Securing Internet of things (IoT) using SDN- enabled Deep learning Architecture

MSc Internship
Cybersecurity

Idehen Jimmy Irivbogbe
19144814

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: ……Irivbogbe Idehen Jimmy.

Student ID: ……19144814……

Programme; ……Cybersecurity…… **Year:** …2021……..

Module: ……MSc Internship……

Supervisor: ……Michael Pantridge……

Submission Due Date: ……16/08/2021……

Project Title: Securing the Internet of things (IoT) using SDN- enabled Deep learning Architecture……

Word Count:……7612……Page Count……21……..

I hereby certify that the information contained in this (my submission) is information pertaining to the research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the project's rear.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. Using other authors' written or electronic work is illegal (plagiarism) and may result in disciplinary action.
I agree to an electronic copy of my thesis being made publicly available on NORMA, the National College of Ireland's Institutional Repository, for a consultation.

Signature: ……

Date: 16/08/2021 ……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on the computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Securing the Internet of things (IoT) using SDN- enabled Deep learning Architecture

Idehen Irivbogbe
19144814

Abstract

In recent years, billions of devices have been connected through the Internet of things (IoT) and continuously shared data. The extreme connectivity of these devices makes the IoT vulnerable to different cyber-attacks, leading to financial and information loss. Due to such threats, the IoT demands a secure infrastructure and is in dire need of security. This work proposes a deep learning model for the detection of cyber threats in IoT. This work used the DNNLSTM algorithm for the detection of threats.

Further, publicly available CICIDS 2018 is used for the training of the proposed algorithm. The proposed model achieved an accuracy of 99.92%, with a recall of 99.50%. This work also compared the proposed model with two other algorithms (GRU and LSTMGRU), trained on the same dataset as well as with existing literature. The proposed model outclassed the other algorithms and existing literature in accuracy and different evaluation metrics.

1 Introduction

Internet of Things (IoT) has become unavoidable as it empowers communication and coordination between numerous devices. The IoT is defined as unique addresses that are assigned to a global network of interconnected devices. IoT devices use different communication protocols and sensing features. These devices have computational abilities to analyse data and provide services. Standard IoT devices include electric switches, doorbells, fire sensors, cameras, video recorders, and almost all real-time device sensors. IoT is a transformation of the world digital technology. It is an archetype that connects millions of digital intelligent devices, prompting the formation of an intellectual environment, such as intelligent health care systems, intelligent ecosystems, smart factories, smart cities, and intelligent vehicular networks [1]. The advanced Internet of Things (IoT) is increasing numerous security concerns. In recent years, we have witnessed fast data growth in IoTs, and due to this increase in data, a considerable number of attacks and threats are also focused on IoT networks [2-3]. IoT contains homogeneous and heterogeneous networks and devices for networking that uses different protocols. It means that vulnerabilities can produce an invisible danger to the IoT devices and the system at large. Cybersecurity exploits numerous concerns in the dynamic IoT devices features in the form of different attacks, i.e., (DoS) Denial of service attacks, (DDoS) Distributed Denial of service attacks, and numerous malware types [4–5]. The attacker always finds vulnerabilities to exploit the system; cybersecurity analysts continuously monitor networks for every vulnerability and threat identification. About 80% of cybersecurity experts spend handling at least one security issue; however, 60% of experts

deal with the network's security operations [5]. Deception attacks and replay attacks have also been described. Industrial level security controls and attack detection techniques are reviewed in [6]. Diverse sorts of security measures in various categories of protocol-following devices have that need to be implemented. In the seamless nature of IoT devices, these security measures are insufficient. However, conventional intrusion detection strategies are deployed to protect devices from attacks, working at the base framework, intrusion detection system (IDS), or firewalls. To secure the IoT infrastructure, there has not yet been an integrated approach invented. Due to the Internet development and the interconnectedness among networking devices, IoT security still remains a significant challenge and poses a severe need for security.

Software-defined network (SDN)-enabled architectures delivers the opportunity to simplify and configure network management. SDN provides an efficient and effective detection without exhaustion and provides a platform for underlying resource-constrained devices that do not overburden to implement a security solution. A network's security system comprises numerous attributes such as antivirus, IDS, and firewalls. IDS generates alerts and identifies unapproved system characteristics like replication, usage, destruction, and modification. Integrating an IDS in a software-defined network for SDN surveillance is one of the best approaches [7].

With the rapid evolution of Artificial Intelligence and features for programming and techniques of the Software-defined network, safety levels may be improved by joining SDNs into AI-based security solutions. Numerous AI-based methods, such as support vector machine (SVM), artificial neural networks (ANNs), Genetic Algorithms, K-Nearest Neighbor, Decision Trees, Naive Bayesian, and fuzzy logic, have been used as network traffic classifiers, with varying degrees of accuracy and ideal results. [8]. The need to present a robust and flexible architecture for threat detection in IoT devices has motivated us to develop an SDN-enabled, hybrid deep learning-based intrusion detection solution.

The author developed a Software-defined Network (SDN) that enabled deep learning-driven, highly scalable, and cost-effective threats detection (IoT). The DNNLSTM classifier was used for experimentation. The proposed algorithm can detect multi-class threats. Two more algorithms, i.e., GRU and LSTMGRU, are used for comparison. Both of these algorithms are trained and evaluated on the same dataset for performance evaluation. For verification, the author also made a comparison of the proposed model with the current work. The evaluation results show that the proposed scheme can detect multi-class threats with a good percentage of accuracy and other evaluation metrics.

2 Related Work

In the existing literature, researchers have proposed various threat detections schemes against attacks. Most existing literature is deep learning, which shows substantial output in numerous computer science fields. In [9], CNN based model to detect intrusion in the network. In real-time unprocessed data, traffic is transformed into an image data format. The number of computation factors was reduced, yielding efficiency increased by transforming

data into image format. In [10], the author proposed a systematic Intrusion detection system for expected SDNs consisting of a seven-layered CNN network for training the dataset. In [11], the (DARPA) Defence Advanced Research Projects Agency trained intrusion detection dataset by use of SVM. Without affecting the network's performance, it provides a high detection rate by combining the flow value and IDSs based packet. The author proposed to preserve the attack feature of input data by combining threat analysis techniques combining long short-term memory (LSTM) in [12]. LSTM introduce classifiers, which distinguish the usual traffic attacks. In [13], the author introduces efficient detection of multi-class threats in IoT enjoinment by DL-enabled malware detection using a hybrid technique by combining Long Short-Term Memory(LSTM) Deep Neural Network(DNN) and Deep Neural Network(DNN).In [14], the author identifies attacks and categorises them using recurrent neural network (RNN) techniques. An efficiency evaluation was prepared for the non-RNN- and RNN based approaches. As given in [15], the author presented an RNN-based anomaly detection system, aside from detecting anomalies by enabling and generating flows in a dynamic access control network. So using an advanced SVM algorithm, DDoS attacks were detected in the SDN environment. In [16], the system is validated using the Hierarchical Task Analysis technique, which validates human errors to attain definite results. The latest machine learning techniques yield a refinement and enhance the efficiency compared to the traditional machine learning techniques, e.g., hybrid A.I. and deep learning schemes. Integrating SDN architecture with techniques based on A.I. has been established to be extremely worthwhile [17]. In [18], the author presented the security model Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System (RTS-DELM-CSIDS). The presented model initially defines security aspects' appropriation by ordering them based on most relevancy and construct a robust intrusion detection system. The presented technique had a precision of 92.73% in validation and 96.22% in training. In [19], the author discussed various detection techniques for cyber-attacks. For denial of service attacks (DDOS) detection, the author recommended an embedded programming technique apart from artificial intelligence and software engineering techniques to improve detection approaches, e.g., zero-day attacks. In recent years, A.I. has played a vital role in intrusion detection systems. In [20], the author proposed to find lousy data injection in smart grids by integrating the features of the power system's inherent physical laws and a traffic flow. This detection model reduces the computational time based on similar features to ensure unknown datasets' precision. The test results indicate detection enhancement rates by 20% over the Chi-square; Additionally, Security-Oriented Cyber-Physical State Estimation (SCPSE) major detectable issues, i.e., Snort false alarms solved by using filters techniques. In [21], to detect adversarial attacks in (SDN) Software-Defined networks, LSTM and CNN-based detection systems are used. In [22], Deep learning approaches have been employed to create a fully connected neural network model and anomaly-based network intrusion detection systems that show manifest outcomes with high accuracy compared to traditional approaches like Adaboost, Random Forest, and SVM. In [23], the deep learning techniques have been demonstrated to having a high detection capability for malicious actions. The (gated recurrent unit–long short-term memory) GRU-LSTM deep learning-based model is executed to detect intrusion in a network-based SDN environment. The author then [24] proposed a method of inspecting the error of reconstruction for the traffic records of the

network by Deep autoencoders. In [25], the author proposed an intelligent attack detection model attaining state-of-the-art performance, with higher efficiency than existing models. In [26], the author proposed a method for vulnerability detection by using a Deep autoencoder and big data visualisation with statistical analysis techniques. The authors proposed a method for detecting cyber threats with a hybrid Intrusion detection technique by combining a deep belief network and genetic algorithm that achieves a 99% accuracy in [27]. The authors proposed a method for cyber threats detection based on convolutional LSTM and Spark machine learning with 97.29% accuracy by using the ISCXUNB dataset in a hybrid detection system [28]. The authors in [29] proposed a detection model that detects suspicious threats in IoT devices using machine learning techniques. This hybrid technique comprises of Isolation Forest, Self-organising map (SOM), One-Class Support Vector Machines (OCSVM Gaussian) and Mixture Modelling (GMM) and with a detection 98% of accuracy. The author in [30] proposed a deep forward neural network and leveraging the deep autoencoders method for the malicious attacks detection in industrial IoTs to achieve 99% accuracy. The author proposed a bio-inspired intrusion technique for crossfire attacks that achieving the rate of 80% detection in [31]. In [32], the authors proposed efficient attack detection by creating LSTM and an ensemble-based GRU method. The protocol Message Queuing Telemetry Transport (MQTT) in the IoT gives 99% accuracy for attack detection. The authors [33] proposed a deep learning-based cybersecurity system to detect threats in IoT networks with 95.5 % accuracy and minimum false-positive rate. IoT traffic behaviour and other network types are comparatively different, as shown in [34]. Traffic for IoT for identifying the devices types whitelisted by using Random Forest-based supervised machine learning algorithm using features extraction method. In [35], the authors proposed the model for gesture recognition. The cognitive perception of gestures is compatible with Frame-level classification. CNNLSTM algorithm is used for gesture recognition. In the study in [36], the authors presented model is appurtenant for distributed deployments and privacy-preserving. This model uses the two-stage hierarchical network intrusion detection (H2ID) technique to detect anomaly detection using a soft-output classifier through the multimodal deep autoencoder (M2-DAE) and attack classification. The authors in [54-55] used Sdn enabled deep learning models for DDoS and Brute force detection in IoT environments. This model was deployed and validated by using the Bot-IoT dataset, including miscellaneous attacks, i.e., Theft, Dos, DDoS, and Scan. The existing literature is presented in detail in Table 1.

Table1. Existing Literature

Ref	Approach	Algorithm	Dataset	Limitations
37	DL and ML techniques are presented for intrusion detection	LSTM,RNN,GRU	Kddcup99	Low Detection accuracy, precision, and recall.
38	DL model is presented for threat detection in intelligent devices	LSTM	Modbus-TCP	Detection complexity is high.
39	Presented a self-learning approach to identify infected devices	GRU	Real-time traffic	Detection accuracy is not good enough.
40	Proposed ML technique for DDoS detection	R.F.	Self-generated using Wireshark	Used machine learning (ML) approaches.

41	Proposed a DL technique for DoS detection in IoT	Deep model	NSL-Kdd	This dataset lack supportive features of IoT.
42	Presented ML technique to detect unknown and known threats	LSTM-RNN	NSL-Kdd	Achieved detection accuracy is low as an ML classifier is used.
43	Presented DL technique to detect botnets	LSTM	CVUT real-time traffic	Unable to identify some sample whether it is malicious or benign.
44	Presented ML techniques for intrusion detection in SDN	SVM	DARPA	Used machine learning (ML) approaches and time complexity is missing.
45	Proposed learning procedures of ANN to detect intrusion by using feed-forward and back learning classifiers	ANN	Internet packet traces	Experiments aren't performed on a proper dataset, and evaluation metrics are missing.
46	Proposed ML techniques for the development of IDS	RTS-DELM-CSIDS	NSL-Kdd	Pre-processing of the dataset is missing those results in Low detection accuracy.
47	Proposed ML technique for the detection of IRC botnet	Baysian, J48, Naïve Bayes	Dartmouth wireless network	Results aren't efficient as the deep learning model.
48	Presented a technique that can detect botnets at the packet level	CNN, RNN	CTU-13 and ISOT	High Detection time.
49	Presented a DL technique based on SDN for DoS detection	RBM	KDD99	Low Detection accuracy.
50	Propose multiple classifiers to improve the rate of learning of algorithms for threat detection	DNN,SVM,J48 and Naivebayes	NSL-Kdd	Low detection accuracy, missing time complexity.

3 Research Methodology

A complete methodology of the proposed work is given in this section as follow:

3.1 Data Processing Module

Usually, Data processing infers the control and collection of things of information to provide valuable data. The data processing in the Failure prediction system means taking the data and then training it or processing it to extract or build a practical model for further use. After that, this model is tested to check that either the build model is accurate or not. Data Processing Module has three main parts, including training data, test data and evaluation factors: Training, Testing and Evaluation.

3.1.1 Training Data

In other words, we can also say that training set, training dataset, or learning set — is the data utilised to prepare a calculation. The preparing information incorporates both input information and the anticipated comparing yield. As it is "ground reality" information, the algorithm can learn how to apply advances such as neural systems to memorise and create complex comes about to make clear choices when afterwards displayed with current

information. For the training purposed, the proposed work used the publicly available CICIDS2018 dataset.

3.1.2 Testing Data

On the other hand, testing data incorporates input information, not the anticipated comparing yield. The testing information is utilised to survey how well your calculation was prepared and assess demonstrate properties. The test dataset may be used to supply an impartial assessment of a last show fit on the preparing dataset.

3.1.3 Evaluation

Deep Learning proceeds to be a progressively essentially component of our lives, whether we're applying the strategies to inquire about or commerce issues. Deep learning models have to grant clear expectations in arranges to form genuine esteem for a given organisation. Whereas preparing a model may be a critical step, how the show generalises on concealed information is a similarly crucial perspective that should be considered in each machine learning pipeline. We got to know whether it works and, subsequently, if able to believe its expectations. Might the demonstrate be merely memorising the information it is bolstered with, and so incapable of forming great expectations on future tests, or tests that it hasn't seen some time recently, usually called the Evaluation.

3.2 Implementation of Data Processing Module

In the data processing module, there are three basic operations, first is to train the data for the trained model, the second is test data, and the third is the Evaluation of the test data that how much it's accurate. When the system starts, it will load the training data set, and after that, it will train the data, and when the system gets the trained model, test data will pass to this trained model to check its accuracy that how much this model is accurate. After the accuracy test, this model will be ready if the precision is acceptable. When the system starts, it will load the training data set, and after that, it will train the data, and when the system gets the trained model, test data will pass to this trained model to check its accuracy that how much this model is accurate. After the accuracy test, this model will be ready if the precision is acceptable. When the system starts, it will load the training data set, and after that, it will train the data, and when the system gets the trained model, test data will pass to this trained model to check its accuracy that how much this model is accurate. After the accuracy test, this model will be ready if the precision is acceptable. When the system starts, it will load the training data set, and after that, it will train the data, and when the system gets the trained model, test data will pass to this trained model to check its accuracy that how much this model is accurate. After the accuracy test, this model will be ready if the precision is acceptable. The whole process of this module can be represented by a flow chart diagram, as shown in Figure 1.

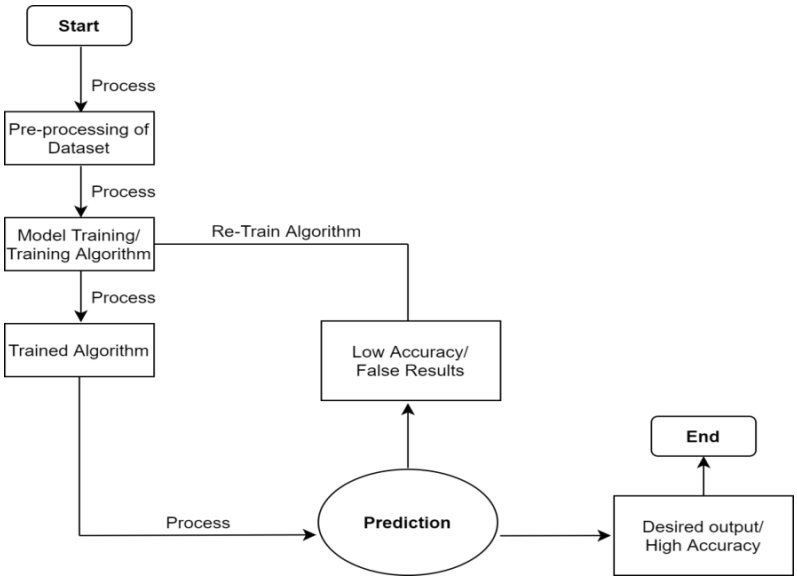

Figure 1. Flow chart of Data processing module

3.3 Deep Learning Module

Deep learning may be an A.I. subset and machine learning, which is being used in complex manufactured neural systems to convey high-tech exactness in assignments such as question discovery, discourse acknowledgement, and dialect interpretation. Deep learning contrasts with conventional machine learning procedures. Consequently, they can learn representations from information like pictures, video or content, with no presentation of hand-coded instructions or human space information. Their exceedingly flexible structures learn specifically from crude news and increment their prescient exactness when given more information. Deep learning relies on numerous later innovations in Artificial Intelligence like Google DeepMind's AlphaGo, self-driving cars, brilliantly voice associates and plentiful more. Analysts and information researchers can altogether speed up profound learning, preparing that may something else take hours and days to fair weeks. In models preparation for arrangement, designers may depend upon GPU-accelerated induction stages for the cloud, implanted gadgets or self-driving cars to provide top-performances and low-latency induction for the foremost computationally-intensive profound neural systems. In this module, I will take the information set and then perform the classification and highlight extraction method simultaneously as the author(Irivbogbe Jimmy) utilises deep learning algorithms.

3.4 Evaluation Methodology

The proposed work employed standard evaluation metrics such as recall, F1-score, precision, training and testing time, Tpr, Tnr, Fpr, and Fnr of the proposed module. The mathematical formulas are given in the below subsections.

3.4.1 *Accuracy*

The accuracy can be calculated by using the mentioned formula.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

3.4.2 ***Recall***

The recall can be calculated by using the mentioned formula.

$$\text{Recall} = \frac{TP}{TP + FN} \quad (2)$$

3.4.3 ***Precision***

The precision can be calculated by using the mentioned formula.

$$\text{Precision} = \frac{Tp}{TP + FP} \quad (3)$$

3.4.4 ***F1-Score***

The F1-score can be calculated by using the mentioned formula.

$$F1 - \text{score} = \frac{2* TP}{2* TP + FP + FN} \quad (4)$$

1. TP = True positive rate
2. TN = True negative rate
3. FP = False positive rate
4. FN = False-negative rate

3.5 Dataset

The dataset has a great impact on the performance evaluation of any detection scheme. Different authors used different datasets in existing literature, such as NSL-KDD, KDDCUP99, MODBUS-TCP etc., for threat detection in IoTs. Some of the datasets used by the authors don't have the supportive features of the IoT. The attackers always try to find local devices by creating web pages and try to control the devices. Further, they also use DNS rebinding to discover these devices. The proposed work used the CICIDS2018 dataset for experimentation. The dataset has the supportive features of the IoT. It contains different attacks as well as benign. Further, the dataset has 82 features. A list of the features is given in Table 3. The proposed work used five different classes of attacks and 1 class of benign. The total distribution of the instances is 80,000. The detail of these classes is given in Table 3.

Table 3. Dataset Description

Classes	Instances	Attack
Benign	65500	-
Brute force	2900	FTP
Bot	2900	
DDoS	2900	Slowloris
	2900	Goldeneye
Infiltration	2900	-
Total	80,000	

4 System Design

4.1 Network Model

A deep learning approach was proposed with an SDN enabled mechanism to detect the cyber risks in the Internet of things, as shown in Figure 2. SDN consists of three planes: the data plane, the control plane, and the application plane. The Data plane is made up of a variety of IoT devices, sensors, and intelligent devices. The most critical aspect of SDN is the control plane. It is completely programmable. All the major decisions of the network are managed here. Lastly, the application plane runs different applications to deliver services for end-users. The proposed model is deployed in the control plane of the SDN because the control plane can extend a lot of networks on the data plane, which provides the solution for heterogeneity between the controller of SDN and IoT devices. Integrating IoT with SDN proposes a perfect way to inspect network traffic to identify and detect cyber-attacks. The proposed model is highly cost-effective.

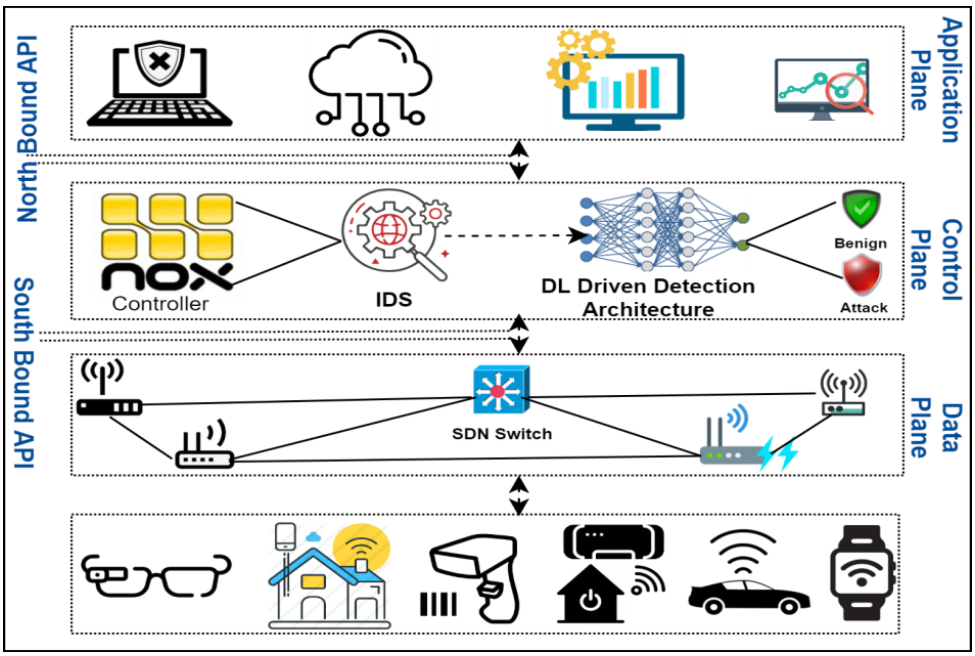

Figure 2 Network Model

4.2 Detection Scheme

The author presents deep learning; SDN enables an approach for the detection of threats in IoT. The DNNLSTM classifier for experimentation. The proposed algorithm can detect multi-class threats. A complete overview of the proposed detection scheme is given in Figure 3. The proposed work used two more algorithms, i.e., GRU and LSTMGRU, for comparison. Both of these algorithms are trained and evaluated on the same dataset for performance evaluation. The proposed model has 01 layers of DNN with 100 neurons and 01 layers of LSTM with 50 neurons. Further, the comparison algorithms, i.e., GRU, have 01 layers with 100 neurons, and LSTMGRU has 01 layers of LSTM with 100 neurons and one layer of GRU with 50 neurons accordingly. The author trained and tested the proposed model using a few false positives, which resulted in a higher detection precision. The experimentations were executed until 05 epochs were attained, having 64 batch volumes to get efficient results. The author also made the differences between the proposed model to the current work in Table 2. For activation function, the proposed work has used RELU for all three algorithms with Adamax as optimiser. These optimum parametric values were achieved after several trials. For the sake of implementation, the proposed work uses the Keras Python framework for TensorFlow at the backend. This uses (GPU) graphical processing unit and a Cuda-enabled version for enhanced performances. The complete description of these algorithms is given in Table 2.

Table2. Algorithms Description

Algorithm	A.F.	Layers	Neurons	Optimiser	Epochs
Cu-DNN-LSTM	Relu	DNN (1)	(100)	Adamax	5
	-	LSTM (1)	(50)		
	-	Dropout	(0.6)		
	Softmax	Output Layer (1)	7		
Cu-GRU	Relu	GRU (1)	(100)	Adamax	5
	-	Dropout	(0.6)		
	Softmax	Output Layer (1)	7		
Cu-LSTMGRU	Relu	LSTM(1)	(100)	Adamax	5
		GRU (1)	(50)		
	-	Dropout	(0.6)		
	Softmax	Output Layer (1)	7		

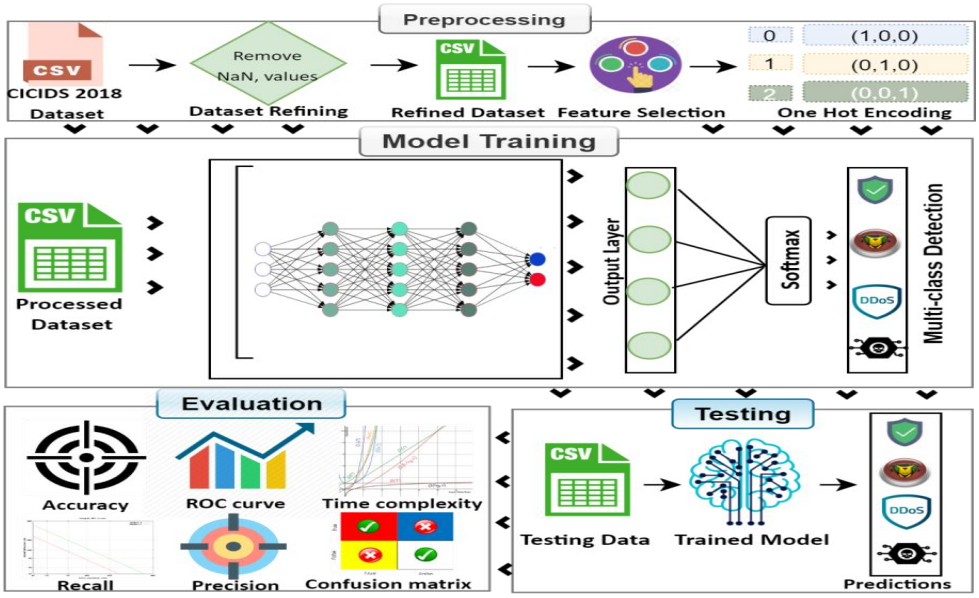

Figure 3. Detection Scheme

4.3 Pre-processing of Dataset

For better results and accuracy, the performance of the proposed model, dataset pre-processing has been done to enhance its quality. The pre-processing plays a vital role in the implementation's output as null values, and infinity values affect the detection accuracy. That is why the pre-processing of the dataset is done. To start, the instances with missing, null and infinity values have been dropped from the dataset. Also, deep learning classifiers input numeric values only, so the conversion of non-numeric values into numeric values is performed. The normalisation of data was also performed to improve the dataset's quality, mainly when values are out of bounds.

Table4. Dataset Features

S. No	Feature	S. No	Feature	S. No	Feature
1	Flow Duration	30	Bwd Iat Avg	59	Down Up Ratio
2	Source IP	31	Fwd Packets Length Max	60	Bwd Packets Length Min
3	Destination IP	32	Fwd Psh Flags	61	Rst Count
4	Timestamp	33	Bwd Iat Max	62	Fwd Packets Bulk Avg
5	Source Port	34	Flow Iat Avg	63	Fwd Iat Avg
6	Destination Port	35	Fwd Urg Flags	64	Subflow Bwd Packets
7	Protocol	36	Bwd Packets	65	Bwd Byt Bulk Avg
8	Total Fwd Packets	37	Packet Length Variance	66	Fwd Act Packets
9	Fwd Packets Length Min	38	Urg Count	67	Atv Max
10	Bwd Packets Length Avg	39	Packet Size Avg	68	Bwd Bulk Rate Avg
11	Flow Iat Std	40	Fwd Bulk Rate Avg	69	Idle Avg
12	Fwd Iat Std	41	Subflow Fwd Byt	70	Ack Count
13	Total Bwd Packets	42	Bwd Urg Flags	71	Fwd Seg Min
14	Fwd Packets Length Avg	43	Packets Len Min	72	Subflow Bwd Byt
15	Bwd Packets Length Std	44	Fin Count	73	Atv Std
16	Flow Iat Max	45	Cwe Count	74	Fwd Packets
17	Fwd iat max	46	Bwd Seg Avg	75	Packets Length Std
18	Total Length of Fwd Packets	47	Bwd Packets Bulk Avg	76	Pst Count
19	Fwd Packets Length Std	48	Fwd Win Byt	77	Fwd Seg Avg

20	Flow Byt Subflow	49	Fwd Hdr Length	78	Idle Min
21	Flow Iat Min	50	Packets Length Avg	79	Active Min
22	Fwd Iat Min	51	Fin Count	80	Idle Max
23	Bwd Packets Length Max	52	Ece Count	81	Active Avg
24	Flow Packets Subflow	53	Fwd Byt Bulk Avg	82	Label
25	Flow Iat Min	54	Subflow Fwd Packets		
26	Bwd Iat Total	55	Bwd Win Byt		
27	Bwd Iat Std	56	Bwd Hdr Length		
28	Bwd Psh Flags	57	Packets Length Max		
29	Bwd Iat Min	58	Syn Count		

5 Implementation

The tools and techniques that were used for the experimentation in this work are described in this section as follow:

5.1 Python

Python is an object-oriented, high-level language for programming with dynamic semantics. Its high-level built-in information structures, blended with dynamic typing and dynamic binding, make it very beautiful for speedy application development and for use as a scripting or glue language to connect present components. Python's simple, handy to study syntax emphasises readability and, for this reason, reduces the cost of software program maintenance.

5.2 Tensor Flow

Tensor Flow is an open-source machine learning library that lets users organise computation to process components with a single API [15]. The T.F. delivers a high-level API for various kinds of layers in ANN, such as a pooling layer, convolutional layer, and fully connected layer. It also offers approaches of applying dropout regularisation and adding activation functions.

The name of Tensor Flow is derived from its main framework: Tensor. In T.F., all the calculations comprise tensors. A tensor is a vector or matrix of n-dimensions that signifies all kinds of data. Every value in a tensor has the same data type with a shape of partially known or known. The shape of the data is the dimensionality of the array or matrix. The edge of the nodes is the tensor, i.e., a method to occupy the operation with data. The single major advantage of T.F. is the abstraction that it delivers for machine learning development. Instead of dealing with the particulars of implementing algorithms or finding suitable conducts to hitch the result of one function to the input, the designer can centre on the complete logic of the application. Some other libraries, i.e., NumPy, pandas, and sklearn are also used in this work.

5.3 Anaconda

Anaconda is an open and accessible distribution specially designed for Python and R languages for different scientific complex computing (data science, significant information processing, predictive analytics, computing, etc.) is ambitious to simplify package management and deployment.

5.4 Spyder

An integrated development environment (IDE) approves pc programmers by integrating necessary tools (e.g., code editor, compiler, and debugger) into a single software package. Users do now not want to set up the language's compiler/interpreter on their machines; an IDE provides the environment itself. Spyder is a dedicated IDE for Python.

6 Evaluation

In this section, the performance of the proposed model is evaluated by standard evaluation metrics. The experimentation is carried out on the CICIDS2018 dataset. This research aims to detect different attacks in the IoT environment. The results of all three algorithms are evaluated. This work evaluated the output of the proposed algorithm with the other two deep learning algorithms and the existing literature.

6.1 Accuracy

For a better assessment, this work presents the model accuracy. Figure 4 shows that the proposed model accuracy of 99.92%, which is far better than the existing literature and the other models. The accuracy has been obtained from the implementation result by implementing the DNNLSTM algorithm on the CICIDS 2018 dataset to train the algorithm for threat detection. The achieved accuracy proves that our proposed model is very efficient.

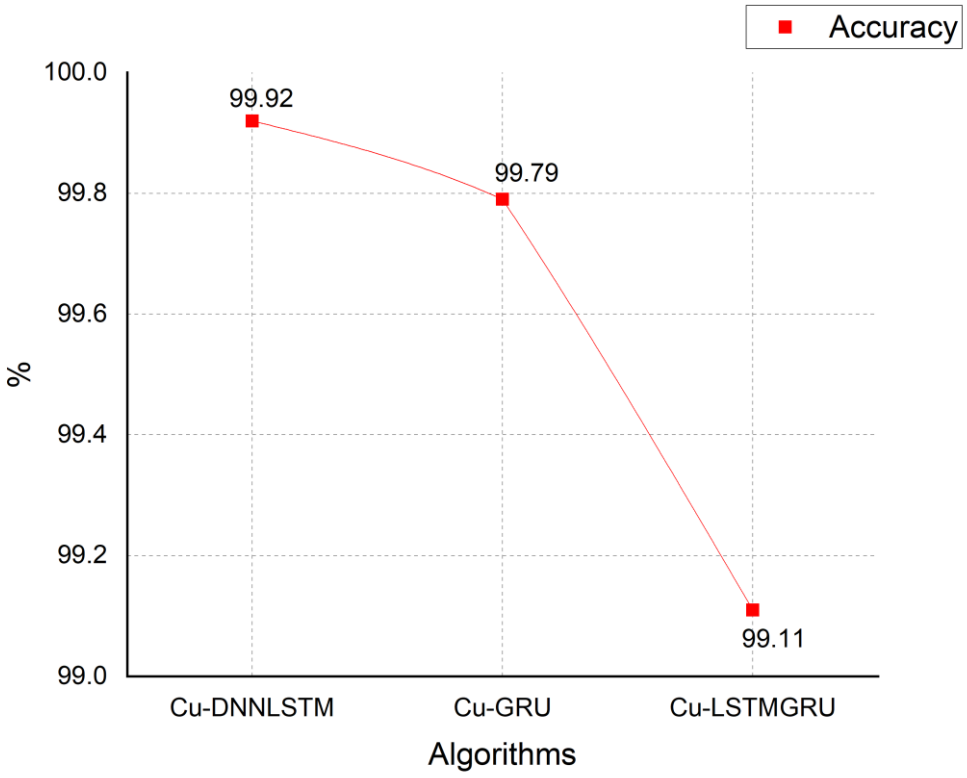

Figure 4 Accuracy of the Model

6.2 Precision, recall and F1-score

For a thorough analysis, the proposed work presents the evaluation metrics of the suggested model results. The proposed model shows a precision of 99.30 percent with recall

of 99.50 percent. Furthermore, the F1- score of the proposed model is 99.5 percent. Figure 5 clearly depicts the precision, recall, and F1-score of the three implemented models.

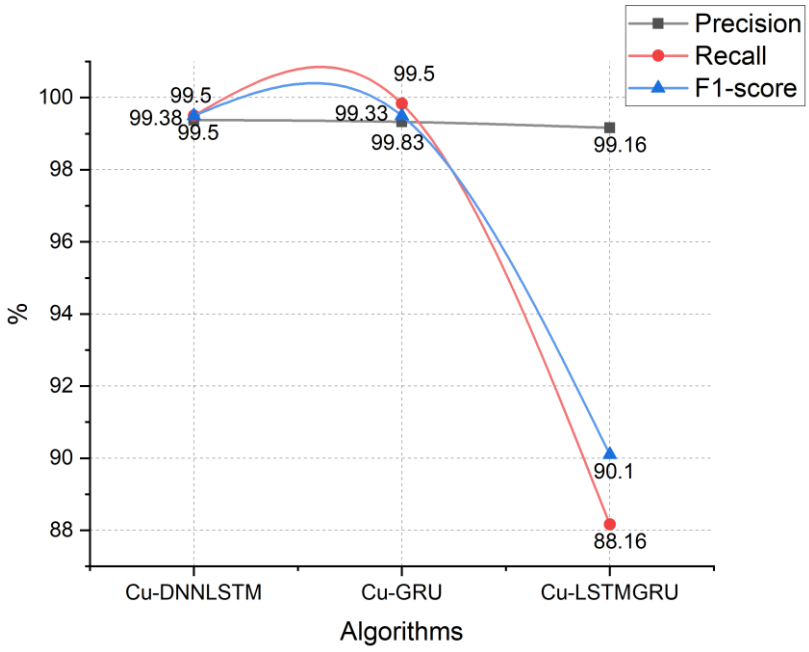

Figure 5 Precision, Recall and F1-score

6.3 Confusion matrix

The confusion matrix is mainly used for classification purposes. The confusion matrix is essential for measuring the F1-score, recall as well as accuracy. It shows the true positive, true negative, false-positive and false-negative rates. A thorough analysis of the confusion matrix shows that the proposed work successfully identified classes correctly. Figure 6 shows the confusion matrix of the three algorithms.

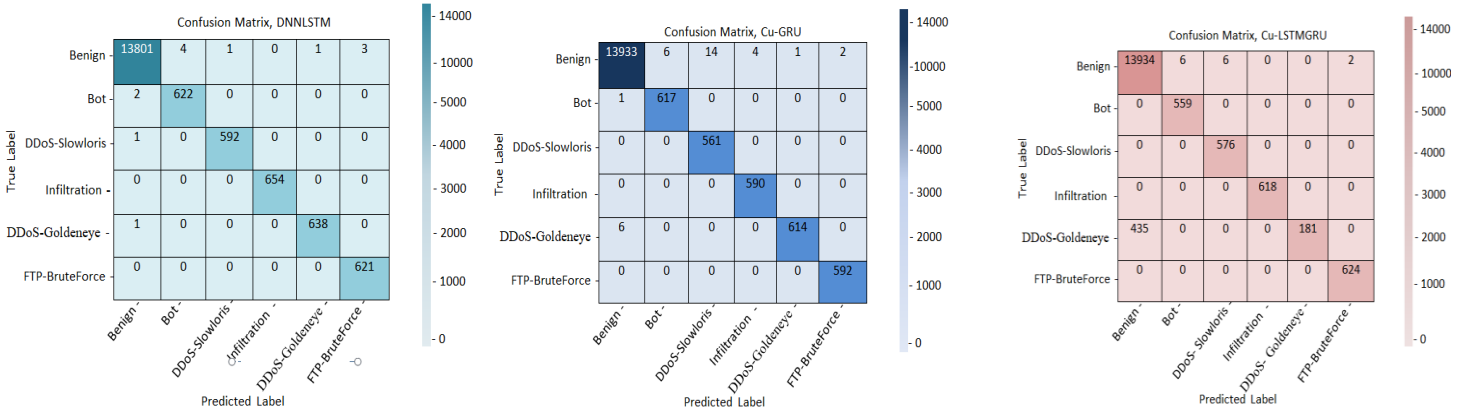

Figure 6 Confusion Matrix of DNNLSTM, GRU, and LSTMGRU

6.4 Fdr, Fnr, Fpr

The proposed work further measure the evaluation metrics for better estimation, i.e., false discover rate (FDR), false-negative rate (FNR), and false-positive rate (FPR). The outcome in figure 7 shows that the model proposed achieved an Fpr of only 0.0070 %, Fnr of 0.0011 % and FDR of only 0.0024 %.

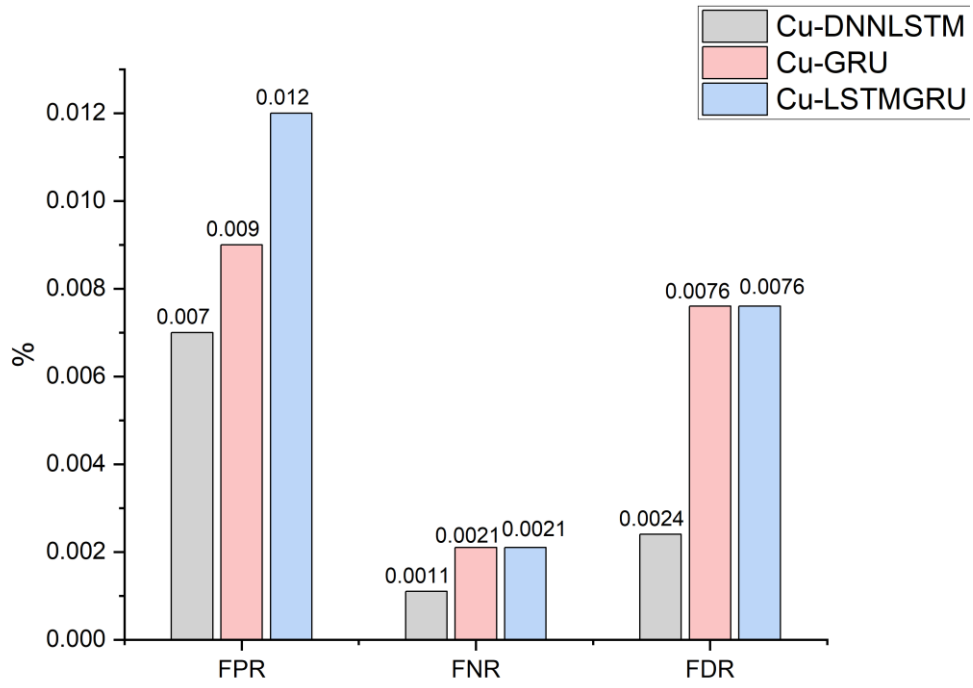

Figure 7 Fdr, Fnr and Fpr

6.5 Tpr, Tnr and Mcc

The Matthews correlation coefficient (MCC) is a more reliable statistical rate that produces a high score only if the prediction obtained good results in all of the four confusion matrix categories (true positives, false negatives, true negatives, and false positives). A matrix on uncertainty was used to get the true negative rate, True positive rate, and Mathews correlation coefficient. Figure 8 clearly shows the values of the tpr, tnr and Mcc, respectively.

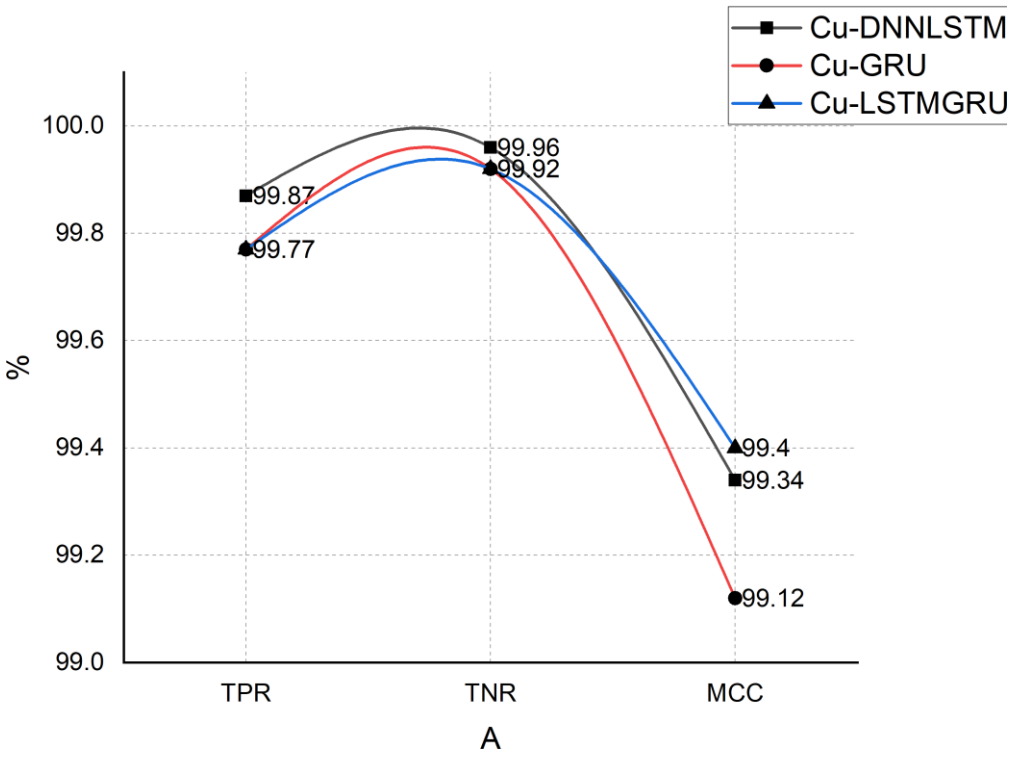

Figure 8 Tpr, Tnr and MCC

6.6 Comparison with existing literature

For a better assessment, the outcome of the proposed model is equated to the existing literature. The proposed model outclassed the other two DL classifiers as well as existing literature. The comparison is made based on standard evaluation metrics, i.e., recall, precision, F1-score, accuracy and other evaluation metrics. The comparison with the other two algorithms is shown in different parts of this work. However, the comparison with existing literature is shown in Table 5 below:

Table 5: Comparison with existing literature

Ref	Accuracy	Algorithm	Dataset	F1-score	Precision	Recall
Proposed	99.92 %	DNNLSTM	CICIDS2018	99.50 %	99.3 %	99.5 %
51	89 %	GRU-RNN	CICIDS2017	99.0 %	99.0 %	99.0 %
52	96.1 %	2L-ZED-IDS	CICIDS2018	-	93.2 %	96.9 %
53	91.5 %	CNN	CICIDS2018	-	-	-

7 Discussion

This research work did the threat detection in an IoT environment by using deep learning algorithms that have been trained and evaluated on publicly available CICIDS 2018 dataset. The dataset has features of IoTs. This work used the DNNLSTM algorithm for threat

detection. Further, this work used two more algorithms, i.e., GRU and LSTMGRU, for comparison. It has been discovered that the proposed model achieved high accuracy with very low false positives and false negatives. Table 6 depicts the results of the three algorithms. The achieved accuracy of the proposed model is 99.92%, with TNR and TPR of 99.96% and 99.87%. However, the accuracy of the GRU and LSTMGRU is 99.79% and 99.11, with TPR of 99.77% and 99.77. A complete comparison is shown in Table 6 accordingly.

Table 6. Comparison of proposed model with other algorithms

Algorithm	***Accuracy***	***Precision***	***Tpr***	***Tnr***	***Fpr***	***Fnr***	***F1-score***	***Recall***
DNNLSTM	*99.92%*	*99.30%*	*99.87%*	*99.96%*	*0.0070%*	*0.0011%*	*99.50%*	*99.50%*
GRU	*99.79%*	*99.33%*	*99.77%*	*99.92%*	*0.0090%*	*0.0021%*	*99.50%*	*99.83%*
LSTMGRU	99.11%	99.16%	99.77%	99.92%	0.012%	0.0021%	90.10%	88.19%

8 Conclusion and Future work

There has been an enormous growth in IoT devices in recent years. This growth has also increased cyber-attacks. As IoT devices are connected through the Internet, so they are vulnerable to such cyber-attacks. Different traditional mechanisms have been employed to protect these devices. However, IoT devices are heterogeneous, and these mechanisms weren't sufficient. There this introduced work SDN enabled deep learning mechanism for the protection of these devices. This work used a DNNLSTM classifier for intrusion detection in IoT devices. The proposed model achieved a detection accuracy of 99.92 percent along with recall and f1-score o5 99.50 %, respectively. This work also used two more algorithms for comparison purposes. However, the proposed model outclassed the other two algorithms and the existing literature in detection accuracy and other evaluation metrics. In the future, the authors aim to use blockchain-enabled mechanisms for IoT security and different deep learning algorithms, as deep learning shows tremendous results in threat detection.

References

1. Mrabet, H.; Belguith, S.; Alhomoud, A.; Jemai, A. A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis. *Sensors* **2020**, *20*, 3625.
2. A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one-class support vector machine and grey wolf optimisation for iot botnet detection," Journal of Ambient Intelligence and Humanized Computing, vol. 11, no. 7, pp. 2809–2825, 2020.
3. T. Hasan, A. Adnan, T. Giannetsos, and J. Malik, "Orchestrating sdn control plane towards enhanced iot security," in 2020 6th IEEE Conference on Network Softwarization (NetSoft). IEEE, 2020, pp. 457–464.
4. Haller, S.; Karnouskos, S.; Schroth, C. The internet of things in an enterprise context. In *Future Internet Symposium*; Springer: Berlin/Heidelberg, Germany, 2008; pp. 14–28.

5. Ferdowsi, A.; Saad, W. Deep Learning for Signal Authentication and Security in Massive Internet-of-Things Systems. *IEEE Trans. Commun.* **2019**, *67*, 1371–1387.
6. Bhunia, S.S.; Gurusamy, M. Dynamic attack detection and mitigation in IoT using SDN. In Proceedings of the 27th International Telecommunication Networks and Applications Conference (ITNAC), Melbourne, VIC, Australia, 22–24 November 2017; pp. 1 – 6.
7. Ben-Asher, N.; Gonzalez, C. Effects of cybersecurity knowledge on attack detection. *Comput. Hum. Behav.* **2015**, *48*, 51–61.
8. Ding, D.; Qing-Long, H.; Yang, X.; Xiaohua, G.; Xian-Ming, Z. A survey on security control and attack detection for industrial cyber-physical systems. *Neurocomputing* **2018**, *275*, 1674–1683.
9. Wu, K.; Chen, Z.; Li, W. A Novel Intrusion Detection Model for a Massive Network Using Convolutional Neural Networks.IEEE Access 2018, 6, 50850–50859.
10. Hu, T.; Niu, W.; Zhang, X.; Liu, X.; Lu, J.; Liu, Y. An Insider Threat Detection Approach Based on Mouse Dynamics and Deep Learning. *Secur. Commun. Netw.* **2019**, *2019*, 1–12.
11. Schueller, Q.; Basu, K.; Younas, M.; Patel, M.; Ball, F. A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018; pp. 1 – 6.
12. S. Qureshi *et al.,* "A Hybrid DL-Based Detection Mechanism for Cyber Threats in Secure Networks," vol. 9, pp. 73938-73947, 2021.
13. Meng, F.; Fu, Y.; Lou, F. A network threat analysis method combined with kernel PCA and LSTM-RNN. In Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), Xiamen, China, 29–31 March 2018 ; pp. 508–513.
14. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). *Int. J. Inf. Syst. Model. Des.* **2017**, *8*, 43–63.
15. Li, H.; Wei, F.; Hu, H. Enabling Dynamic Network Access Control with Anomaly-based IDS and SDN. *Secur. Softw. Def. Netw. Funct. Virtual.* **2019**, 13–16.
16. Latah, M.; Toker, L. Artificial intelligence enabled software-defined networking: A comprehensive overview. *IET Netw.* **2019**, *8*, 79–99.
17. Oo, M.M.; Kamolphiwong, S.; Kamolphiwong, T. The design of SDN based detection for distributed denial of service (DDoS) attack. In Proceedings of the 2017 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand, 15–18 November 2017; pp. 1 – 5.
18. Raiyn, J. A survey of cyber attack detection strategies. *Int. J. Secur. Appl.* **2014**, *8*, 247–256.
19. Haider, A.; Muhammad, A.K.; Abdur, R.; Muhib, U.R.; Hyung, S.K. A Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System. *CMC-Comput. Mater. Cont.* **2021**, *66*, 1785–1798.
20. Liu, T.; Yanan, S.; Yang, L.; Yuhong, G.; Yucheng, Z.; Dai, W.; Chao, S. Abnormal traffic-indexed state estimation: A cyber–physical fusion approach for smart grid attack detection. *Future Gener. Comput. Syst.* **2015**, *49*, 94–103.

21. Huang, C.-H.; Lee, T.-H.; Chang, L.-h.; Lin, J.-R.; Horng, G. Adversarial Attacks on SDN-Based Deep Learning IDS System. *Int. Conf. Mobile Wirel. Technol.* **2019**, *513*, 181–191.
22. Baek, S.; Kwon, D.; Kim, J.; Suh, S.C.; Kim, H.; Kim, I. Unsupervised Labeling for Supervised Anomaly Detection in Enterprise and Cloud Networks. In Proceedings of the 2017 IEEE 4th International Conference on Cyber Security and Cloud Computing (CSCloud), New York, NY, USA, 26–28 June 2017; pp. 205–210.
23. Dey, S.K.; Rahman, M.M. In Flow based anomaly detection in software defined networking: A deep learning approach with feature selection method. In Proceedings of the 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (iCEEiCT), Dhaka, Bangladesh, 13–15 September 2018; pp. 630–635.
24. Dawoud, A.; Shahristani, S.; Raun, C. A Deep Learning Framework to Enhance Software Defined Networks Security. InProceedings of the 2018 32nd International Conference on Advanced Information Networking and Applications Workshops (WAINA), Krakow, Poland, 16–18 May 2018; pp. 709–714.
25. Fu, Y.; Lou, F.; Meng, F.; Tian, Z.; Zhang, H.; Jiang, F. An Intelligent Network Attack Detection Method Based on RNN. In Proceedings of the 2018 IEEE Third International Conference on Data Science in Cyberspace (DSC), Guangzhou, China, 18 – 21 June 2018; pp. 483–489.
26. Arora, K.; Chauhan, R. Improvement in the performance of deep neural network model using learning rate. In Proceedings of the Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1 – 5.
27. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722.
28. Khan, M.; Karim, M.; Kim, Y. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network. *Symmetry* **2019**, *11*, 583.
29. Bhatt, P.; Morais, A. HADS: Hybrid anomaly detection system for iot environments. In Proceedings of the International Conference on Internet of Things, Embedded Systems and Communications (IINTEC), Hamammet, Tunisia, 20-21 December 2018; pp. 191–196.
30. Sisinni, E.; Saifullah, A.; Han, S.; Jennehag, U.; Gidlund, M. Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Trans. Ind. Inf.* **2018**, *14*, 4724–4734.
31. Mansour, A.; Azab, M.; Rizk, M.R.; Abdelazim, M. Biologically-inspired SDN-based intrusion detection and prevention mechanism for heterogeneous IoT networks. In Proceedings of the IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON), Vancouver, BC, Canada, 1–3 November 2018; pp. 1120–1125.
32. Alaiz-Moreton, H.; Aveleira-Mata, J.; Ondicol-Garcia, J.; Muñoz-Castañeda, A.L.; García, I.; Benavides, C. Multiclass Classification Procedure for Detecting Attacks on MQTT-IoT Protocol. *Complexity* **2019**, *2019*.
33. Narayanadoss, A.R.; Truong-Huu, T.; Mohan, P.M.; Gurusamy, M. Crossfire attack detection using deep learning in software defined ITS networks. In Proceedings of the

89th Vehicular Technology Conference (VTC2019-Spring), Kuala Lumpur, Malaysia, 28 April–1 May 2019; pp. 1 – 6.

34. Meidan, Y.; Bohadana, M.; Shabtai, A.; Ochoa, M.; Tippenhauer, N.O.; Guarnizo, J.D.; Elovici, Y. Detection of unauthorised IoT devices using machine learning techniques. *arXiv* **2017**, arXiv:1709.04647.
35. Tsironi, E.; Barros, P.; Weber, C.; Wermter, S. An analysis of Convolutional Long Short-Term Memory Recurrent Neural Networks for gesture recognition. *Neurocomputing* **2017**, *268*, 76–86.
36. Bovenzi, G.; Giuseppe, A.; Domenico, C.; Valerio, P.; Antonio, P. A Hierarchical Hybrid Intrusion Detection Approach in IoT Scenarios. 2020.
37. Vinayakumar, R.; Soman, K.P.; Poornachandran, P. Evaluation of Recurrent Neural Network and its Variants for Intrusion Detection System (IDS). Int. J. Inf. Syst. Model. Des. 2017, 8, 43–63.
38. M. Saharkhizan, A. Azmoodeh, A. Dehghantanha, K.-K. R. Choo, and R. M. Parizi, "An ensemble of deep recurrent neural networks for detecting iot cyber attacks using network traffic," IEEE Internet of Things Journal,2020.
39. T. D. Nguyen, S. Marchal, M. Miettinen, H. Fereidooni, N. Asokan, and A.-R. Sadeghi, "Dïot: A federated self-learning anomaly detection system for iot," in 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS). IEEE, 2019, pp. 756–767.
40. X. Li, G. Zhang, Z. Wang, and W. Zheng, "Hyconv: Accelerating multiphase cnn computation by fine-grained policy selection," IEEE Transactions on Parallel and Distributed Systems, vol. 30, no. 2, pp. 388–399, 2018.
41. A. A. Diro and N. Chilamkurti, "Distributed attack detection scheme using deep learning approach for internet of things," Future Generation Computer Systems, vol. 82, pp. 761–768, 2018.
42. Meng, F.; Fu, Y.; Lou, F. A network threat analysis method combined with kernel PCA and LSTM-RNN. In Proceedings of the 2018 Tenth International Conference on Advanced Computational Intelligence (ICACI), Xiamen, China, 29–31 March 2018; pp. 508–513.
43. P. Torres, C. Catania, S. Garcia, and C. G. Garino, "An analysis of recurrent neural networks for botnet detection behavior," in 2016 IEEE biennial congress of Argentina (ARGENCON). IEEE, 2016, pp. 1–6.
44. Schueller, Q.; Basu, K.; Younas, M.; Patel, M.; Ball, F. A hierarchical intrusion detection system using support vector machine for SDN network in cloud data center. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC), Sydney, NSW, Australia, 21–23 November 2018; pp. 1–6.
45. E. Hodo , et al. , Threat analysis of IoT networks using artificial neural network intrusion detection system, in: Proceedings of the International Sympo- sium on Networks, Computers and Communications (ISNCC), Yasmine Hammamet, 2016, pp. 1–6.
46. Haider, A.; Muhammad, A.K.; Abdur, R.; Muhib, U.R.; Hyung, S.K. A Real-Time Sequential Deep Extreme Learning Machine Cybersecurity Intrusion Detection System. CMC-Comput. Mater. Cont. 2021, 66, 1785–1798.

47. L. Carl et al., "Using machine learning technliques to identify botnet traffic," in Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE, 2006.
48. A. Pektas and T. Acarman, "Botnet detection based on network flow summary and deep learning," International Journal of Network Management, vol. 28, no. 6, p. e2039, 2018.
49. A. Dawoud, S. Shahristani, and C. Raun, "Deep learning and software defined networks: towards secure iot architecture," Internet of Things, vol. 3, pp. 82–89, 2018.
50. Arora, K.; Chauhan, R. Improvement in the performance of deep neural network model using learning rate. In Proceedings of the Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 21–22 April 2017; pp. 1–5.
51. Tang, T.A.; McLernon, D.; Mhamdi, L.; Zaidi, S.A.R.; Ghogho, M. Intrusion Detection in Sdn-Based Networks: Deep Recurrent Neural Network Approach in Deep Learning Applications for Cyber Security. In *Deep Learning Applications for Cyber Security*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 175–195.
52. Catillo, M.; Rak, M.; Villano, U. 2L-ZED-IDS: A Two-Level Anomaly Detector for Multiple Attack Classes. In Proceedings of the AINA Workshops 2020, Caserta, Italy, 15–17 April 2020; pp. 687–696.
53. Kim, J.; Kim, J.; Kim, H.; Shim, M.; Choi, E. CNN-Based Network Intrusion Detection against Denial-of-Service Attacks. Electronics 2020, 9, 916.
54. Javeed, D., Gao, T., & Khan, M. T. (2021). SDN-Enabled Hybrid DL-Driven Framework for the Detection of Emerging Cyber Threats in IoT. *Electronics*, *10*(8), 918.
55. Javeed, D., Gao, T., Khan, M. T., & Ahmad, I. (2021). A Hybrid Deep Learning-Driven SDN Enabled Mechanism for Secure Communication in Internet of Things (IoT). *Sensors*, *21*(14), 4884.