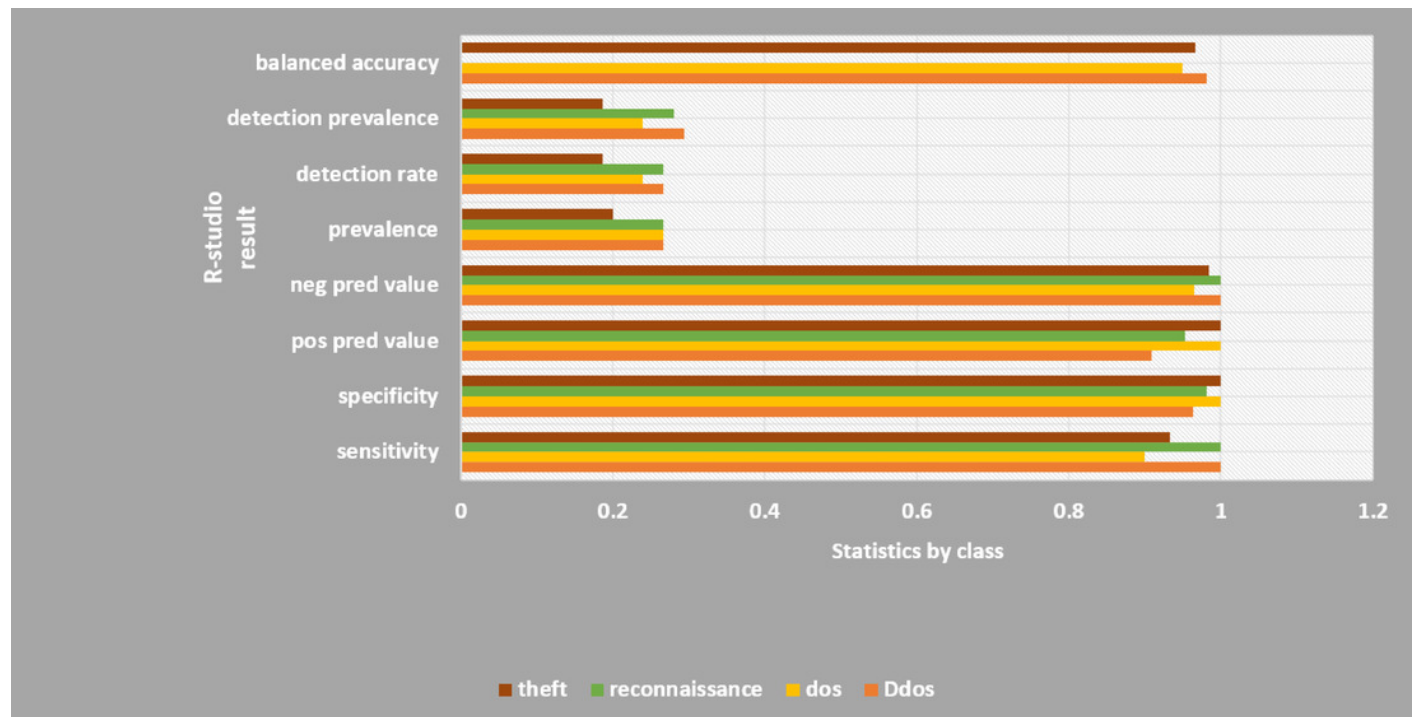
Figure 14

Graphical comparison of random forest in this study with random committee in other study.