SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)

Article *in* Computer Communications · July 2020

DOI: 10.1016/j.comcom.2020.07.006

5 authors, including:

Shahzana Liaqat
COMSATS University Islamabad
1 PUBLICATION **5** CITATIONS
SEE PROFILE

Adnan Akhunzada
Technical University of Denmark
100 PUBLICATIONS **1,208** CITATIONS
SEE PROFILE

Computer Communications 160 (2020) 697–705

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

SDN orchestration to combat evolving cyber threats in Internet of Medical Things (IoMT)

Shahzana Liaqat [a], Adnan Akhunzada [b,1], Fatema Sabeen Shaikh [c], Athanasios Giannetsos [b], Mian Ahmad Jan [d,e,*]

[a] COMSATS University, Islamabad, Pakistan
[b] DTU Compute, Technical University of Denmark, Denmark
[c] Computer Information Systems Department, College of Computer Science and Information Technology, Imam Abdulrahman Bin Faisal University, Kingdom of Saudi Arabia
[d] Informetrics Research Group, Ton Duc Thang University, Ho Chi Minh City, Viet Nam
[e] Faculty of Information Technology, Ton Duc Thang University, Ho Chi Minh City, Viet Nam

ARTICLE INFO

Keywords:
Internet of Medical Things (IoMT)
Industrial Internet of Things (IIoT)
Deep Learning (DL)
Botnet detection
Hybrid deep learning models
Software Defined Networking (SDN)

ABSTRACT

Internet of Medical Things (IoMT) is now worth a billion dollar market. While offering enormous benefit, the prevalent and open environment of IoMT ecosystem can be a potential target of varied evolving cyber threats and attacks. Further, extensive connectivity of IoMT devices and their dynamic massive heterogeneous communication can create a new attack surface for sophisticated multivector malware attacks. There is a dire need to protect the forthcoming IoMT industrial revolution from varied evolving cyber threats and attacks. The authors propose a hybrid DL-driven SDN-enabled IoMT framework leveraging Convolutional Neural Network (CNN) and Cuda Deep Neural Network Long Short Term Memory (cuDNNLSTM) for a timely and efficient detection of sophisticated multivector malware botnets. For comprehensive evaluation, a state-of-the-art IoMT dataset and standard performance metrics have been employed. For verification purpose, we compare our proposed framework with our constructed hybrid DL-driven architectures and benchmark algorithms. Our proposed technique outperforms in terms of detection accuracy and testing efficiency. Finally, we also perform 10-fold cross validation to utterly show unbiased results.

1. Introduction

Internet of Things (IoT) is an enduring evolving technological paradigm connecting billions of smart objects [1,2] resulting in smart ecosystems such as smart factories, smart cities, smart health, smart home, smart vehicular networks, smart grids and Industrial Internet of Things (IIoT). Consequently, IoT is becoming and indispensable part of any emerging computing and networking paradigm. Currently, the latest revolution of Industrial IoT is growing tremendously resulting in huge monetary benefits and automation [3]. IIoT has the ability to enhance industrial safety, quality control, automation and production flow management.

On the contrary, open and prevalent environment of the Internet of Medical Things (IoMT) can be a potential primary target for various cyber threats and attacks [4]. Heterogeneous and dynamic nature of IoT devices magnifies the possibilities of cyber exploits exponentially that may leads to Denial of Service (DoS), Distributed Denial of Service (DDoS), advance persistent threats and attacks, data-injection attacks and sophisticated malware botnet attacks to entirely jeopardize the availability and confidentiality of available data, processes, or even throw the whole ecosystem into chaos [5–7]. Hence, IoMT besides leveraging huge benefits is vulnerable to varied evolving cyber threats such as key logging, phishing, identity theft and malicious bot proliferation [8]. Likewise, the digital landscape of the IoMT is also prone to complex hacking approaches, physical security threats, and set of varied devices to be simply compromised by botnets [1]. Consequently, attacks launched against IoMT can have devastating effects with severe damages compare to traditional industries and enterprises [4]. Further, attack detection is radically divergent from existing mechanisms due to IoT special service requirements (i.e., resource limitations, low latency, scalability, distribution and mobility) [9]. Hence, IoMT network desperately need an adaptive, flexible, dynamic cost-effective, well-timed detection mechanism against varied prevalent evolving cyber threats.

* Corresponding author: Mian Ahmad Jan.
E-mail addresses: adnak@dtu.dk (A. Akhunzada), mianjan@tdtu.edu.vn (M.A. Jan).
[1] Equal-first author: Adnan Akhundzada.

https://doi.org/10.1016/j.comcom.2020.07.006
Received 27 April 2020; Received in revised form 1 July 2020; Accepted 3 July 2020
Available online 11 July 2020

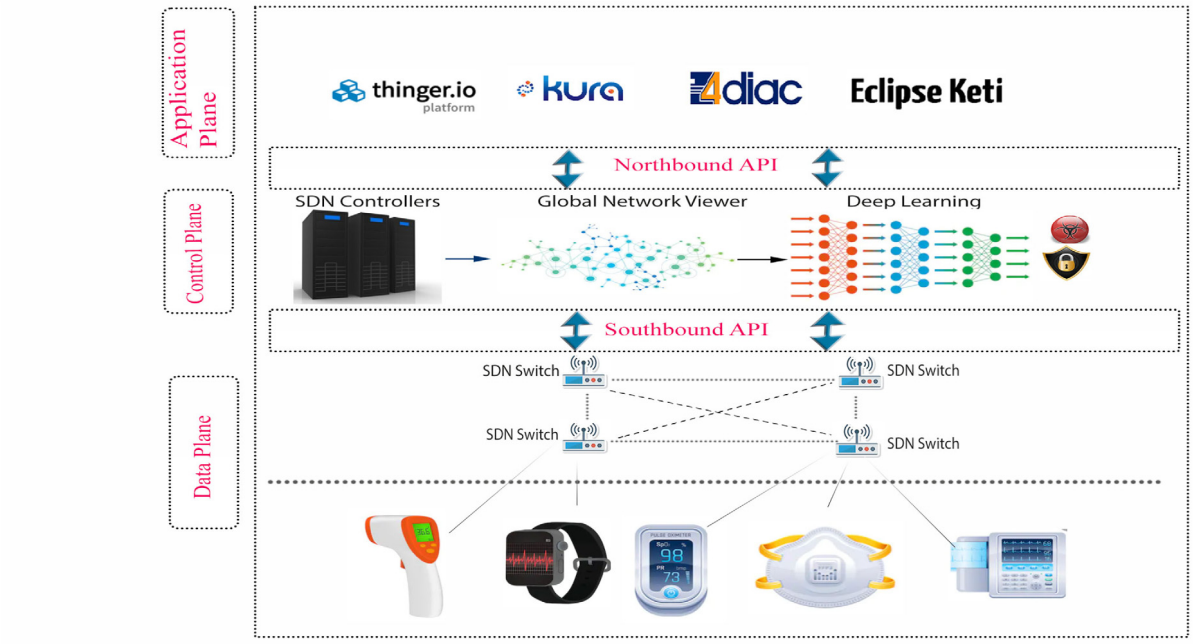


Fig. 1. DL-driven SDN-enabled IoMT detection framework.

1.1. Motivation

IoMT infrastructure comes at the cost of severe cyber threats and attack countermeasures. Multivector malicious bots are one of the most prevalent and sophisticated cyber threats in IoT environment. Botnet, is a captivating platform to potentially launch almost all kind of network attacks on IoMT [10], such as leveraging traditional malicious codes and syndicate attack methods that makes it a remarkable platform for the skilled adversaries. Consequently, botnets have the ability to launch various types of attacks, such as Distributed Denial of Service (DDoS), keylogging, identity theft, phishing, reconnaissance and bot proliferation [9,11,12]. Eventually, there is a dire need to enhance protection mechanisms against various evolving malware bot threats and attacks.

To the best of our knowledge, this is the first effort that comprehensively tackle sophisticated IoMT multi-vector evolving cyber threats using a hybrid DL-driven SDN-enabled framework. Moreover, the authors propose a highly scalable, adaptive, cost effective, well-timed detection framework leveraging the underlying IIoT resources without exhaustion is a novel breakthrough.

1.2. Contributions

The main contributions of the paper are manifold.

(1) The authors propose a highly efficient and scalable hybrid DL-driven (i.e., cuDNNLSTM-CNN) SDN-enabled framework to detect sophisticated and malicious multivector evolving IoMT botnets. Further, the proposed SDN-enabled mechanism is designed that do not place extra burden on the underlying IoMT resources.
(2) A current state-of-the-art publicly available IoMT dataset (i.e., Bot-IoT dataset) is employed for a comprehensive evaluation of the proposed mechanism.
(3) Standard performance metrics have been utilized to thoroughly evaluate our proposed mechanism (i.e., accuracy, precision, recall, F1-score, ROC, FNR, FPR, FDR, MCC etc.).
(4) For verification purpose, we compare our proposed technique with our constructed hybrid DL driven architectures (i.e., hybrid DNN–GRU, LSTM–GRU). We also provide a comprehensive comparison with current benchmark algorithms.
(5) Our proposed mechanism outperforms both in terms of detection accuracy, and computational complexity.
(6) Finally, a 10-fold cross validation is also performed to explicitly show unbiased results.

1.3. Organization

The remainder of this paper is structured as follows. Background and related work is provided in Section 2. Section 3 describes a detailed overview of our proposed scheme (i.e., network model, hybrid DL-Driven architecture, time complexity of the proposed model description and pre-processing of the dataset). Section 4 elaborates the experimental setup and performance evaluation metrics. Section 5 presents experimental results and discussions. Section 6 concludes the paper with future remarks.

2. Background and related work

Software defined networking (SDN) is considered a promising next generation networking paradigm. SDN basically comprises of three planes (i.e., application plane, control plane and data plane and their corresponding APIs (i.e., southbound and northbound). A comprehensive and detailed architecture of SDN is explored in our published work [13–17]. The power of SDN lies in its centralized control intelligence that is the control plane. The controller is central decision maker and has the ability to view abstractly and govern the whole underlying topological view (i.e., central and global network view) [18]. Moreover, the controller is programmable and can customize various functionalities [17]. Precisely, the control plane has the capability and potential to extend many underlying networks at the data plane such as software defined vehicular networks, software defined edge computing architectures, software defined fog computing architectures, software defined IoT architectures and so on. The literature is evident of varied SDN-enabled computing architectures [9,19–21]. There are few computing architectures that shows an infrastructure plane extended from data plane to show more clarity. However, together explicitly; it is known as data plane. Consequently, SDN-enabled frameworks can enhance the potentials of underlying highly dynamic and heterogeneous environment of Industrial IoT as shown in Fig. 1. that also depicts

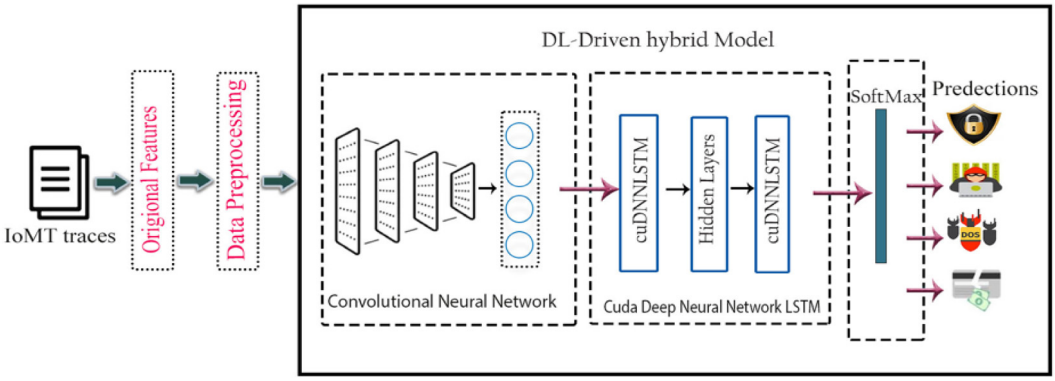


Fig. 2. DL-driven hybrid detection model.

Table 1
Features table.

Sr. no	Features names	Sr. no	Features names	Sr. no	Features	Sr. no	Features
1	pkSeqID	12	State	23	Dpkts	34	TnP-Per-Dport
2	Stime	13	State-number	24	Sbytes	35	AR-P-Proto-P-SrcIP
3	Flgs	14	Ltime	25	Dbytes	36	AR-P-Proto-P-DstIP
4	Proto	15	Seq	26	Rate	37	N-IN-Conn-P-SrcIP
5	proto-number	16	Dur	27	Srate	38	N-IN-Conn-P-DstIP
6	Saddr	17	Mean	28	Drate	39	AR-P-Proto-P-Sport
7	Sport	18	Stddev	29	TnBPSrcIP	40	AR-P-Proto-P-Dport
8	Daddr	19	Sum	30	TnBPDstIP	41	Pkts-P-State-P-Protocol-P-DestIP
9	Dport	20	Min	31	TnP-PSrcIP	42	Pkts-P-State-P-Protocol-P-SrcIP
10	Pkts	21	Max	32	TnP-PDstIP		
11	Bytes	22	Spkts	33	TnP-PerProto		

our proposed hybrid DL-driven framework. A major breakthrough of SDN-enabled hybrid DL-driven IoMT frameworks is that it leverages the underlying resource constrained nature IoT devices without exhaustion. Our proposed detection mechanism is highly scalable that can simply be customized and extended to any commercial controller such as Floodlight, POX, and Open daylight etc. Authors in [22] propose a hybrid intrusion detection system based on Spark Machine Learning and Convolutional-LSTM. The proposed technique achieves 97.29% detection accuracy with 0.71% false alarm rate utilizing the ISCX-UNB dataset for evaluation. The study in [23] presents a hybrid IDS for improved IoT security that combines Genetic Algorithm and Deep Belief Network for the cyber threats detection. This approach achieves approximately 99% accuracy. The article [5] is based on using deep auto encoder and deep forward neural networks for the identification of malicious activities in industrial internet of things. This model scores a detection accuracy of 99%.

Bhatt et al. [24] employed a hybrid detection module (HDM) comprising of One class Support Vector Machine (OCSVM), Self-Organizing Maps (SOM), Gaussian Mixture Model (GMM), and Isolation Forest with 98% detection accuracy. A synthetic MQTT Dataset [25] is used to detect multi-class attacks by applying a GRU-LSTM ensemble model that nearly achieves 99% per class accuracy. Deep learning based IDS in [26] employs Gated Recurrent Neural Networks (GRU) for identifying intrusion in IoT network and achieve an overall accuracy of 98.91% having False Alarm Rate of 0.76% that needs to be reduced for improved efficiency. In [27], the authors propose a bio-inspired SDN-based IDS for cross fire attacks. [28] presents a novel SDN originated IoT network security architecture, SeArch, leveraging deep learning models for intelligent threat detection in IoT network that scores 95.5% of average accuracy employing Stacked Autoencoder (SAE) and Self Organizing Map (SOM). The work done in [29], demonstrates an artificial neural network-based approach to train a network packet inspector for the identification of malicious packets from the IoT devices.

3. Methodology

The section presents the complete methodology of the proposed work.

3.1. Network model

The SDN control plane has the capability to extend IoMT at the data plane. Our proposed hybrid DL-Driven SDN-enabled IoMT detection framework is part of the control plane as shown in Fig. 1. The proposed detection module is essentially deployed on the control plane of the SDN. The reasons of placement of our proposed module on the control plane are manifold. Firstly, the control plane is programmable and represents the core centralized intelligence of the SDN. It maintains the global view of the entire underlying networks and all the forwarding decision are made here at the control plane. Secondly, the proposed framework at the control plane can simply leverage the underlying IIoT constrained devices without exhaustion that makes it a more suitable breakthrough for IoMT. Further, the proposed system is highly manageable and scalable being centralized, customized, and extended to any commercial SDN controller. Some architectures divide the data plane into two (i.e., data plane and infrastructure plane). However, infrastructure is explicitly part of the data plane. Therefore, we have data plane that implicitly carries infrastructure plane that may comprises of Varied IoMT devices, wireless technologies, sensors, and various mobile and smart devices etc. A complete explanation of the SDN architecture is provided in our published work [14–16].

3.2. Proposed Hybrid DL-driven architecture

The proposed Hybrid DL architecture comprises of CNN and cuDNNLSTM to detect sophisticated multi vector malware botnets in IIoT environment. A schematic architecture of our hybrid CNN-cuDNNLSTM model is shown in Fig. 2. Initially, we trained our Model on 2D-CNN to do feature extraction by applying two layers known as convolution and pooling layer that results on feature maps. This process helps the

model to learn the spatial features efficiently. On the contrary, 2D-CNN is unable to figure-out the critical inter-dependency of the features due to lack of temporal information. Therefore, we introduced the cuDNNLSTM layers after the CNN layers, to learn the spatial as well as temporal features in a more robust way. By employing this method, we managed to overcome the vanishing and exploding gradient dilemma efficiently that has mainly resulted in a higher detection accuracy with reduced False Positives. Finally, temporal feature vectors are passed as input to the Softmax classifier as a probabilistic function. The complete architecture of the proposed hybrid architecture is given in Table 1.

(1) *Convolutional Neural Network (CNN)* Convolutional Neural Network (CNN) is a multi-layered architecture comprises of convolutional layers, Pooling layers and fully-connected layers. It is a feed-forward neural network that works in two parts: (a) Feature extraction, and (b) predictions. CNN facilitates in achieving Feature Extraction in order to retrieve distinguishing features. Feature extraction can be expressed as Eq. (1) where $\bigotimes$ indicates the convolutional operation and W^{jk} represents weights (i.e., weight of the jth layer in the kth feature map) initialized randomly and then trained with CNN model. F_{j-1} is defined as the output of the $j-1$ layer and F_k is the output of the kth feature map in the convolutional layer. Subsequently, we make final predictions on basis of these extracted features. Hence, it is important to realize that feature extraction is crucial for classification problems. CNN facilitate in adjusting the weights and biases of neurons. In addition, the idea behind the convolutional layers is to extract the spatial patterns and reducing the noise of the original signal by applying the convolutional operations. A detailed explanation of the CNN is given in [30–32]. We have further employed cuDNN-enabled CNN that essentially improves the overall computational complexity of CNN.

$$F_k = \sigma(\sum W^{jk} \bigotimes F(j-1)) \tag{1}$$

(1) *Cuda Deep Neural Network Long Short Term Memory (cuDNNLSTM)*

The complexity of scaling up Recurrent Neural networks stems from the dependence of the state computation on time. In common architecture of RNNs, such as Long Short Term Memory (LSTM) and Gated Recurrent Units (GRU), the computation of each step is postponed until the previous step has been completed. Such sequential dependencies result in slower recurrent networks which can limits their applicability. cuDNNLSTM is a fast LSTM implementation backed by cuDNN [25]. cuDNN is a GPU accelerated library which enables fast and easy multi-threading for LSTM networks with high sequence modelling capacity. In addition, it also performs fast matrix multiplication to improve the overall performance. LSTM neural networks have the ability to overcome the long-term dependency problem as they can memorize information for a longer time period. LSTM layers are made up of recurrently connected memory blocks which helps the model to forget the previous states and replace it with new information [33]. Hence, system learns gradually to its maximum capability. cuDNNLSTM has the ability to overcome the BasicLSTMCell sequential dependency problem [34]. The pseudo code of the proposed hybrid architecture is given in Algorithm 1.

3.3. Time complexity of hybrid CNN-cuDNNLSTM algorithm

We have analysed the time complexity of our proposed DL-Driven hybrid detection model. Since it is a hybrid architecture, we compute the time complexity of the CNN-2D and cuDNNLSTM separately. Subsequently, we added both time complexities to show overall time complexity of the proposed DL-driven hybrid model. Time complexity can be calculated using the following equations:

$$CNN2D = O(\sum_{d}^{f=1} n_{l-1}.s_l^2.n_l.o_l^2) \tag{2}$$

where l is the index of convolutional layer while f denotes its depth.n_l is the number of filters in the lth convolutional layer. n_{l-1} presents the number of input channels of lth layer and s_l denotes the spatial size of the feature map, whereas; the basic architecture of LSTM is local in time and space [35] and therefore, the complexity of each weight is $O(1)$. Consequently, the complexity of CNN-cuDNNLSTM per time step can be computed using Eq. (3).

$$O(\sum_{d}^{f=1}(n_{l-1}.s_l^2.n_l.o_l^2) + w) \tag{3}$$

for calculating the complexity of training process of CNN-cuDNNLSTM detection model can be written as Eq. (4):

$$O((\sum_{d}^{f=1}(n_{l-1}.s_l^2.n_l.o_l^2) + w).i.e.,k) \tag{4}$$

where i denotes the input length, e is the number of epochs and k expresses the number of folds. Hence, our proposed algorithm has O complexity in the symbolic asymptotic notation.

Algorithm 1 Hybrid CNN-cuDNNLSTM detection model

Input:

nth iiot features and botnet ground truth labels:

$$X_n^{iot}, Y_n^{iot}$$

CNN2D.layers = C; *cuDNNLSTM.layers* = M; *k-Folds* = K; *epochs* = e;

Output:

obtain the Error E and predictions P
for all K :=1 to 10 **do**
  for epochs :=1 to e **do**
    if select.layer[C] = *Convolutional layer* **then**
      Randomly generate the weights w and bias b;
      Extract features;
      if select.layer[C] = *Max Pooling layer* **then**
        extract the feature maps according to the equation (1);
      else
        convert the 2-D Feature into the 1-D feature vector
      end if
      if select.layer[M] = *cuDNNLSTM* **then**
        Randomly generate the w and b of *cuDNNLSTM*;
        Compute the Hidden layers of *cuDNNLSTM*;
        Compute the output of Hybrid *CNN2D-cuDNNLSTM*;
      end if
      end for

3.4. Dataset description

Selecting an appropriate dataset contributes significantly in evaluating the performance of a detection system. Extant researchers have employed KDD99 [36], NSL-KDD [37–39], KDD CUP99[27,40], UNSW-NB15 [41], CICIDS 2017 [42] for intrusion detection in IoTs that lacks supportive IoT features and are mainly composed of missing realistic traffic, and IoT traces. That is why, we selected a current state-of-the-art publicly available Bot-IoT dataset [8]. The dataset comprises of realistic IoT network traffic, as the traffic is recorded from varied dedicated IoMT devices with real IoT traces. Bot-IoT dataset has 72 million records captured in an IoT simulated environment. We have used a down scaled version of this dataset that initially contains 668522 attack and 477 legitimate traffic instances, each instance or record with

Table 2
Description of system model.

Algorithm	Layers	Neurons/Kernal	AF/LF	Optimizer	Epochs	Batch-size
CNN-cuDNNLSTM	*Conv Layer*(2)	(60,20)	*RelU/CC-E*	*Adam*	5	32
	MaxPool layer	(1,1)	–			
	Dropout	(0.1)	–			
	Flatten		–			
	cuDNNLSTM(2)	(60,20)	–			
	Merge Layer		–			
	Output Layer	4	*softmax*			
DNN–GRU	*dense layer*(2)	(10,10)	*RelU/CC-E*	*Adam*	5	32
	Dropout(2)	(0.35,0.35)	–			
	GRU(3)	(10,15,5)	–			
	Dropout(3)	(0.35,0.35,0.35)	–			
	Merge Layer		–			
	Output layer	4	*softmax*			
LSTM–GRU	*dense layer* (2)	(15,10)	*RelU/CC-E*	*Adam*	5	32
	Dropout(2)	(0.35,0.35)				
	GRU(2)	(15,10)				
	Dropout(2)	(0.35,0.35)				
	Merge Layer		–			
	Output layer	4	*softmax*			

AF = Activation Function. LF = Loss Function. CC-E = categorical cross-enotropy.

42 features as shown in Table 1. We have up-sampled the legitimate traffic instances to 2400. The dataset is multi-nominal carrying multiple classes of attacks. However, we have labelled our dataset with three main attack classes (i.e., DDoS, theft and reconnaissance) (see Table 2).

3.5. Data pre-processing

Pre-processed data is achieved by applying the following steps.

(1) *Data Transformation*

Initially we have dropped the rows which contain missing values (i.e., nans, blanks etc.,) from the dataset as it can drastically impact the quality of data and the quality of the proposed evaluation model. It was observed that, each instance in Bot-IoT dataset contain 40 numeric and 7 non-numeric features. DL-enabled algorithms normally process data in the form of numeric matrix. Therefore, we have converted all of the non-numeric features such as flgs, proto, state and category to numeric values by using sklearn's label encoder function. In next step, we performed one hot encoding on the output label 'category' as numeric ordering of categories can degrade the performance of the model and may produce unexpected results. Finally, we have also converted two more features 'saddr' and 'daddr' IP values that contains both IPV4 and IPV6 addresses to numeric values. we have observed that 'sport' and 'dport' are in hexadecimal format, therefore; we need to convert these hexadecimal values to integers to feed the dataset accordingly.

(2) *Data Normalization*

To increase the effectiveness of IDS, there is a need to shift all values to a scaled version as it depurates the effect of gross influence. This process is referred as normalization [28]. MinMaxScaler function have been utilized to perform the normalization on the feature vectors of Bot-IoT dataset. The minmax normalization is based on the following equation:

$$\frac{X_i - \min(X)}{\max(X) - \min(X)} \tag{5}$$

3) *Up-sampling Dataset*

We have also addressed class imbalance problem in the dataset. We observed that there is only 477 IoT normal traffic, in contrast, to 668522 attack traffic is present in the selected file UNSW_2018_IoT Botnet_Full5pc_4. We up-sampled the Normal traffic to create a balanced dataset. After up-sampling, the total number of normal records is 2400. Now the total number of IoT traces are 670445 (i.e., attack and normal).

4. Experimental setup and performance evaluation metrics

4.1. Experimental setup

We have trained all the proposed hybrid DL models using Keras with python version 'Python 3.7.3'. In addition, we have configured our PC server with GPU based Tenserflow and Nvidia cuDNN library to enable parallel processing and fast matrix multiplication. All experimentations have been carried out on a single PC server equipped with Intel(R) Core (TM) i7-8750H CPU @ 2.21 GHz processor, 16 GB RAM and Nvidia GeForce GTX 1060 6GB graphics card.

4.2. Performance evaluation metrics

We have employed standard performance evaluation metrics (i.e., accuracy, precision, recall, ROC, F1-Score, testing and training time, True Negative Rate (TNR), False Negative Rate (FNR), False Discovery Rate (FDR), False Positive Rate (FPR), Matthews Correlation Coefficient (MCC) and False Omission Rate (FOR). True Positive (TP), True Negative (TN), False Positive (FP) and False Negative (FN) are being noticed from confusion matrix. Elements of the confusion matrix = FP, FN, TP, TN where FP (false positives) indicates the number of normal instances misclassified as anomalous; FN represent attacks which are incorrectly identified as normal; TN and TP represent correctly classified attacks and normal instances. Whereas, ROC curve plots the visualized performance for the comparison of true positive rate and false positive rate. Mathematical formulas for basic evaluation metrics are as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Precision = \frac{TP}{TP + FP} \tag{7}$$

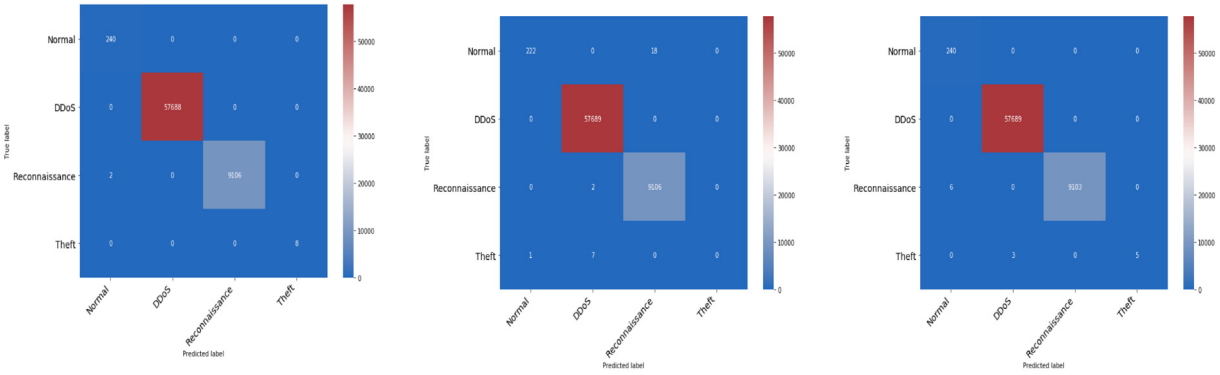
$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$F1 - score = \frac{2 * TP}{2 * TP + FP + FN} \tag{9}$$

$$FPR = \frac{FP}{FP + TN} \tag{10}$$

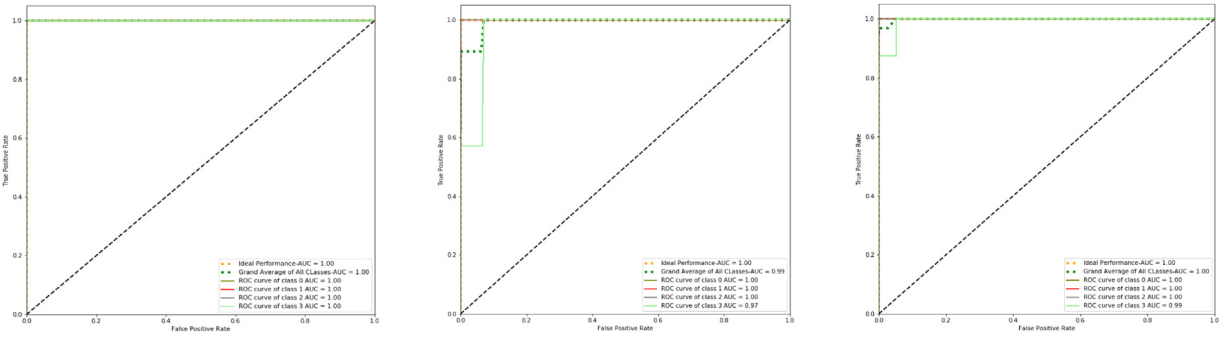
5. Results and discussions

To analyse the performance of our proposed Hybrid DL-Driven model (i.e., CNN-cuDNNLSTM), we thoroughly compared it with our constructed hybrid DL-driven models (i.e., DNN–GRU, LSTM–GRU). A correlation of the results obtained by applying the aforementioned standard and extended evaluation metrics are thoroughly analysed and detailed in this section. Additionally, a comprehensive comparison of the proposed model is also provided with current benchmarks in Table 4.



(a) Hybrid CNN-cuDNNLSTM Confusion Matrix. (b) Hybrid DNN-GRU Confusion Matrix. (c) Hybrid LSTM-GRU Confusion Matrix.

Fig. 3. Confusion Metrics.



(a) CNN-cuDNNLSTM ROC (b) DNN-GRU ROC (c) LSTM-GRU ROC

Fig. 4. ROC Curves.

5.1. Cross validation

To show explicitly unbiased results, we have employed 10-fold cross validation presented in Table 3. The results of each fold is displayed for more clarity. Further, the average results of the 10-fold for varied performance metrics are shown in their corresponding subsections.

5.2. Confusion matrix analysis

A thorough analysis of the confusion matrices evidently shows that our proposed technique outperforms from the rest of our constructed two hybrid DL-driven architectures as shown in Fig. 3. Our proposed technique identifies correctly the three different classes of attacks. On the contrary, LSTM–GRU performs comparatively better than DNN–GRU.

5.3. ROC analysis

An ROC curve plots the visualized performance for the comparison of true positive rate and false positive rate. The ROCs as shown in Fig. 4. presents precisely better performance of the proposed algorithm compared to the rest of the hybrid DL-driven architectures.

5.4. Accuracy, precision, recall and F1-score

For a detailed performance assessment, Fig. 5. presents the detection accuracy, precision, recall and F1-score. It depicts clearly that our proposed mechanism outperforms in terms of the crucial performance metrics compared to LSTM–GRU, and DNN–GRU. However, Fig. 6. clearly demonstrate the average per-class accuracy of the varied attacks that also depicts that CNN-cuDNNLSTM produced outclass results as compared to DNN–GRU and LSTM–GRU.

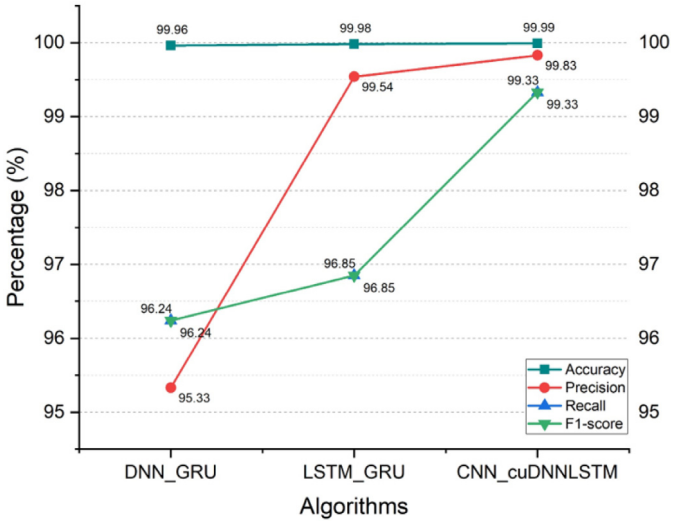


Fig. 5. Accuracy, precision, recall, F1-measure.

5.5. TPR, TNR, MCC analysis

We calculated the TPR, TNR, and MCC values from the confusion matrix for the detailed assessment and analysis. Fig. 7. exhibit clearly that our proposed mechanism shows better results in terms of TPR, and MCC. However, the value of TNR is almost same for LSTM–GRU and CNN-cuDNNLSTM compared to DNN–GRU.

Table 3
10-Folds results for hybrid DL algorithms.

Folds	Accuracy (%)			Precision (%)			Recall (%)			F1-Score (%)		
	Hyb-1	*Hyb-2*	*Hyb-3*	*Hyb-1*	*Hyb-2*	*Hyb-3*	*Hyb-1*	*Hyb-2*	*Hyb-3*	*Hyb-1*	*Hyb-2*	*Hyb-3*
1	99.99	99.99	99.99	99.73	98.12	100	100	100	100	100	100	100
2	100	99.99	99.18	100	99.21	98.63	100	100	99.09	100	100	99.09
3	99.98	99.98	100	100	99.74	100	99.11	100	100	100	97.01	100
4	99.96	99.99	99.18	99.75	100	99.08	99.54	100	100	98.32	100	100
5	100	98.36	99.46	100	99.08	97.08	100	99.08	100	100	99.08	100
6	98.51	99.97	99.15	98.96	99.32	98.54	99.21	100	99.09	100	100	99.09
7	99.95	99.96	99.19	99.47	96.21	97.55	100	99.09	99.09	99.45	100	99.09
8	100	99.18	99.25	100	99.12	100	100	100	100	100	99.09	100
9	99.97	99.95	99.17	100	98.65	100	100	100	99.09	100	100	99.09
10	99.96	99.96	99.36	96.34	100	100	99.11	99.54	100	100	100	100

HYB-1 = Hybrid CNN-cuDNNLSTM. HYB-2 = Hybrid DNN–GRU. HYB-3 = Hybrid GRU-LSTM.

Table 4
Comparison with benchmarks.

Parameters	H. Muna [43]	Y. Li [44]	R. Vinaya [45]	Our work
Dataset	UNSW-NB 15	NSL-KDD	DMD-2018	Bot-IoT dataset
Algorithm	DAE-DFFNN	Multi-CNN	DGA	CNN-cuDNNLSTM
Binary_class	✓	✓	✓	✓
Multi_class	✓	✓	–	✓
Cuda Enabled	–	–	–	✓
10-fold	–	–	–	✓
Accuracy by Class	✓	✓	–	✓
Average Accuracy	99	86.95	99.2	99.99
Precision	–	89.56	85.0	99.83
Recall	–	87.25	99.2	99.33
F1-score	–	88.41	91.5	99.33
Testing Time	55000(ms)	–	–	3000(ms)
FPR	8.2	13.45	–	5.99
Evaluation Metrics(*others*)	–	–	–	✓

Others = *TNR, FNR, FDR, FOR, MCC.*

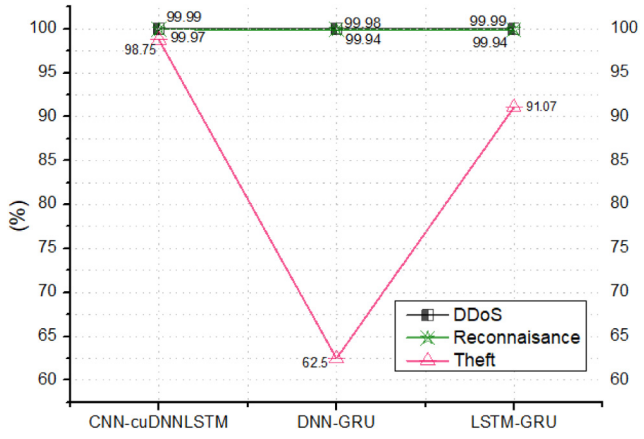


Fig. 6. Per-class Accuracy.

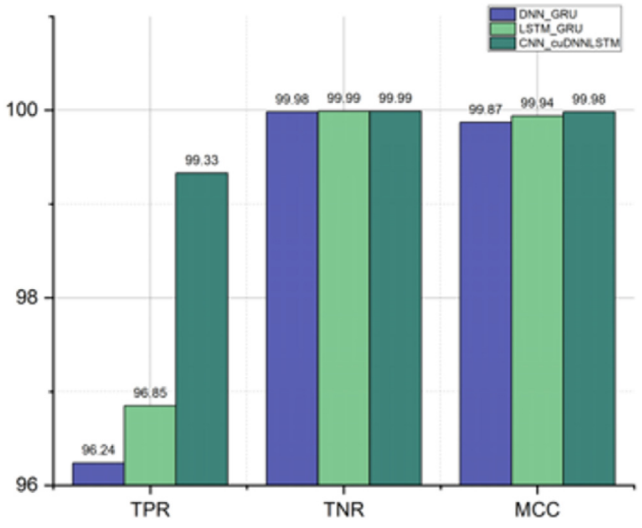


Fig. 7. TPR, TNR, MCC.

5.6. FPR, FNR, FDR and FOR analysis

For the rigorous assessment and analysis, we calculated the FPR, FNR, FDR and FOR, values from the confusion matrix. Fig. 8. demonstrates clearly that our proposed mechanism shows better results in terms of FPR, FNR, FDR and FOR. Conversely, LSTM–GRU performs comparatively well than the DNN–GRU.

5.7. Speed efficiency

We investigated the proposed hybrid deep learning model in the context of total elapsed time it requires. There are two major phases (i.e., testing and training). Since Training time is performed offline, we usually do not consider it. However, testing time represents the actual efficiency of the model (i.e., testing time represent the total time elapsed when the model is deployed in real time). For more clarity, we have calculated both average testing and training time. The results in Fig. 9. clearly shows that our proposed mechanism is quite efficient and computationally in-expensive both in testing and training time. On the other hand, DNN–GRU is time efficient compared to LSTM–GRU.

6. Conclusion

IoMT demands a reliable, dynamic, flexible, faster and secure network infrastructure for its exponential growth. In this paper, we introduced a novel Hybrid DL-driven SDN-enabled IoMT detection framework to combat sophisticated multivector botnet attacks (i.e., DDoS,

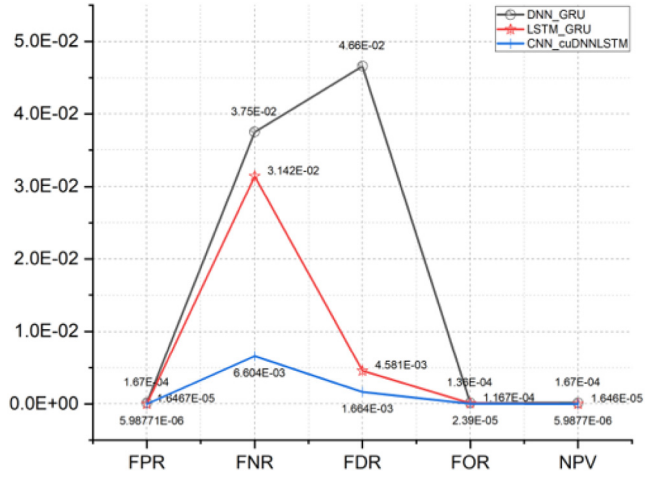


Fig. 8. FPR, FNR, FDR, FOR.

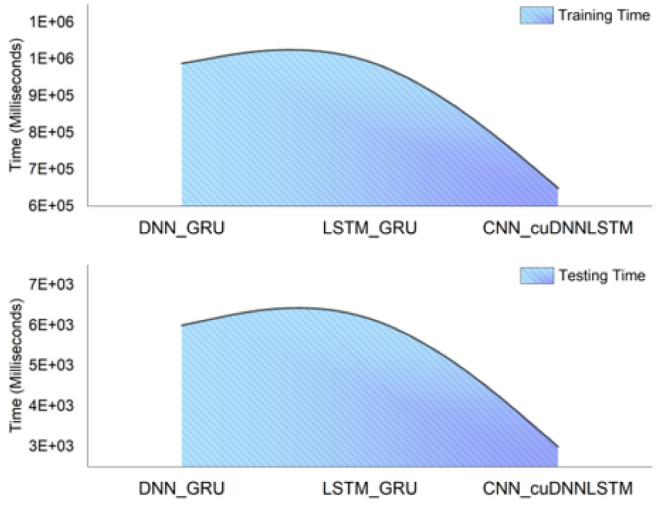


Fig. 9. Training and testing time.

theft and reconnaissance). The proposed mechanism is highly scalable, cost-effective and proficient. Besides, the proposed SDN-enabled IoMT framework leverages the underlying IoT resource constrained devices without exhaustion. Comprehensive evaluation and comparison with current benchmarks and our constructed GPU accelerated hybrid DL driven architectures (i.e., DNN–GRU and LSTM–GRU), the proposed mechanism outperforms in terms of detection accuracy and speed efficiency. Finally, we endorse varied hybrid DL-driven architectures for rigorous evaluation in the emerging computational paradigms and IoT ecosystems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work was supported by the European Commission, under the ASTRID and FutureTPM projects; Grant Agreements no. 786922 and 779391, respectively.

References

[1] G. Aceto, V. Persico, A. Pescapé, A survey on information and communication technologies for industry 4.0: state of the art, taxonomies, perspectives, and challenges, IEEE Commun. Surv. Tutor. (2019).
[2] B. Martinez, C. Cano, X. Vilajosana, A square peg in a round hole: the complex path for wireless in the manufacturing industry, IEEE Commun. Mag. 57 (4) (2019) 109–115.
[3] J. Li, Z. Zhao, R. Li, H. Zhang, AI-Based two-stage intrusion detection for software defined iot networks, IEEE Internet Things J. 6 (2) (2018) 2093–2102.
[4] I. Makhdoom, M. Abolhasan, J. Lipman, R.P. Liu, W. Ni, Anatomy of threats to the internet of things, IEEE Commun. Surv. Tutor. 21 (2) (2018) 1636–1675.
[5] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: challenges, opportunities, and directions, IEEE Trans. Ind. Inf. 14 (11) (2018) 4724–4734.
[6] S.S. Bhunia, M. Gurusamy, Dynamic attack detection and mitigation in IoT using SDN, in: 2017 27th International Telecommunication Networks and Applications Conference (ITNAC), IEEE, 2017, pp. 1–6.
[7] A. Ferdowsi, W. Saad, Deep learning for signal authentication and security in massive internet-of-things systems, IEEE Trans. Commun. 67 (2) (2018) 1371–1387.
[8] N. Koroniotis, N. Moustafa, E. Sitnikova, B. Turnbull, Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset, Future Gener. Comput. Syst. 100 (2019) 779–796.
[9] R. Chaudhary, G.S. Aujla, S. Garg, N. Kumar, J.J. Rodrigues, SDN-Enabled multi-attribute-based secure communication for smart grid in iiot environment, IEEE Trans. Ind. Inf. 14 (6) (2018) 2629–2640.
[10] C. Esposito, X. Su, S.A. Aljawarneh, C. Choi, Securing collaborative deep learning in industrial applications within adversarial scenarios, IEEE Trans. Ind. Inf. 14 (11) (2018) 4972–4981.
[11] M. Yin, X. Chen, Q. Wang, W. Wang, Y. Wang, Dynamics on hybrid complex network: Botnet modeling and analysis of medical iot, Secur. Commun. Netw. 2019 (2019).
[12] J. Slay, Towards developing network forensic mechanism for botnet activities in the iot based on machine learning techniques, in: Mobile Networks and Management: 9th International Conference, MONAMI 2017, Melbourne, Australia, December 13–15, 2017, Proceedings, vol. 235, Springer, 2018, p. 30.
[13] S. Haider, A. Akhunzada, I. Mustafa, T.B. Patel, A. Fernandez, K.-K.R. Choo, J. Iqbal, A deep CNN ensemble framework for efficient DDoS attack detection in software defined networks, IEEE Access 8 (2020) 53972–53983.
[14] A. Akhunzada, A. Gani, N.B. Anuar, A. Abdelaziz, M.K. Khan, A. Hayat, S.U. Khan, Secure and dependable software defined networks, J. Netw. Comput. Appl. 61 (2016) 199–221.
[15] A. Akhunzada, E. Ahmed, A. Gani, M.K. Khan, M. Imran, S. Guizani, Securing software defined networks: taxonomy, requirements, and open issues, IEEE Commun. Mag. 53 (4) (2015) 36–44.
[16] A. Akhunzada, M.K. Khan, Toward secure software defined vehicular networks: taxonomy, requirements, and open issues, IEEE Commun. Mag. 55 (7) (2017) 110–118.
[17] A. Abdelaziz, A.T. Fong, A. Gani, U. Garba, S. Khan, A. Akhunzada, H. Talebian, K.-K.R. Choo, Distributed controller clustering in software defined networks, PLoS One 12 (4) (2017).
[18] S. Al-Rubaye, E. Kadhum, Q. Ni, A. Anpalagan, Industrial internet of things driven by SDN platform for smart grid resiliency, IEEE Internet Things J. 6 (1) (2017) 267–277.
[19] M. Du, K. Wang, An SDN-enabled pseudo-honeypot strategy for distributed denial of service attacks in industrial internet of things, IEEE Trans. Ind. Inf. (2019).
[20] M. Ojo, D. Adami, S. Giordano, A SDN-iot architecture with NFV implementation, in: 2016 IEEE Globecom Workshops (GC Wkshps), IEEE, 2016, pp. 1–6.
[21] W. Ren, Y. Sun, H. Luo, M. Guizani, A novel control plane optimization strategy for important nodes in SDN-iot networks, IEEE Internet Things J. 6 (2) (2018) 3558–3571.
[22] M.A. Khan, M. Karim, Y. Kim, et al., A scalable and hybrid intrusion detection system based on the convolutional-LSTM network, Symmetry 11 (4) (2019) 583.
[23] Y. Zhang, P. Li, X. Wang, Intrusion detection for iot based on improved genetic algorithm and deep belief network, IEEE Access 7 (2019) 31711–31722.
[24] P. Bhatt, A. Morais, HADS: Hybrid anomaly detection system for iot environments, in: 2018 International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), IEEE, 2018, pp. 191–196.
[25] H. Alaiz-Moreton, J. Aveleira-Mata, J. Ondicol-Garcia, A.L. Muñoz-Castañeda, I. García, C. Benavides, Multiclass classification procedure for detecting attacks on MQTT-iot protocol, Complexity 2019 (2019).
[26] O. Ibitoye, O. Shafiq, A. Matrawy, Analyzing adversarial attacks against deep learning for intrusion detection in iot networks, 2019, arXiv preprint arXiv:1905.05137.
[27] A. Mansour, M. Azab, M.R. Rizk, M. Abdelazim, Biologically-inspired SDN-based intrusion detection and prevention mechanism for heterogeneous iot networks, in: 2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), IEEE, 2018, pp. 1120–1125.

[28] A.R. Narayanadoss, T. Truong-Huu, P.M. Mohan, M. Gurusamy, Crossfire attack detection using deep learning in software defined its networks, in: 2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring), IEEE, 2019, pp. 1–6.
[29] J. Yoon, Using a deep-learning approach for smart iot network packet analysis, in: 2019 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW), IEEE, 2019, pp. 291–299.
[30] X. Li, G. Zhang, Z. Wang, W. Zheng, HyConv: accelerating multi-phase CNN computation by fine-grained policy selection, IEEE Trans. Parallel Distrib. Syst. 30 (2) (2018) 388–399.
[31] M. Duan, K. Li, C. Yang, K. Li, A hybrid deep learning CNN–ELM for age and gender classification, Neurocomputing 275 (2018) 448–461.
[32] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, A.L. Yuille, Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs, IEEE Trans. Pattern Anal. Mach. Intell. 40 (4) (2017) 834–848.
[33] B. Fernandes, F. Silva, H. Alaiz-Moretón, P. Novais, C. Analide, J. Neves, Traffic flow forecasting on data-scarce environments using ARIMA and LSTM networks, in: World Conference on Information Systems and Technologies, Springer, 2019, pp. 273–282.
[34] M. Müller, Optimizing recurrent neural network language model GPU training.
[35] E. Tsironi, P. Barros, C. Weber, S. Wermter, An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition, Neurocomputing 268 (2017) 76–86.
[36] A. Dawoud, S. Shahristani, C. Raun, Deep learning and software-defined networks: towards secure iot architecture, Internet of Things 3 (2018) 82–89.
[37] A.A. Diro, N. Chilamkurti, Distributed attack detection scheme using deep learning approach for internet of things, Future Gener. Comput. Syst. 82 (2018) 761–768.
[38] B.A. Tama, K.-H. Rhee, An integration of pso-based feature selection and random forest for anomaly detection in iot network, in: MATEC Web of Conferences, vol. 159, EDP Sciences, 2018, p. 01053.
[39] A. Abeshu, N. Chilamkurti, Deep learning: the frontier for distributed attack detection in fog-to-things computing, IEEE Commun. Mag. 56 (2) (2018) 169–175.
[40] J. Kim, H. Kim, et al., An effective intrusion detection classifier using long short-term memory with gradient descent optimization, in: 2017 International Conference on Platform Technology and Service (PlatCon), IEEE, 2017, pp. 1–6.
[41] Y. Zhou, M. Han, L. Liu, J.S. He, Y. Wang, Deep learning approach for cyberattack detection, in: IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2018, pp. 262–267.
[42] K.F. Xylogiannopoulos, P. Karampelas, R. Alhajj, Detecting ddos attacks on multiple network hosts: Advanced pattern detection method for the identification of intelligent botnet attacks, in: Developments in Information Security and Cybernetic Wars, IGI Global, 2019, pp. 121–139.
[43] A.-H. Muna, N. Moustafa, E. Sitnikova, Identification of malicious activities in industrial internet of things based on deep learning models, J. Inf. Secur. Appl. 41 (2018) 1–11.
[44] Y. Li, Y. Xu, Z. Liu, H. Hou, Y. Zheng, Y. Xin, Y. Zhao, L. Cui, Robust detection for network intrusion of industrial IoT based on multi-CNN fusion, Measurement 154 (2020) 107450.
[45] R. Vinayakumar, M. Alazab, S. Srinivasan, Q.-V. Pham, S.K. Padannayil, K. Simran, A visualized botnet detection system based deep learning for the internet of things networks of smart cities, IEEE Trans. Ind. Appl. (2020).