Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 2845446, 12 pages
https://doi.org/10.1155/2022/2845446

Research Article

Detection of Botnet Attacks against Industrial IoT Systems by Multilayer Deep Learning Approaches

Mohammed Mudassir[1] **Devrim Unal**[2] **Mohammad Hammoudeh**[3] **and Farag Azzedin**[3]

[1] *Department of Mechanical and Industrial Engineering, Qatar University, PO Box 2713, Doha, Qatar*
[2] *KINDI Center for Computing Research, Qatar University, PO Box 2713, Doha, Qatar*
[3] *Information & Computer Science Department, King Fahd University of Petroleum & Minerals, Saudi Arabia*

Correspondence should be addressed to Mohammad Hammoudeh; m.hammoudeh@kfupm.edu.sa

Received 15 January 2022; Revised 17 March 2022; Accepted 11 April 2022; Published 17 May 2022

Academic Editor: Mohammad R Khosravi

Industry 4.0 is the next revolution in manufacturing technology that is going to change the production and distribution of goods and services within the following decade. Powered by different enabling technologies that are also being developed simultaneously, it has the potential to create radical changes in our societies such as by giving rise to highly-integrated smart cities. The Industrial Internet of Things (IIoT) is one of the main areas of development for Industry 4.0. These IIoT devices are used in mission-critical sectors such as the manufacturing industry, power generation, and healthcare management. However, smart factories and cities can only function when threats to cyber security, data privacy, and information integrity are properly managed. In this regard, securing IIoT devices and their networks is vital to preserving data and privacy. The use of artificial intelligence is an enabler for more secure IIoT systems. In this study, we propose high-performing deep learning models for the classification of botnet attacks that commonly affect IIoT devices and networks. Evaluation of results shows that deep learning models such as the artificial neural network (ANN), the long short-term memory (LSTM), and the gated recurrent unit (GRU) can successfully be used for classifications of IIoT malware attacks with an accuracy of up to 99%.

1. Introduction

The Industrial Internet of Things (IIoT) is the latest technological development in manufacturing and production that is being adopted rapidly all over the world. The IIoT is a part of the more general Internet of Things (IoT) network, which is characterized by its ability to connect billions of devices, appliances, sensors, equipment, and systems and to enable communication among these connected objects or "Things." The IoT market is expected to grow rapidly within the next decade, and its market value is estimated to be at least USD 2 trillion. The bulk of the objects in IIoT is low-powered devices with limited resources (such as battery and processing power). They need support systems for data analytics and security. From industrial equipment to home appliances, most electric-powered devices are becoming smart, interactive, and connected. IIoT is integral to creating cyberphysical systems (CPS) where physical processes are sensed, monitored, controlled, and commanded by humans or computer systems over the Internet. The IIoT is more focused on industrial applications such as smart factories, smart manufacturing, and Industry 4.0. IIoT is often integrated with other IoT networks and applications such as smart cities, smart transportation, smart grids, smart agriculture, smart healthcare, and other smart things. While the definition of Industry 4.0 varies, most notably it is the latest technological revolution in manufacturing that is at least supported by IIoT, CPS, and 5G networks, 3D printing, augmented and virtual reality, simulation, smart contracts, and sustainability measures [1].

As much of the enabling technologies for IIoT are based on IoT, they share a lot of similarities when it comes to security, privacy, and integrity. Although IIoT-enabled devices bring convenience to individuals and companies, this may come at the expense of their privacy [2–4]. Since IIoT devices are equipped with hardware and software that can potentially track user behavior, it is a necessity to design policies and technical solutions to ensure that the privacy, safety, and freedom of the users are always preserved. With numerous devices being connected to the Internet every day, IIoT opens up a broad platform for multifaceted cyberattacks. Common concerns related to IIoT devices include data theft, loss of privacy, and the possibility of abuse through unauthorized access that can take over control of the devices [5]. Many researchers are working on mitigating the various security problems related to IIoT devices and networks. Some of the regular attacks encountered in IIoT devices and networks include the distributed denial of service (DDoS) attacks over different communication protocols, data theft through keylogging and exfiltration, tracking through fingerprinting, and scanning for open ports over the network [6]. Many of these attacks on IIoT devices and networks are performed through botnets [7]. A botnet consists of several Internet-connected devices where each of the devices runs one or more bots. As the botnet infects other IIoT devices, the network of infected devices grows to make the botnet more computationally powerful and carry out larger attacks [8].

Furthermore, the vast applications of IIoT in critical industries and businesses have made them prone to cybercrimes where malicious agents try to override the security systems [9]. The risks involved in the potential overtake of the IIoT devices are enormous. The hazards involved in hacking include the theft of confidential information, the privacy of the public, and in some cases cyberattacks that can even result in loss of lives such as sabotage of medical equipment. For IIoT systems within Industry 4.0, it can mean disruption of production and services, stealing trade secrets, and leakage of confidential business data, all of which could lead to huge financial losses [10]. Hence, it is very important to provide layers of security over the IIoT devices to prevent any loss of data. In recent times, the number of IIoT attacks, especially the attacks carried out by the botnets, has increased substantially. Since there are many types of attacks over various protocols and devices, it becomes increasingly difficult to secure the IIoT devices and networks. Machine learning and deep learning have recently started to gain grounds for malware detection to help with this problem.

Botnets are assumed to be the biggest threats to IIoT networks. A Gartner report estimates that by 2025, the number of IoT devices will reach 50 billion [11]. This vast IoT network is a lucrative target for malicious agents. Many intrusion detection products and services are currently available in the market that offers various levels of protection against IIoT devices. However, there are new threats that emerge every day, and it is important to search for detection methods that are comprehensive, intelligent, and adaptive. Recent advances in machine learning and deep learning show promising results in the classification of attacks [12]. The superiority of deep learning models compared to conventional methods of detection is that they can learn from unstructured data without supervision [13]. Consequently, attacks that are new or can avoid signature-based methods can still be detected by deep learning-based models.

One of the shortcomings to using ML models for classifying malware and network traffic is that the ML model often fail to correctly identify classes that are minority in the train set. [14, 15] used a number of sampling techniques such as oversampling, undersampling, and others for improving the identification of the minority classes.

In this paper, we present deep learning models for the classification of malicious packets originating from IIoT devices. Our results show that deep learning models trained on balanced dataset can give a highly accurate classification of malware data with good precision and recall.

2. Background and Related Work

IIoT devices are often connected to the network and are controlled remotely through a user interface [16]. All of the IIoT devices are based on four characteristics which include a feedback mechanism, a few communication protocols, a control system, and some security layers. The signal to control the system is sent through the interface to the controlling device. The IIoT devices operate based on the signal received and send the feedback back to the interface. This feedback is sent through the sensors placed within the devices. These sensors convert the physical data into electronic signals and send it to the interface through the control systems [17].

2.1. Industry 4.0. Industry 4.0 is the next evolution in manufacturing processes. It is highly integrated across all levels of operation. Figure 1 shows an overview of the Industry 4.0 ecosystem. It is supported by IIoT that allows connectivity between all devices, sensors, machines, and operators. Industry 4.0 allows a high level of autonomy through smart factories. From production to quality control to final delivery of the product, little human intervention is required. Product defects can be identified using computer vision. Additive manufacturing can produce complex designs while reducing material wastage. The operators can be informed of the production processes through wearables. For example, the 3D printer could send a notification to the operator's smartwatch once the fabrication is complete.

Due to the large number of devices, protocols, and systems present in the IIoT network, it is a lucrative target for malware and botnets. For instance, if malware infects the 3D printer, it could alter the design, change the print parameters, and cause damage to the product. Due to the nature of 3D printing, some defects introduced by malware may not be readily noticeable and this could create a hazard for the end-user of the product [10].

Industry 4.0 can also utilize smart contracts and blockchain to help with preserving the integrity of the systems, processes, and operations. For instance, a product design can be cryptographically signed and verified with blockchain to preserve its integrity [18].

2.2. Architecture. The systems behind the functioning IIoT devices are quite complex and based on different kinds of

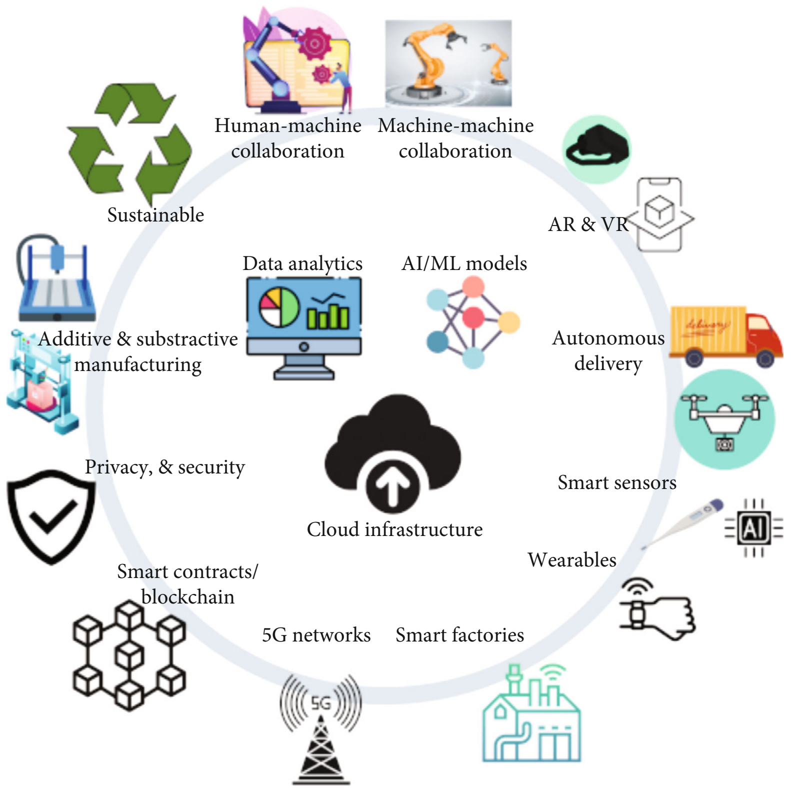

Figure 1: Overview of Industry 4.0 ecosystem. All devices are connected and integrated with IIoT. The data is transmitted to the cloud for analysis. Threats can be detected using ML-based detection methods.

layers. Depending on the model, the IIoT architecture may be based on three or five layers. The three-layer model includes a perception layer, a network layer, and an application layer. Additionally, most of the recent IIoT systems are based on the Service-Oriented Architecture (SOA) architecture [19].

The perception layer is a layer that is based on the hardware objects of the IIoT system, and hence, it is also called the object layer. This consists of physical sensors and measures the parameters controlled by the system. This data perceived by the physical sensor is then converted into the electronic signal by the electronic circuits and then transmitted to the interface through the network layer.

The network layer transmits the data from the device to the interface controlled by the user. It also transfers the data or the set values input by the user to the device. This layer is also called the transport layer. It is this layer that is most prone to hacking and intrusion and must be provided with protective systems to protect the device from any external control. The layer must be provided with the methods to prevent the intrusion. The network layer is based on connection protocols which are done using any wireless communication methods like NFC, Bluetooth, and Wi-Fi technology.

The application layer varies from service to service. This is the main interface that is available to the end-user through which he enters the commands and asks the device to perform accordingly. This layer must also be provided with security measures to protect the device from intervention from an outside source.

2.3. Attacks. The cyberattacks into the IIoT-based devices are of many types based on which type of layer they are attacking and the severity of the attack. These attacks make the IIoT solutions vulnerable and the main hurdle in the widespread use of the systems. The types of cyberattacks into the IIoT devices include [20] denial of service (DoS) attack, flooding, blackhole attack, Sybil attack, clone attack, and sinkhole attack among various others and combination thereof.

DoS attack is the one in which the application layer, which is the user interface, is no longer in control of the legitimate user. This attack is through the communication protocol followed by the system, which is either Bluetooth [21], Wi-Fi, or NFC technology. This attack also affects the hardware devices as well. DoS attack performed at a large scale through botnets is called distributed denial of service (DDoS) attack [22, 23]. Flooding is the one in which the cyber hackers take control of the interface over the network and show its presence by displaying the “Hello” message over the interface. A blackhole attack is one in which the route of the connection is changed, and the user is unable to access the device from the connection source [24]. Sybil attack is the one in which the multiple connection routes are created and the original information which is to be transmitted is corrupted. In the clone attack, similar connection routes are generated by the attackers which causes the data which is transmitted to be lost and get corrupted. A sinkhole attack is the one in which the original connection node acts as a sink and attracts and corrupts the surrounding connection nodes. Yavuz et al. proposed highly-scalable deep learning methods for the detection of IIoT routing attacks with high accuracy and precision results on decreased rank, hello-flood, and version number modification attacks [25].

2.4. Communication Protocols. The communication protocols followed by the IIoT devices are often insecure and unreliable. Therefore, there is a need for security layers to prevent any intrusion in the communication system. The communication protocols followed by IIoT devices include [26] IPv6, 6LoWPAN, User Datagram Protocol (UDP), Quick UDP Connection (QUIC), Datagram Transport Layer (DTLS), CNN (Content-Centric Networking), and Constrained Application Protocol (CoAP).

IPv6 communication technology has become standardized over the past few years because of its universality and ease of use. IPv6 is better than the other communication protocols as it provides a higher speed for the data packet transmission. In this protocol, the data which is to be transmitted does not need to be passed to network-address translators (NAT) as compared to the other protocols such as IPv4.

6LoWPAN communication protocol further reduces the data packet size by compressing it, and hence, this makes the data transmission faster and more reliable. This communication protocol has a working range of 2.4 GHz frequency range with the transfer rate of as fast as 250 kilobytes per second.

UDP provides a simpler communication protocol between the device and the interface due to its lightweight, which reduces the lag between the data communication. It is mostly used for live communication such as live processing of process-plant parameters as it has less overhead. QUIC is the more advanced version of UDP. As the name suggests, it is faster than a conventional UDP connection and hence more reliable. It also allows multiple complex connections between the two nodes, and hence, data can be transmitted faster.

DTLS communication protocol is used where private connections are required as this allows the data transmission privately without any external influence.

CNN communication protocol allows the data-centric transmission between the device and the user interface without any external noise. This is the most effective and reliable protocol for the accurate transmission of data.

CoAP communication protocol is used for application-specific purposes. It uses the HTTP server for the communication between the device and the interface. HTTP server with the URL provides the Web access to the communication, and hence, the interface is easy to understand and has a vast atmosphere.

2.5. Intrusion Detection. The intrusion detection system is based on the same concept as the working of the IIoT devices. The intrusion detection can be placed as a separate layer on the top of the IIoT architecture or it can be embedded into the application and connection layer of the IIoT architecture. The main concept behind the working of intrusion detection is that it assigns unique identifiers to the data nodes emerging from a specific network. All the data nodes which have different identifiers, not recognized by the system, are rejected, and the user is informed about the breaches.

There are many intrusion detection methods to spot any malicious activity on IIoT devices. Some of the most common intrusion detection methods which are used commonly by businesses and industries [27] include detection based on signatures, anomaly-based detection, detection based on specifications, detection using machine learning, deep learning, and a combination of approaches.

A signature-based detection system is used for the communication and data packets transmitted through the connection layer. These detection systems detect any abnormality in the data packets transmitted through the network and give an alert based on the data. This is a very operative and fast method to detect any intrusion within the system. This method works based on the attack signatures identified to it based on the past data. It investigates the past for the signatures of the data which caused the intrusion and looks for the same intrusion signatures in the future and notifies if it happens again. One of the drawbacks of this method is that it cannot detect any new intrusion occurring in the future as the system will not identify the signature as the intrusion [24]. The method is based on machine learning and statistical tools, and therefore, to apply the system on any device, the system must be fed with the previously known intrusion signatures, to begin with, and it continuously learns the new intrusions based on the inputs provided to it by the user. The algorithm can also be modified to detect any new data packet as an intrusion. This is more stringent and will also detect the new data input by the user as an intrusion. Some researchers have also modified the system, and instead of detecting the data packets, they have made the system recognize the energy consumed by the specific signature. This method is more reliable, stringent, and fast as compared to the previous one [26].

The drawback of the signature-based detection approach is that it cannot detect new intrusions into the system. Anomaly-based detection mitigates this problem as this is based on the anomaly or irregular data packets instead of the signatures. Any new data packet trying to enter the system that does not match with the regular attributes will be detected as an anomaly. This will make the system more secure, but the users must keep their data consistent so that the user data is not itself corrupted [28]. This method is also effective in detecting sinkhole attacks. If the data packets taken in by the system are large as compared to the normal usage, the system will detect this as an anomaly and will inform the user about the intrusion [29].

The specification-based intrusion detection method is based on the instructions provided to the system, and the system data packets will follow the instructions. This set of instructions will prevent any data packet not following the instructions as an anomaly. This method is effective for the DoS attacks in which the user is prevented from controlling the application. This approach is very much dependent on the specification set for the data packets.

Machine learning finds various applications in the field of intrusion detection. The applications are programmed to learn through past intrusions. This is possible through the machine learning application. If any similar intrusion is done again on the system, the program stops it immediately and informs the user about it [30]. Deep learning is different from machine learning in the sense that machine learning consists of a single algorithm that enables the machine to learn from past instances. In contrary to that, deep learning is a part of machine learning and consists of many layers of algorithms which are called ANN (artificial neural network) [31]. In intrusion detection, especially for IIoT devices and networks, unsupervised deep learning is not heavily dependent on past intrusions. If provided unstructured data, it can function well to detect any future intrusions into the layers of the IIoT. Deep learning is like the functioning of the human brain. Deep learning-based intrusion detection algorithms identify the differences between the required data packets and the intrusions by themselves and based on their learnings, preventing the intrusions from happening in the future [32, 33].

Shafiq et al. [34] used four different machine learning classifiers (random forest, support vector machine, decision tree, and Naive Bayes) for IoT botnet attack classification using the dataset developed by Koroniotis [35]. Their reported accuracy was higher than 99% for classifying some of the selected attacks. All models performed well with over 98% accuracy across all attack classes. Within healthcare IoT, [36] built a testbed that monitors the patients' biometrics and collects network flow metrics for providing them treatment and medical diagnostics and used different machine learning methods for training and testing against the dataset which included man-in-the-middle cyberattacks.

To make intrusion detection more advanced, a combination of the abovementioned intrusion detection methods is used. Each method has its specific features, and hence, to protect the system from multiple cyberattacks, a combination of the methods can be used. This approach provides more stringent protection as compared to the individual approaches.

3. Methodology

A structured and labeled dataset of IIoT botnet attack data is used for training the machine learning models. The machine learning models are developed in Python 3.8 using Keras, Tensorflow, and Scikit-Learn libraries. The data is scaled before training used the machine learning models. The overview of the methodology is shown in Figure 2.

3.1. Dataset. The IIoT botnet attack dataset was developed by [35]. It consists of several types of attacks including DoS, DDoS, theft, and reconnaissance. The DoS and DDoS attacks contain 3 different protocols such as the HTTP, TCP, and UDP. Theft includes keylogging and stealing data. Reconnaissance includes fingerprinting of the operating system and scan of open ports. Overall, the attacks can be categorized into 10 different classes as shown in Table 1. There are about 37 features. The complete labeled dataset is about 16 gigabytes. 5% of this dataset is considered for this study. Nevertheless, this smaller sampled dataset contains approximately 3.6 million records. The dataset is randomly sorted into two sets— training and testing. 80% of the data is allocated to the training set while the remaining 20% is allocated to the validation set. Figure 3 shows the labels of the attacks and their frequencies in the original dataset. The imbalance is present in the original dataset where the classes of attacks are imbalanced. This affects the deep learning models as they need sufficient training data in recognizing the attacks appropriately. Although some techniques such as oversampling from classes with fewer samples can be used in some instances, we kept the statistics of the sampled dataset to accurately reflect the original dataset, as this also represents the real-life attack scenarios, with some types of attacks being more frequent than others. Table 1 shows the number of records for each of the classes of attacks. Figure 4 shows that the sampled dataset is representative of the original dataset in terms of the attack classes being proportional compared against the total number of records in each dataset.

3.2. Imbalance Correction. The dataset created by [35] suffers from heavy class imbalance. This affected the performance of the deep learning models in correctly identifying the threats in multiclass classification. To improve the model performance, a balanced dataset is created using the techniques suggested by [15]. The Python imbalanced-learn module has been used for undersampling the majority class to create a balanced dataset with equal number of cases from each class [37].

3.3. ML Models. Three different kinds of deep learning models are used for this study: the artificial neural network (ANN), gated recurrent unit (GRU), and long short-term memory (LSTM).

3.3.1. Artificial Neural Network. ANN is commonly used for classification problems in supervised learning. The ANN consists of an input layer, an output layer, and several hidden layers which consist of neurons. The hyperparameters are tuned manually for optimal performance as shown in Table 2. The loss function is *categorical_crossentropy* which is used for multiclass classification problems along with the *accuracy* metric. The rectified linear unit (ReLu) is used for activations in all the layers except the output layer which uses *softmax* to give the multiclass outputs. The hidden layers and the number of hidden layers can be tuned manually to have better performance. The optimizer is *adam*, which is a gradient-based optimizer that is popular in machine learning problems for its fast convergence. A batch size of 64 and an epoch of 200 were used for training the models. The initial learning rate was set at 0.001 and adapted to lower rates as the training progressed over several epochs. The training set is scaled using *RobustScaler* as it improves the performance of the ANN.

3.3.2. Long Short-Term Memory. LSTM network is a state-of-the-art recurrent neural network that can learn from both long- and short-term dependencies and is more robust to the vanishing gradient problem in deep-layered networks. This deep learning algorithm is quite robust for modeling time-dependent data [38]. Since IIoT devices transfer data through packets over some time, the attack features can be considered time-dependent. For example, during a DDoS attack, the IIoT traffic might experience higher latency. These would result in a longer duration for data transfer,

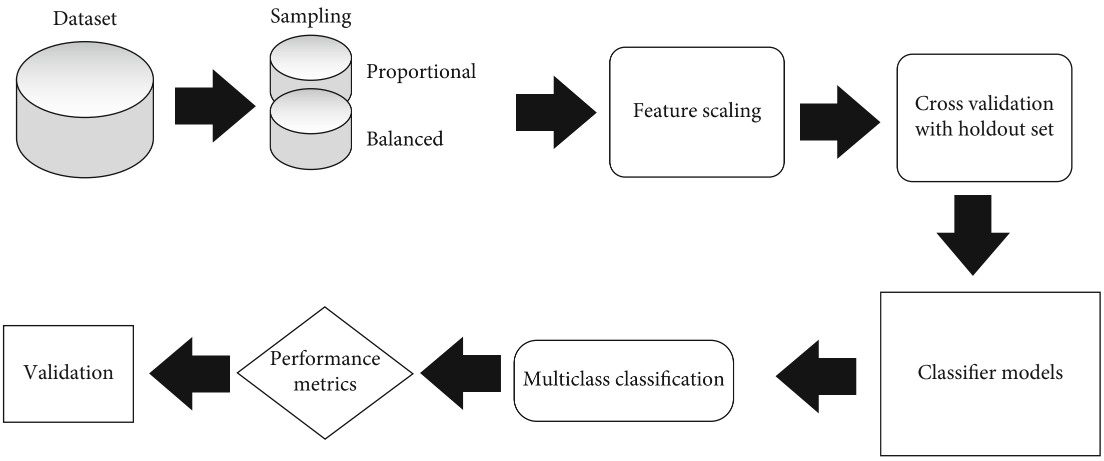

Figure 2: Overview of the methodology.

Table 1: Records of different attack types contained within the sampled dataset.

ID	Category	Frequency	Percent
0	Normal	477	0.01
1	dos_http	1485	0.04
2	dos_tcp	615800	16.79
3	dos_udp	1032975	28.16
4	ddos_http	989	0.03
5	ddos_tcp	977380	26.64
6	ddos_udp	948255	25.85
7	rcn_fngrprnt	17914	0.49
8	rcn_scan	73168	1.99
9	theft_data	6	1.6×10^{-4}
10	theft_keylog	73	1.99×10^{-3}
Total		3668522	100%

and the attack might be picked up by a well-trained LSTM model. The LSTM block, which is analogous to the neuron of the ANN, has three gates. These gates—forget (f), input (i), and output (o) gates—are represented by sigmoid functions. In the LSTM block, C_{t-1} is the cell state or memory from the previous block. X_t is the vector input, C_t is the cell state of the present block, h_{t-1} is the previous block output, and h_t is the output of the current block. Element-wise Hadamard product is performed at the ⊗ junction. Likewise, the element-wise summation is done at the + junction. The LSTM gates and memory equations are given by (1) to (6). The features are scaled using a min-max scaler before training. Table 3 shows the LSTM model hyperparameters used for training our models.

$$f_t = \sigma_g\left(W_f x_t + U_f h_{t-1} + b_f\right), \quad (1)$$

where f_t is the activation vector of the forget gate, σ_g is the sigmoid function, W and U are the weight matrices, and b is the bias vector.

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i), \quad (2)$$

where i_t is the activation vector of the input or update gate.

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o), \quad (3)$$

where o_t is the activation vector of the output gate.

$$\tilde{c}_t = \sigma_h(W_c x_t + U_c h_{t-1} + b_c), \quad (4)$$

where the activation vector of the cell input is given by c_t, and σ_h is the hyperbolic tangent (tanh) function.

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t, \quad (5)$$

where c_t is the cell state vector.

$$h_t = o_t \otimes \sigma_h(c_t), \quad (6)$$

where h_t is the output vector of the LSTM block or the hidden state vector.

3.3.3. Gated Recurrent Unit. GRU is a recurrent neural network that is very similar to the LSTM yet simpler in its design. Instead of the 3 gates that LSTM utilizes, the GRU uses 2 gates: update and reset gates. It also does not have a separate cell state or memory. Instead, it uses the hidden state for transferring information. The update gate serves the function of both forget and input gates in that it decides what new information to consider and what information to forget. The reset gate is used for controlling the amount of past information to forget. Table 4 shows the hyperparameters used for training our GRU model.

$$z_t = \sigma_g(W_z x_t + U_z h_{t-1} + b_z), \quad (7)$$

$$r_t = \sigma_g(W_r x_t + U_r h_{t-1} + b_r), \quad (8)$$

$$\widehat{h}_t = \phi_h\left(W_h x_t + U_h\left(r_t \otimes h_{t_1}\right) + b_h\right), \quad (9)$$

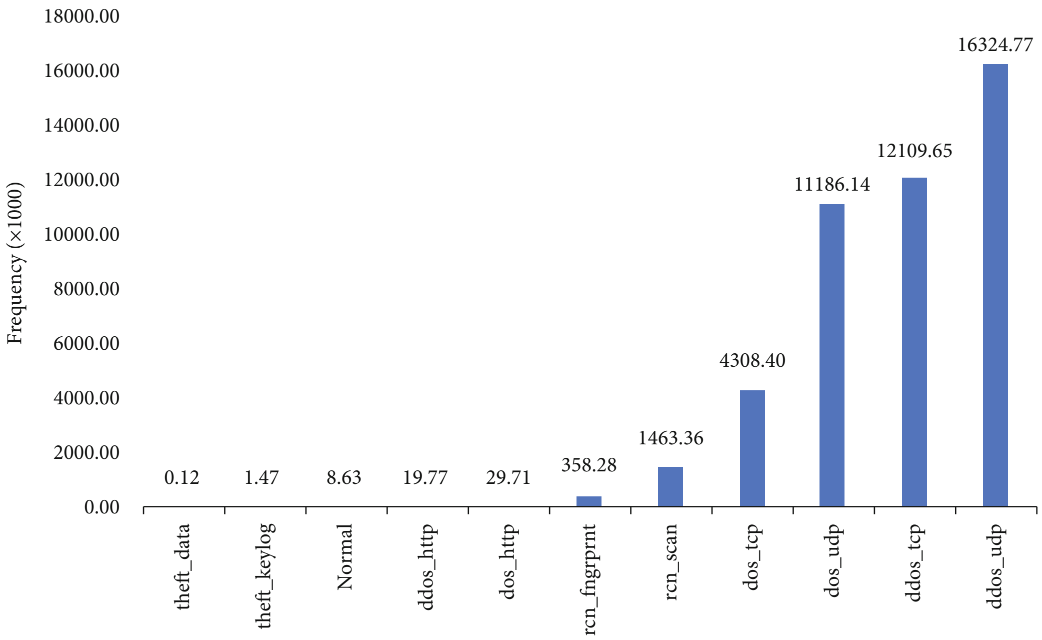

Figure 3: Bar chart of the original dataset.

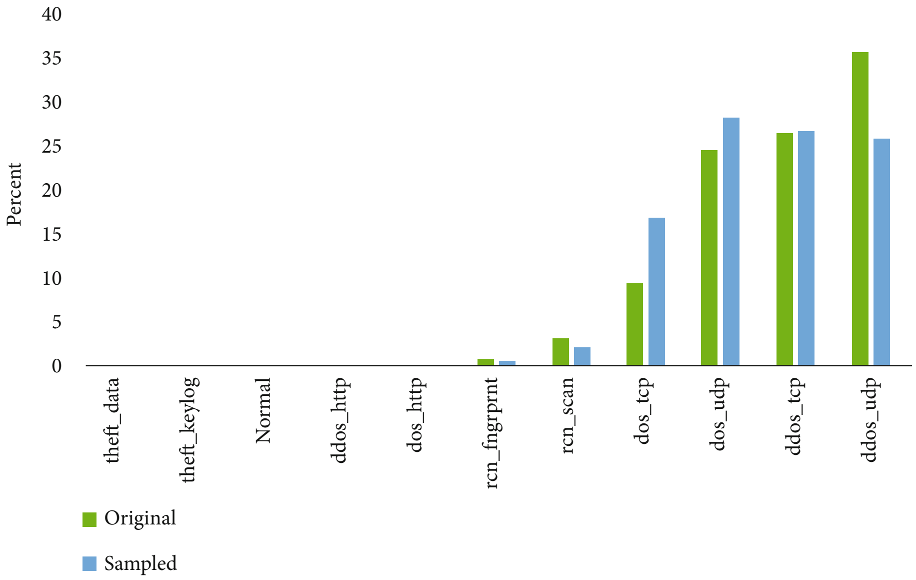

Figure 4: Comparison between the original dataset and the sampled dataset shows that the sampled dataset represents the original dataset in terms of proportionality.

$$h_t = (1 - z_t) \otimes h_{t-1} + z_t \otimes \widehat{h}_t, \tag{10}$$

where σ_g is the sigmoid function, ϕ_h is the hyperbolic tangent, x_t is the input vector, h_t is the output vector, $\widehat{h}_t$ is the candidate activation vector, z_t is the update gate vector, r_t is the reset gate vector, W and U are the parameter matrices, and b is the bias vector.

4. Results and Discussion

With the rapid growth of IIoT devices, it has become imperative to develop secure systems that can mitigate attacks against IIoT networks. Botnet attacks are regularly targeted towards these networks and devices for stealing data, denying legitimate users from accessing services, and invading user privacy. Traditional signature-based malware detection is not sufficient to protect against these threats. There were some previous studies such as [34, 35] which applied classical learning methods for botnet detection, such as decision trees, naive Bayes, and SVM. However, these models are not suitable for training on large amounts of data.

The deep learning classification models can be evaluated using different performance indicators (PI). The indicators are accuracy (11), F-1 score (12), and area under the receiver

Table 2: Settings of the ANN.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
Hidden layers	2
Neurons per hidden layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

Table 3: Settings of the LSTM.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
LSTM layers	2
LSTM blocks per layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

Table 4: Settings of the GRU.

Hyperparameters	Options
Loss function	categorical_crossentropy
Metric	Accuracy
Activations	ReLu & Softmax
GRU layers	2
GRU blocks per layer	(100,100)
Optimizer	Adam
Batch size	64
Learning rate	0.001
Epochs	200

Table 5: Average performance scores of ML classification models.

Metrics	ANN	GRU	LSTM
Accuracy %	99	98	98
AUC-ROC score	0.85	0.83	0.84
Precision	0.98	0.99	0.98
Recall	0.99	0.98	0.98
F1-score	0.98	0.98	0.98
Cohen's Kappa (k)	0.98	0.98	0.98

operating characteristic curve (AUC-ROC). These PI are based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The most commonly reported PI is accuracy. However, as the records of attack classes are imbalanced, F1-score may provide better insight. F1-score close to 1 indicates that the model performs well in both precision (13) and recall (14). AUC-ROC score indicates how good the model is in differentiating between the true positives and the true negatives. AUC-ROC score of 0.5 means that the model does not discriminate between classes. AUC-ROC score closer to 1 indicates that the model is good at making a distinction between classes, while scores less than 0.5 suggest that the model performs worse than a random classification [39]. Cohen's Kappa (k) is another classification metric that can be used to compare the test set classifications against the predicted set classifications. The k indicates the level of agreement between these two sets by a number between -1 and 1 with 1 being in perfect agreement. k of 0 implies that there is no agreement between the two sets despite having some probability, and a k value of -1 implies that the agreement is arbitrarily worse than random. The k is given by (15).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{11}$$

$$F1 - \text{score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{12}$$

where precision and recall are given by (13) and (14), respectively.

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{13}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{14}$$

$$k = \frac{\rho_0 - \rho_e}{1 - \rho_e}, \tag{15}$$

where ρ_0 is the observed agreement similar to (11), and ρ_e is the probability of agreement by chance calculated from the classes present in the dataset.

Table 5 summarizes the average performance of the deep learning models concerning the various classification metrics. On average, the three deep learning models performed well with ANN reporting 99% accuracy, and both LSTM and GRU reporting 98%. In terms of AUC-ROC score, ANN scored 0.85, followed by LSTM with 0.84, and GRU with 0.83. GRU reported the highest precision of 0.99, and ANN and LSTM both reported 0.98. ANN reported the highest recall with 0.99, and LSTM and GRU both reported 0.98. All three models reported the same F1-score of 0.98. The three models also reported the same Cohen's Kappa of 0.98.

Tables 6, 7, and 8 show the performance of the deep learning models with respect to each attack type. When using the proportionally sampled dataset, all the models could not identify *theft_data* correctly in the test set. However, upon inspection, the predicted classification *theft_data* was misclassified as another type of attack and not as normal

Table 6: Precision of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	1	1	1	1	1	1
dos_http	0	1	1	1	0.93	1
dos_tcp	0.99	1	0.94	1	0.94	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.98	0	0.98	0.46	0.97
ddos_tcp	1	1	0.98	1	0.99	1
ddos_udp	1	1	0.99	1	1	1
rcn_fngrprnt	0	0.97	1	1	1	1
rcn_scan	0.64	0.98	1	1	1	1
theft_data	0	0.87	0	0.89	0	0.88
theft_keylog	1	1	0	0.97	0	0.98

^{B}Balanced dataset.

Table 7: Recall of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	0.14	0.84	0.7	0.97	0.72	0.98
dos_http	0	0.96	0.05	0.97	0.18	0.99
dos_tcp	0.98	1	0.97	1	0.98	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.89	0	0.91	0.09	0.96
ddos_tcp	0.98	1	0.96	1	0.96	1
ddos_udp	1	1	1	1	1	1
rcn_fngrprnt	0	0.97	0.99	1	1	1
rcn_scan	1	1	1	1	1	1
theft_data	0	0.88	0	0.85	0	0.84
theft_keylog	0.19	0.92	0	0.93	0	0.91

^{B}Balanced dataset.

Table 8: F1-score of the deep learning models with respect to each attack type.

Classes	ANN	ANN^B	LSTM	LSTM^B	GRU	GRU^B
Normal	0.25	0.91	0.82	0.98	0.84	0.99
dos_http	0	0.98	0.10	0.98	0.30	0.99
dos_tcp	0.98	1	0.95	1	0.96	1
dos_udp	1	1	1	1	1	1
ddos_http	0	0.93	0	0.94	0.15	0.96
ddos_tcp	0.99	1	0.97	1	0.97	1
ddos_udp	1	1	0.99	1	1	1
rcn_fngrprnt	0	0.97	0.99	1	1	1
rcn_scan	0.78	0.99	1	1	1	1
theft_data	0	0.87	0	0.87	0	0.86
theft_keylog	0.32	0.96	0	0.95	0	0.94

^{B}Balanced dataset.

traffic. This poor performance with regard to *theft_data* can be attributed to the low number of records for this attack class in the sampled dataset as shown in Table 1. ANN is the only model to identify *theft_key* with a precision of 1, recall of 0.19, and F1-score of 0.32. All three models identified the *rcn_scan* with ANN reported F1-score of 0.78 and both LSTM and GRU reporting F1-Score of 1. ANN could not correctly classify *rcn_fngrprnt*, while both LSTM and

Table 9: Performance of ML and DL models compared.

Works	Models and performance
Shafiq et al. [34]	Models have accuracy ≥99%. Models include decision tree, random forest, SVM, and naive Bayes.
Koroniotis et al. [35]	SVM, LSTM, and RNN models have accuracy ≥98%.
Our work	Deep LSTM, GRU, and ANN models perform with accuracy of 98% – 99%.

GRU were able to classify it with F1-score of 0.99 and 1, respectively. All models classified *ddos_udp* correctly with F1-score of 1. For *ddos_tcp*, ANN has had F1-score of 0.99, and LSTM and GRU both have received 0.97. GRU somewhat classified the *ddos_http* with an F1-score of 0.16, where both ANN and LSTM failed to classify it correctly. *dos_udp* was correctly classified by all 3 models. *dos_tcp* was classified by ANN with F1-score of 0.98, LSTM with F1-score of 0.95, and GRU with F1-score of 0.96. ANN could not correctly classify *dos_http*, whereas LSTM and GRU classified it with F1-score of 0.1 and 0.3, respectively. Lastly, all models classified the normal traffic well in precision; however, the recall performance dropped with ANN scoring 0.14, and LSTM and GRU scoring 0.7 and 0.72, respectively. When using the balanced dataset with equal samples from each of the classes, the results showed significant improvement in terms of precision, recall, and F1-score.

Compared to previous works that used ML and DL models, our models have performed well with accuracy ≥ 98% as shown in Table 9. The cited works differ in the use of different types of models and feature selection methods.

Deep learning models are preferred over classical (such as linear models and shallow ANN) machine learning models for big data since classical models take a significantly longer time to train on them. IIoT systems generate huge amounts of data in short periods, because of a large number of deployed IIoT devices. Considering Table 5 and Tables 6 to 8, it can be seen that deep learning models are promising in classifying IIoT attacks and can be potentially used for securing the IIoT network against previously unknown threats, thus protecting zero-day attacks.

In this work, two types of deep learning models are used for classifying the IIoT botnet attacks: the deeply connected neuron-based ANN and the recurrent neural network-based LSTM and GRU. All three models performed well in the selected performance measures across different attack types. The ANN had the highest average accuracy of 99% although it misclassified some attacks into the wrong category. LSTM and GRU are almost similar in performance; however, GRU performed slightly better in classifying some of the attacks such as *ddos_http* and *dos_http*. The poor performance of the models in precision and recall of identifying minority classes was fixed by balancing the dataset with equal size of classes. As for Industry 4.0, training deep learning models is computationally expensive. Thus, it may need to be optimized for deploying on IIoT systems and networks.

5. Conclusion

In this work, three different types of deep learning-based models—LSTM, GRU, and ANN—have been used for classifying ten different IIoT botnet attacks covering various communication protocols and devices. All the models are shown to have high performance with more than 98% classification accuracy. The implication of this study is that deep learning models can be used for IIoT malware detection especially within the context of novel threats that often elude the conventional methods. While the deep learning models may fail to identify minority classes, this can be fixed or improved by training the models on balanced dataset. Undersampling the majority classes have helped in correcting the imbalance in this case.

As the smart factories become more connected, the threats to people's data and privacy increase through sophisticated malware attacks and botnets. Deep learning models can be used for protecting these devices and networks by identifying the threats. The main advantage of these models is that they perform better as they learn from the big data produced by the billions of IIoT connected devices. In future works, areas of research that could be explored further include federated learning for IIoT networks as well as novel approaches to share threat analytics between devices and networks. Furthermore, different types of IoT datasets can be merged together to create a comprehensive IoT system dataset that can be used for training ML and DL models and provide security using federated learning and edge computing. For instance, healtcare IoT dataset [36] can be merged with IIoT datasets to extend the range and variety attacks on IIoT systems.

Data Availability

Previously reported IoT Botnet attack data (Bot-IoT dataset) were used to support this study and are available at: https://research.unsw.edu.au/projects/bot-iot-dataset. These prior studies (and datasets) are cited at relevant places within the text as reference [35].

Conflicts of Interest

The authors declare that they have no conflict of interest.

References

[1] E. Oztemel and S. Gursev, "Literature review of industry 4.0 and related technologies," *Journal of Intelligent Manufacturing*, vol. 31, no. 1, pp. 127–182, 2020.

[2] L. Zhou, K.-H. Yeh, G. Hancke, Z. Liu, and C. Su, "Security and privacy for the industrial Internet of Things: an overview of approaches to safeguarding endpoints," *IEEE Signal Processing Magazine*, vol. 35, no. 5, pp. 76–87, 2018.

[3] W. Ren, X. Tong, J. Du et al., “Privacy-preserving using homomorphic encryption in mobile IoT systems,” *Computer Communications*, vol. 165, pp. 105–111, 2021.

[4] A. A. Abd El-Latif, B. Abd-El-Atty, S. E. Venegas-Andraca et al., “Providing end-to-end security using quantum walks in IoT networks,” *IEEE Access*, vol. 8, pp. 92687–92696, 2020.

[5] M. M. Ogonji, G. Okeyo, and J. M. Wafula, “A survey on privacy and security of Internet of Things,” *Computer Science Review*, vol. 38, article 100312, 2020.

[6] G. De La Torre Parra, P. Rad, K.-K. R. Choo, and N. Beebe, “Detecting Internet of Things attacks using distributed deep learning,” *Journal of Network and Computer Applications*, vol. 163, article 102662, 2020.

[7] I. Ali, A. I. A. Ahmed, A. Almogren et al., “Systematic literature review on IoT-based botnet attack,” *IEEE Access*, vol. 8, pp. 212220–212232, 2020.

[8] A. Marzano, D. Alexander, O. Fonseca et al., “The evolution of bashlite and mirai iot botnets,” in *2018 IEEE Symposium on Computers and Communications (ISCC)*, Natal, Brazil, June 2018.

[9] R. Kour, “Cybersecurity issues and challenges in Industry 4.0,” in *Applications and Challenges of Maintenance and Safety Engineering in Industry 4.0*, pp. 84–101, IGI Global, 2020.

[10] J. Prinsloo, S. Sinha, and B. von Solms, “A review of Industry 4.0 manufacturing process security risks,” *Applied Sciences*, vol. 9, no. 23, p. 5105, 2019.

[11] S. Dange and M. Chatterjee, “IoT Botnet: the largest threat to the IoT Network,” in *Data Communication and Networks. Advances in Intelligent Systems and Computing, vol 1049*, L. C. Jain, G. A. Tsihrintzis, V. E. Balas, and D. K. Sharma, Eds., pp. 137–157, Springer, Singapore, 2020.

[12] L. Gupta, T. Salman, A. Ghubaish, D. Unal, A. K. Al-Ali, and R. Jain, “Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach,” *Applied Soft Computing*, vol. 118, p. 108439, 2022.

[13] F. Hussain, R. Hussain, S. A. Hassan, and E. Hossain, “Machine learning in IoT security: current solutions and future challenges,” *IEEE Communications Surveys & Tutorials*, vol. 22, no. 3, pp. 1686–1721, 2020.

[14] D. Krishnan and P. Babu, “Imbalanced classification for botnet detection in Internet of Things,” in *Next Generation of Internet of Things. Lecture Notes in Networks and Systems, vol 201*, R. Kumar, B. K. Mishra, and P. K. Pattnaik, Eds., pp. 595–605, Springer, Singapore, 2021.

[15] S. Bagui and K. Li, “Resampling imbalanced data for network intrusion detection datasets,” *Journal of Big Data*, vol. 8, no. 1, p. 6, 2021.

[16] M. Hammoudeh, J. Pimlott, S. Belguith et al., “Network traffic analysis for threats detection in the Internet of Things,” *IEEE Internet of Things Magazine*, vol. 3, no. 4, pp. 40–45, 2020.

[17] S. H. Jafier, “Utilizing feature selection techniques in intrusion detection system for Internet of Things,” in *ICFNDS '18: Proceedings of the 2nd International Conference on Future Networks and Distributed Systems*, pp. 1–3, New York, New York, USA, June 2018.

[18] Q. Wang, X. Zhu, Y. Ni, G. Li, and H. Zhu, “Blockchain for the IoT and in dustrial IoT: a review,” *Internet of Things*, vol. 10, article 100081, 2020.

[19] M. Hammoudeh, G. Epiphaniou, S. Belguith et al., “A service-oriented approach for sensing in the internet of things: intelligent transportation systems and privacy use cases,” *IEEE Sensors Journal*, vol. 21, no. 14, pp. 15753–15761, 2020.

[20] L. Liu, B. Xu, X. Zhang, and X. Wu, “An intrusion detection method for Internet of Things based on suppressed fuzzy clustering,” *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, 2018.

[21] M. Zubair, D. Unal, A. Al-Ali, and A. Shikfa, “Exploiting bluetooth vulnerabilities in e-health IoT devices,” in *ICFNDS '19: Proceedings of the 3rd International Conference on Future Networks and Distributed Systems*, pp. 1–7, New York, NY, USA, July 2019.

[22] T. A. Tuan, H. V. Long, L. H. Son, R. Kumar, I. Priyadarshini, and N. T. Son, “Performance evaluation of Botnet DDoS attack detection using machine learning,” *Evolutionary Intelligence*, vol. 13, pp. 283–294, 2019.

[23] I. Ghafir, V. Prenosil, M. Hammoudeh et al., “BotDet: a system for real time botnet command and control traffic detection,” *IEEE Access*, vol. 6, pp. 38947–38958, 2018.

[24] H. Sedjelmaci, S. M. Senouci, and M. A. Abu-Rgheff, “An efficient and lightweight intrusion detection mechanism for service-oriented vehicular networks,” *IEEE Internet of Things Journal*, vol. 1, no. 6, pp. 570–577, 2014.

[25] F. Y. Yavuz, D. Unal, and E. Gul, “Deep learning for detection of routing attacks in the Internet of Things,” *International Journal of Computational Intelligence Systems*, vol. 12, no. 1, pp. 39–58, 2018.

[26] E. Anthi, L. Williams, and P. Burnap, “Pulse: an adaptive intrusion detection for the internet of things,” in *Living in the Internet of Things: Cybersecurity of the IoT - 2018*, p. 4, London, UK, 2018.

[27] S. Madhawa, P. Balakrishnan, and U.-m. Arumugam, “Data driven intrusion detection system for software defined networking enabled industrial Internet of Things,” *Journal of Intelligent & Fuzzy Systems*, vol. 34, no. 3, pp. 1289–1300, 2018.

[28] F. Yulong, Z. Yan, J. Cao, O. Kone, and X. Cao, “An automata based intrusion detection method for Internet of Things,” *Mobile Information Systems*, vol. 2017, Article ID 1750637, 13 pages, 2017.

[29] A. Tabassum, A. Erbad, and M. Guizani, “A survey on recent approaches in intrusion detection system in IoTs,” in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1190–1197, Tangier, Morocco, June 2019.

[30] S. U. Jan, S. Ahmed, V. Shakhov, and I. Koo, “Toward a lightweight intrusion detection system for the Internet of Things,” *IEEE Access*, vol. 7, pp. 42450–42471, 2019.

[31] C. M. Liu, Y. Zhang, R. Chen, L. X. Xiao, and J. D. Zhang, “Research on intrusion detection for the Internet of Things based on clone selection principle,” *Advanced Materials Research*, vol. 562-564, pp. 1982–1985, 2012.

[32] R. Chen, C. M. Liu, and C. Chen, “An artificial immune-based distributed intrusion detection model for the Internet of Things,” *Advanced Materials Research*, vol. 366, pp. 165–168, 2011.

[33] H. Naeem and A. A. Bin-Salem, “A CNN-LSTM network with multi-level feature extraction-based approach for automated detection of coronavirus from CT scan and X-ray images,” *Applied Soft Computing*, vol. 113, article 107918, 2021.

[34] M. Shafiq, Z. Tian, A. K. Bashir, X. Du, and M. Guizani, “CorrAUC: a malicious bot-IoT traffic detection method in IoT

network using machine-learning techniques," *IEEE Internet of Things Journal*, vol. 8, no. 5, pp. 3242–3254, 2021.

[35] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Generation Computer Systems*, vol. 100, pp. 779–796, 2019.

[36] A. A. Hady, A. Ghubaish, T. Salman, D. Unal, and R. Jain, "Intrusion detection system for healthcare systems using medical and network data: a comparison study," *IEEE Access*, vol. 8, pp. 106576–106584, 2020.

[37] G. Lemaitre, F. Nogueira, and C. K. Aridas, "Imbalanced-learn: a python toolbox to tackle the curse of imbalanced datasets in machine learning," *Journal of Machine Learning Research*, vol. 18, no. 17, pp. 1–5, 2017.

[38] M. Mudassir, S. Bennbaia, D. Unal, and M. Hammoudeh, "Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach," *Neural Computing and Applications*, pp. 1–15, 2020.

[39] J. N. Mandrekar, "Receiver operating characteristic curve in diagnostic test assessment," *Journal of Thoracic Oncology*, vol. 5, no. 9, pp. 1315-1316, 2010.