

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/349341567>

A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT)

Article in Computer Communications · February 2021

DOI: 10.1016/j.comcom.2021.01.013

CITATIONS

0

2 authors:



Soneila Khan

COMSATS University Islamabad

1 PUBLICATION 0 CITATIONS

SEE PROFILE

READS

148



Adnan Akhunzada

Technical University of Denmark

100 PUBLICATIONS 1,208 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Smart Cities [View project](#)



Optimizing SIEM throughput [View project](#)



A hybrid DL-driven intelligent SDN-enabled malware detection framework for Internet of Medical Things (IoMT)

Soneila Khan ^{a,1}, Adnan Akhunzada ^{b,*}

^a COMSATS University, Islamabad, (CUI), Pakistan

^b DTU Compute, Cybersecurity Section, Technical University of Denmark (DTU), Denmark

ARTICLE INFO

Keywords:

Deep learning (DL)
Internet of Things (IoT)
Internet of Medical Things (IoMT)
Factory-of-things (FoT)
Hybrid deep learning architecture
Executable malware's
Software Defined Networks (SDN)

ABSTRACT

The rise of Internet of Things (IoT) has envisioned smart connectivity of billions of varied devices with high computational and processing capabilities resulting in the evolution of Internet of Medical Things (IoMT). This radical increase in the digital landscape of IoMT that processes considerably large amount of valuable data is primarily a potential attack target for varied cyber adversaries. One of the most prevalent, sophisticated and new evolving cyber threats for IoT ecosystems are multifaceted malicious malwares. The authors propose a highly scalable hybrid (deep learning) DL-driven SDN-enabled framework for efficient and timely detection of sophisticated IoMT malwares. Further, the proposed framework leverages the underlying IoMT resource constrained devices without exhaustion. We employed state-of-the-art publicly available dataset for a comprehensive evaluation of the proposed mechanism. Further, standard metrics have been employed to rigorously evaluate the performance of the proposed technique. For verification purpose, we compare our proposed mechanism with our constructed hybrid DL-driven architectures comprised of state-of-the-art DL-algorithms and current benchmarks. The proposed scheme outperforms in terms of high detection accuracy and speed efficiency. We also employed 10-fold cross validation to explicitly show unbiased results.

1. Introduction

Internet of Things (IoT) have aided in the integration of numerous devices (i.e. homogeneous and heterogeneous) for the provision of new services and automation of distinct procedures in various domains, starting from everyday activities to critical infrastructures. The IoT paradigm has brought an incredible increase in evolving new opportunities and technical challenges in the emerging industrial revolution and Internet of Medical Things (IoMT). Subsequently, IoMT is a breakthrough to create an environment for smart services and manufacturing by introducing IoT capabilities to medical applications and process management. Further, heterogenic IoT devices are deployed by IoMT networks to satisfy a wide range of user requirements [1]. On the contrary, the rapid increase in IoMT has raised various security concerns. Combining the Internet of Things (IoT) concept with industrial process management have reshaped the modern industry with great automation and tremendous revenue creation. Yet, it has become a potential target of varied cyber hackers and attacks due to the prevalent and open Internet of Medical Things (IoMT) [2,3]. Cyber-attacks on modern smart industries are of great concern as the losses caused by sophisticated cyber-attacks are extremely huge. Leveraging IoMT has revolutionized the modern industries, however; the consequences

of this emerging technology leads to various security issues for its absolute adaption [4]. Security becomes a more prime concern when people's lives are involved and more importantly when things are held responsible for controlling machinery and embedded systems in future industries [5]. Security is, therefore; an arresting challenge to be thoroughly addressed for the absolute adoption and exponential growth of IIoT [4,6,7].

Recently, IoT ecosystems and industries have been severely targeted with various sophisticated cyber malware automated attacks such as Stuxnet attack against the nuclear systems of Iran [8], German steel mill blast furnace attack [9], Saudi Aramco oil company attack [10], and Mirai [11]. The literature is evident of various sophisticated malware automated attacks with disastrous impact on underlying IoT ecosystems and Internet of Medical Things (IoMT).

1.1. Motivation

Malware detection is an overbearing concern as they are highly prevalent, sophisticated and have the capability to penetrate IoT ecosystems. Thus, malwares have the potential to simply take over the entire IoMT system. Malware threats are not new, yet hunting malware cyber

* Corresponding author.

E-mail addresses: soneila.niazi@hotmail.com (S. Khan), adnak@dtu.dk (A. Akhunzada).

¹ All the authors have equally contributed to the article.

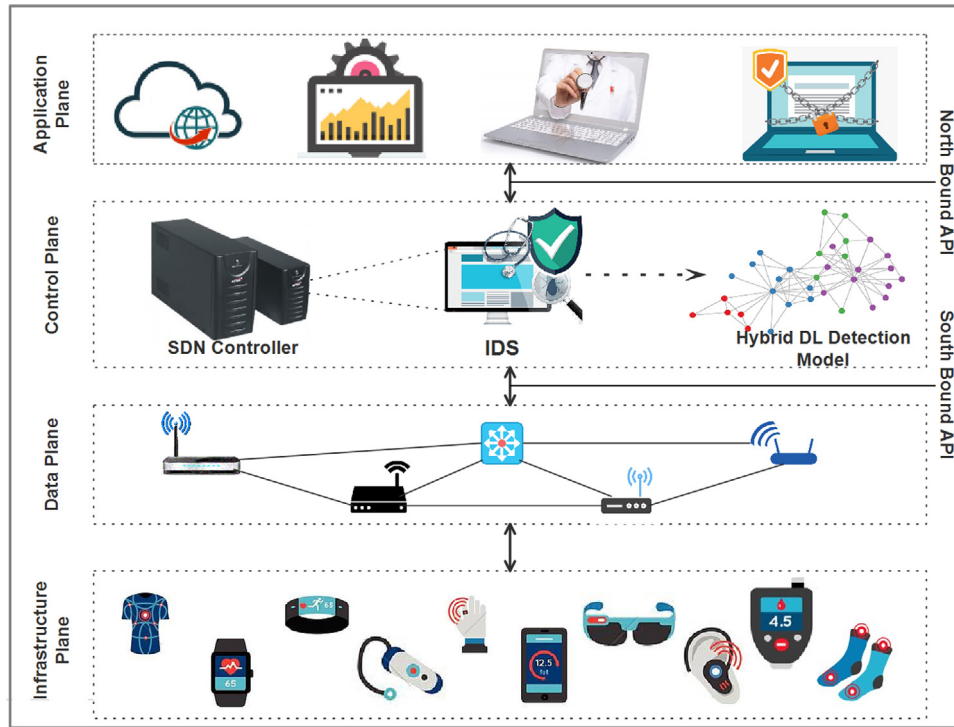


Fig. 1. SDN-enabled architecture for IoMT.

threats continues to be an arresting challenge [12]. As the IoT devices are becoming prevalent, the lack of security for such devices may result in the complete compromise of these devices against drastic and persistent malware attacks [13]. Kaspersky Lab reported that most IoT devices examined in 2016, were unsafe and potentially exploitable. Consequently, IoMT devices can easily be targeted with potential malwares like Hijime [11] and Mirai. Providing security to such complex networks and their innumerable access protocols is a real challenge. Limited storage and processing capabilities of the IoMT devices makes it extremely difficult to implement practical protection mechanisms. As shown in Fig. 1, we propose a highly scalable cost-effective, and efficient SDN-enabled IoMT malware detection framework that does not place any extra burden on the underlying IoT resource constrained devices.

1.2. Contributions

The main contributions of the paper are manifold. The authors propose a highly scalable SDN-enabled malware detection framework leveraging a hybrid DL-architecture (i.e. Hybrid CNN-LSTM) for Internet of Medical Things (IoMT). The proposed framework is a major breakthrough that simply does not place any extra burden on the underlying IoMT resource constrained network. The proposed mechanism has been comprehensively evaluated with current state-of-the-art publicly available IoT dataset (i.e., IoT Malware Dataset) explained later in Section 3. Besides, We employed standard and extended performance evaluation metrics (i.e., detection accuracy, precision, recall, F1-score, ROC, TPR, TNR, MCC etc.) to rigorously evaluate the proposed mechanism. For verification purpose, the proposed mechanism is compared with our constructed hybrid DL-driven state-of-the-art architectures (i.e., CNN-GRU and LSTM-GRU) and current benchmarks. Our proposed mechanism outperforms in terms of detection accuracy and computational complexity. Finally, to show that the results are completely unbiased, we also employed a 10-fold cross validation.

1.3. Organization

The remainder of this document is structured as follows; background and related work of the paper is presented in Section 2. Section 3 gives a comprehensive overview of the proposed Methodology that covers proposed DL architectures, dataset description, experimental setup and standard evaluation metrics. Section 4 comprises of experimental results and discussions. Section 5 concludes the paper.

2. Background and related work

The promising SDN paradigm is regarded as the next generation networking architecture. SDN comprises of three planes (i.e. application plane, control plane and data plane with their corresponding northbound and southbound APIs). A complete description of the SDN's entire architecture is detailed in our published articles [14,15]. The SDN control plane gives a complete global view of the entire underlying network topology and can handle heterogeneous networks at the data plane. The centralized control intelligence of the SDN and powerful abstraction of the controller makes it a great place to implement and customize various APIs with varied functionalities affecting the underlying networks at the control plane. The control plane is programmable to be extended to implement various APIs for different underlying computing architectures such as SDN enabled Fog-computing architecture, SDN enabled edge computing architecture and SDN enabled IoT architecture. The literature is evident of using SDN controller, being innovative and programmable can be extended to IoMT [3,16]. Likewise, SDN control plane can provide any solution for dealing with IoT node heterogeneity between SDN controller and linked IoMT nodes via Open-Flow switches. The integration of SDN and IoMT, therefore; offers a precise solution for inspecting network traffic to identify suspicious activities, sophisticated threats and attacks. To encounter evolving sophisticated IoMT malwares timely and efficiently, we propose a scalable, cost effective and efficient SDN-enabled detection framework for the underlying Internet of Medical Things (IoMT). Consequently, our proposed detection module can be

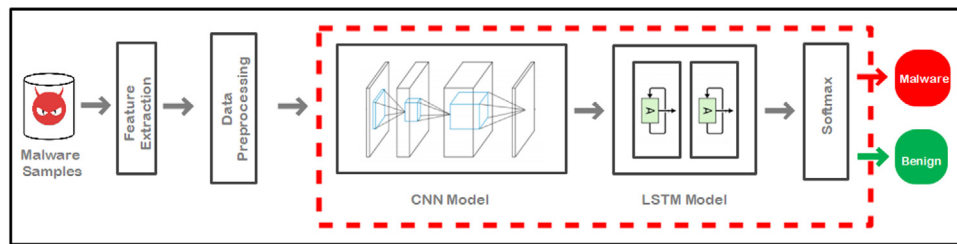


Fig. 2. Proposed hybrid deep learning model.

customized and extended to any SDN commercial controller such as Floodlight, POX, and Open daylight etc., to implement a centralized malware detection framework for IoMT. Our proposed SDN-enabled framework is depicted in Fig. 1. HaddadPajouh et al. [17] proposed a deep RNN based approach for encountering IoT malwares with 98.18% detection accuracy. Authors in [16] present a network IDS using deep auto encoders to detect IoT malware bots. The proposed work performs well in terms of TPR (True Positive Rate) (i.e. 100%) and FPR (False Positive Rate) (i.e. 0.007 to 0.01).

In [18] the authors employed a Convolutional-LSTM (Long Short Term Memory) based hybrid intrusion detection system on ISCX-UNB dataset. The proposed technique scores 97.29% accuracy with FAR (False Alarm Rate) 0.71%.

The article [3] creates an IoT and IoBT (Internet of Battlefield Things) malware dataset and apply Deep Eigen Space learning for an efficient detection of sophisticated malwares and achieves 99.68% detection accuracy. An ensemble-based learning technique is proposed in [19] for the detection of malware botnet attacks against DNS, HTTP and MQTT protocols in IoT. Standard IoT botnet datasets (i.e. UNSW-NB15 and NIMS) have been utilized for the evaluation of the proposed techniques with detection accuracy lying in between 95.25% and 99.86%.

Dovom et al. [20] utilizes opcodes of the program's sequences that were transmuted for the classification of IoT malwares. The authors have used fuzzy and fast fuzzy patterns that acquire detection accuracy of 98.83%. Further, the study [21] presents a deep learning model based on the visualization of gray scale images for the classification of varied malwares. The malware's content is represented as images and their proposed solution implements convolution neural network to predict malwares with 98.94% detection accuracy. In [22], a Bi-directional LSTM based solution has been proposed to efficiently detect botnet activities in the consumer IoT devices. The proposed algorithm achieves 99.99% accuracy.

A study in [23] presents a technique for the detection of malwares where the malicious codes are converted to gray scale images. CNN is used as a malware detection technique that achieves detection accuracy of 94.5%, along with 94.6% precision and 94.5% of recall rate. Further, the article [24] proposed a deep transfer learning-based approach for the classification of malwares achieving an accuracy rate of 99.25%, and FPR and TPR of 0.030 and 98.15% respectively. A graph based deep learning model is proposed in [25] for the classification of IoT malwares. The proposed scheme scores 97.13% accuracy, 11.26% of TPR and 1.55% FPR. The authors of [7] proposed a deep learning approach for the identification of malicious activities in IIoTs. The proposed scheme achieves a detection rate of 99% and FPR of 1.8% respectively. The idea of hybrid DL driven algorithms for evolving malware threats in IIoT is still in its infancy. Though, it has been exceptionally successful in the areas of image recognition, natural language translation, and classification etc. Likewise, DL-driven models have demonstrated exceptional outcomes in the presence of abundant training data.

3. Methodology

3.1. Network model

The control plane of SDN is capable of extending IoMT at the data plane earlier mentioned in detail. The proposed hybrid CNN-LSTM algorithm is an extended module of the controller (i.e., the proposed detection module is effectively deployed at the SDN control plane). Besides, the proposed module can be customized for any commercial SDN controller such as Floodlight, OpenDaylight, PoX etc. The control plane is entirely programmable and is capable to provide any solution for dealing with IoT node heterogeneity between SDN controller and linked IoMT nodes via Open-Flow switches. The integration of SDN and IoMT, therefore; offers a precise solution for inspecting network traffic to identify suspicious activities, sophisticated threats and attacks. As it maintains the global network view, all decisions for the underlying network and devices take place here. The proposed framework leverages the underlying IoMT constrained devices without exhaustion that makes it a more suitable revolution for IoMT. The data plane comprises of various IoMT devices, sensors with varied mobile and smart devices. Our previous published work completely explains the SDN architecture [14,15].

3.2. Proposed hybrid deep learning algorithm

We propose a hybrid deep learning model for the detection of malwares in Internet of Medical Things (IoMT). The proposed Hybrid DL architecture comprises of CNN (Convolution Neural Network) and LSTM (Long Short Term Memory) models to detect sophisticated malware attacks in IIoT environment. Our proposed hybrid model (i.e. CNN-LSTM) is represented in Fig. 2. Firstly, we train a CNN model for the significant extraction of features. Due to the lack of temporal data, the CNN model is unable to figure out the interdependence of the features. Therefore, we propose a Hybrid CNN-LSTM model to train the system in a more robust manner for the effective learning of the features. By adopting this strategy we were able to overcome the gradient ascend and descend problem, which resulted in giving a higher detection accuracy and lower false positive rates.

We propose a hybrid DL-driven framework (i.e. CNN-LSTM) for prevalent malware detection in Industrial Internet of things. Both the models are combined using an Add() function of the Merge layer of keras. The Add function serves as an input to the merge layer. It takes the outputs of CNN and LSTM model and returns a single output tensor to the next layer which then acts as a single input tensor to perform a binary classification for the prediction of malwares.

Our proposed hybrid model is depicted in Fig. 2. Each model consists of multiple layers. CNN comprises of two convolutional layers with a combination of 64 and 50 filters, each filter is of size 1. The next consecutive layer is a maxpooling layer with size 1. Flatten layer is added as the last layer of CNN model. Further, we add two LSTM layers with 70 and 50 neurons. Each followed by a dropout of 0.1 in order to alleviate the overfitting of the model. Relu is used as the activation function in all layers of the model except the output layer that employs Softmax. We use binary cross entropy as a loss function.

Table 1
Extracted features.

Sr.no	Feature name	Type	Sr.no	Feature name	Type	Sr.no	Feature name	Type
1	call	ins	9	str.GetThreadLocale	str	17	sym.imp.KERNEL32.dll HeapAlloc	imp
2	push	ins	10	str.Runtime ErrorProgram	str	18	sym.imp.str_str	imp
3	pop	ins	11	str.FlsSetValue	str	19	sym.imp.strlen	imp
4	leave	ins	12	str.FlushProcessWriteBuffers	str	20	sym.imp.memcpy	imp
5	shl	ins	13	str.ComparreStringEx	str	21	sym.imp.free	imp
6	cmp	ins	14	str.EnumSystemLocalesEx	str	22	sym.imp._fprintf_chk	imp
7	eax	ins	15	str.IsValidLocaleName	str	23	sym.imp.KERNEL32.dll_GMH	imp
8	jbe	ins	16	str.Kernel32.dll	str	24	sym.imp.KERNEL32.dll_SetUnhandledException	imp

ins = Instructions.

str = Strings.

imp = Imports.

GMH = GetModuleHandler.

A detailed designed description of both models is given in Table 2. These experiments are executed till 100 epochs with a batch size of 32. Multiple experiments have been carried out to achieve optimized and efficient results of the proposed DL-driven hybrid architecture.

(1) Long Short Term Memory (LSTM)

LSTM, an enhanced RNN is a renowned AI based neural network architecture proposed by Hochreiter and Schmidhuber in 1997. LSTM is mainly used for time series prediction and solving the problems of gradient descent in the back-propagation for prolonged dependencies. It is a composition of states consisting of three gates responsible for adding, deleting and preserving cells information based on the requirement [26]. In order to handle long term dependency, the LSTM employs gating mechanisms that optimizes the flow of information. Forget gate is the first sigmoid gate, it searches the gates that have not been used for a long time. This gate generates a value between 0 and 1, 0 relates to the less referred value, whereas; 1 corresponds to the important and the most referred value. If the value reaches 0 the cell gets deleted and is considered as an unused cell. An Input gate, the next layer of an LSTM determines which information should be stored in the cell, this gate is also responsible for information updating process [27]. The output gate works on filtered cells. It produces an output of 0 or 1, this value is passed to the tanh layer which yields results between 1 and -1. Following are the equations corresponding to the gates and cell states.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (1)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2)$$

$$\tilde{C}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

(2) Convolution Neural Network (CNN)

Convolution Neural Networks (CNN) have achieved outstanding results in the fields of computer vision, speech recognition and natural language processing and now it is progressing in security domain. CNN is capable of learning important features automatically compared to conventional feature selection algorithms [23]. It is a collection of interconnected processing components intended to transform the set of inputs to the set of required outputs. Input, output and hidden layers are main components of a convolution network that performs multiple operations on input data that includes convolution, pooling and flattening layers that consequently create fully connected neural network.

3.3. Comparison with state-of-the-art

To prove the effectiveness of our proposed hybrid DL driven architecture (i.e., CNN-LSTM), we compare it with our constructed hybrid DL driven mechanisms (i.e., CNN-GRU and LSTM-GRU). The complete architecture of our proposed techniques with other constructed hybrid

DL driven architectures are shown in Table 2. We also compare our proposed mechanism with current state-of-the-art DL algorithms as shown in Table 4. A comprehensive evaluation of our proposed hybrid DL algorithm with our constructed hybrid DL driven architectures, and current benchmarks shows promising results in terms of detection accuracy with a trivial trade-off in speed efficiency. The detailed results and analysis can be seen in Section 5.

Algorithm 1 Proposed Hybrid CNN-LSTM Detection Model

1: nth iomt features and malware labels:

$$X_n^{iomt}, Y_n^{iomt}$$

CNN layers = c; LSTM layers = l; k-Folds = k; epochs = e;

2: get the Error E and predictions P

3: for all k := 1 to 10 do

4: for epochs := 1 to e do

5: if select.layer[c] = Convolutional layer then

6: generate the weights w and bias b randomly;

7: Extract features;

8: end if

9: if select.layer[c] = Max Pooling layer then

10: extract the feature maps;

11: else

12: generate a feature vector

13: end if

14: if select.layer[l] = LSTM then

15: Randomly generate the w and b of LSTM;

16: Compute the Hidden layers of LSTM;

17: Compute the output of Hybrid CNN-LSTM;

18: end if

19: end for

20: end for

3.4. Dataset

We used current state-of-the-art IoT malware publicly available dataset for comprehensive evaluation. The dataset comprises of benign and malware IoT samples (i.e., 128 malware and 1089 benign files) represented in a raw format (i.e. benign samples are provided in the form of ELF (Executable Likable Format) files, whereas; malwares samples are given as sequences of opcodes). All files are extracted and then the samples of ELF are decompiled. Python script is used to disassemble the files in order to generate sequences of opcodes using Radare2 [28]. Radare2 is a robust reverse engineering and binary analysis framework. The disassembled files are now present in a raw format. To generate a feature vector, we use bag of words technique. The feature vector is finally used for the classification of malware and benign samples. Fig. 3 shows the detailed procedure of extracting features from samples of malware.

The Bag of Words technique is performed on critical opcode instructions that were selected from all generated opcode sequences. Finally,

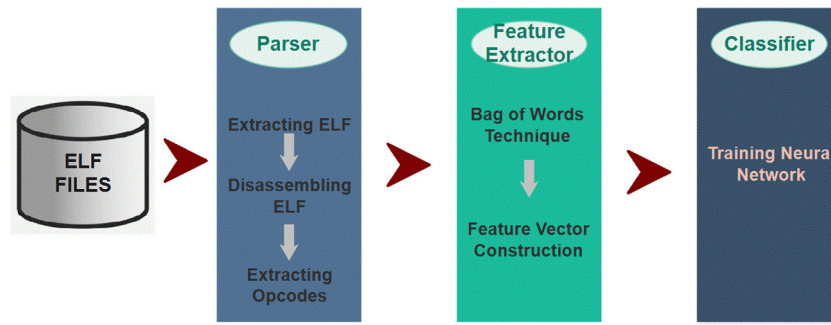


Fig. 3. Opcode extraction.

Table 2

Description of hybrid system model.

Algorithm	Layers	Neurons/Kernel	AF/ LF	Optimizer	Epochs	Batch-size
CNN-LSTM	Conv Layer(2)	(64,50)	RelU/BC-E	Adam	100	32
	MaxPool layer	(1,1)	–			
	Flatten	–	–			
	LSTM layer(2)	(70,50)	–			
	Dropout(2)	(0.1,0.1)	–			
	Merge Layer	–	–			
	Dense	70	–			
	Dropout	(0.1)	–			
CNN-GRU	Output Layer	2	softmax	Adam	100	32
	Conv Layer (2)	(15,10)	RelU/BC-E			
	MaxPool Layer(2)	(1,1)	–			
	Flatten	–	–			
	GRU layer(3)	(20,15,10)	–			
	Dropout(3)	(0.5,0.6,0.2)	–			
	Merge Layer	–	–			
	Dense Layer	20	–			
LSTM-GRU	Dropout	(0.3)	–	Adam	10	32
	Output Layer	2	softmax			
	GRU Layer (3)	(30,23,15)	RelU/BC-E			
	Dropout(3)	(0.3,0.2,0.3)	–			
	LSTM Layer(2)	(20,15)	–			
	Dropout(2)	(0.3,0.1)	–			
	Merge Layer	–	–			
	Dense layer	10	–			
	Dropout	0.3	–			
	Output Layer	2	softmax			

AF = Activation Function.

LF = Loss Function.

BC-E = Binary cross-entropy.

a word dictionary of size 284 is created from the selected opcodes. The resultant vectors are categorized into imports, instructions, libraries and string types. A detailed description of these features is provided in Table 1.

3.5. Experimental setup

We trained all the proposed hybrid DL models using Keras and ‘Python 3.7.3’. All experimentation has been carried out on Intel(R) Core (TM) i7-85550U CPU @ 1.80 GHz processor and 16 GB RAM, running Spyder (python).

3.6. Standard evaluation metrics

Standard evaluation performance metrics (i.e., confusion matrix, precision, recall, accuracy, F1-score and ROC curves etc.) have been employed to comprehensively evaluate our proposed technique. To calculate certain vital parameters, we need to first calculate TP, TN, FN, and FP explained below for subsequent calculation of complete evaluation metrics.

The number of malicious records labeled as malicious are considered as **True Positive (TP)**.

True Negative (TN) is the number of benign records predicted as benign.

False Positive (FP) the normal or benign samples wrongly classified as malicious ones or attack fall in this category.

False Negative (FN) are the malicious samples that are misclassified as normal.

Accuracy determines the number of samples correctly identified by a classifier, divided by the number of all applications for malware and benign.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

Precision is the ratio of malware predicted to be properly labeled as malware. It is defined as:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

Recall also referred as the detection rate is the number of malware samples from each class that are correctly predicted.

$$Recall = \frac{TP}{TP + FN} \quad (7)$$

F1-Score is defined as the harmonic mean of precision and recall, and is defined as follows:

$$F1 - score = \frac{2 * TP}{2 * TP + FP + FN} \quad (8)$$

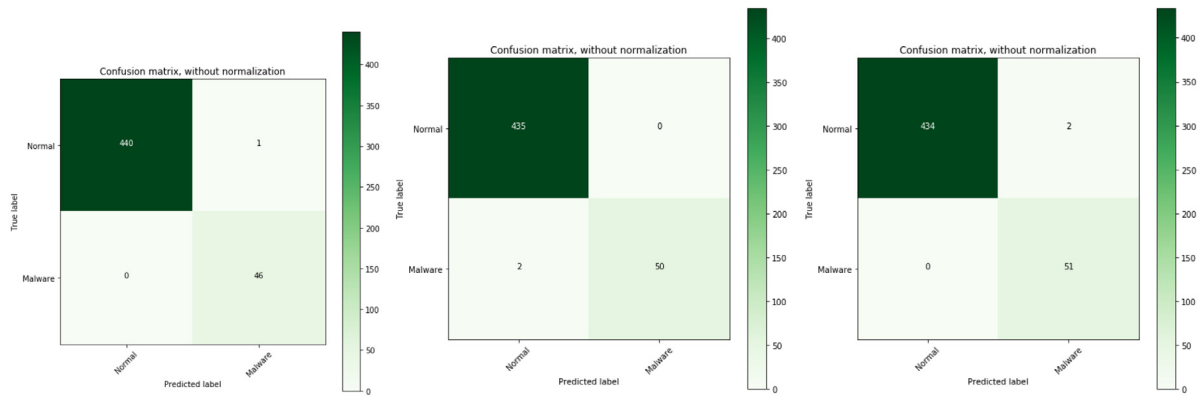


Fig. 4. Confusion matrices for hybrid CNN-LSTM, CNN-GRU and LSTM-GRU.

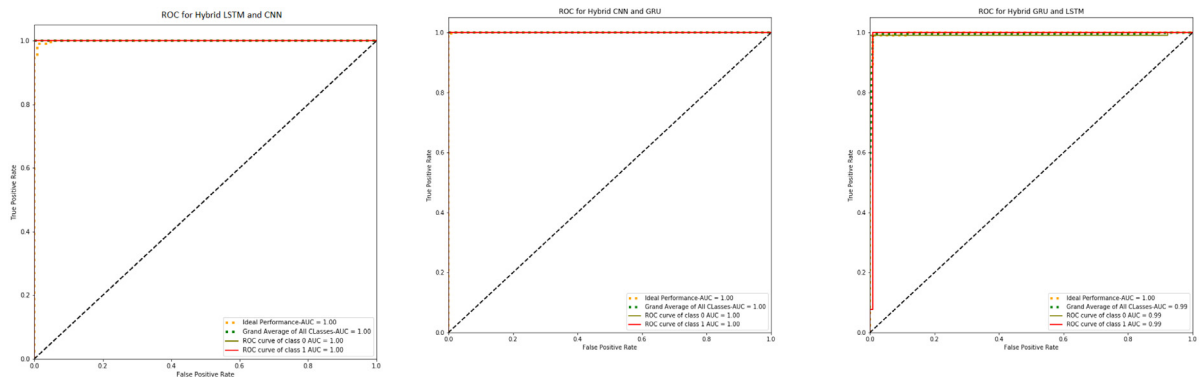


Fig. 5. AU-ROC curve for hybrid CNN-LSTM, CNN-GRU and LSTM-GRU.

Confusion Matrix To define the general output of a classification model, confusion matrix is used. Confusion Matrix can be presented in binary or multi-class representations. It is helpful for calculating precision, accuracy, recall and AUC-ROC curve values.

AUC-ROC

The degree of separability is expressed by AUC-ROC and primarily reflects the output of multi-class classification problems. To differentiate between multiple classes, this is one of the most critical evaluation metrics. The higher AUC is better for the model to predict 0s as 0s, 1s as 1s. The ROC curve diagram shows relationship between TP and FP rate.

4. Results and discussions

For a comprehensive evaluation of the proposed Hybrid DL architecture (i.e. Hybrid CNN-LSTM), we compared it with two benchmark hybrid DL-driven models (i.e. CNN-GRU, and GRU-LSTM). The proposed hybrid DL-driven models have been rigorously evaluated with standard metrics.

4.1. Cross validation

We have employed 10-fold cross validation to utterly show unbiased results. Table 3 presents a detailed description of each fold. Further, average results of 10-folds for the main evaluation metrics have been carried out and depicted accordingly in different sections.

4.2. Confusion matrices

Fig. 4 clearly depicts varied confusion matrices of corresponding hybrid DL-driven architectures. A comprehensive analysis of the confusion matrices demonstrates that our proposed hybrid architecture outperforms the rest of the two hybrid DL architectures.

4.3. Analyzing ROC

An ROC is regarded to be an important parameter for the graphical observation in the intrusion detection systems. ROC presents a relationship between the True Positive and False Positive rates. Fig. 5 represents the ROC curves for our implemented hybrid detection DL architectures which clearly depict the relation between TP and FP rates.

4.4. Accuracy, Precision, Recall and F1-Score

Our proposed (i.e. Hybrid CNN-LSTM) model in terms of detection accuracy, precision, recall and F1-Score shows an outclass performance compared to other two hybrid DL algorithms as shown in Fig. 6. Detailed results for Accuracy, Precision, Recall and F1-Score of each fold are represented in Table 3.

4.5. Analyzing TNR, TPR and MCC

We have represented the performance of TPR (True Positive Rate), TNR (True Negative Rate) and MCC (Matthews Correlation Coefficient) in graph 7. The graph positively shows that the proposed Hybrid DL-model (i.e. CNN-LSTM) performs better than the other two hybrid benchmark DL models (see Figs. 6 and 7).

4.6. FNR, FPR, FOR and FDR

Fig. 8 presents FPR (False Positive Rate), FNR (False Negative Rate), FDR (False Discovery Rate) and FOR (False Omission Rate) for a detailed comparison. It clearly depicts comparison of the three Hybrid architectures in terms of the aforementioned metrics. Our proposed model (i.e. Hybrid LSTM-CNN) outperforms from the rest of the two current hybrid DL benchmarks as shown in the graph.

Table 3
10-folds results for hybrid DL algorithms.

Folds	Accuracy			Precision			Recall			F1-Score		
	CNN-LSTM	CNN-GRU	GRU-LSTM	CNN-LSTM	CNN-GRU	GRU-LSTM	CNN-LSTM	CNN-GRU	GRU-LSTM	CNN-LSTM	CNN-GRU	GRU-LSTM
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

Table 4
Comparison with current benchmarks.

	Detection accuracy	Detection algorithm	Time complexity	Hybrid technique	Dataset
Proposed model	99.83%	Hybrid CNN-LSTM	1.2 s	Yes	Internet of Things
[29]	98.60%	Hybrid DAE-CNN_S	59.6 min	Yes	Malware dataset
[30]	92.10%	Hybrid DBN-DAE	1.243 s	Yes	KDDCUP 99
[3]	99.68%	CNN	NA	No	Internet of Things
[17]	99.18%	LSTM	NA	No	IoT application dataset
[7]	99%	DAE	0.06 s	No	NSLKDD and UNSW-NB15

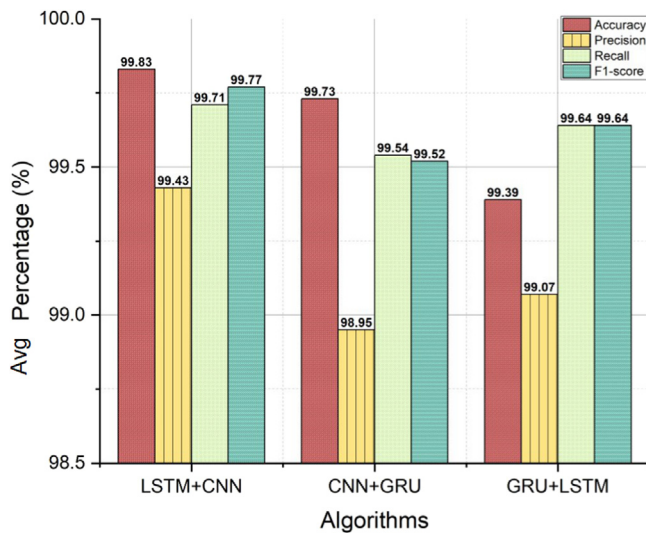


Fig. 6. Accuracy, Precision, Recall, F1-Measure.

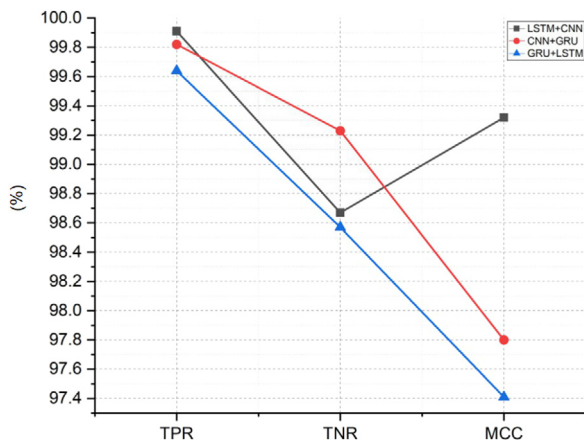


Fig. 7. Analyzing TPR, TNR, MCC for Hybrid CNN-LSTM, CNN-GRU and LSTM-GRU.

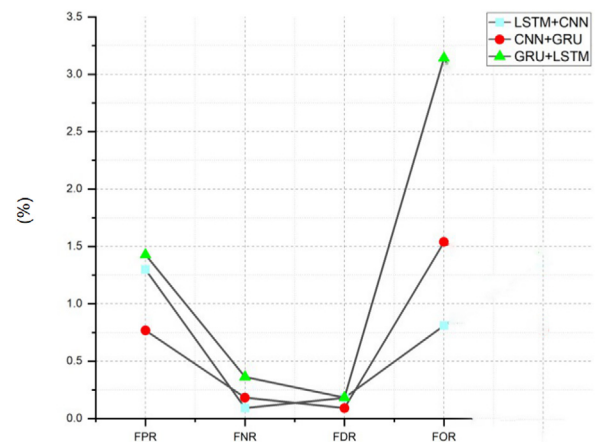


Fig. 8. Analyzing FPR, FNR, FDR, FOR for Hybrid CNN-LSTM, CNN-GRU and LSTM-GRU.

Moreover, our proposed model obtains an FNR, FPR, FOR and FDR of 0.091%, 1.3%, 0.81% and 0.18% respectively. Whereas, the hybrid model of CNN-LSTM achieves these results as 0.18%, 0.77%, 1.54% and 0.09%. Another hybrid model that is GRU-LSTM performs with the results 0.36%, 1.4%, 3.14% and 0.18% respectively.

4.7. Training and testing time

Fig. 9 shows the training and testing time of the three implemented hybrid models and draws a comparison in the time taken by each algorithm in the training and testing phase respectively. Our proposed hybrid model clearly outperforms in terms of testing time.

5. Conclusion

The open and prevalent environments of the Internet of Medical Things (IoMT) having high end digital assets and the involvement of human lives makes it very challenging to protect it from the evolving cyber threats, attacks and vulnerabilities. Consequently, there is a dire need to protect the IoMT ecosystem from the evolving drastic and persistent cyber threats. We propose a highly scalable and efficient hybrid DL SDN-enabled IoMT malware detection framework that simply does

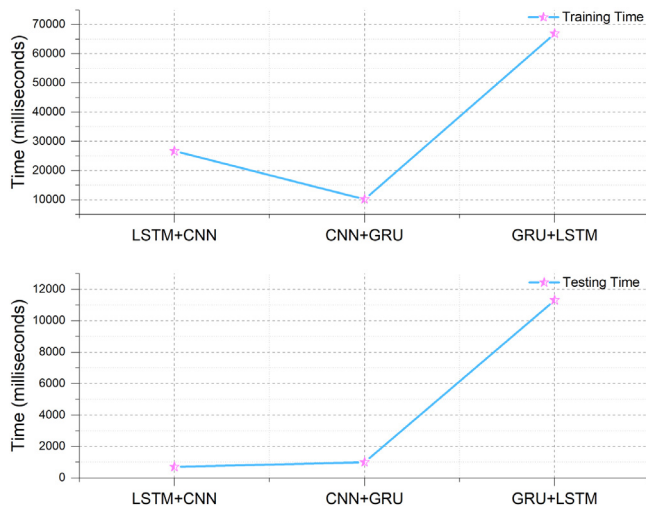


Fig. 9. Training and testing time.

not place any extra burden on the underlying IoT resource constrained agents. The proposed mechanism outperforms in terms of detecting sophisticated IoMT malware's for subsequent mitigation and prevention. Moreover, our proposed work also achieves low computational complexity. We endorse hybrid DL-architectures for the emerging 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.

Acknowledgments

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

References

- [1] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: Challenges, opportunities, and directions, *IEEE Trans. Ind. Inf.* 14 (11) (2018a) 4724–4734.
- [2] T. Ban, R. Isawa, S.-Y. Huang, K. Yoshioka, D. Inoue, A cross-platform study on emerging malicious programs targeting IoT devices, *IEICE Trans. Inf. Syst.* 102 (9) (2019) 1683–1685.
- [3] A. Azmoodeh, A. Dehghantanha, K.-K.R. Choo, Robust malware detection for internet of (battlefield) things devices using deep eigenspace learning, *IEEE Trans. Sustain. Comput.* 4 (1) (2018) 88–95.
- [4] E. Sisinni, A. Saifullah, S. Han, U. Jennehag, M. Gidlund, Industrial internet of things: Challenges, opportunities, and directions, *IEEE Trans. Ind. Inf.* 14 (11) (2018b) 4724–4734.
- [5] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, P. Faruki, Network intrusion detection for IoT security based on learning techniques, *IEEE Commun. Surv. Tutor.* (2019).
- [6] J. Wan, J. Li, M. Imran, D. Li, et al., A blockchain-based solution for enhancing security and privacy in smart factory, *IEEE Trans. Ind. Inf.* (2019).
- [7] 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.
- [8] J.-Y. Keller, D. Sauter, Monitoring of stealthy attack in networked control systems, in: 2013 Conference on Control and Fault-Tolerant Systems (SysTol), IEEE, 2013, pp. 462–467.
- [9] R.M. Lee, M.J. Assante, T. Conway, German steel mill cyber attack, *Ind. Control. Syst.* 30 (2014) 62.
- [10] J. Leyden, Hack on saudi aramco hit 30,000 workstations, oil firm admits-first hacktivist-style assault to use malware?(the register), 2012.
- [11] S. Edwards, I. Profetis, Hajime: Analysis of a decentralized internet worm for IoT devices, *Rapidity Netw.* 16 (2016).
- [12] 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.
- [13] 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.
- [14] A. Akhuzada, 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.
- [15] A. Akhuzada, 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.
- [16] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, Y. Elovici, N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders, *IEEE Pervasive Comput.* 17 (3) (2018) 12–22.
- [17] H. Haddadpajouh, A. Dehghantanha, R. Khayami, K.-K.R. Choo, A deep Recurrent Neural Network based approach for Internet of Things malware threat hunting, *Future Gener. Comput. Syst.* 85 (2018) 88–96.
- [18] 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.
- [19] N. Moustafa, B. Turnbull, K.-K.R. Choo, An ensemble intrusion detection technique based on proposed statistical flow features for protecting network traffic of internet of things, *IEEE Internet Things J.* (2018).
- [20] E.M. Dovom, A. Azmoodeh, A. Dehghantanha, D.E. Newton, R.M. Parizi, H. Karimipour, Fuzzy pattern tree for edge malware detection and categorization in IoT, *J. Syst. Archit.* 97 (2019) 1–7.
- [21] C.D. McDermott, F. Majdani, A.V. Petrovski, Botnet detection in the internet of things using deep learning approaches, in: 2018 International Joint Conference on Neural Networks (IJCNN), IEEE, 2018, pp. 1–8.
- [22] T. Kim, B. Kang, M. Rho, S. Sezer, E.G. Im, A multimodal deep learning method for android malware detection using various features, *IEEE Trans. Inf. Forensics Secur.* 14 (3) (2018) 773–788.
- [23] P. Liu, An intrusion detection system based on convolutional neural network, in: Proceedings of the 2019 11th International Conference on Computer and Automation Engineering, ACM, 2019, pp. 62–67.
- [24] L. Chen, Deep transfer learning for static malware classification, 2018, arXiv preprint arXiv:1812.07606.
- [25] A. Abusnaina, A. Khormali, H. Alasmay, J. Park, A. Anwar, U. Meteriz, A. Mohaisen, Breaking graph-based IoT malware detection systems using adversarial examples: poster, in: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks, ACM, 2019, pp. 290–291.
- [26] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, *IEEE Trans. Neural Netw. Learn. Syst.* 28 (10) (2016) 2222–2232.
- [27] T. Nathezhtha, V. Yaidehi, Cloud insider attack detection using machine learning, in: 2018 International Conference on Recent Trends in Advance Computing (ICRTAC), IEEE, 2018, pp. 60–65.
- [28] Radare2, 2006, <https://github.com/radareorg/radare2>.
- [29] D. Carlin, A. Cowan, P. O'kane, S. Sezer, The effects of traditional anti-virus labels on malware detection using dynamic runtime opcodes, *IEEE Access* 5 (2017) 17742–17752.
- [30] Y. Li, R. Ma, R. Jiao, A hybrid malicious code detection method based on deep learning, *Int. J. Secur. Appl.* 9 (5) (2015) 205–216.