International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Attack Detection and Mitigation in IoT-Fog Architecture: Handling Class Imbalance Problem

Navnath B. Pokale[1], Pooja Sharma[2], Deepak T. Mane[3]

Submitted: 28/05/2023 **Revised**: 08/07/2023 **Accepted**: 25/07/2023

Abstract:-The occurrence of data breaches and cyberattacks has greatly increased across numerous companies, organizations, & industries as a result of the exploitation of security holes in IoT devices. There are more zero-day threats presently since more IoT devices are being linked and employ the various protocols. In the realms of big data and cyber-security, DL (deep learning) has shown to be the most effective technique. because it can extract and learn deep features from known assaults and identify novel attacks. Adopting the DL based assaults identification is the greatest crisis in IoT -Fog architecture since it endures with poor or low degree of data privacy. Thereby, this paper focuses on both attack detection and mitigation of attacker in network. The process starts with the class imbalance problem solving via advanced class imbalanced processing. Subsequently, as the extraction of handcrafted features give addition information related to attack behavior, this paper intends to extract the features like correlation-based features, raw data, improved entropy-based features, as well as statistical features. Attack detection will take place based on the retrieved features trained with the DL combo architecture; a novel hybrid detection model combining Bi-GRU and LSTM, detecting the presence of attack in network. However, it is important to mitigate the attacker existing in the network, and hence a new entropy based mitigation procedure is followed in this article. Finally, the results and discussion section shows the efficiency of proposed work over the conventional methods in terms of different performance analysis.

Keywords—*Attack Detection; IoT-Fog Architecture;Bi-GRU; LSTM; Mitigation.*

Introduction

The prevalence of IoT [9] devices in modern technology including IoVT, IoMT, smart grids, and smart electrical appliances has given rise to a great deal of assaults due to their importance in resource sharing. While physical devices like sensors and actuators provide on-demand cloud administration, their concentration is dangerous. With all of this in mind, delivering cloud services to the IoT presents significant issues in terms of data abstention, data security, data obtrusion, and data shielding.

[1]School of Engineering and Technology,D Y Patil Univeristy, Ambi, Pune- 410506, Maharashtra, India
[3]Vishwakarma Institute of Technology, Pune-411037, Maharashtra, India
nbpokale@gmail.com[1], poojasharma861984@gmail.com[2], dtmane@gmail.com[3]
Corresponding Author: Deepak T. Mane (dtmane@gmail.com)

To provide services near to the network's edge and address cloud-based IoT issues, the abstraction layer Fog is utilized. Between the cloud and client devices, a distributed decentralized paradigm called fog has developed [13], enabling it to offer services with lower latency and network bandwidth consumption. For the smart devices to communicate, noncellular network protocols including LoRa, COAP, LoRaWAN, and MQTT are necessary. Due to their low latency and low bandwidth usage, these protocols are beneficial to end users [15] [16]. Fog communication protocols are used to share and store data acquired from end devices so that it may be quickly retrieved later. The fog layer/fog-node is more vulnerable to assaults when exchanging data. Also, the fog nodes have a poor degree of data privacy and are susceptible to attacks like probing, DDoS, man-in-the-middle, port scan attacks, and many others [14], showing its low level of data security. The fog layer therefore requires an attack

detection mechanism, and to specify this, it requires a security system in the fog layer.

As the online usage has greatly increased, a significant amount of information has moved, leading to an increased number of oddities. Attacks' causes are also consistently growing in proportion. To provide secure forms of assistance to end users, several associations are continuously working on network attack [17] [18] detection. Also, the risk of information infringement is increased due to the widespread usage of IoT and cloud services via the fog layer. Here, it is necessary to provide or set up a more secure system using DL techniques that can effectively identify threats. With the growth of the internet, the general population is turning to modern innovations that use ML and DL techniques [19] to predict, recognize, or arrange network behavior.

Attack detection is now the most recent trend and area of investigation for cyber dangers. ML methods are heavily used at first to identify assaults, however they are not allowed for massive amounts of data. As DL handles several layers with a high DR, it is used to discriminate attacks in the fog layer in order to surpass the limit of ML. When an attacker is discovered, the fog node updates the cloud with the node's behavior, classifying it as malicious, nonmalicious, or multilabel. The use of DL to classify various attacks has resulted in binary classifications of usual and abnormal behavior as well as multilabel classifications that are delivered to the cloud for node behavior updates [20]. Moreover, DL has a top-notch detection rate. It is absurd to expect to do complicated DL computations given the resource-constrained nature of IoT. Considering this, DL is appropriate to perform on a fog node or fog layer with great accuracy. Number of DL methods are in progress to detect the attacks including RNN, LSTM [3], which gives satisfactory results on accurate detection. With the consideration, this paper intends to propose a new hybrid model with the combination of Bi-GRU and LSTM, and the major contribution of the paper is as follows:

1. Dealing with class imbalance problem with advanced class imbalance processing that enhances the further performance of detection.
2. Contributing with the extraction of handcrafted features includingimproved entropy-based features, raw data, statistical features, and correlation-based features.
3. Presented a hybrid classifier with the combination of Bi-GRU and LSTM for attack detection, and once the presence of attack is detected, followed a new mitigation strategy to eradicate the attacker from the network by following modified entropy based mitigation process.

The article isordered as: Section II covers the review. An overview of the attack detection scheme in IoT-Fog is given in Section III. Section IV establishes the preprocessing through enhanced feature extraction and class imbalance processing of the proposed model. The identification using hybrid classifiers and an enhanced mitigation mechanism are shown in Section V. Sections VI and VII offer the results & conclusion.

Literature Review

A. Related works

In 2021, Manimurugan*et al.* [1] has implemented a combined infrastructure for the IoT-Fog-Cloud computing idea. This study provides the INB classifer utilizing the PCA technique focused on NIDS. The UNSW-NB15 dataset has been used to assess the assault detection system. The PCA approach was used to establish the dataset properties, whereas the INB classifier was determined to categorize attacks. This approach has been recommended in order to improve performance analysis based on accuracy, DR, recall, and precision & to raise the efficiency of anomaly detection.

In 2020, Samy *et al.* [2] has adopted a comprehensive framework for detecting theattack that was powerful, distributed, and larger DR to recognize different IoT cyber-attacks using DL. The proposed frameworks distributed structure, and powerful computational power to enable it to create an attack detector for fog nodes. The suggested framework has a greater detection accuracy in multi-class classification, a higher detection rate for binary classification, and can detect a wide range of cyberattacks.

In 2023, Sanju*et al.* [3] has suggested a hybrid metaheuristics-DL strategy to improve IoT devices' ability to identify intrusions. The intrusion detection in the IoTmight be improved by using a sophisticated metaheuristics method using an ensemble of RNNs. The GRU and LSTM models, which make up RNNs, have been employed to identify various sorts of threats in IoT systems. As used in this study, feature selection has been carried out using HHO and fractional derivative mutation. Publicly accessible datasets were used to evaluate the suggested strategy, and the empirical study showed that it performs better than the other comparable methods with respect to accuracy as well as effectiveness. The suggested

approach offered a potential overall method for improving IDS in IoT.

In 2023, Mirdula*et al.* [4] has invested a DL algorithm-based approach to increase network security for the Internet of Things. The IDS taken into account in this study was Network IDS, which examines information on user behavior based on deep learning, digital twins, and manufacturer usage descriptions. Users in smart buildings and IoT devices are automatically connected through the Intelligent Communication system. The pattern of aberrant or anomalous traffic at the device level will be predicted while traffic is occurring using MUD profiles, dynamic user behavior, and IoT device traffic data. Python software is used to implement the MUD-ML-based model and test the outcomes.

In 2021, Sudheera*et al.* [5] has developed ADEPT, a distributed framework to spot and detect every stage of a coordinated attack. The procedure for Adept has three steps. The network traffic was initially inspected locally of IoT devices to check anomalies compared to their typical features. Last but not least, they utilize a ML technique to distinguish various attack phases in the generated warnings utilizing characteristics that include both pattern-level &alert-level information. They conduct in-depth tests using simulated and accurate network traffic, and the outcomes show how effective the suggested methodology was in identifying and locating the attack-stage.

In 2023, Rania *et al.* [6] has established a SATIDS based on a better LSTM network. The suggested system distinguishes between malicious and lawful communications, recognizes the category of attacks, and specifies the kind of sub-attack with high performance. Two of the most recent realistic datasets, ToN-IoT and InSDN, were used to train and evaluate the suggested system in order to demonstrate its efficacy. They examined and contrasted its performance with that of other IDSs. The experimental findings demonstrate that the suggested approach outperforms competitors in identifying a wide variety of assaults. For the ToN-IoT dataset, it achievedhigh accuracy, detection rate, and precision.

In 2022, Bhukya*et al.* [7] has suggested a new DIDS approach to DL. This DIDS learning model includes the prediction of unidentified assaults to manage the computational load in huge networks and boost throughput with a minimal false alarm rate. The evaluation of our suggested method against existing algorithms reveals that it identifies threats sooner than existing techniques. The 99% accuracy rate for detecting the assault has been attained while the processing time has also been decreased.

In 2021, Kumar *et al.* [8] accomplishes an innovative hybrid feature reduction approach for efficient threat detection in IoT networks. With this method, 3 different feature sets were obtained by performing feature ranking using gain ratio, RF mean, correlation coefficient, &decrease accuracy. Three well-known ML techniques, including XGBoost, RF, and KNN, are then used to the resulting reduced feature set to identify cyberattacks. The accuracy, DR, F1 score, and precision of the proposed framework have been investigated and compared with few existing techniques.

B. Review

The assault detection methodology for IoT-Fog is given in Table 1. Threats & cyberattacks have a significant impact on applications for intelligent IoT. Due to expanding threats and vulnerabilities, many conventional approaches of IoT security are presently insufficient. If the next-generation IoT system is to have a dynamic and current security system, it must make use of the capabilities of artificial intelligence, particularly ML and DL solutions. They spoke about how to exploit unstructured data to identify attack trends and protect IoT devices. They take into account the problems brought up in this field and aim to find, via future research and development, the best methods for safeguarding IoT networks & devices. Finding the right learning strategy for a certain IoT security scenario could take some time. This approach is used by different learning algorithms to provide a range of outputs depending on the caliber of the input. If the wrong learning approach is applied, the model's effectiveness, accuracy, and labor needs may be impacted. Additionally, duplicated IoT security data might result in the collection of irrelevant data and the drawing of incorrect conclusions. If the IoT data are lacking in some manner, such as by not being representative, being of low quality, having irrelevant qualities, or being too tiny for training, they may perform badly, be less accurate, or even be entirely useless. There are problems with IoT-based systems as a result of bad management. Since software developers typically strive to find out how to extract relevant data from sensors, the issue emerges. The fact that the task was completed is all that matters, regardless of how the data was collected. When there is ambiguity, hackers may find it simpler to breach a system and steal critical user data. Thus, developers must start focusing on data collecting.

Table 1 Precedingschemes Regarding Onattack Detection In Iot-Fog: Merits&Demerits

Author	Schemes	Merits	Demerits
Manimuruganet *al.* [1]	INB classifier	✓ High detection rate, recall, precision, and accuracy	➢ The typical FPR problems in the detection of anomalies were not resolved by using a novel method.
Samy *et al.* [2]	DL scheme	✓ Better detection accuracy, detection rate and response time	➢ A collaborative computing environment including Apache Spark and other datasets were not used for contrasting the suggested attack detection.
Sanjuet *al.* [3]	HHO-EFDM algorithm	✓ High accuracy, AUC-ROC, precision, recall, and F1-score	➢ Need to improve the proposed model by incorporating other types of data such as network topology and device metadata.
Mirdulaet *al.* [4]	MUD-DL framework	✓ Better accuracy, delay, CPU usage, and throughput.	➢ The proposed model was not used with advancements in the field to achieve the required efficiency..
Sudheeraet *al.* [5]	ADEPT model	✓ High F1-score, precision, recall, and accuracy	➢ Experiments with simulated IoT network traffic using the most recent pertinent data sets and realistic Mirai attack scenarios were required.
Rania *et al.* [6]	SATIDS based an improved LSTM network	✓ Higher accuracy, detection rate, F1-score, and precision	➢ Need to focus on refining the model of adopted scheme by exploiting feature selection, detect zero-day attacks on IoT system.
Bhukyaet *al.* [7]	DL model	✓ Lower computational overhead, increased throughput, and low false alarm rate.	➢ The cyber attacks as these devices do not have dedicated memory space.
Kumar *et al.* [8]	RF, KNN, and XGBoost	✓ High precision accuracy, DR, and F1 score	➢ The need to use large data processing technologies to expand the suggested architecture in a real network traffic streaming scenario.

Overview ofattack detection scheme in IoT-Fog

Security and privacy are the most important problems in IoT-Fog architecture. It suggests that merging IoT with fog and cloud computing might give IoT applications for smart cities a stronger platform. The fact that an IoT network has limited resources makes it more prone to security breaches than other types of networks. The network resource may be compromised, rendered inoperable or otherwise harmed during an attack or information collecting probe known as an exploit. Automated attack detection model is in need to detect and mitigate the attacks, and this paper intends to propose an automated attack detection system with DL technique by following the given steps:

(i) Preprocessing
(ii) Feature Extraction
(iii) Attack detection
(iv) Mitigation

❖ Preprocessing- Preprocessing: This stage process with the improved class imbalance processing to address the class imbalance issue.

❖ Feature Extraction- After that, the process of feature extraction is carried out, which involves the extraction of entropy-based features, correlation-based features, statistical features, enhanced and raw features.

❖ Attack Detection- Based on the characteristics retrieved, an attack detection process will be carried out. To accomplish this, a novel

hybrid detection model combining Bi-GRU and LSTM is presented.

❖ Mitigation- After an attack has been identified, the attacker in the network must be neutralized. An improved entropy based mitigation approach will be used in this case.

Fig. 1 determines the IoT-Fog architecture, and Fig. 2 describes proposed attack detection scheme

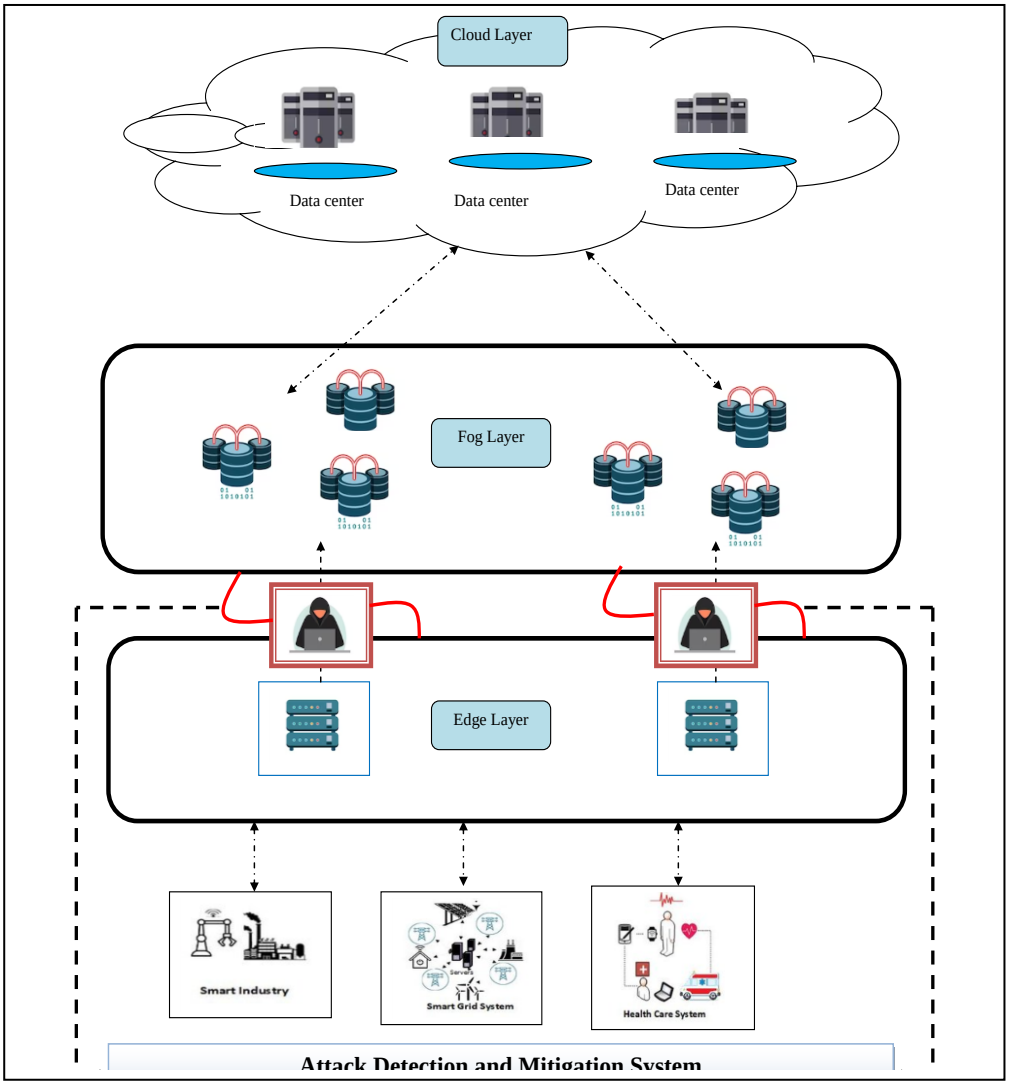


Fig. 1. Representation of IoT-Fog architecture

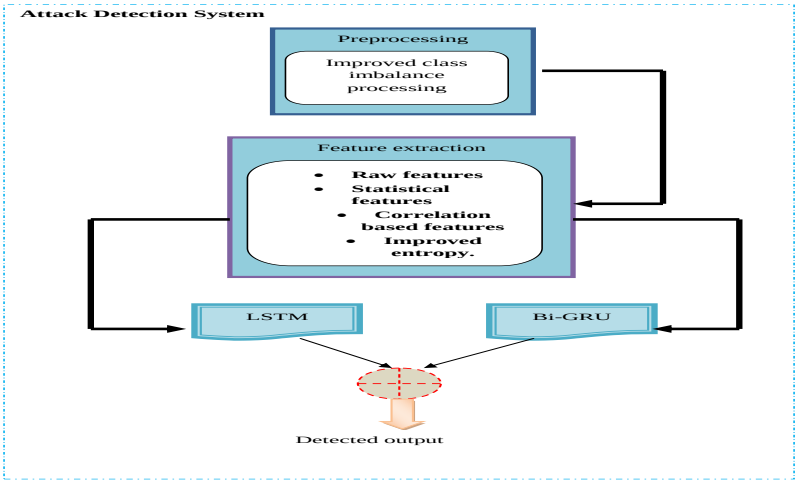


Fig. 2. Representation of proposed attack detection scheme

II. PREPROCESSING VIA IMPROVED CLASS IMBALANCE PROCESSING AND FEATURE EXTRACTION PROCESS OF PROPOSEDSCHEME

A. Preprocessing

The problem of class imbalance often arises when some classes are considerably more prevalent than others. Standard classifiers frequently become overwhelmed by the extensive classes in such situations and disregard the small ones. Hence, it is important to solve the issue, and hence an improved class imbalance processing is introduced to handle the issue in dataset Φ_{in} .

Improved class imbalance processing: An imbalanced classification data set has uneven class proportions. When the distribution of classes is not balanced, or when certain classes appear substantially more frequently than the others, this phenomenon is known as imbalanced data and commonly occurs in real-world applications.

The pseudo-code of **improved class imbalance processing** is given in Algorithm 1.

Algorithm 1: Pseudo-code of **improved class imbalance processing**			
Input: C =Entire count of classes, $\lvert D\rvert = N$ is Entire count of samples, Training set D = each labels length, D_i =data			
Output: $I_{resam} = \text{int}\left(\frac{N}{C}\right)$			
For $i \leftarrow 1$ to C do			
	If $\lvert D_i\rvert < I_{resam}$, then		
		D_i' = smote (D_i , I_{resam})	
	End if		
		If $\lvert D_i\rvert > I_{resam}$, then	
		G_k = Improved kernel K-means (D_i , C), $k = 1,2,...,C$	
			For $k \in 1$ to C do

			G'_k = resample (G_k , $\frac{I_{resam}}{C}$)
			End for
			$D'_i = Concatenate(G'_k)$
		End if	
	$D' = Concatenate(D'_i)$		
End for			
Return D'			

Here, improved kernel K-means is used for cluster the data. The adaptation is done in kernel, where the sigmoid kernel is used. Three types of kernals are used like polynomial kernel K_P , Gaussian kernel K_g , and sigmoid kernel K_S . Thus, the proposedkernal calculation is given as in Eq. (1).

$$Kernal = \frac{K_p * w_1 + K_g * w_2 + K_S * w_3}{3} \quad (1)$$

Here, w is the weight parameter calculated by sinusoidal map x_{j+1} in Eq. (2).

$$x_{j+1} = a x_i^2 \sin(\pi x_j) \quad (2)$$

Here, $j = 3$ number of kernals, $a = 2.3$, $x_1 \to [0, Mean(K_P)]$, $x_2 \to [0, Mean(K_g)]$, and $x_3 \to [0, Mean(K_S)]$.Here, the preprocessed data is labeled as Φ_{PRE} .

B. Feature extraction

Extracting handcrafted features is very important as they give addition information from the original data given. With the prerocessed data Φ_{PRE} , it extracts certainfeature that includes:

- ✓ Raw data features
- ✓ Improved Entropy based feature
- ✓ Statistical features
- ✓ Correlation based features

(i) Raw data:

Raw data is sometimes referred to as atomic data, original data, and source data. It is information that hasn't been prepared for use. The raw features are specified as $\Re$.

(ii) Improved Entropy based features:

Entropy is uncertainty/ randomness in the data, the more the randomness the higher will be the entropy. Information gain uses entropy to make decisions. It is calculated to determine the data set uncertainty level. In addition, Eq. (3) determines the entropy formulation.

$$E(y) = -\sum_{l=1}^{q} p(y_l) * \log_2(p(y_l)) \quad (3)$$

Here, $y \to$ data value, $p(y_l) \to$ probability of occurrence of value l .

As per improved entropy, the Shannon entropy is used.

$$E'(y) = -\sum_{l=1}^{q} p(y_l) * \log_2(p(y_l)) + Mean(WMI) \quad (4)$$

Where, WMI is the weighted mutual information it is determined below.

$$WMI(X:Y) = [H(X) - H(X/Y)] * W_z \quad (5)$$

Here, $H(X)$ is the entropy of X, and $H(X/Y)$ is the conditional entropy.

$$W_z = \sin(\pi z) \quad (6)$$

Where, z indicates the $Mean[H(X)]$.

(i) Statistical features:

Statistical features are those features of the dataset that can be defined and calculated via statistical analysis. The statistical features include mean, median and SD are extracted.

*Mean (Average):*The sum of all values for the entire count is used to calculate the mean.

$$Mean = \overline{A} = \frac{1}{B} \sum_{b=1}^{B} A_b \quad (7)$$

In Eq. (7), $A \rightarrow$ observed value, $B \rightarrow$ number of values, & $\overline{A} \rightarrow$ mean.

Median: The center values of a dataset are chosen as the median in this method. The median of a dataset is determined as the mean of the two values in Eq. (8) when the dataset's center contains two values.

$$Median(A) = \begin{cases} A\left(\frac{B}{2}\right) & if B \text{ is odd} \\ \frac{A\left(\frac{B-1}{2}\right) + A\left(\frac{B+1}{2}\right)}{2} & \text{if } B \text{ is even} \end{cases} \quad (8)$$

SD: A group of dispersion values or the level of variability is evaluated. Eq. (9) is utilized to calculate the SD.

$$\sigma = \sqrt{\frac{1}{B-1}\sum_{b=1}^{B}\left(A_b - \overline{A}\right)^2} \quad (9)$$

In Eq. (10), the overall extracted features ϑ are provided.

$$\vartheta = \left[\overline{A} \;\; Median(A) \;\; \sigma\right] (10)$$

(ii) Correlation feature

The most popular method for determining a linear correlation is to use a correlation-based feature .

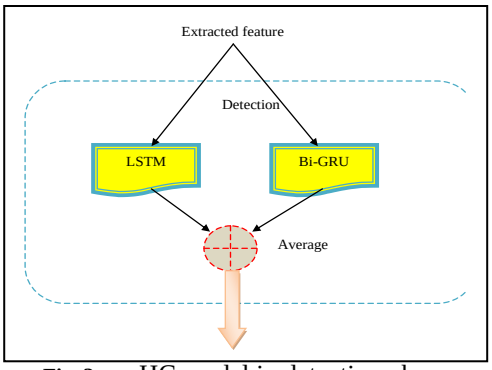


Fig. 3. HC model in detection phase

coefficient. Typically, the Pearson correlation coefficient is calculated using Eq. (11).

$$Corr(u,v) = \frac{\sum\left(u_e - \overline{u}\right)\left(v_e - \overline{v}\right)}{\sqrt{\sum\left(u_e - \overline{u}\right)^2\left(v_e - \overline{v}\right)^2}} \quad (11)$$

Here, $u_e \rightarrow$ value of u variable in a sample, $v_e \rightarrow$ value of v variable in a sample, $\overline{u} \rightarrow$ mean of the values of the u-variable, $\overline{v} \rightarrow$ mean of the values of the v-variable.

In Eq. (12), the overall features $\aleph$ are specified.

$$\aleph = \left[\Re \;\; E'(y) \; \vartheta \;\; Corr(u,v)\right] (12)$$

III. Detection via Hybrid Classifiers and Improved Mitigation Process

A.Attack Detection

Hybrid classifier (HC): According to the proposed work, HC is the combined form of two DL models namelyLSTM &Bi-GRU that trains with the extracted feature set $\aleph$. The procedure is as follows: The features $\aleph$ are given as the input into the Bi-GRU and LSTM classifiers. To get the final detected result, the intermediate results from the two classifiers are averaged. Fig. 2 represents the architecture of detection procedure.

LSTM:LSTM is an RNN extension that was created to get around the long-term dependence problem unlike RNN, it can retain data for a very long time.The cell state is the main element of an LSTM. The gates are employed to protect the cell state by utilizing the sigmoid function to add or remove information from it. As comparable layers for input and output $\left(O_t, M_{t-1}, L_{t-1}\right)$ and $\left(L_t, C_t\right)$, think of the cell state and hidden state as M and L. The outputs, input, and forget gates are chosen as necessary U_t, V_t, Z_t at the time t. Most of the data is filtered.

$$Z_t = \xi\left(\hat{W}_m O_t + h_m + \hat{W}_n L_{t-1} + h_n\right) \quad (13)$$

Both a weight matrix as well as a bias parameter are expressed as $\left(\hat{W}_n, h_n\right)$and $\left(\hat{W}_m, h_m\right)$. The gate activation function is selected to use the sigmoid operation (ξ). The input gate is used by the LSTM cell to combine the appropriate data as determined by Equations (14), (15), and (16). Here, bias settings, weight matrices, $\left(\hat{W}_{\hat{x}}, h_{\hat{x}}\right)$and $\left(\hat{W}_{\hat{y}}, h_{\hat{y}}\right)$are used to map the input and hidden layers to the cell gate. For

the hidden and input layers correspond to Q_t, the weight and bias are $\left(\hat{W}_c, h_c\right)$and $\left(\hat{W}_d, h_d\right)$.

$$R_t = \tanh\left(\hat{W}_{\hat{y}} O_t + h_{\hat{y}} + \hat{W}_{\hat{x}} L_{t-1} + h_{\hat{x}}\right) \quad (14)$$

$$Q_t = \xi\left(\hat{W}_c O_t + h_c + \hat{W}_d L_{t-1} + h_d\right) \quad (15)$$

$$M_t = Z_t M_{t-1} + Q_t O_t \quad (16)$$

In Eq. (17) and (18), the LSTM obtain the hidden layer (output) from the output gate.

$$U_t = \xi\left(\hat{W}_s O_t + h_s + \hat{W}_r \tilde{H}_{t-1} + h_r\right) \quad (17)$$

$$L_t = U_t \tanh\left(M_t\right) \quad (18)$$

The weight and bias parametersare $\left(\hat{W}_r, h_r\right)$& $\left(\hat{W}_s, h_s\right)$to map the hidden & input layers to U_t their respective locations. DT_{LSTM} specifies the output of LSTM.

Bi-GRU:A paradigm for sequence processing called BiGRU uses two GRUs. The input is processed by one in a forward manner and the other in a backward way. With only input and forget gates, it is a two-way RNN. Where $F = \left(f_1, f_2, ..., f_{\tilde{n}}\right)$and f the current word concatenating vector for determing the forward GRU. The forward GRU is determined in the manner described below:

$$\tilde{A} = \kappa\left(\tilde{w}_{z\tilde{A}} z_{\tilde{t}} + \tilde{w}_{\tilde{r}\tilde{A}} \tilde{r}_{\tilde{t}-1} + \tilde{c}_{\tilde{A}}\right) \quad (19)$$

$$K = \kappa\left(\tilde{w}_{zK} z_{\tilde{t}} + \tilde{w}_{\tilde{r}K} \tilde{r}_{\tilde{t}-1} + \tilde{c}_K\right) \quad (20)$$

$$\tilde{L} = \tanh\left(\tilde{w}_{z\tilde{L}} z_{\tilde{t}} + \tilde{w}_{\tilde{r}\tilde{L}}\left(\tilde{A}\Theta\tilde{r}_{\tilde{t}-1}\right) + \tilde{c}_{\tilde{L}}\right) \quad (21)$$

$$\tilde{r} = \left(1-K\right)\Theta\tilde{r}_{\tilde{t}-1} + K\Theta\tilde{L} \quad (22)$$

Here, $\tilde{c}$ and $\tilde{w} \rightarrow$ bias vector & weight matrix, $\tilde{r}_{\tilde{t}}$ $\rightarrow$ the hidden state $\tilde{t}$, $\Theta \rightarrow$ element-wise multiplication, $\kappa \rightarrow$ sigmoid function,and $z_{\tilde{t}} \rightarrow$ source word vector at $\tilde{t}$. The outcomes of the forward and backward GRUs $\rightarrow \vec{r}_{\tilde{A}}$ & $\overleftarrow{r}_{\tilde{A}}$. Thus, the Bi-GRU result $\rightarrow DT_{\tilde{A}}^{Bi-GRU} = \left[\vec{r}_{\tilde{A}}, \overleftarrow{r}_{\tilde{A}}\right]$.

The overall detected outcome $\rightarrow O*$ is given in Eq. (23).

$$O* = \frac{DT_{\tilde{A}}^{Bi-GRU} + DT_{LSTM}}{2} \quad (23)$$

B. Enhanced Mitigation process

Mitigation is the procedure of removing attacker node from the network. Considering the architecture with set of nodes including attacker and non-attacker nodes. The process of finding the attacker node based on the behaviorof the nodes.

- For each node, the entropy is calculated based on corresponding features.
- Setting a threshold value 't'
- If the obtained entropy is high than the threshold value 't', it is considered to benon-attacker node.
- If the entropy is low than the threshold value 't', it is considered as the attacker nodes. Thereby, it is mandatory to remove the corresponding nodes during this mitigation process.

The proposed entropy is determined in accordance with the enhanced mitigation approach using Eq. (24).

$$En = \frac{1}{1-\beta}\log_2 \sum_{\hat{i}=1}^{\hat{n}} J_{\hat{i}}^{\beta} + Normalized\left(Corr(u,v) + \gamma_o\right) \quad (24)$$

Where, $Corr(u,v)$ is the correlation coefficient as per Eq. (11). Here, $\gamma_o \rightarrow$ encumbered value. Where, $\varepsilon_{\hat{i}}$ is calculated using zig zag map, and $\tilde{F}_{\hat{i}}$ is feature value.

$$\gamma_o = \frac{\sum \varepsilon_{\hat{i}} \tilde{F}_{\hat{i}}}{\sum \varepsilon_{\hat{i}}} \quad (25)$$

$$\varepsilon_{\hat{i}} = \begin{cases} -\hat{m}\left(\tilde{x}_{\tilde{a}} + \dfrac{2}{|\hat{m}|}\right), & -1 < \tilde{x}_{\tilde{a}} \le \dfrac{1}{|\hat{m}|} \\ \hat{m}\tilde{x}_{\tilde{a}}, & -\dfrac{1}{|\hat{m}|} < \tilde{x}_{\tilde{a}} \le \dfrac{1}{|\hat{m}|} \\ -\hat{m}\left(x_n + \dfrac{2}{|\hat{m}|}\right), & \dfrac{1}{|v|} < \tilde{x}_{\tilde{a}} \le 1 \end{cases} \quad (27)$$

Results & Discussions

C. Simulation setup

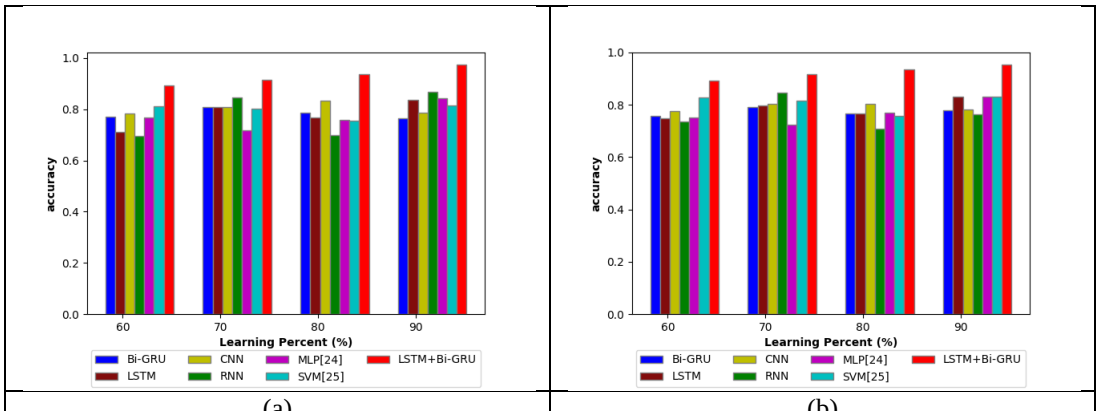
The PYTHON simulation tool was used to evaluate the performance of the LSTM + Bi-GRU model to prior approaches used for attack detection and mitigation in the IoT-Fog framework. Additionally, data from [21], [22], and [23] were collected. With varying learning percentages of 60, 70, 80, and 90, the LSTM + Bi-GRU model was evaluated against prior approaches such as Bi-GRU, LSTM, CNN, RNN, MLP [24], and SVM [25] models.

D. Dataset Description

This work includes 5 datasets. Here, the first three datasets are UNSW 2,3,4 from [21], fourth dataset from [22], and fifth dataset from [23]. The row and column size includes Dataset1 = 7342 X 45, Dataset2 = 8292 X 45, Dataset3 = 7750 X 45, Dataset4 = 6000 X 79, and Dataset5 = 11730 X 46. The attributes are "Dst Port, Protocol, Timestamp Flow, Duration, Tot FwdPkts, Tot BwdPkts, TotLenFwdPkts, TotLenBwdPkts, FwdPkt Len Max, FwdPkt Len Min, FwdPkt Len Mean, FwdPkt Len Std, BwdPkt Len Max, BwdPkt Len Min, BwdPkt Len Mean, BwdPkt Len Std Flow, Byts/s, Flow Pkts/s, Flow IAT Mean, Flow IAT Std Flow, IAT Max, Flow IAT Min, Fwd IAT Tot, Fwd IAT Mean, Fwd IAT Std, Fwd IAT Max, Fwd IAT Min, Bwd IAT,TotBwd, IAT Mean, Bwd IAT Std, Bwd IAT Max, Bwd IAT Min, Fwd PSH Flags, Bwd PSH Flags, Fwd URG Flag, Bwd URG Flags, Fwd Header Len, Bwd Header Len, FwdPkts/s, BwdPkts/s, Pkt Len Min, Pkt Len Max, Pkt Len Mean, Pkt Len Std, Pkt Len Var, FIN Flag Cnt, SYN Flag Cnt, RST Flag Cnt, PSH Flag Cnt, ACK Flag Cnt, URG Flag Cnt, CWE Flag Count ECE, Flag Cnt Down/Up Ratio, Pkt Size Avg, Fwd Seg SizeAvg, Bwd Seg Size Avg, FwdByts/b Avg, FwdPkts/b Avg, FwdBlk Rate Avg, BwdByts/b Avg, BwdPkts/b Avg, BwdBlk Rate Avg, SubflowFwdPkts, SubflowFwdByts, SubflowBwdPkts, SubflowBwdByts, Init Fwd Win Byts, Init Bwd Win Byts, Fwd Act Data Pkts, Fwd Seg Size Min, Active Mean, Active Std Active Max, Active Min Idle, Mean, Idle Std Idle Max, Idle Min, Label, flow_duration Header_Length, Protocol Type, Duration, Rate, Srate, Drate, fin_flag_number, syn_flag_number, rst_flag_number,psh_flag_number, ack_flag_number, ece_flag_number, cwr_flag_number, ack_count,syn_count,fin_count, urg_count, rst_count, HTTP, HTTPS,DNS, Telnet, SMTP, SSH, IRC, TCP, UDP, DHCP, ARP, ICMP, IPv LLC, Tot sum, Min, Max, AVG, Std, Tot size, IAT, Number, Magnitue, Radius, Covariance, Variance, Weight, and label".

E. Positive metricsevaluation of LSTM + Bi-GRU to preceding models

In Fig. 4 to Fig. 7, the LSTM + Bi-GRU scheme is evaluated to earlier approaches like Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] model for positive metrics (Precision, Specificity, Sensitivity, & Accuracy) for 5 datasets. Additionally, for dataset 1 in Figure 3, the LSTM + Bi-GRU approach at a learning rate of 90%produced maximal accuracy (0.96), while other strategies including the Bi-GRU, LSTM, CNN, RNN, MLP, and SVM models each achieved a lower accuracy number. Similarly, employing HC and increased features for dataset 2 in Fig. 5, the LSTM + Bi-GRU model sensitivity produces superior results (0.90) in 60% learning percentages. But the Bi-GRU, LSTM, CNN, RNN, MLP, and SVM models have lower sensitivity levels. Further, the LSTM + Bi-GRU model outperformed earlier models including the Bi-GRU, LSTM, CNN, RNN, MLP [24], and SVM [25] model for dataset 3 in terms of attack detection specificity (about 0.92 at 80% learning percentage). Similarly, the LSTM + Bi-GRU technique outperformed the Bi-GRU, LSTM, CNN, RNN, MLP, and SVM [24, 25] models with better attack detection for dataset 4in IoT-Fog holds maximum precision at a learning percentage of 70%, in Fig.7.The proposed scheme holds maximum accuracy, sensitivity, specificity, and precision at learning percentages of 90% for dataset 5 thanks to improved features and the HC (Bi-GRU & LSTM) concept; however, the existing methods, including the LSTM + Bi-GRU strategy outperformed the Bi-GRU, LSTM, CNN, RNN, MLP [24], and SVM [25] model each attained minimal values. It is feasible to get a better recommended approach for attacks detection in IoT-Fog architecture by employing the improved features, & HC.



(a) (b)

1

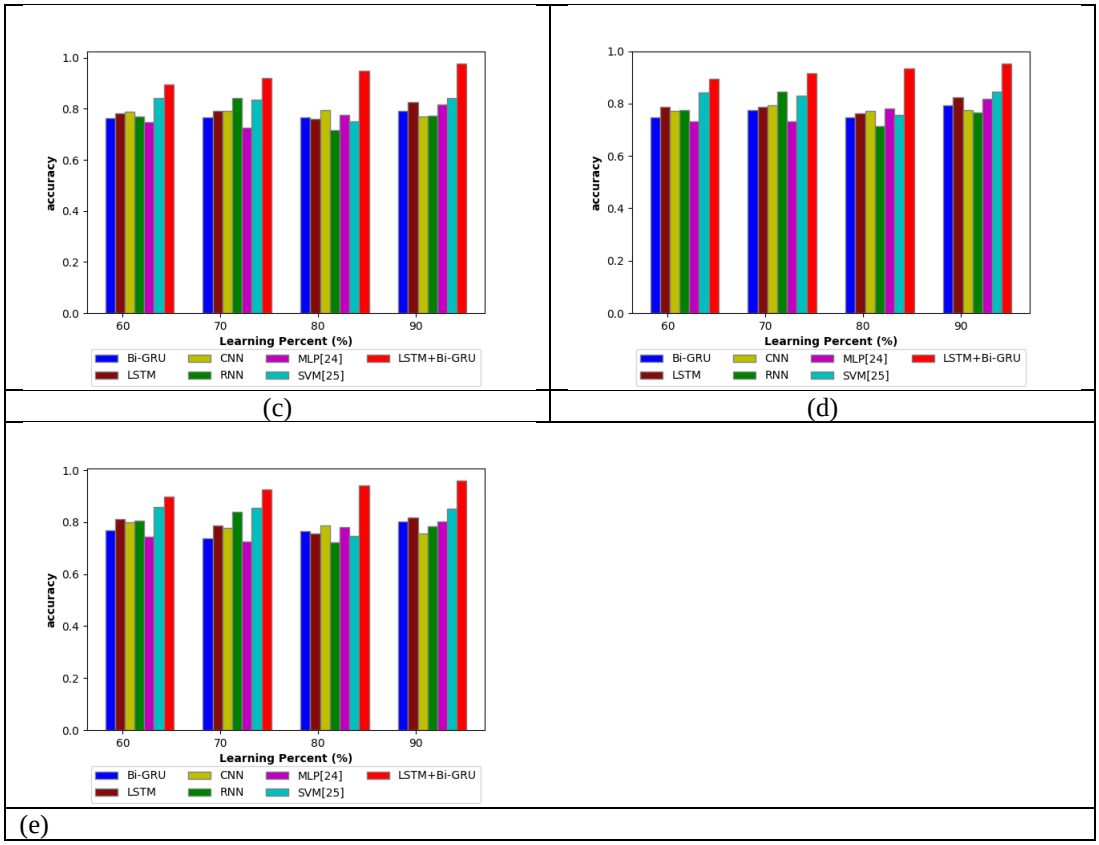
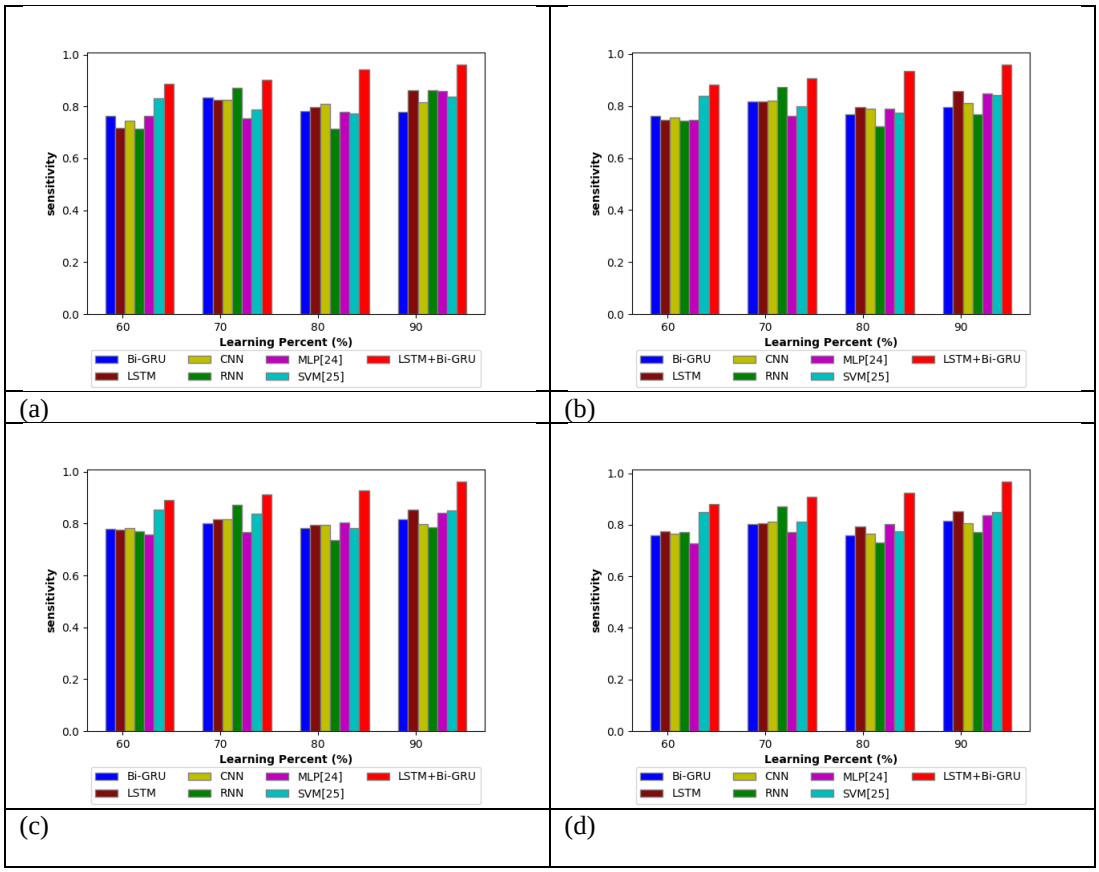
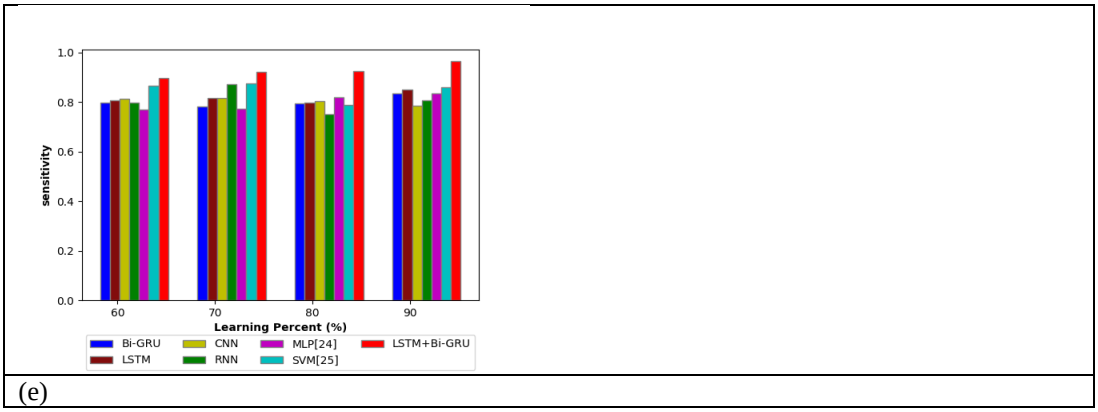


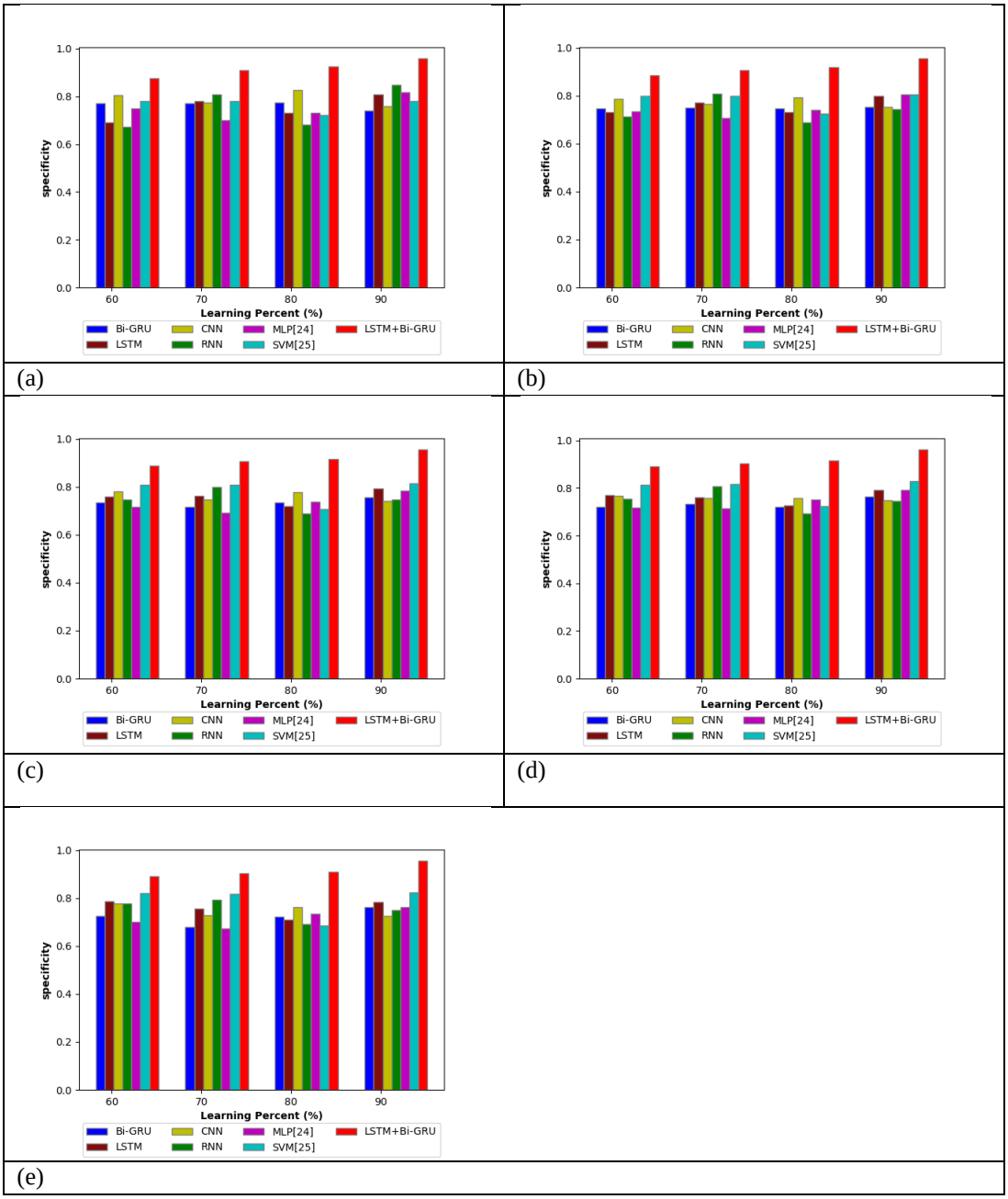
Fig. 4. Accuarcy metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5





(e)

Fig. 5. Sensitivity metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5



(a) (b)

(c) (d)

(e)

Fig. 6. Specificity metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5

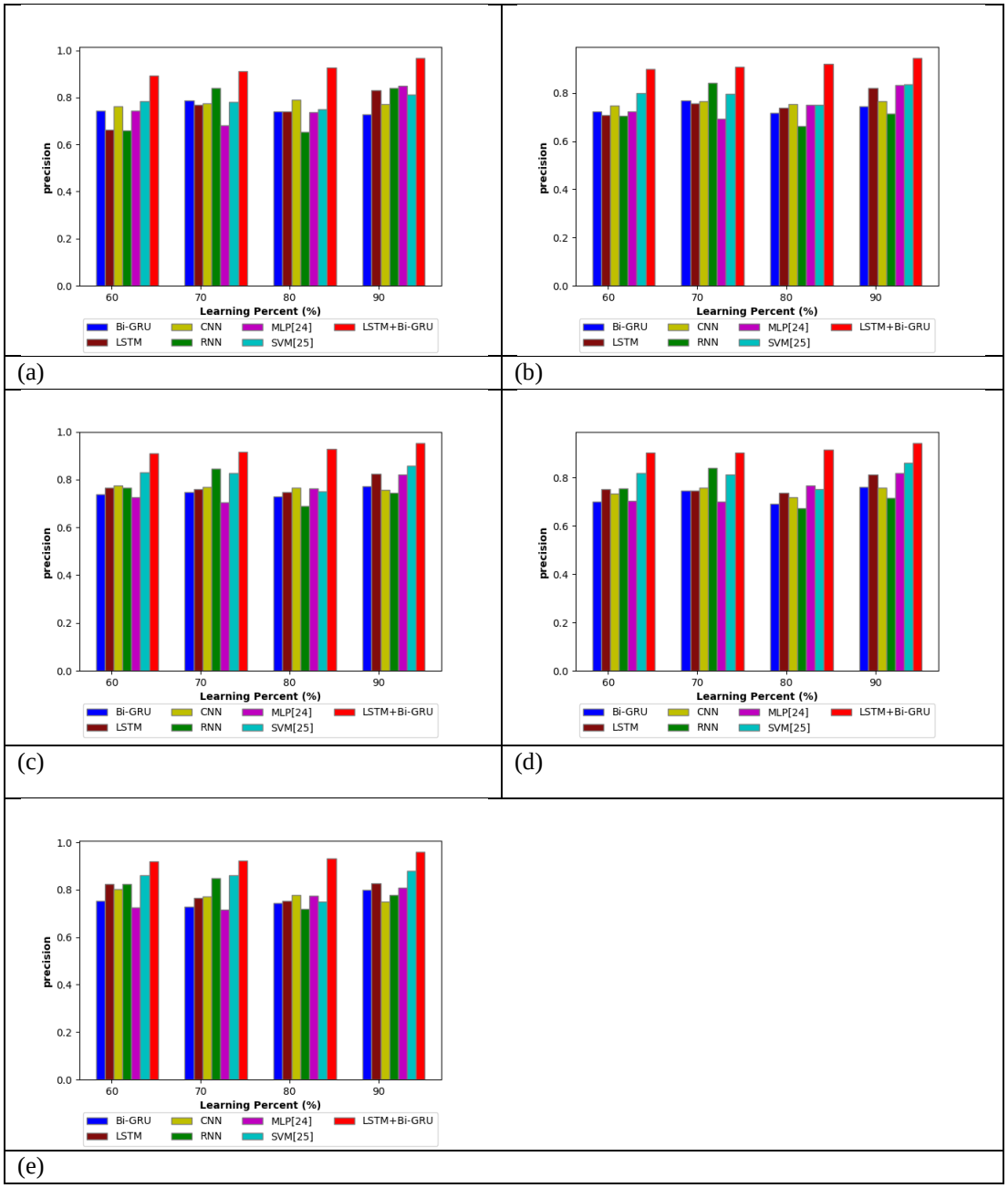


Fig. 7. Precision metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5

F. Negativemetrics evaluation of LSTM + Bi-GRU to preceding models

The LSTM + Bi-GRU scheme is determined to traditional schemes like Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] model, respectively based on negative measures (FNR, & FPR) for 5 datasets is represented in Fig. 8& Fig. 9. While compared to other methods such as Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] model in Fig. 7,the LSTM + Bi-GRU model produced improved attack detection results with a minimal FPR of 0.09 at an 80% learning rate for dataset 3. Additionally, the LSTM + Bi-GRU model outperforms other models for dataset 5 in Fig. 9 with the lowest FNR value (0.04) and highest outcomes for attack detection and mitigation at a learning rate of 90%. As a result, the chosen LSTM + Bi-GRU method has shown betterment than other models for attack detection and mitigation while reducing the negative error value.

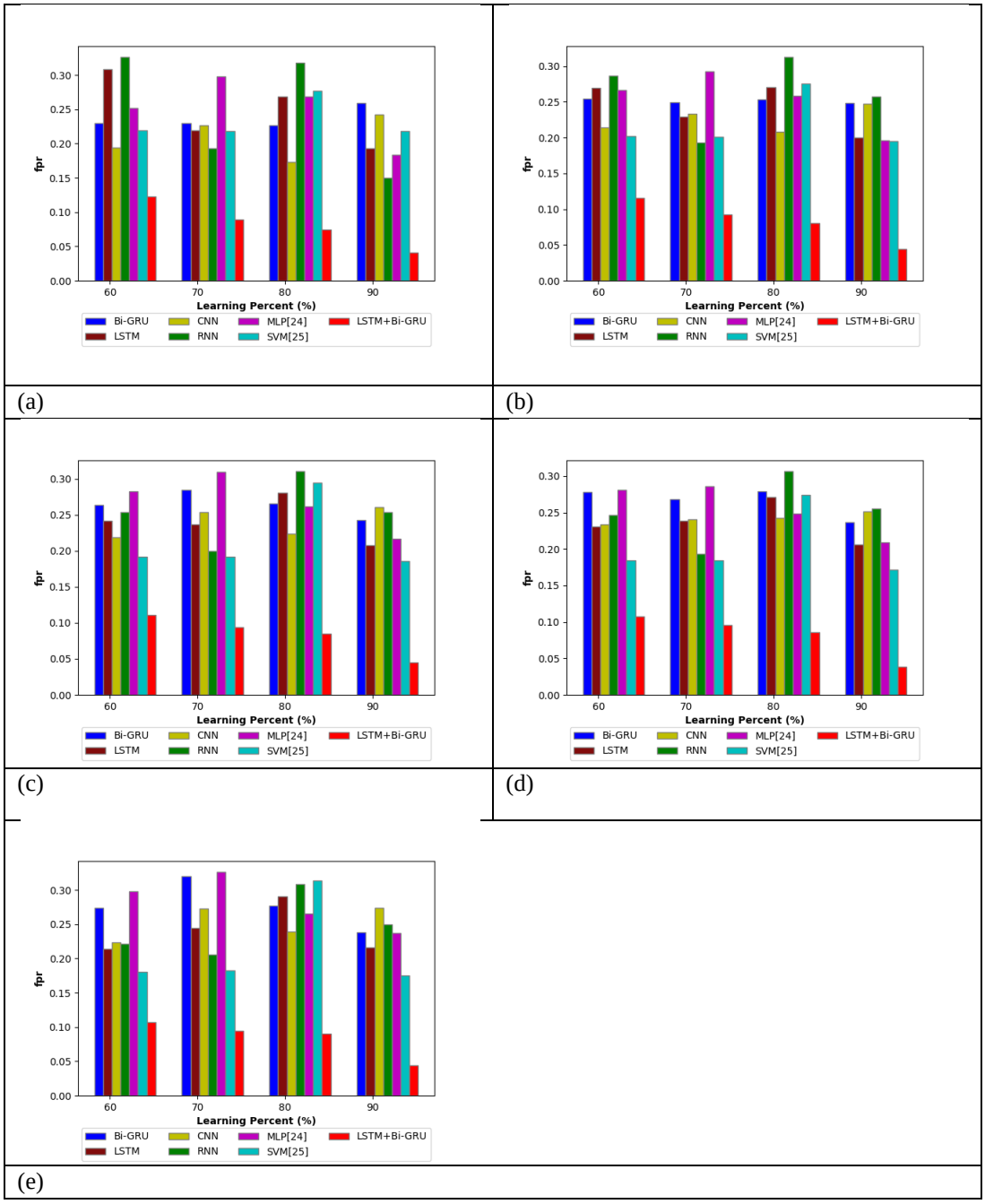
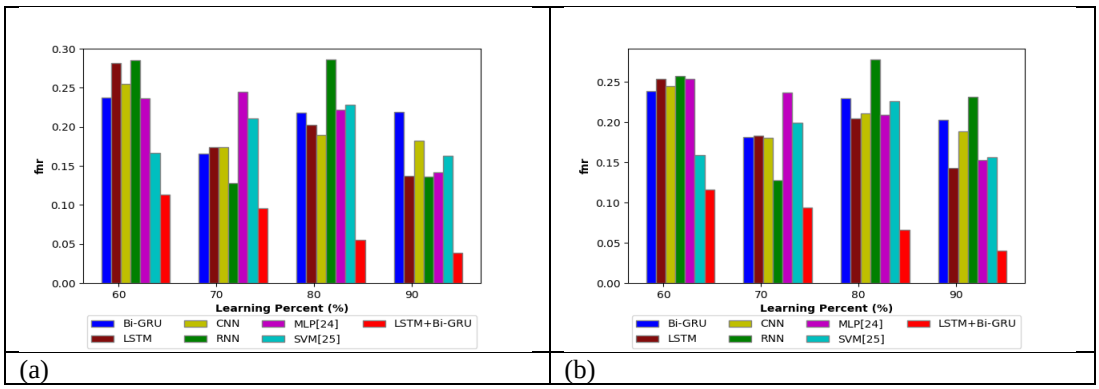


Fig. 8. FPR metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5



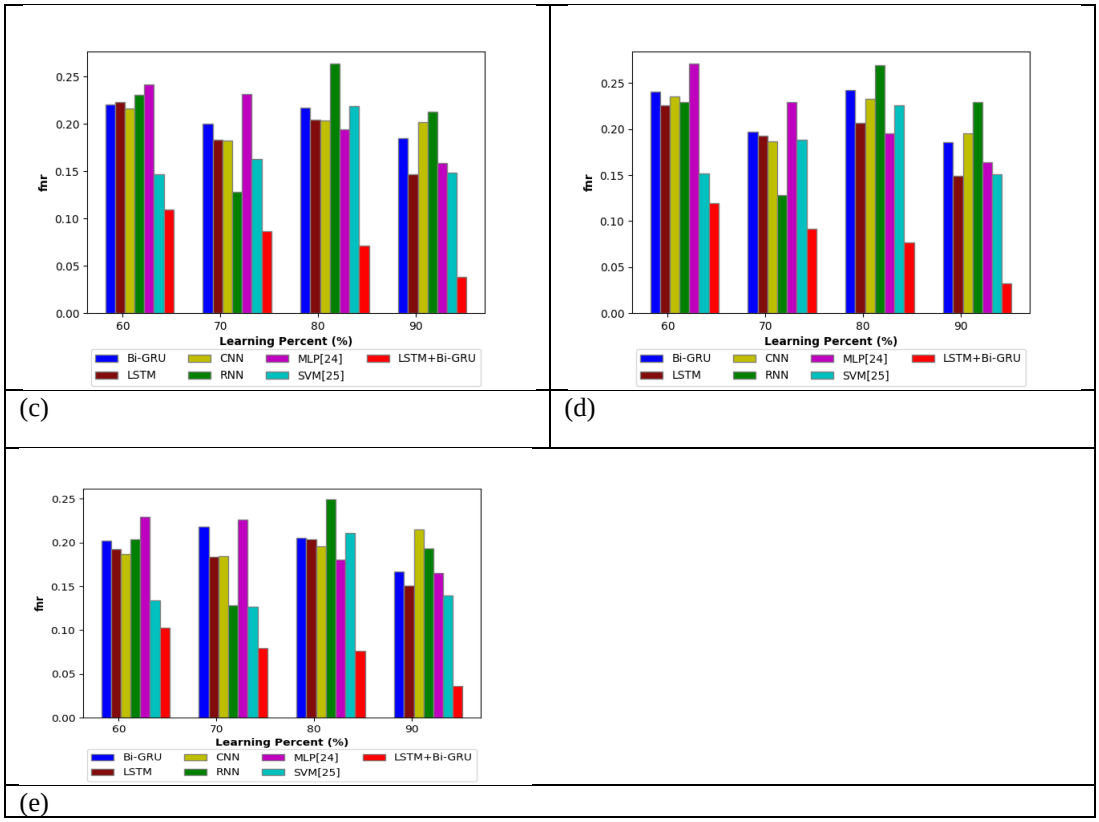
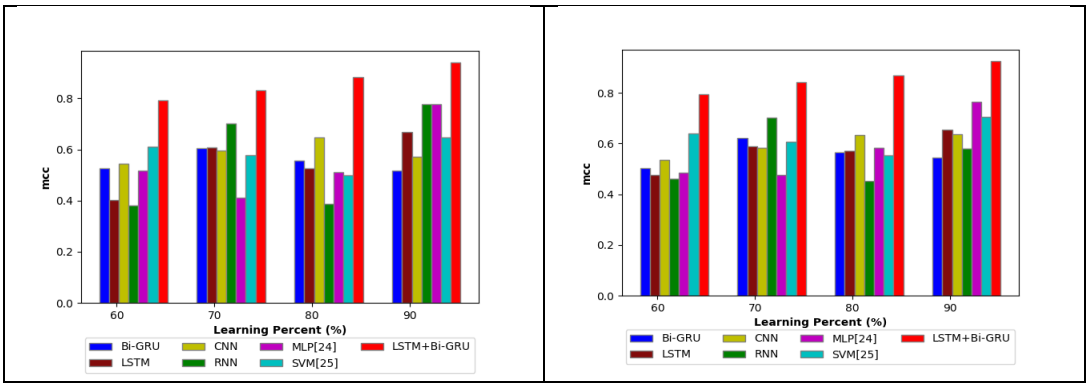


Fig. 9. FNR metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5

G. Other metrics evaluation of LSTM + Bi-GRU to preceding models

The LSTM + Bi-GRU scheme is evaluated to traditional schemes like Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] model, correspondingly for other measures (MCC, NPV &F-measure) for 5 datasets is shown in Fig. 10, 11 and 12 Furthermore, at learning percentage 90% for dataset 1, the NPV of the LSTM + Bi-GRU strategy achieves improved attack detection performance of 0.96, which is much higher than earlier schemes like Bi-GRU, LSTM, CNN, RNN, MLP [24], and SVM [25] model. The LSTM + Bi-GRU has attained maximum MCC (approximately 0.95), outperforming other methods like the Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] model in terms of attack detection results in 90% of learning percentage for dataset 3 in Fig. 11.When the learning rate reached 70%, the LSTM + Bi-GRU scheme employed enhanced features and the HC concept to get a higher F-measure for detecting assaults in IoT-Fog. The suggested approach considerably enhances outcomes for various metrics for identifying attacks in IoT-Fog architecture as a consequence of the additional features and HC.



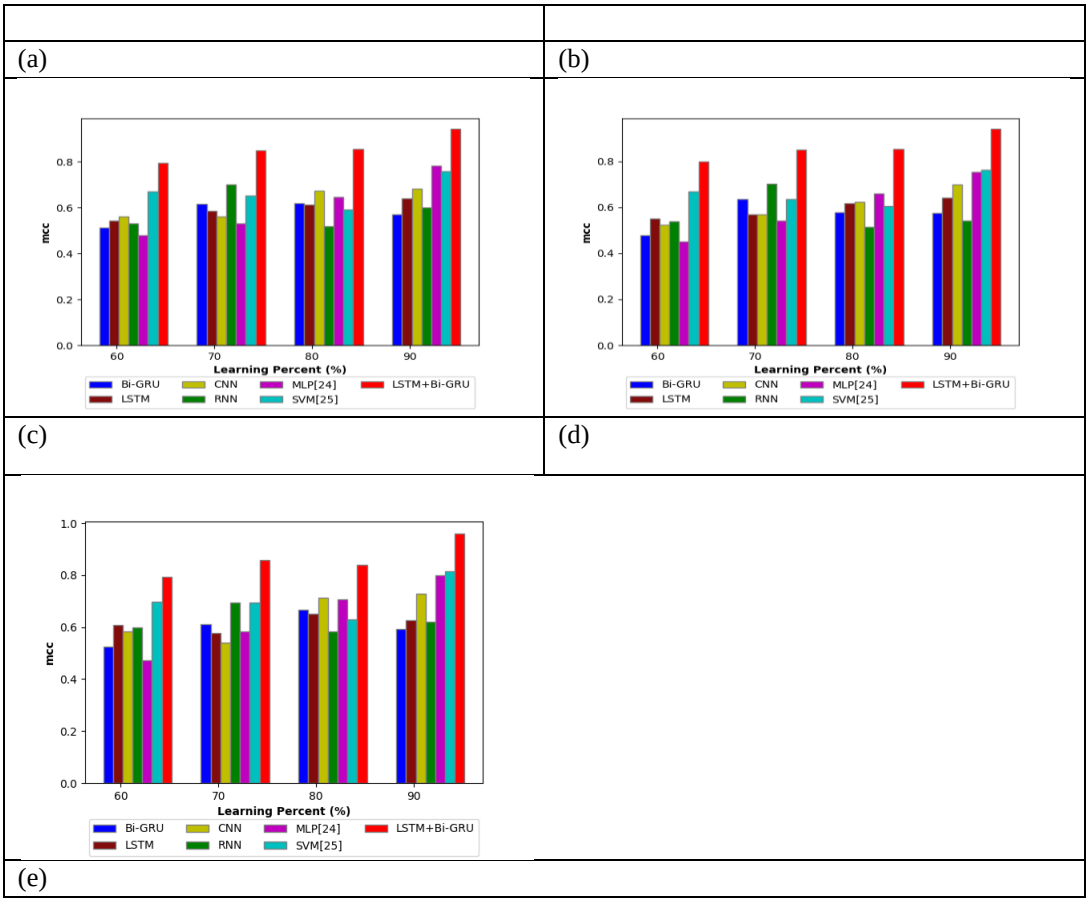
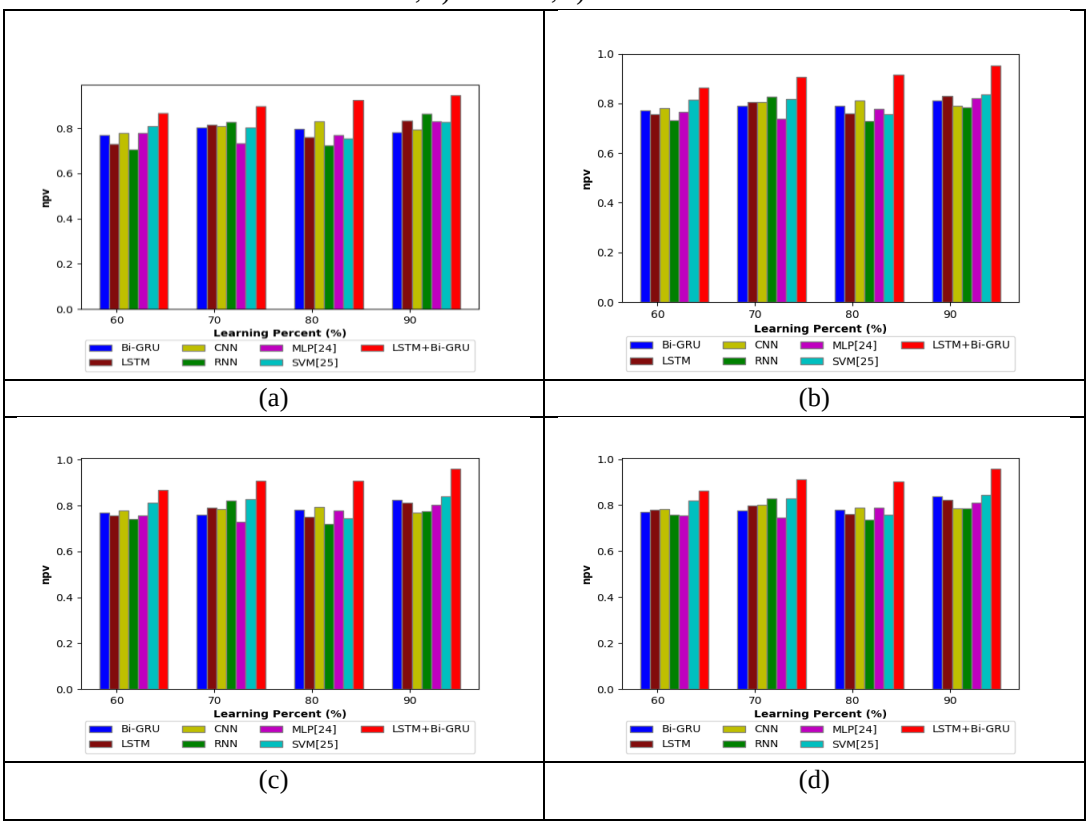


Fig. 9. MCC metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5



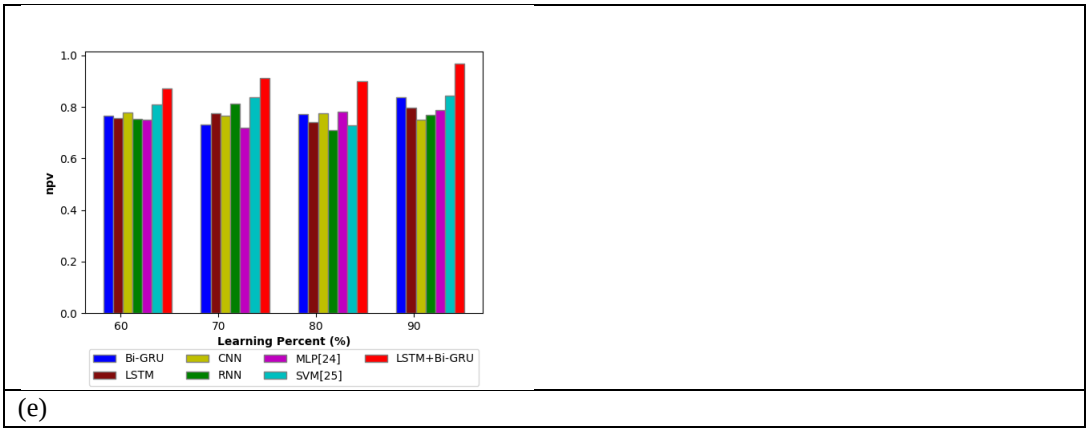


Fig. 10. NPV metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset 5

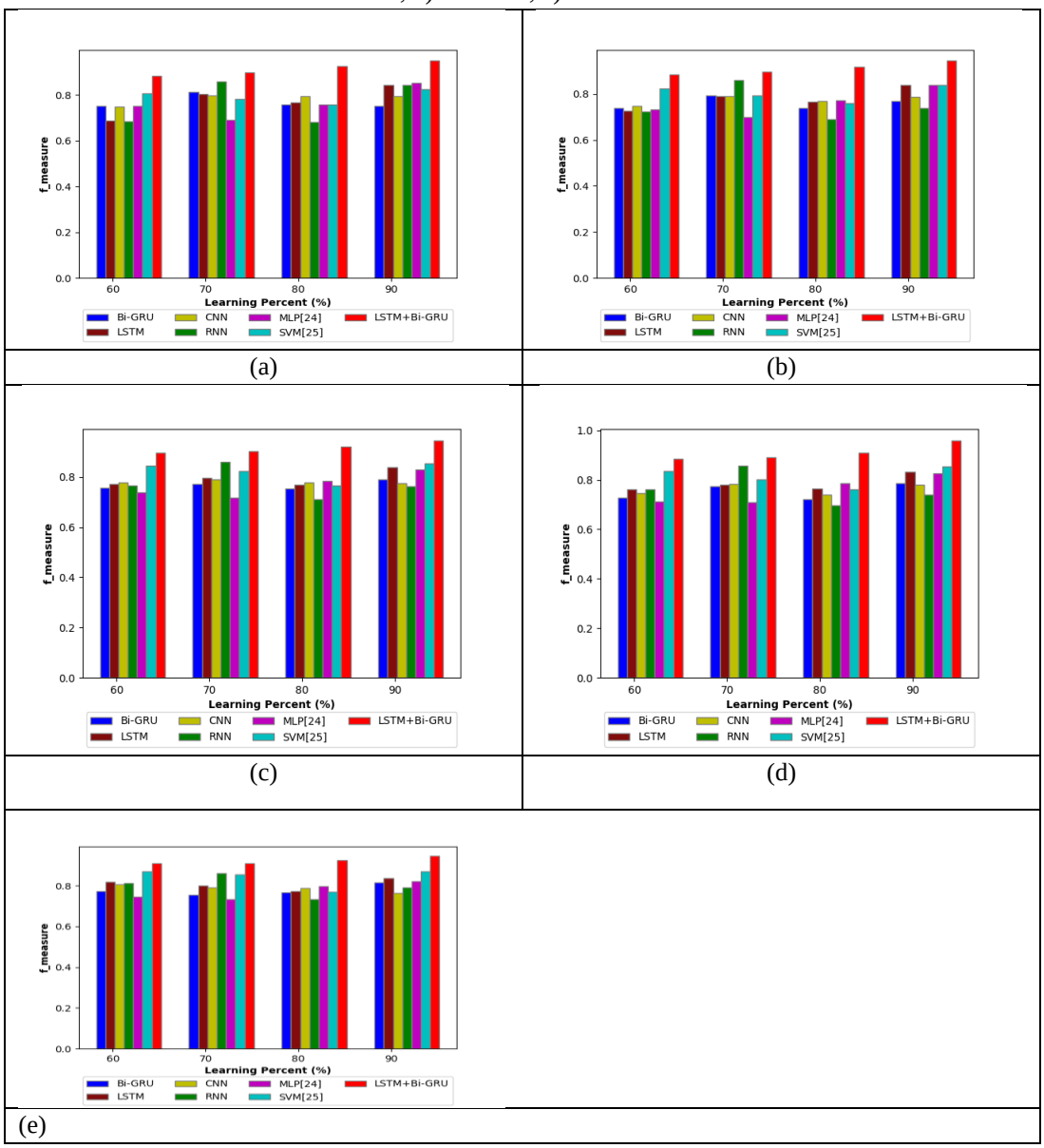


Fig. 11. F-measure metrics evaluation of LSTM + Bi-GRU to preceding models for a) dataset 1, b) dataset 2, c) dataset 3, d) dataset 4, e) dataset

H. Ablation Study of LSTM + Bi-GRU to preceding models

Table II determines the ablation study of adopted scheme (LSTM + Bi-GRU) over model without feature extraction, mode without conventional entropy, and model without class imbalancing, respectively for 5 datasets. Additionally, the selected strategy (LSTM + Bi-GRU) achieves better results (0.933) for dataset 2 than models without feature extraction, modes without conventional entropy, and models without class imbalancing, respectively. To detect attacks using a hybrid classifier and improved feature concepts, the MCC value of the adopted scheme (LSTM + Bi-GRU) achieves the highest results (0.8542); however, models without feature extraction, modes without conventional entropy, and models without class imbalancing, respectively, achieve lower MCC values. Furthermore, in dataset 5, the chosen strategy (LSTM + Bi-GRU) achieves a minimal FNR value of 0.076 and performs better at identifying attacks than models without feature extraction, modes without conventional entropy, and models without class imbalancing, respectively. This demonstrates that the chosen strategy outperforms bestattacksdetection using HC in IoT-Fog.

Table 1 Ablation Study Of Proposed Vs Extant Models

Dataset 1				
Metrics	**Proposed model**	**Model without feature extraction**	**Model without conventional entropy**	**Model without class imbalancing**
MCC	0.884119	0.589122	0.701579	0.443961
NPV	0.925593	0.808255	0.827934	0.737068
FPR	0.074702	0.230112	0.193207	0.295611
Accuracy	0.936967	0.805809	0.845651	0.721822
FNR	0.055584	0.177175	0.12781	0.240801
Sensitivity	0.944416	0.822825	0.87219	0.759199
Specificity	0.925298	0.769888	0.806793	0.704389
Precision	0.927956	0.769971	0.841293	0.687481
F-measure	0.924707	0.795076	0.858547	0.695725
Dataset 2				
Metrics	**Proposed model**	**Model without feature extraction**	**Model without conventional entropy**	**Model without class imbalancing**
NPV	0.914964	0.808545	0.779309	0.759599
FPR	0.080107	0.220614	0.252875	0.27563
FNR	0.06607	0.195983	0.202925	0.222884
Sensitivity	0.93393	0.804017	0.797075	0.777116
Specificity	0.919893	0.779386	0.747125	0.72437
Accuracy	0.936356	0.80296	0.776568	0.747488
Precision	0.922695	0.760511	0.752244	0.722105
F-measure	0.916727	0.779285	0.774031	0.736167
MCC	0.868228	0.607989	0.576562	0.530287
Dataset 3				
Metrics	**Proposed model**	**Model without feature extraction**	**Model without conventional entropy**	**Model without class imbalancing**
NPV	0.906766	0.7882	0.778323	0.759193
FPR	0.084953	0.247816	0.253169	0.278849
FNR	0.07117	0.207842	0.198986	0.213704
Sensitivity	0.92883	0.792158	0.801014	0.786296
Specificity	0.915047	0.752184	0.746831	0.721151
Accuracy	0.949777	0.781379	0.778532	0.753547
Precision	0.927831	0.744014	0.758584	0.735779

F-measure	0.920439	0.765587	0.77884	0.752049
MCC	0.854222	0.591926	0.60731	0.594086
Dataset 4				
Metrics	**Proposed model**	**Model without feature extraction**	**Model without conventional entropy**	**Model without class imbalancing**
NPV	0.904336	0.788111	0.768656	0.762487
FPR	0.085511	0.239703	0.250793	0.276067
FNR	0.076556	0.191232	0.188596	0.204686
Sensitivity	0.923444	0.808768	0.811404	0.795314
Specificity	0.914489	0.760297	0.749207	0.723933
Accuracy	0.935745	0.791739	0.784511	0.757497
Precision	0.917433	0.767129	0.779998	0.73874
F-measure	0.908747	0.785894	0.795372	0.753034
MCC	0.852338	0.640605	0.643324	0.623098
Dataset 5				
Metrics	**Proposed model**	**Model without feature extraction**	**Model without conventional entropy**	**Model without class imbalancing**
NPV	0.898568	0.818094	0.765155	0.755101
FPR	0.089799	0.21166	0.253982	0.287478
F-measure	0.924151	0.826811	0.795751	0.757305
MCC	0.840215	0.645351	0.642006	0.617881
Specificity	0.910201	0.78834	0.746018	0.712522
Accuracy	0.943198	0.82573	0.782917	0.753908
Precision	0.932967	0.805632	0.781696	0.740087
FNR	0.076271	0.152492	0.188737	0.206037
Sensitivity	0.923729	0.847508	0.811263	0.793963

I. Statistical evaluation of LSTM + Bi-GRU to preceding models

In Table III, the statistical evaluation of the LSTM + Bi-GRU model utilizing the accuracy measure is analyzed with earlier models like the Bi-GRU, LSTM, CNN, RNN, MLP [24] and SVM [25] models, in accordance with five case scenarios. For dataset 1, the LSTM + Bi-GRU strategy produced the highest results (0.974), whereas models like the Bi-GRU (0.810), LSTM (0.837), CNN (0.834), RNN (0.867), MLP (0.842), and SVM (0.814) model produced the worst results. For identifying the assault for dataset 5, the LSTM + Bi-GRU technique likewise achieved the greatest performance (0.9321) when analyzed to other approaches like Bi-GRU, LSTM, CNN, RNN, MLP [24], and SVM [25] model. Thus, the LSTM + Bi-GRU strategy outperformed the preceding schemes for all scenarios to detect theattacks in IoT-Fog.

TABLE I. STATISTICAL EVALUATION OF LSTM + BI-GRU TO PRECEDING MODELS

Dataset 1					
Metrics	**Worst**	**Best**	**Mean**	**Median**	**SD**
MLP [24]	0.71826	0.842641	0.771363	0.762276	0.045035
SVM [25]	0.755843	0.814182	0.79605	0.807088	0.023793
CNN	0.782167	0.834994	0.803668	0.798755	0.020712
RNN	0.694626	0.867804	0.777112	0.773009	0.080026
Bi-GRU	0.764742	0.810189	0.783172	0.778879	0.017621
LSTM	0.711894	0.837749	0.781003	0.787184	0.047127
LSTM + Bi-GRU	0.892174	0.974465	0.929644	0.925969	0.030339
Dataset 2					

Metrics	Worst	Best	Mean	Median	SD
MLP [24]	0.725384	0.830146	0.768898	0.76003	0.038681
SVM [25]	0.757136	0.830646	0.807847	0.821803	0.029874
CNN	0.777021	0.80364	0.791316	0.792302	0.011803
RNN	0.707485	0.845651	0.762907	0.749246	0.051751
Bi-GRU	0.758523	0.792688	0.774605	0.773605	0.012881
LSTM	0.74924	0.831272	0.785799	0.781342	0.031446
LSTM + Bi-GRU	0.893798	0.953547	0.924893	0.926113	0.022365
Dataset 3					
Metrics	**Worst**	**Best**	**Mean**	**Median**	**SD**
MLP [24]	0.72533	0.816613	0.766183	0.761395	0.03423
SVM [25]	0.751864	0.842897	0.818239	0.839098	0.038417
CNN	0.769126	0.79595	0.785961	0.789383	0.010112
RNN	0.715428	0.842462	0.775482	0.772019	0.045058
Bi-GRU	0.763229	0.791359	0.771632	0.76597	0.01145
LSTM	0.761138	0.825311	0.790043	0.786861	0.023293
LSTM + Bi-GRU	0.896805	0.97684	0.93602	0.935218	0.030122
Dataset 4					
Metrics	**Worst**	**Best**	**Mean**	**Median**	**SD**
MLP [24]	0.732508	0.817651	0.766432	0.757785	0.035615
SVM [25]	0.758429	0.84711	0.819644	0.836518	0.035967
CNN	0.771876	0.795217	0.778965	0.774383	0.009555
RNN	0.714603	0.845651	0.775368	0.770608	0.046673
Bi-GRU	0.7464	0.794799	0.766038	0.761477	0.020189
LSTM	0.763662	0.824795	0.790595	0.786962	0.021918
LSTM + Bi-GRU	0.895422	0.952629	0.925141	0.926257	0.021339
Dataset 5					
Metrics	**Worst**	**Best**	**Mean**	**Median**	**SD**
MLP [24]	0.725276	0.80308	0.763469	0.76276	0.03085
SVM [25]	0.746592	0.857327	0.828632	0.855303	0.047384
CNN	0.755927	0.799569	0.780605	0.783462	0.016055
RNN	0.723372	0.839272	0.788057	0.794792	0.042335
Bi-GRU	0.738212	0.802947	0.768659	0.766738	0.022984
LSTM	0.756923	0.819351	0.794287	0.800436	0.024471
LSTM + Bi-GRU	0.899812	0.960132	0.932148	0.934323	0.022337

Conclusion

In IoT-Fog architecture, this study has offered an automated attack detection model. The phases in the process were "(i) preprocessing, (ii) feature extraction, (iii) attack detection, and (iv) mitigation". Preprocessing was the initial stage that handled the class imbalance problem with enhanced class imbalance processing. Correlation-based features, enhanced entropy-based features, statistical features, and Raw data were included in the feature extraction process. Based on the gathered features, attack detection will take place; for this purpose, a unique hybrid detection model combining Bi-GRU and LSTM was given. Once an assault was discovered, the network attacker had to be minimized. In this instance, a better mitigation strategy will be applied. A number of metrics were utilized to compare the selected approach against potential alternatives. For dataset 1, the LSTM + Bi-GRU strategy produced the highest results (0.974), whereas models like the Bi-GRU (0.810), LSTM (0.837), CNN (0.834), RNN (0.867), MLP (0.842), and SVM (0.814) model produced the lowest results.

References

[1] Manimurugan, S. IoT-Fog-Cloud model for anomaly detection using improved Naïve Bayes and principal component analysis. J Ambient Intell Human Comput (2021). https://doi.org/10.1007/s12652-020-02723-3

[2] A. Samy, H. Yu and H. Zhang, "Fog-Based Attack Detection Framework for Internet of Things Using Deep Learning," IEEE Access, vol. 8, pp. 74571-74585, 2020, doi: 10.1109/ACCESS.2020.2988854.

[3] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks" Journal of Engineering ResearchAvailable online 19 June 2023In press, corrected proofArticle 100122

[4] MirdulaSRoopa M, "MUD enabled deep learning framework for anomaly detection in IoT integrated smart building," e-Prime - Advances in Electrical Engineering, Electronics and Energy10 June 2023Volume 5 (Cover date: September 2023)Article 100186.

[5] K. L. K. Sudheera, D. M. Divakaran, R. P. Singh and M. Gurusamy, "ADEPT: Detection and Identification of Correlated Attack Stages in IoT Networks," IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6591-6607, 15 April15, 2021, doi: 10.1109/JIOT.2021.3055937.

[6] Rania A. ElsayedReem A. HamadaShaimaa Ahmed Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection" Ain Shams Engineering Journal3 March 2023Volume 14, Issue 10 (Cover date: October 2023)Article 102211.

[7] BhukyaMadhuM. Venu Gopala ChariVeerenderAerranagula, "Intrusion detection models for IOT networks via deep learning approaches", Measurement: Sensors12 December 2022Volume 25 (Cover date: February 2023)Article 100641.

[8] Kumar, P., Gupta, G.P. & Tripathi, R. Toward Design of an Intelligent Cyber Attack Detection System using Hybrid Feature Reduced Approach for IoT Networks. Arab J Sci Eng 46, 3749–3778 (2021). https://doi.org/10.1007/s13369-020-05181-3

[9] H. Al-Hamadi, I. -R. Chen, D. -C. Wang and M. Almashan, "Attack and Defense Strategies for Intrusion Detection in Autonomous Distributed IoT Systems," IEEE Access, vol. 8, pp. 168994-169009, 2020, doi: 10.1109/ACCESS.2020.3023616.

[10] L. Liu, X. Xu, Y. Liu, Z. Ma and J. Peng, "A Detection Framework Against CPMA Attack Based on Trust Evaluation and Machine Learning in IoT Network," IEEE Internet of Things Journal, vol. 8, no. 20, pp. 15249-15258, 15 Oct.15, 2021, doi: 10.1109/JIOT.2020.3047642.

[11] I. Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," IEEE Access, vol. 9, pp. 103906-103926, 2021, doi: 10.1109/ACCESS.2021.3094024.

[12] L. Vu, Q. U. Nguyen, D. N. Nguyen, D. T. Hoang and E. Dutkiewicz, "Deep Transfer Learning for IoT Attack Detection," IEEE Access, vol. 8, pp. 107335-107344, 2020, doi: 10.1109/ACCESS.2020.3000476.

[13] M. Hossain and J. Xie, "Third Eye: Context-Aware Detection for Hidden Terminal Emulation Attacks in Cognitive Radio-Enabled IoT Networks," IEEE Transactions on Cognitive Communications and Networking, vol. 6, no. 1, pp. 214-228, March 2020, doi: 10.1109/TCCN.2020.2968324.

[14] D. -C. Wang, I. -R. Chen and H. Al-Hamadi, "Reliability of Autonomous Internet of Things Systems With Intrusion Detection Attack-Defense Game Design," IEEE Transactions on Reliability, vol. 70, no. 1, pp. 188-199, March 2021, doi: 10.1109/TR.2020.2983610.

[15] A. Y. Khan, R. Latif, S. Latif, S. Tahir, G. Batool and T. Saba, "Malicious Insider Attack Detection in IoTs Using Data Analytics," IEEE Access, vol. 8, pp. 11743-11753, 2020, doi: 10.1109/ACCESS.2019.2959047.

[16] Kumar, P., Gupta, G.P. & Tripathi, R. A distributed ensemble design based intrusion detection system using fog computing to protect the internet of things networks. J Ambient Intell Human Comput 12, 9555–9572 (2021). https://doi.org/10.1007/s12652-020-02696-3

[17] Babu, M.R., K.N.Veena Implementing optimized classifier for distributed attack detection and BAIT-based attack correction in IoT. Int J Syst Assur Eng Manag (2021). https://doi.org/10.1007/s13198-021-01115-w

[18] Krishna, E.S.P., Thangavelu, A. Attack detection in IoT devices using hybrid metaheuristic lion optimization algorithm and firefly optimization algorithm. Int J Syst Assur

Eng Manag (2021). https://doi.org/10.1007/s13198-021-01150-7

[19] Fotohi, R., Pakdel, H. A Lightweight and Scalable Physical Layer Attack Detection Mechanism for the Internet of Things (IoT) Using Hybrid Security Schema. Wireless PersCommun 119, 3089–3106 (2021). https://doi.org/10.1007/s11277-021-08388-1

[20] Duraisamy, A., Subramaniam, M. Attack Detection on IoT Based Smart Cities using IDS Based MANFIS Classifier and Secure Data Transmission Using IRSA Encryption. Wireless PersCommun 119, 1913–1934 (2021). https://doi.org/10.1007/s11277-021-08362-x

[21] https://cloudstor.aarnet.edu.au/plus/index.php/s/2DhnLGDdEECo4ys?path=%2FUNSW-NB15%20-%20CSV%20Files

[22] https://www.unb.ca/cic/datasets/index.html

[23] https://www.kaggle.com/datasets/solarmainframe/ids-intrusion-csv.

[24] Ahmed, S.; Khan, Z.A.; Mohsin, S.M.; Latif, S.; Aslam, S.; Mujlid, H.; Adil, M.; Najam, Z. Effective and Efficient DDoS Attack Detection Using Deep Learning Algorithm, Multi-Layer Perceptron. Future Internet 2023, 15, 76. https://doi.org/10.3390/fi15020076

[25] Alibrahimi, Fuqdan&Amintoosi, Haleh. (2020). A Hybrid Method of Genetic Algorithm and Support Vector Machine for DNS Tunneling Detection.

[26] Dr. S.A. Sivakumar. (2019). Hybrid Design and RF Planning for 4G networks using Cell Prioritization Scheme. International Journal of New Practices in Management and Engineering, 8(02), 08 - 15. https://doi.org/10.17762/ijnpme.v8i02.76

[27] Jackson, B., Lewis, M., González, M., Gonzalez, L., & González, M. Improving Natural Language Understanding with Transformer Models. Kuwait Journal of Machine Learning, 1(4). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/152

[28] Kshirsagar, P. R., Reddy, D. H., Dhingra, M., Dhabliya, D., & Gupta, A. (2022). A review on application of deep learning in natural language processing. Paper presented at the Proceedings of 5th International Conference on Contemporary Computing and Informatics, IC3I 2022, 1834-1840. doi:10.1109/IC3I56241.2022.10073309 Retrieved from www.scopus.com