Computers, Materials & Continua
DOI: 10.32604/cmc.2022.027212
Article

Tech Science Press

Securing Consumer Internet of Things for Botnet Attacks: Deep Learning Approach

Tariq Ahamed Ahanger[1,*], Abdulaziz Aldaej[1], Mohammed Atiquzzaman[2], Imdad Ullah[1] and Mohammed Yousuf Uddin[1]

[1]College of Computer Engineering and Science, Prince Sattam bin Abdulaziz University Al-Kharj, 11942, Saudi Arabia
[2]School of Computer Science, University of Oklahoma Norman, 73019-6151, United States
*Corresponding Author: Tariq Ahamed Ahanger. Email: t.ahanger@psau.edu.sa
Received: 13 January 2022; Accepted: 19 April 2022

Abstract: DDoS attacks in the Internet of Things (IoT) technology have increased significantly due to its spread adoption in different industrial domains. The purpose of the current research is to propose a novel technique for detecting botnet attacks in user-oriented IoT environments. Conspicuously, an attack identification technique inspired by Recurrent Neural networks and Bidirectional Long Short Term Memory (BLRNN) is presented using a unique Deep Learning (DL) technique. For text identification and translation of attack data segments into tokenized form, word embedding is employed. The performance analysis of the presented technique is performed in comparison to the state-of-the-art DL techniques. Specifically, Accuracy (98.4%), Specificity (98.7%), Sensitivity (99.0%), F-measure (99.0%) and Data loss (92.36%) of the presented BLRNN detection model are determined for identifying 4 attacks over Botnet (Mirai). The results show that, although adding cost to each epoch and increasing computation delay, the bidirectional strategy is more superior technique model over different data instances.

Keywords: Internet of Things; deep learning security; DDoS attack; botnet

1 Introduction

With an anticipated 43 billion gadgets linked to the Internet by 2023, the Internet of Things (IoT) is predicted to revolutionize the next decade of enhanced connectedness. IoT technology's main goal is to link devices to the internet, resulting in smart gadgets that can gather, store and share data without the need for human participation [1]. Many of these IoT gadgets are geared at consumers who place low costs and convenience of use over security [2,3]. As a result of these commercial incentives, IoT manufacturers have omitted crucial security elements, resulting in a swath of unsecured gadgets including internet camcorders and video recording boxes [4,5]. Inherent computational limits, the usage of default passwords and insecure protocols are all common sources of vulnerabilities and attacks [6,7]. The fast spread of vulnerable IoT devices, as well as the simplicity with which attackers may find them via internet services like shodan, creates a rapid-growing ocean of vulnerabilities

[8,9]. By compensating and exploiting a significant number of susceptible IoT-oriented gadgets, unauthenticated users can launch significant attacks against internet resources including Distributed Denial of Service (DDoS), Phishing, and Spamming [10–12]. The current increase in IoT-based DDoS attacks is expected to continue unless IoT makers assume responsibility and include security procedures into the equipment [13,14]. Fig. 1 [15] shows the statistical visualization of attacked sites in 2011. Moreover, the IoT poses a variety of concerns since it can transform into a major platform for futuristic internet-oriented attacks [16]. Due to the quantifiable elevation of DDoS attacks over targeted user-level IoT networks, minimal understanding of inherited vulnerability among devices is another major concern [17]. The aforementioned issue is exacerbated by the reduced user interface over IoT networks, making it nearly hard for customers to detect and recognize attacks in corresponding home networks [18].

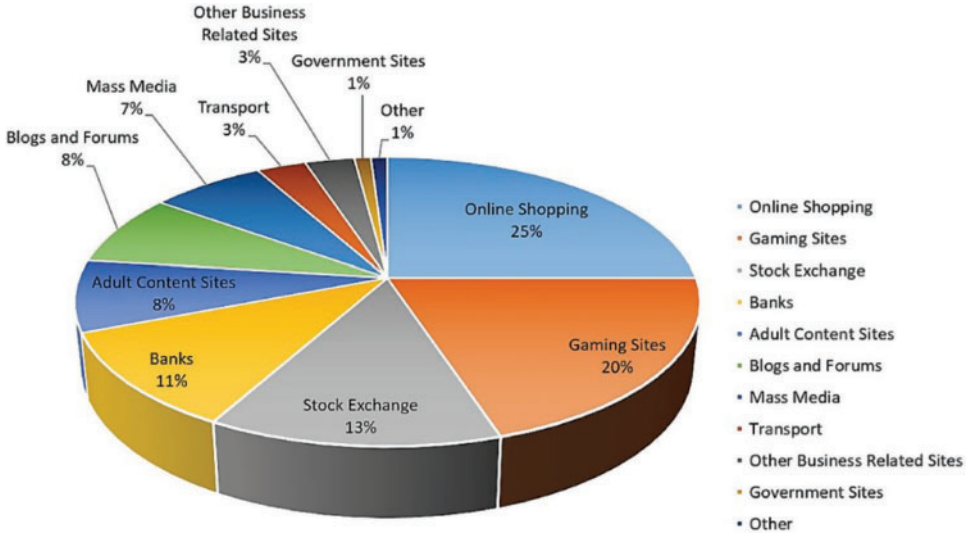

Figure 1: Statistical visualization for attacked sites

Table 1: Abbreviation used

Abbreviation	Description
BLRNN	Recurrent Neural networks and Bidirectional Long Short Term Memory
DL	Deep Learning
DDoS	Distributed Denial of Service
CC	Control and Command
MIPS	Million Instructions Per Second
ML	Machine Learning
NN	Neural Network
LSTM	Long Short Term Memory
RNN	Long Short Term Memory
DNS	Domain Name Server
AES	Average Absolute Error

(Continued)

Table 1: Continued

Abbreviation	Description
ACK	Acknowledgment
UDP	User Datagram Protocol

1.1 Research Domain

A fundamental review of concerned domain-oriented research has been performed to build a secure sandboxed botnet environment to substantiate IoT-botnet attack detection. In the current study, an IoT-oriented camcorder is attacked to launch a series of DDoS attacks on a predetermined device. The camcorder did not show any signs of vulnerability during the infection process or attacks and it continued to work normally. The camcorder could still be accessed remotely and overall processing was not impacted. Video streaming remained as responsive as before, indicating that, in the absence of visible symptoms of infection, identification of botnet activity would be a challenging task within users' complex IoT networks. Current botnet identification techniques, including flow-based intrusion determination and impression, have failed to stop the proliferation of IoT botnets. The absence of protocols inside consumer IoT networks is mostly due to code alternations rendering vulnerabilities useless [19,20].

1.2 Research Contribution

The major purpose of the presented study is to propose a method for botnet activity detection in user oriented IoT networks. Conspicuously, Recurrent Neural Network inspired Deep Bidirectional Long Short Term Memory (BLRNN) used to create a unique detection model. Identification was done at the data packets, with an emphasis on text identification, which is generally ignored by flow-based detection procedures. For text recognition and conversion, the word embedding technique was utilized and it proved to be an excellent way for anticipating attack paths. The accuracy and loss of the BLRNN detection model were compared with the LRNN and other state-of-the-art Deep Learning (DL) models. Based on the aforementioned aspects, the following are the primary contributions of the research paper:

1. Formulating a consumer IoT scenario for attack vectors and botnet traffic.
2. Proposing a DDoS identification technique for text determination of characteristics inside attack vectors.
3. Deploying the proposed Recurrent Neural Network based DDoS attack models to identify and forecast infected IoT networks.
4. Assessing the performance of the proposed DL-inspired botnet attack detection model in comparison to the state-of-the-art DL techniques.

Paper Structure Section 2 discusses botnet activity in the IoT and how DL may be used to identify attacks. The botnet architecture utilized to construct the botnet dataset is described in Section 3. It also explains how to develop a botnet detection model using a BLRNN in conjunction with the word embedding approach. The procedure of data gathering and pre-processing is described in Section 4. Section 5 assesses the experimental findings and performance enhancement by comparative analysis with the state-of-the-art DL techniques. Section 6 concludes the paper with future research directions.

2 IoT: Security Perspective

DDoS attacks are one of the severally widespread and damaging security attacks on the cyber world. Recent years have seen severe DDoS attacks over the internet. Over 99 Gbps of attacks were registered in 2016 with an increase of 139%, with 3 attacks exceeding 299 Gbps [21]. In the 4th quarter of 2017, 90% of DDoS attacks used an effective technique of multi-vector, which was deployed over the IoT network [22]. In the last few years, the Mirai botnet is considered the major IoT network-specific DDoS attack. Mirai is a malware code that infects underneath IoT-network to create a robotic network termed as botnet. Botnet comprises compromised IoT devices (also referred to as bots). The bots are remotely controlled by an attacker (botmaster) via a control and command (CC) server, causing them to engage in DDoS attacks on Internet targets. The Mirai botnet was utilized to launch a 620 Gbps DDoS attack over a website, krebsonsecurity.com. Moreover, it was also responsible for subsequent DDoS attacks over DNS provider (DYN) and French business (OVH) resulting in the compromise of nearly 1,00,000 devices in the IoT network, resulting in the bandwidth elevation of 1.2 Tbps. The DYN attack caused considerable internet disturbance. Moreover, it left numerous high-profile websites unavailable, including GitHub, Twitter, Reddit and Netflix. Mirai's infection mechanism and spread strategy are depicted in Fig. 2. A CC server, a Scanner server and bots (compromised device in IoT-network) make up the Mirai infrastructure. Several devices in IoT-network executing busy-box (Linux version), get infected and spread by using weak security credentials.

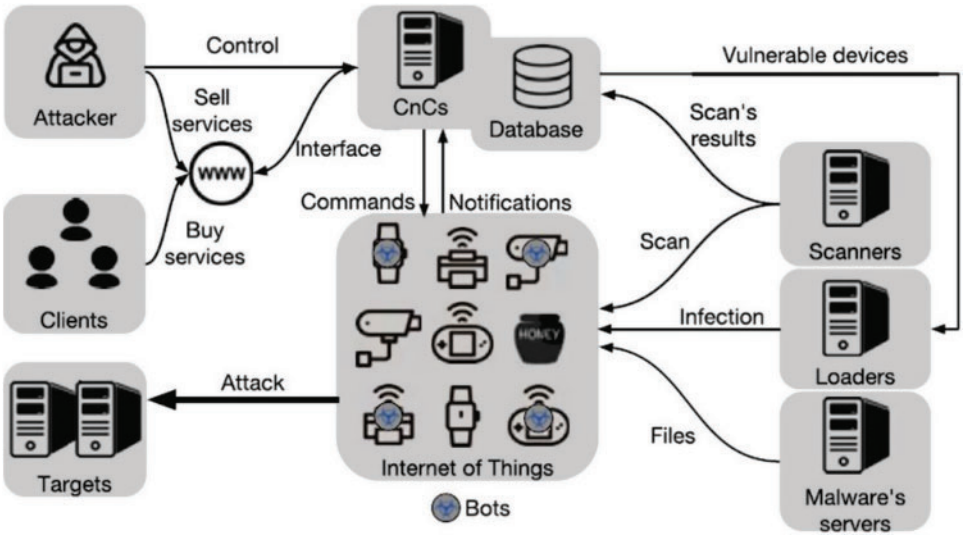

Figure 2: Systematic workflow of IoT botnet attack

1. An attacker (botmaster) begins the procedure by communicating with the scanner server and launching the scanner. c component to search for susceptible devices in IoT-network over the internet that has telnet services and ports 23 or 2323 exposed.
2. The virus initializes a brute force technique for a successful login utilizing default user-name and password, after discovering a susceptible device.
3. If the login procedure is successful, the corresponding user-name and password along with device details are transmitted to the CC server. The scan server will use these to log in and deliver malware to the susceptible device.

4. The CC server sends an infect command to the scanner server, which includes all relevant information such as login credentials, IP address and device architecture. Arm, MIPS, Sparc, and PowerPC are some of the hardware architectures supporting Mirai
5. The acquired information is used by the scanner server to sign in and tell the susceptible device to connect to the scanner server through TFTP or WGET. After connection the bot downloads and executes the malware. After being infected, the first compromised IoT device joins the Mirai botnet.
6. To evade detection, the malware version of the binary file is deleted and memory execution proceeds. Attack instructions may now be sent by the botmaster, with specified criteria of the target and attack duration.
7. ACK packet flood (ack), Recursive DNS (DNS), UDP flood (UDP), SYN packet flood (syn) and GRE flood (GRE IP) are among the 10 DDoS attack methods included in the virus. Other types of DDoS attacks are depicted in Tab. 2 [15] with detailed summarization.
8. The initial bot is currently exploring for more susceptible devices over IoT-network with Telnet services and open ports (2323 or 23), to perform the infection procedure again and expand the botnet. The conceptual view of the botnet infection procedure is depicted in Fig. 3.
9. The CC server receives new information on susceptible IoT devices.
10. The scanner server along with the loader server receives a fresh infection command. The detailed procedure for infection is presented in Fig. 4. Specifically, 3 technique are used including Central Source-based Method, Back Propagation Method and Automatized Propagation Technique.
11. On the newly discovered susceptible IoT device, the necessary hardware binary is loaded.
12. The CC server sends out the appropriate command for executing the attack.
13. The attack is carried out in collaboration with the first bot by the newly created 2nd bot.
14. The process of looking for other susceptible IoT devices is repeated to extend the botnet.

The detailed comparative analysis with state-of-the-art works is depicted in Tab. 3.

Table 2: DDoS attack summary

DDoS attack	Exploitation attack type	IP spoofing	Attack attribute	Attack effect
TCP SYN attack	Protocol exploiting resource depletion	spoofed	TCP 3-way handshaking	Consumer Resource
HTTP flood	Resource Depletion Protocol	Non-Spoofed	HTTP GET and POST Request	Consumer Resource
TCP PUSH+ACK	Resource Depletion Protocol	Spoofed and Non-Spoofed	Sets PUSH and ACK	Memory Flooding
slowlories	Resource Depletion Protocol	Non-Spoofed	Keep HTTP connection open	Consumer sockets
SIP flood	Resource Depletion Protocol	Non-Spoofed	HTTP GET and POST Request	Consumer sockets
Slow read	Resource Depletion Protocol	Non-Spoofed	Slow rate for data read	Consumer sockets

(Continued)

Table 2: Continued

DDoS attack	Exploitation attack type	IP spoofing	Attack attribute	Attack effect
Land Attack	Resource depletion using malformed packets	spoofed	Sets I to all quality if service bits	Sets I to all quality if service bits
DNS flooding	Flooding-based infrastructure attack	spoofed	Exploits DNS response message	Exploit user's network bandwidth
NTP amplification	Amplification based bandwidth depletion attack	spoofed	Exploits NTP commands	Exploit user's network bandwidth
DNS amplification	Amplificationbased bandwidth depletion	Spoofed	Exploits DNS commands	Exploit user's network bandwidth
Fraggle	Protocol exploited bandwidth depletion attack	spoofed	Exploits ICMP request packet	Exploit user's network bandwidth
UDP Flood	Protocol exploited bandwidth depletion	spoofed	Sends large stream of UDP packets to random port	Exploit user's network bandwidth

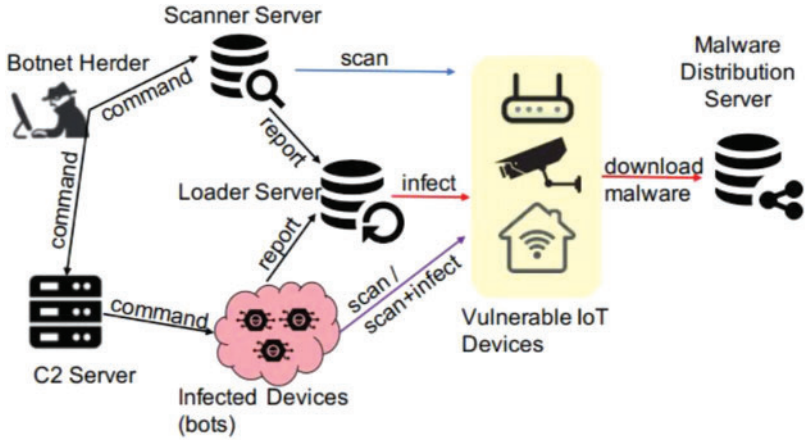

Figure 3: IoT botnet infection

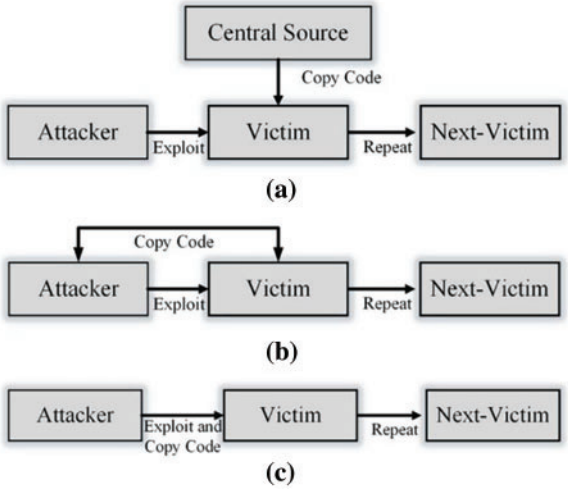

Figure 4: Attacker code transmission: (a) Central source-based method; (b) Back propagation Method; (c) Automatize propagation technique

Table 3: State-of-the-art comparison (1 Available, 0 Not Available)

References	Chen et al. [21]	Agrawal et al. [12]	Saharkhizan et al. [22]	Ray et al. [23]	Li et al. [14]	Puat et al. [24]	Snehi et al. [17]	Proposed
DDoS Attack	1	1	1	1	1	1	1	1
IoT Domain	0	1	1	0	1	0	0	1
Quantification	1	1	1	1	1	0	1	1
Mirai Attack	1	1	1	1	0	0	1	1
Deep Learning	0	0	0	0	1	0	0	1
Botnet	0	0	0	0	0	1	1	1
Security	0	0	0	0	0	1	0	1
Reliability	0	0	0	0	0	0	0	1
Stability	0	0	1	0	0	0	0	1

3 DL Techniques for Attack Detection

The growing use of IoT networks in a widespread service domain, as well as growing computing and processing capabilities, make them a lucrative target for attacks like network packets and malware targeted to infiltrate specific IoT devices. Because of the unique service needs, such as low latency, resource specialization, dispersed nature, and mobility, to name a few, attack detection in IoT systems differs significantly from conventional approaches. As a result, traditional network attack detection is restricted in its application to IoT security issues. According to Kaspersky Lab, the majority of IoT devices analyzed in 2016 were unsafe, with default passwords or vulnerabilities that were readily exploited by Mirai and Hajime malware. Because of the inclusion of numerous IoT protocols, a significant number of zero-day attacks are constantly appearing. The majority of these attacks are minor variations of previously known cyber-attacks, making them difficult to identify even with powerful computational intelligence processes like classical machine learning systems. Due to the broad diversity and spread of IoT devices, it is cumbersome to incorporate Machine Learning (ML)

for assessing cyber-security, particularly for devices under IoT-network having minimal computation capacity. The success of DL in numerous big data sectors, on the other hand, has sparked significant interest in cybersecurity. Advancements in computing hardware and the incorporation of novel Neural Network (NN) library including Tensorflow and Theano have made DL procedures more practical. Additionally, the access of heterogeneous training data instances has contributed to the efficacy of DL procedures. In the disciplines of image processing, pattern recognition and computer vision, DL offers numerous improvements in traditional artificial intelligence applications. DL networks are capable of significantly improving classification and prediction accuracy in these difficult tasks. DL's key advantage is the lack of human attribute processing, pre-computation, and data reduction capability, which make it possible to use DL even in networks with limited resources. It indicates that DL's capacity to self-learn leads to improved accuracy and quicker processing, which may be used to efficiently identify novel distributed attacks in IoT systems. It is critical for securing the IoT networks due to a variety of vulnerabilities, including jamming, spoofing, replaying, and eavesdropping, as well as resource limits including unsafe programming languages, and out-of-memory accesses. The major goal of the presented study is to utilize an effective DL technique to identify botnet attacks. Botnet attacks have been effectively mitigated using other ML and evolutionary computing approaches. As an illustration, swarm intelligence can be used to remove any fixed master-slave connection between bots and to autonomic bot operational duties. Botnets' changing behavior often allows them to get beyond standard detection methods. However, by identifying the behavior adopted by botnets throughout their life cycle, the development of behavioral detection tools has aided in coping with the continual shift in botnet activity. For example, all bots must connect to the CC server to receive the next instructions. It can lead to detection systems if watched over a lengthy period. The requirement of properly dealing with enormous data sequences is one of the consequences of long-term network traffic observation. In general, RNN backed up by the LSTM network, is successful in identifying diverse sequential states that are temporally variant, compensating delays between input and output. The proposed model is effective for jobs with decision-making services in machine translation, natural language processing, and image recognition. Moreover, it has been demonstrated to be an efficient framework for tagging tasks. Furthermore, a bidirectional LSTM provides 2 distinct layers to collect ubiquitous historical and futuristic data. The major objective of the presented research is the use of the LSTM network for assessing variable traffic to identify attacks. The detailed procedure is depicted in Tab. 4.

Table 4: Botnet attack identification algorithm

Botnet attack identification technique
Input: Data input values for DDoS attacks ***Objective:*** Estimate Performance
Data Processing unitToDrop = 19% Format Parsing Describe tokenization ***Do*** Format Parsing ***for*** row = 1 to n, do

(Continued)

Table 4: Continued

Botnet attack identification technique
Tokenized conversion to integer format
Perform Indexing
Dictionary creation for indices
padding with Max 31
Additional Attribute Injection
End For Loop
Until Dataset Return = True
Data Training and Testing
Validation
Sequential=Model
Assign Cell = 0
Sigmoid Activation
Assign mae = loss
Optimize using Adam
epochs = 150
Form New LSTM and BLSTM module
Add LSTM
Initialize Dense Layer
Add Dense layer
Set Activation Function
Perform Compilation
Do
for epoch=1 to 150
Estimate Validation
Estimate Accuracy
End for
End
Return Accuracy and Validation
Output: performance Estimation

4 Research Framework

A full description of the botnet environment and procedure implementation is provided to aid the repeatability of the presented study.

4.1 Simulation Setup

For simulation, a secure sandboxed environment was constructed as shown in Fig. 5. The presented technique included a CC server, a scanner server, a loader server, and a utility server for DNS data analysis. All relevant traffic was mirrored to a packet sniffer device through port Tap0, which is captured for subsequent examination. For launching an attack over the target device (Raspberry Pi), 2 Sricam SP009b Wireless HD IP WiFi camcorder executing busybox utility were utilized to formulate bots. GitHub repository was used for obtaining the Mirai source code. Although changes in source

code were minimally performed to achieve a faithful simulation of a Mirai attack, some configuration alterations were necessary to meet the legal restrictions.

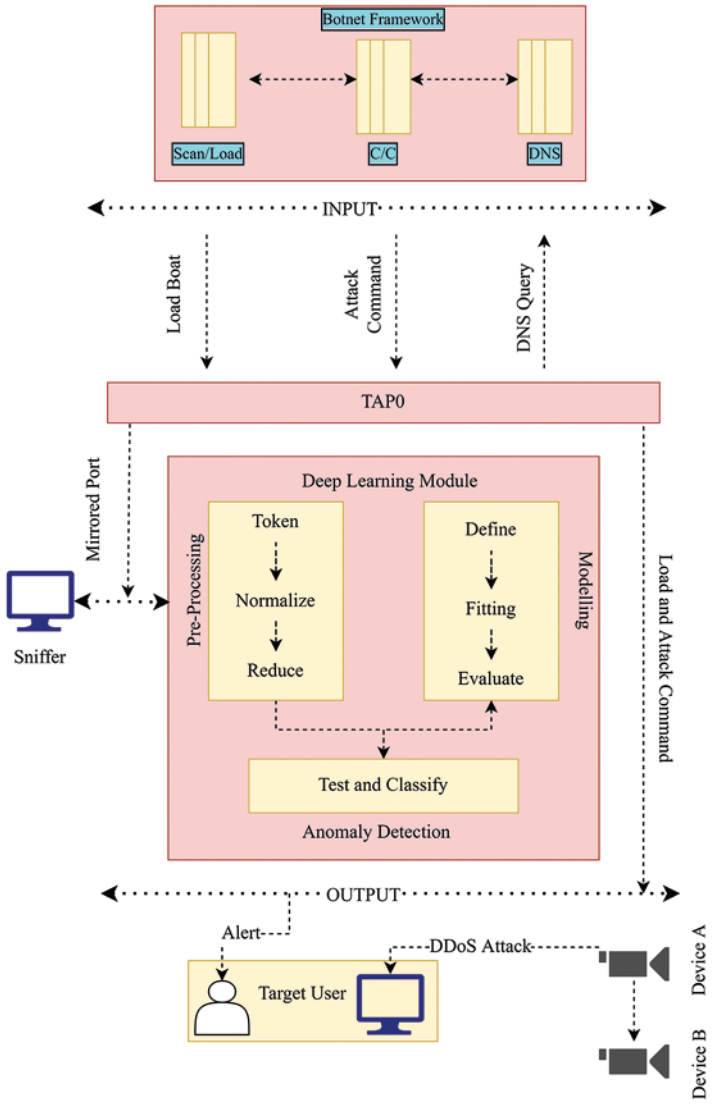

Figure 5: DL Technique for attack detection

4.1.1 Configuration of CC Server

Necessary apt-get-install was used to install the vital programs. Several domains were established and attached to the main programs. The default values of (NULL,'user1','password', 1, 1, 1, 1, 0, 0, 29, '') were inserted. Once set up, the root program was updated to add the default user name and passwords. ARM compilers were installed for the underneath binary hardware, and required output paths were added. The CC server was used to transmit data regarding connections, hindrances, and data transmission status. Using build.sh function, the needed binary source codes were built and saved in the output directory.

4.1.2 Scan Loader Server Configuration

apt-get-install-apache2.y command was used to install Apache. The binary architecture files that had been prepared were transferred to the bin directory. The networking address of the scanner server was inserted into the main file and complete access was granted using the chmod command. build.sh command was executed to compile the loading file and place it in the loader directory. IP addresses that are excluded were adjusted in scanner file to fit in the proposed model. It reduces the IP address range accessible for the scanner server and ensures the utilized range was permitted in the proposed environment. The scanner server networking address was added to scanListen file. Moreover, the default port number for brute force attacks was set to 49212. The scanListen was relocated to the loader server from the tools directory using go-build command. Because the camcorder utilized in the current research didn't come with wget, tftp was installed separately. Using touch command, a tftp configuration was built, with tftpboot defined as the location where the computational files will be stored for subsequent delivery of the data-load.

4.1.3 Domain Name Server Parameters

To determine the networking address of the CC server, the Mirai virus requires accessibility to a DNS. Conspicuously, Bind9 software was utilized to construct the 2 necessary domain reports. The bots will interact with the CC server and present the information from the device in the IoT network.

4.2 Word Embedding Mechanism

To transform string data obtained in recorded packets into a format useable by the BLRNN, the created model employs a unique application of Word Embedding. The experimental configuration described in the previous section was employed to produce the dataset used in the presented research. It includes Mirai botnet traffic including Infect, Scan, Attack traffic and Control, Camcorder data produced in the proposed model. In the feature of the dataset, there were attributes including Length, Source, Time, Protocol, Destination and cumulative payload data. Some characteristics, such as No. and Time, were eliminated since they did not provide many opportunities for data analysis. The attribute depiction is detailed in Tab. 5. The majority of the gathered data was stored as shown in Tab. 6. Henceforth, the framework could read and interpret the textual feature. DL technique of conventional Recurrent Neural Networks (RNN) and Long Short Term Memory (LSTM) only work with numerical data. However, [24] showed that a Deep Bidirectional LRNN (BLRNN) can be employed to provide promising text identification results. Furthermore, the efficacy was enhanced when a BLRNN was used with word embedding to co-relate real numbers and textual vectors. It proves to be an excellent approach for modeling and detecting sequential text. Based on the aforementioned beneficial aspects, the current study provides a detection technique and paradigm for attack identification in the IoT network. The presented approach initializes by turning every letter in a token and integer format. Moreover, a dictionary of token words and the corresponding index was produced, and text was replaced with the matching index number. It was necessary to keep the sequential order to comprehend each attack type, therefore an array of indices was established. The protocol and length attributes must be included in the array since attacks are frequently linked to the protocol utilized and the length of the recorded packet. To transform and construct a dictionary of tokens and corresponding indices, a word embedding procedure was employed once more. Words were then added to the array, in addition to the integral length attribute. Labels for every kind of intercepted packet were converted from string to integer and inserted in the formulated array. Moreover, Keras package with a wrapper Tensorflow and Theano API were utilized to complete the procedure. Simultaneously, the Keras function was used to transform the text into the index, generate

a two-dimensional list, and formulate a dictionary. Finally, because deep neural networks demand equal-length arrays, identification of the maximum length of a phrase was necessary. Moreover, all the arrays were padded with 0 to make the total length equal to 32. The data was separated into train data and test data. After processing and reshaping into the 3-dimensional array, the format needed by the LSTM layer was acquired. The detailed procedure is depicted in Tab. 1.

Table 5: Model attributes

Variables	Measure
Activation function	Sigmoid
Epochs	150
Dense layer total unit	7
Loss	Mean Absolute Error
BLSTM layer total units	19
Optimizer	Adam

Table 6: Data attributes

	Attack	Mirai	Normal	Clean
UDP	8270	1465	476413	500431
DNS	7605	3307	477300	501385
Mirai	0	4001	487565	484367
ACK	56333	5261	477450	521778

4.3 DL-Inspired Modeling Approach

The use of DL to botnet attack identification in the IoT network is the main contribution of the current study. As mentioned previously, word embedding was utilized to translate textual data in a tokenized integral format for usage with the DL technique. The detection model is assessed using numerous DL techniques and compared with state-of-the-art techniques based on a variety of attacks connected with the Mirai botnet [25,26]. To create identification models, neural network layers with sigmoid activation are added to the proposed technique. The proposed neural network is constructed using the Average Absolute Error (AAE) function and the Adam optimizer for 150 epochs. The suggested detection approach, as illustrated in Fig. 5, is divided into 3 phases. The Preprocessing phase modifies characteristics to ensure that the data representation is appropriate for the procedures that will be employed. Before normalization and elimination of packets with missing data, the data is tokenized using the word embedding approach. To design, fit and assess the detection model, numerous procedures are employed in the training phase. Finally, the created dataset is evaluated in the Anomaly Detection phase to see how successful the model is in terms of statistical parameters.

5 Experimental Simulation

5.1 Data Source

A dataset is required with a combination of botnet communication, pathways and device traffic corresponding to the IoT network to test the proposed detection methods. Because there are presently no publicly available datasets that meet all 3 criteria, data was generated synthetically. Infected IoT devices may use 10 different attack channels in the Mirai botnet software to launch DDoS attacks on targets. 4 attack vectors, including Synchronize (SYN), Domain Name System (DNS) flood, User Datagram Protocol (UDP) flood, and Acknowledgment (ACK) flood attacks, were chosen for the presented approach. Normal traffic generated by the camcorder, as well as CC communications between the server and the bot were also intercepted. The tcp-dump instruction was used to acquire data instances and create the necessary dataset. The instruction specifies that the acquisition divides the data into 5 files and a maximal acquisition size of 1 GB. The required data was taken in 5 distinct captures, which were afterward accumulated in a single dataset. Initial data acquisition comprised of data traffic generated from the IoT device for 2 h, including normal device network connectivity as well as 2 remote camcorder connectivity for realtime streaming. Mobax-term is incorporated to establish a secure connection to the CC server before launching the MYSQL database. Secondly, remote connectivity was utilized to log in to the CC server, allowing the compromised IoT IP camera to issue attack commands. The model acquired the next attacked camcorder via port numbers 2323 and 23, as well as the initialization of the scanning process. Thirdly, a remote connection was utilized before scanning the network device availability for further infection using the loader server. The capture was a flooding attack, in which the CC server delivered the infection command and the infected IoT device transmits burst UDP packets to its target for the 60s. The fourth capture-type was alone flooding attack, in which the CC server generated attack instruction. This resulted in the burst transmission of DNS packets for the 60s to the targeted device. Finally, the capture was alone flooding attack of the ACK packets, in which 60s burst transmission was made to the infected device. The captured files were transformed to CSV format. Labels of DNS, mirai, UDP, ACK and Norm were denoted to the acquired data. Further, the acquired data is fed to the attack identification model for determining the performance enhancement of the proposed framework. After missing data packets have been removed, the normalized data represents the cumulative samples. Even though the mirai recorded packets as depicted in Tabs. 7 and 8 looked to be similarly sophisticated, the information in the packets remained the same.

Table 7: Packet structure

Packet	Source	Destination	Time	Size	Protocol
ACK	192.168.249.39	192.168.249.11	3.40	89	DNS
ACK	192.168.249.39	192.168.249.39	0.15	449	UDP
ACK	192.168.249.39	192.168.249.49	0.15	59	TCP
ACK	192.168.249.39	192.168.249.59	0.01	5955	TCP

Table 8: Structure of ACK packet

Packet	Source	Destination	Time	Size	Protocol
ACK	192.168.249.39	192.168.249.11	3.40	455	DNS
ACK	192.168.249.39	192.168.249.11	0.83	455	UDP
ACK	192.168.249.39	192.168.249.11	0.83	455	TCP
ACK	192.168.249.39	192.168.249.11	0.83	59	TCP

5.2 Comparative Analysis

A set of 4 tests were conducted and compared with the state-of-the-art DL techniques for detecting the performance of attack identification. Because unidirectional LRNNs can only save historical data, the goal of the comparative analysis is to analyze whether utilizing bidirectional LRNN, which can collect contextual information from temporal instances, can perform with superiority in comparison to the state-of-the-art DL techniques. Test 1 splits captured data in training data and validation data, which was used to train each model with a total of 19 iterations. Initially, loss function and average accuracy were calculated and depicted in Tab. 9. For the Test-1 statistical parameters are compared to the conventional LRNN and BLRNNN model. The mathematical model for BLRNN is presented as $h_t = BLRNN_{enc}(x_t, h_{t-1})_t = BLRNN_{dec}(t, s_{t-1})$, such that $c_i = \sum_{j=1}^{T_x} \propto_{ij} h_j \propto_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})} e_{ij} = a(s_{i-1}, h_j)$, such that a is an specific attention function Both models produced great accuracy and prediction for DNS, UDP, and mirai attack types. ACK attacks, on the other hand, yielded fewer favorable findings, even though this attack had the most samples. It is due to the type and data complexity of the sequence number. On the other hand, there was an integral data format, where contiguous packet sequence numbers (48582-30047, 30047-48582) were interlinked, and consistency existed in size and length. However, several packets were not synchronized which may have caused the detection model to miss this pattern, resulting in a lower detection rate and a substantially larger loss measure. Test-2 used UDP, ACK, mirai, DNS and Norm attacks to realize a multi-dimensional attack environment due to its vulnerable behavior. The impact of all attacks for the identification of accuracy and loss function is shown in Tab. 9. Test-3 comprises concatenating norm, UDP, mirai, and DNS capture to build a multi-dimensional attack scenario. When the ack attack is eliminated, the model's overall detection accuracy and prediction are quite good, as shown in Tab. 9. Test-4 provided another validation of the proposed model, in which 3 ack attacks were performed simultaneously to increase the data size and evaluate the variance in accuracy and prediction. When provided with a larger sample size, Tab. 9 reveals that increasing the sample size increases overall validation accuracy to 95.23%, with BLSTMRNN delivering the minimal loss function, indicating that the presented model was able to better anticipate attack traffic. Even though the bidirectional technique adds epoch-overhead and enhances temporal delay, the trajectory demonstrates a superior model over time. The brief analysis report is presented in Tab. 10.

Table 9: Performance results

Attribute	Train	Validate	BLSTM accuracy	BLSTM loss	LSTM accuracy	LSTM loss
Multi-Vector (3 ACK)	357423	141178	91.132402	0.05078	91.132402	0.131247
Multi-Vector (without ACK)	284432	101885	96.410022	0.224182	96.410022	0.224182
Multi-Vector (1 ACK)	308776	115083	90.840001	0.730202	90.840001	0.73027
ACK	300273	110404	92.654087	0.7476	93.654087	0.747662
DNS	280511	200763	97.377178	0.005342	97.377178	0.005342
UDP	280001	200430	97.471033	0.01452	97.41033	0.014556
Mirai	276050	107307	98.887881	0.000708	98.460504	0.016664

Table 10: Brief analysis report

Botnet	192.168.249.39:80/bins
Peer-to-Peer	FALSE
Architecture	x86
Av Class	mirai|10,linux|9,server|4,backdoor|4
TCP Scanning	2323,23
TCP,CNC	192.168.249.39:1024

5.3 Prediction Efficiency

The estimated results for determining prediction efficiency based on different techniques are shown in Tab. 11. It displays the results for comparative analysis between the proposed attack detection with the state-ofthe-art DL techniques in the current domain. It is to be mentioned that only the attack detection technique is altered for performance assessment while the remaining model is unaltered. Moreover, a variety of approaches such as Adaptive Boosting (AdaBOOST), Random Forest Regressor (RFR), Support Vector Regressor (SFR), Decision Tree Regressor (DTR), K-Nearest Neighbor Regressor (K-NNR), Artificial Neural Network (ANN), Gradient Boosted Decision Trees (GBDT) and Adaptive Neuro-Fuzzy Inference System (ANFIS) were applied. Statistical parameters of Accuracy, Specificity, Fmeasure ad Sensitivity are evaluated. Mathematically,

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN}$$

where TP, TN, FP, FN indicates True Positive, True Negative, False Positive and False Negative respectively. From the computed results, it can be seen that in the current scenario, the presented model can register an enhanced accuracy measure (98.4%) which is far superior to other DL techniques for detecting DDoS attacks. It shows the enhanced precision of the presented technique. Similarly, for the specificity measure, the presented model can acquire a better value of 98.7% in comparison to state-of-the-art techniques. It shows that the presented methodology is better is predicting attacks correctly. As for the sensitivity analysis, the maximal measure of 99.0% is registered for the presented model depicting a significant improvement of the proposed model. F-Measure is the harmonic mean between specificity and sensitivity. In the current study, the presented model can register an enhanced measure of 99.0% depicting significant improvement in comparisons to the state-of-the-art DL techniques. Overall the enhanced statistical parameters for detecting attacks indicate significant improvement for the proposed model.

Table 11: Statistical results: Comparative analysis of DL-based attack detection

Techniques	Methods	Accuracy	Specificity	Sensitivity	F-measure
Base-line techniques	k-NNR	89.2	89.7	90.2	91.7
	ANN	91.3	92.5	94.3	95.6
	DTR	90.7	91.8	93.2	94.4
	SVR	89	89.6	90.7	93.8
	ADABOOST	93.8	95.8	97.9	97.3
Ensemble technique	RFR	93.8	94.3	96.2	96
	GBDT	97.2	97.2	98.3	96.3
	ANFIS	96.2	97.5	98.7	98.2
	SGDR	92	93.2	95.8	96.2
	Proposed	98.4	98.7	99	99

5.4 Reliability Analysis

Only the prediction model is changed to assess the proposed system's dependability, while the rest of the model remains unchanged. The suggested system has been subjected to several simulations, the results of which are shown in Fig. 6. The total system accuracy is determined by the dependability trends for the state-ofthe-art DL model for attack detection. The suggested methodology surpasses the existing techniques based on the variable number of data instances. It is noteworthy to depict that cumulative data instances are comprised for the computation of the reliability analysis. Moreover, the dataset is updated with 10% falsified data for accurate prediction of the attack. Furthermore, only KNN, SVR, and DTR models are compared for reliability analysis due to feasibility. As can be observed from the graphical results, the presented approach achieves average dependability of 92.36% in the current situation in comparison to 88.15% for SVR, 89.25% for K-NNR, and 87.02% for DTR. Furthermore, the constant high dependability value demonstrates the suggested technique's improvement in prediction attacks effectively. As a result, the presented model of DDoS attack detection is more trustworthy than the state-ofthe-art prediction technique in the current context [27,28].

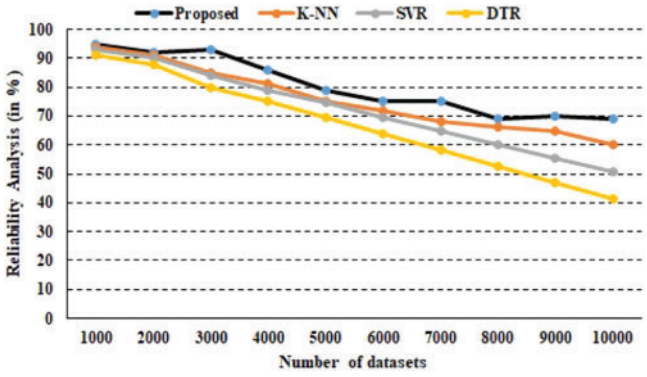

Figure 6: Reliability analysis

5.5 Stability Assessment

Stability is a metric used to gauge a system's overall performance. To put it another way, it is utilized to estimate overall stability across heterogeneous data instances. It is specifically quantified in terms of Average Absolute Shift (AAS) having a value between 0 and 1, with 0 denoting least stability and 1 denoting greatest stability. The findings obtained for the suggested model for stability estimate are shown in Fig. 7. The provided DDoS attack detection approach achieves a minimal measure of 0.59 and a maximal measure of 0.83 for AAS, with an average of almost 0.76. Henceforth, the suggested system is extremely reliable for processing huge datasets for detecting attacks in rea-time.

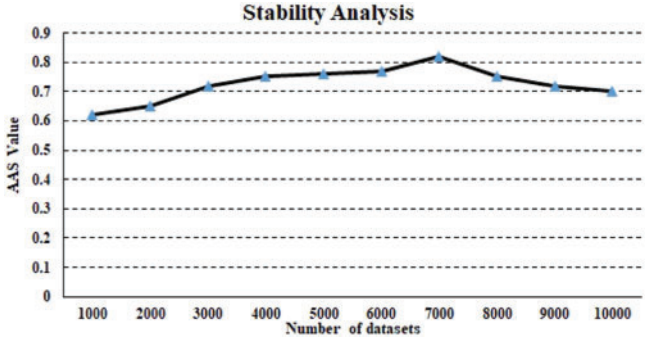

Figure 7: Stability analysis

6 Conclusion

The current research presents a DL-based technique for botnet detection utilizing a Bidirectional Long Short Term Memory Recurrent NN in combination with efficient word embedding. The performance of the presented technique was compared with state-of-the-art DL techniques to determine superiority in terms of statistical parameters for the accumulated data instances. The data instance comprises historical data instances. Based on 4 categories of attacks for mirai botnet malware, the current model acquired enhanced heterogeneous measures of the statistical parameters. Specifically, Accuracy (98.4%), Specificity (98.7%), Sensitivity (99.0%), F-measure(99.0%) and Data loss (92.36%) are acquired with minimal validation loss function value was registered. The current research demonstrated that increasing the data size might enhance statistical measures. The favorable

findings show that the presented unique DL model for botnet detection in the IoT is successful. Moreover, the performance can be further enhanced by focusing botnet identification at the packet level and employing text identification on elements that are ordinarily disregarded. Furthermore, while the bidirectional technique increases processing time and adds cost to each epoch, it looks to be a superior technique on an over-temporal basis. Several prospective research directions have been presented. First, a larger dataset can be created, which will include all 10 attack pathways employed by the Mirai botnet software. A modified version of the mirai code can be built for comparison with current signature and flow-based anomaly detection approaches to demonstrate the potential of the proposed model.

Funding Statement: The authors extend their appreciation to the Deputyship for Research and Innovation, Ministry of Education in Saudi Arabia for funding this research work through the project number (IF-PSAU-2021/01/17795).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] M. H. Kashani, M. Madanipour, M. Nikravan, P. Asghari and E. Mahdipour, "A systematic review of IoT in healthcare: Applications, techniques, and trends," *Journal of Network and Computer Applications*, vol. 192, no. 5, pp. 103164, 2021.

[2] M. A. Rahim, M. A. Rahman, M. M. Rahman, A. T. Asyhari, M. Z. A. Bhuiyan *et al.,* "Evolution of IoT-enabled connectivity and applications in automotive industry: A review," *Vehicular Communications*, vol. 27, no. 100285, pp. 1–15, 2021.

[3] J. P. Shim, R. Sharda, A. M. French, R. A. Syler and K. P. Patten, "The Internet of Things: Multi-faceted research perspectives," *Communications of the Association for Information Systems*, vol. 46, no. 1, pp. 511–536, 2020.

[4] F. Y. Yavuz, "Deep learning in cyber security for Internet of Things," Master's thesis, Fen Bilimleri Enstitüsü, 2018.

[5] J. Canedo and A. Skjellum, "Using machine learning to secure IoT systems," in *14th Annual Conf. on Privacy, Security and Trust (PST)*, Auckland, New Zealand, IEEE, pp. 219–222, 2016.

[6] M. A. Aref, S. K. Jayaweera and S. Machuzak, "Multi-agent reinforcement learning based cognitive anti-jamming," in *IEEE Wireless Communications and Networking Conf. (WCNC)*, San Francisco, CA, USA, pp. 1–6, 2017.

[7] A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for Internet of Things," *Future Generation Computer Systems*, vol. 82, no. 6, pp. 761–768, 2018.

[8] A. Saied, R. E. Overill and T. Radzik, "Detection of known and unknown DDoS attacks using artificial neural networks," *Neurocomputing*, vol. 172, no. 7, pp. 385–393, 2016.

[9] Y. Chen, Y. Zhang and S. Maharjan, "Deep learning for secure mobile edge computing," 2017. [Online]. Available: https://arxiv.org/abs/1709.08025.

[10] M. K. Alzaylaee, S. Y. Yerima and S. Sezer, "DL-Droid: Deep learning based android malware detection using real devices," *Computers & Security*, vol. 89, no. 5, pp. 101663, 2020.

[11] M. Aazam, S. Zeadally and K. A. Harras, "Deploying fog computing in industrial Internet of Things and industry 4. 0," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 10, pp. 4674–4682, 2018.

[12] P. Agrawal and B. Trivedi, "AndroHealthCheck: A malware detection system for android using machine learning," in *Computer Networks, Big Data and IoT Lecture Notes on Data Engineering and Communications Technologies*, pp. 35–41, 2021. https://doi.org.10.1007/978-981-16-0965-7_4.

[13] A. M. Zarca, M. Bagaa, J. B. Bernabe, T. Taleb and A. F. Skarmeta, "Semantic-aware security orchestration in SDN/NFV-enabled IoT systems," *Sensors*, vol. 20, no. 13, pp. 1–24, 2020.

[14] D. Li, L. Deng, Z. Cai and A. Souri, "Blockchain as a service models in the internet of things management: Systematic review," *Transactions on Emerging Telecommunications Technologies*, pp. 4139, 2020. https://doi.org/10.1002/ett.4139.
[15] T. Mahjabin, Y. Xiao, G. Sun and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," *International Journal of Distributed Sensor Networks*, vol. 13, no. 12, pp. 1550147717741463, 2017.
[16] M. Elsisi, M. Q. Tran, K. Mahmoud, D. E. A. Mansour, M. Lehtonen *et al.,* "Towards secured online monitoring for digitalized GIS against cyber-attacks based on IoT and machine learning," *IEEE Access*, vol. 9, pp. 78415–78427, 2021.
[17] M. Snehi and A. Bhandari, "Vulnerability retrospection of security solutions for software-defined cyber-physical system against DDoS and IoT-DDoS attacks," *Computer Science Review*, vol. 40, no. 12, pp. 100371, 2021.
[18] C. Xenofontos, I. Zografopoulos, C. Konstantinou, A. Jolfaei, M. K. Khan *et al.,* "Consumer, commercial, and industrial IoT (In)security: Attack taxonomy and case studies," *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 199–221, 2022.
[19] X. Zhang, X. Sun, X. Sun, W. Sun and S. K. Jha, "Robust reversible audio watermarking scheme for telemedicine and privacy protection," *Computers, Materials and Continua*, vol. 71, no. 2, pp. 3035–3050, 2022.
[20] X. Zhang, W. Zhang, W. Sun, X. Sun and S. K. Jha, "A robust 3-D medical watermarking based on wavelet transform for data protection," *Computer Systems Science and Engineering*, vol. 41, no. 3, pp. 1043–1056, 2022.
[21] Y. Chen, W. Zheng, W. Li and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition. Letters*, vol. 144, pp. 1–5, 2021.
[22] M. Saharkhizan, A. Azmoodeh, H. HaddadPajouh, A. Dehghantanha, R. M. Parizi *et al.,* "A Hybrid deep generative local metric learning method for intrusion detection, handbook. big data privacy, pp. 343–357, 2020. [Online]. Available: https://link.springer.com/book/10.1007/978-3-030-38557-6.
[23] A. Ray, S. Rajeswar and S. Chaudhury, "Text recognition using deep BLSTM networks," in *ICAPR 8th Int. Conf. Advances in Pattern Recognition*, Kolkatat, India, pp. 1–6, 2015.
[24] H. A. M. Puat and N. A. A. Rahman, "IoMT: A review of pacemaker vulnerabilities and security strategy," in *Journal of Physics: Conf. Series*, Mangalore, India, vol. 1712, 2020.
[25] C. D. Mcdermott, F. Majdani and A. V. Petrovski, "Botnet detection in the internet of things using deep learning approaches," in *Int. Joint Conf. on Neural Networks (IJCNN)*, Brazil, Rio de Janeiro, pp. 1–8, 2018.
[26] T. Trajanovski and N. Zhang, "An automated and comprehensive framework for IoT Botnet Detection and Analysis (IoT-BDA)," *IEEE Access*, vol. 9, pp. 124360–124383, 2021.
[27] B. K. Mishra, A. K. Keshri, D. K. Mallick and B. K. Mishra, "Mathematical model on distributed denial of service attack through Internet of Things in a network," *Nonlinear Engineering*, vol. 8, no. 1, pp. 486–495, 2019.
[28] C. Kolias, G. Kambourakis, A. Stavrou and J. Voas, "DDoS in the IoT: Mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80–84, 2017.