A Hybrid Deep Learning Model with Self-Improved Optimization Algorithm for Detection of Security Attacks in IoT Environment

Amit Sagu 1, Nasib Singh Gill 1, Preeti Gulia 1, Jyotir Moy Chatterjee 2 and Ishaani Priyadarshini 3,*

1 Department of Computer Science & Applications, Maharshi Dayanand University, Rohtak 124001, India
2 Department of Information Technology, Lord Buddha Education Foundation, Kathmandu 44600, Nepal
3 School of Information, University of California, Berkeley, CA 94720-5800, USA
* Correspondence: ishaani@ischool.berkeley.edu

Abstract: With the growth of the Internet of Things (IoT), security attacks are also rising gradually. Numerous centralized mechanisms have been introduced in the recent past for the detection of attacks in IoT, in which an attack recognition scheme is employed at the network’s vital point, which gathers data from the network and categorizes it as “Attack” or “Normal”. Nevertheless, these schemes were unsuccessful in achieving noteworthy results due to the diverse necessities of IoT devices such as distribution, scalability, lower latency, and resource limits. The present paper proposes a hybrid model for the detection of attacks in an IoT environment that involves three stages. Initially, the higher-order statistical features (kurtosis, variance, moments), mutual information (MI), symmetric uncertainty, information gain ratio (IGR), and relief-based features are extracted. Then, detection takes place using Gated Recurrent Unit (GRU) and Bidirectional Long Short-Term Memory (Bi-LSTM) to recognize the existence of network attacks. For improving the classification accuracy, the weights of Bi-LSTM are optimally tuned via a self-upgraded Cat and Mouse Optimizer (SU-CMO). The improvement of the employed scheme is established concerning a variety of metrics using two distinct datasets which comprise classification accuracy, and index, f-measure and MCC. In terms of all performance measures, the proposed model outperforms both traditional and state-of-the-art techniques.

Keywords: Bi-LSTM; GRU; IoT; optimization; security attack detection

1. Introduction

IoT has impacted our everyday lives due to its supreme advancement and it is a rising technology that provides an exclusive connection for making available automated operations and services in diverse fields [1]. IoT is perceived as a method with a suitable system interrelated via servers, sensors, and different software. Due to their lesser processing power, IoT appliances are deployed in several security areas. As additional devices are deployed on IoT, security and privacy issues are gaining progressive deliberation [2]. The IoT attacks are leading to implausible losses to IoT networks and even threatening human security. Moreover, IoT networks contain limited characteristics that make attack exposure more difficult. Initially, the range of platforms, hardware, software, and protocols exposes diverse vulnerabilities. Subsequently, higher-rate and lower-rate attacks are widely used by IoT hackers to hack legitimate data [3]. These lower-rate attacks are complicated to notice and reside on the network. Finally, attackers become more intelligent and can vigorously modify their attack strategies as per ecological feedback to avoid detection, making it more complex for defenders to find out consistent patterns to recognize attacks [4].

As the number of IoT devices increases, attackers have many opportunities to negotiate them through malicious email, secrecy attacks, and denial of service (DoS) attacks, amongst other types of attacks [5]. Attacks can come from the channels that unite the IoT elements. The protocols deployed in IoT systems include security issues that affect the whole system.
IoT systems are also susceptible to renowned network attacks such as DoS and spoofing of software and appliances [6].

Certain reports declared that 70% of IoT devices are subjected to different network attacks that cause diverse vulnerabilities, such as encryption and password security. The domain wherein IoT is broadly used are smart homes, intellectual transport systems, agriculture, hospitals, earthquake discovery, and so on [7,8]. For the malevolent agents, IoT appliance is vulnerable making the IoT devices a launcher pad for attacks on varied domains [9]. Thereby, a protected IoT infrastructure is essential for protecting IoT devices from attacks. In this paper, a hybrid deep learning model that identifies security attacks in an IoT environment is implemented. Two deep learning model is used: GRU and Bi-LSTM, weights of Bi-LSTM are optimally tuned via a self-upgraded cat and mouse optimizer (SU-CMO) algorithm to refine classification accuracy. Furthermore, two separate datasets are utilized to compare the proposed model’s classification accuracy to other existing schemes.

The present paper follows mainly three steps which are as under:

- Suggests a new attack detection model in IoT, where various diverse features are derived.
- Deploys hybrid classifiers such as GRU and Bi-LSTM with an optimization strategy to detect attacks.
- Exploits an SU-CMO model to choose the optimal weights in Bi-LSTM.

Section 2 reviews the work. Section 3 briefs a stepwise description of the proposed model and Section 4 explains about the extraction of diverse features. Section 5 depicts SU-CMO-based hybrid classification. Section 6 illustrates outcomes and the conclusion is presented in Section 7.

2. Literature Review

2.1. Related Work

In 2020, Mandal et al. [1] stated that ML supported various fields for service betterment. Moreover, for having interaction among humans in a single device, IoT acts a major role. Moreover, due to the emergence of digital technology, conveying data with no human communication is possible. In the aspect of privacy and security patterns, ML was deployed. Here, the recognition of intrusion and the challenges of security in IoT were concentrated more. Moreover, a variety of attacks were analyzed depending on surface attacks. Here, the major aim was to develop the effectiveness of recognizing the attacks by executing the ML approach.

In 2021, Kan et al. [10] proposed an attack detection technique for an IoT system depending upon Adaptive APSO-CNN. The APSO-CNN optimized the 1-D CNN constraints. For the fitness of composing technique, the cross-entropy losses of CNN were considered. Additionally, an assessment technique was defined that measured both the prediction and predicted labels to evaluate the anticipated APSO-CNN algorithm. From the simulated outcomes, the efficiency and reliability of APSO-CNN were proven regarding attack detection.

In 2020, Nimbalkar et al. [11] explored various attacks in IoT due to the vulnerabilities in devices. They stated that recognition of attack was a dreary procedure for ML techniques due to the existence of traffic features in IoT systems. Their work offered a feature selection for intrusion detection systems for the revelation of DoS and DDoS attacks. Using the inclusion operation and union operation, the sunsets of features in the developed system were obtained. In the end, the enhancement of deployed scheme was proven.

In 2020, Pecor et al. [12] explained the contribution of IoT to the everyday lives of humans. They established the recognition of traffic in the network and the classification of the detected network. In this work, a large dataset was introduced for detecting the traffic in the network. With the aid of the DN model, they scrutinized binary and multinomial classification.

In 2020, Rahman et al. [13] developed a Scalable ML model for detecting intrusions in IoT-facilitated Smart Cities. Their work addressed the limitations of centralized IDS by...
proposing semi-distributed and distributed techniques. Additionally, feature extraction and feature selection were performed. For allocating the tasks, parallel ML techniques were developed. Their results obtained provided accuracy and building time performance on attack detection.

In 2020, Atul et al. [14] exposed an effective method for sharing and relocating information for digital communication. Certain system challenges were mentioned namely, failure in service, abnormality, and security barriers. This work analyzed and presented a communication pattern utilizing the EASH approach. The abnormality sources of the communiqué paradigm were differentiated by employing the ML technique. The performance, accuracy, and effectiveness were calculated for the developed method.

In 2021, Krishna and Thangavelu et al. [15] examined the DOS attack in IoT systems. The security issues and attacks, which occurred in IoT devices, were demonstrated. For detecting the attack, two algorithms such as hybrid ML-F were proposed. Here, the developed scheme has attained the utmost performance in classifying the attacks correspondingly.

In 2020, Gu et al. [16] examined the security and privacy issues in IoT with much concentration. They described that IoT attacks were causing implausible defeat to IoT networks and intimidating the security of humans. Here, a reinforcement learning-oriented attack detection scheme was proposed that detected the attack pattern and its conversion. In this work, the IoT traffic features were also explored and entropy-based metrics were used to forecast the attacks in IoT. Furthermore, widespread experiments were performed over the IoT dataset, and the efficiency of the developed model was revealed.

In 2022, Gopali et al. [17] by employing a recurrent neural network including LSTM, can identify anomalies in the Internet of Things environment. In addition, the authors evaluate and contrast deep learning strategies i.e., CNN, reporting on their effectiveness. In this study, it is observed that the LSTM model with the fastest learning rate had the highest accuracy, although it required a longer time to be trained.

In 2022, Ahmed et al. [18] offered a comprehensive analysis of numerous low-rate distributed denial of service (LDDoS) detection algorithms that are being used for SDN. The distributed denial of service (DDoS) assault has recently evolved into the LDDoS attack, which is more difficult to detect and creates more of a challenge. The authors demonstrate that techniques employing deep learning combined with a hybrid model, such as CNN-LSTM and CNN-GRU, may achieve the desired results.

In 2022, Abbas et al. [19] demonstrated that traditional approaches and technologies are ineffective in addressing new security concerns and difficulties, and how machine learning as a promising technology enables the creation of a wide variety of effective ways that can improve the safety and security of the IoT.

2.2. Review

Table 1 shows a review of the approaches used in the detection of attacks in IoT. In [1], an ML algorithm was used to find out the attacks that occurred in the IoT network. High accuracy and lower false rate are identified using the machine learning classification algorithm. However, some of the security issues are not considered in this approach. APSO-CNN was performed in [10], which performs effective and reliable detection of attack in IoT networks, but it does not differentiate the complicated task of interruption. JRip classifier was deployed in [11] and achieved a higher performance, accuracy, and detection rate, but the detection was applied for only a particular dataset. NN was exploited in [12], which offered higher accuracy and performance rate. However, layer specification was not performed using this analysis. In [13], the MLP algorithm was used to detect the attack, which provided a high level of performance and a higher feature set. Furthermore, it tests only detection rate but not efficiency. In [14] attack detection was performed using the EASH algorithm in which a higher rate of accuracy is found. ML-F was used in [15] which provided a higher detection of attacks in IoT networks. However, it does not categorize some of the attacks. The Markov Decision technique was used in [16] for the quick detection
of attacks in IoT networks. This attained high accuracy for the feature set. However, ANN was not detected accurately.

Table 1. Reviews on conventional IoT attack detection models.

<table>
<thead>
<tr>
<th>Author</th>
<th>Deployed Schemes</th>
<th>Features</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandal et al. [1]</td>
<td>ML algorithm</td>
<td>High accuracy</td>
<td>Some security issues were not considered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lower false rate</td>
<td></td>
</tr>
<tr>
<td>Kan et al. [10]</td>
<td>APSO-CNN</td>
<td>Effective and reliable</td>
<td>Do not differentiate complicated tasks from interruption</td>
</tr>
<tr>
<td>Pushparaj et al. [11]</td>
<td>JRip classifier</td>
<td>Higher performance</td>
<td>Only a particular dataset is preferred</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Improved accuracy</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>High detection rate</td>
<td></td>
</tr>
<tr>
<td>Pecor et al. [12]</td>
<td>NN</td>
<td>Higher accuracy and performance</td>
<td>Layers specification is not provided</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Arafatur et al. [13]</td>
<td>MLP</td>
<td>High level of performance</td>
<td>Test only for attack detection but not efficiency</td>
</tr>
<tr>
<td></td>
<td></td>
<td>High ranking of feature</td>
<td></td>
</tr>
<tr>
<td>Jyoti et al. [14]</td>
<td>EASH</td>
<td>Attain a higher accuracy rate</td>
<td>Tested for detection rate only</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Krishna et al. [15]</td>
<td>ML-F</td>
<td>High accuracy</td>
<td>Two attack categories were not considered</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Low detection time</td>
<td></td>
</tr>
<tr>
<td>Gu et al. [16]</td>
<td>Markov Decision</td>
<td>High accuracy for a feature set</td>
<td>ANN was not detected accurately</td>
</tr>
<tr>
<td>Gopali et al. [17]</td>
<td>LSTM, CNN</td>
<td>Higher accuracy rate</td>
<td>Higher training time</td>
</tr>
<tr>
<td>Ahmed et al. [18]</td>
<td>CNN, GRU, LSTM</td>
<td>Higher detection rate</td>
<td>Supplementary details issues are not provided.</td>
</tr>
<tr>
<td>Abbas et al. [19]</td>
<td>Random Forest, Knn</td>
<td>Offers extensive review</td>
<td>Comparison results are not provided</td>
</tr>
</tbody>
</table>

3. A Stepwise Description of the Proposed Model

The developed attack detection model comprises three essential phases.

- Initially, features including “kurtosis, variance, moments, mutual information, symmetric uncertainty, information gain ratio, and relief-based features” are derived.
- These features are then subjected to optimized GRU and BI-LSTM that recognizes the presence of attacks.
- Here, the weights of BI-LSTM are optimally tuned via SU-CMO.

Figure 1 depicts the overall framework of the proposed model below.
4. Extraction of Diverse Features

This work extracts the following features from the input data. A brief explanation of the features is as follows:

Kurtosis [20]: “It is a measure that identifies whether the data are light-tailed or heavy-tailed and related to the normal distribution”. Datasets with smaller kurtosis offer minor outliers or tails. The arithmetic formulation of kurtosis KS for univariate data such as \( Y_1, Y_2, \ldots Y_k \), is articulated in Equation (1).

\[
KS = \frac{\sum_{i=1}^{k} (Y_i - \bar{Y})^4}{\Gamma^4}.
\] (1)

The standard deviation is calculated by \( k \) value available in the denominator whilst computing the kurtosis.

Variance [21]: It is defined as the mean squared disparity amongst every data point and the center of distribution computed by mean.

Moment [22]: It is the instant probability distribution along with arbitrary variables in probability theory and statistics. It is the ordinary value of a specified integer power difference of the arbitrary variable from the mean. The moments with higher order are related to the shape and spread distribution of the location. The \( m^{th} \) moment linked to the central moment of a real-valued arbitrary variable \( Q \) is the quantity \( Q_m = \rho(Q) \right\{ \left( Q - \rho(Q) \right)^m \} \), in which, \( \rho \) stands for the expectation operator. The \( m^{th} \) moment regarding the mean \( Q \) is portrayed for a continuous UPD with \( f(y) \) PDF. The moment is given in Equation (2).

\[
\text{Moment} = Q_m = \rho \left[ \left( Q - \rho(Q) \right)^m = \int_{-\infty}^{\infty} \left( y - Q \right)^m f(y)dy \right].
\] (2)

MI Features: It is defined as the calculation of exchanged information among two ensembles of random variables \( N \) and \( Z \) It is formulated as revealed in Equation (3), in which, \( \rho \) signifies probability.

\[
MI = \sum \rho(N,Z) \log_2 \frac{\rho(N,Z)}{\rho(N) \rho(Z)}.
\] (3)

Symmetrical uncertainty: SU computes the features depending upon evaluated SU correlation metrics amongst the class and feature. The MI is calculated as in Equation (4), wherein, MI implies MI, \( f e \) stands for features, \( cl \) stands for class, and \( P \) implies probability function. Further, SU is formulated as in Equation (5), in which \( En \) refers to the entropy function.

\[
MI(f,e,cl) = \sum P(f,e,cl) \log_2 \frac{P(f,e,cl)}{P(fe)P(cl)}
\] (4)

\[
SU(f,e,cl) = 2(MI(f,e,cl))/(En(f,e)En(cl)).
\] (5)

IGR [23]: It is the ratio between information gain (IG) and split information (SI) value as shown in Equation (6), where \( K \) refers to a random variable, \( b \) refers to an attribute and in equation (7) \( U(ti) \) refers to the number of times \( ti \) occurs, \( U(t) \) refers to entire event counts and \( t \) refers to event sets.

\[
IGR(K,b) = \frac{IG(K,b)}{SI(K)}
\] (6)

\[
IGR(K,b) = \frac{\sum_{i=1}^{n} P(K) \log P(K) - \left( \sum_{i=1}^{n} P(K|b) \log P(K|b) \right)}{-\sum_{i=1}^{n} \frac{U(ti)}{U(t)} \log_2 \left( \frac{U(ti)}{U(t)} \right)}.
\] (7)
**Relief Features**: It was modeled for application to binary classification issues with discrete or numerical features. It is modeled as in Equation (8), where $E_i$ points out the feature vector, the closer same class instance is termed “nearHit”, and the closer different class instance is termed “nearMiss”.

$$R_i = R_i - (E_i - \text{nearHit}_i)^2 + (E_i - \text{nearMiss}_i)^2.$$  

(8)

The derived features are signified by $Fe$, which are then subjected to hybrid classification.

---

**5. SU-CMO-Based Hybrid Classification**

5.1. Hybrid Classifiers

**LSTM classifier**: It [24] includes a sequence of recurring LSTM cells. Each cell of LSTM encompassed 3 units, such as “forget gate, the input gate, and the output gate”. Presume variables $Z$ and $D$ that indicate hidden and cell state in order. $(X_t, D_{t-1}, Z_{t-1})$ and $(Z_t, D_t)$ indicate input and output layers.

At time $t$, the output, input and forget gate implies $O_t$, $I_t$, $F_t$ in that order. LSTM chiefly exploits $I_t$ for sorting the data. The sorted data indicate specified partial features connected to the previous gaze direction; $F_t$ is formulated as shown in Equation (9).

$$F_t = \sigma(J_{IF}X_t + I_{IF} + J_{ZF}Z_{t-1} + L_{ZF}).$$  

(9)

In Equation (9), $(J_{ZF}, L_{ZF})$ and $(J_{IF}, I_{IF})$ points out weight and bias constraints to map hidden and input layers to forget gate and activation function is signified by $\sigma$.

Input gate is exploited by LSTM as revealed in Equations (10)–(12), wherein, $(J_{ZG}, L_{ZG})$ and $(J_{IG}, I_{IG})$ implies weight and bias constraints to map hidden and input layers to the cell gate respectively. $(J_{ZI}, L_{ZI})$ and $(J_{II}, I_{II})$ imply weight and bias constraints to map hidden and input layers to $I_t$.

$$G_t = \tanh(J_{IG}X_t + I_{IG} + J_{ZG}Z_{t-1} + L_{ZG})$$  

(10)

$$I_t = \sigma(J_{II}X_t + L_{II} + J_{ZI}Z_{t-1} + L_{ZI})$$  

(11)

$$D_t = F_tD_{t-1} + I_tG_t$$  

(12)

$$O_t = \sigma(J_{IO}X_t + L_{IO} + J_{ZO}Z_{t-1} + L_{ZO})$$  

(13)

$$Z_t = O_t \tanh(D_t).$$  

(14)

In addition, the LSTM cell obtains the output hidden layer from the output gate as shown in Equations (13) and (14), in which, $(J_{ZO}, L_{ZO})$ and $(J_{IO}, I_{IO})$ represents weight and bias to map the hidden and input layer to $O_t$. Accordingly, the weights of LSTM represented by $J$ are optimally elected by the proposed SU-CMO model.

**Bi-GRU [25]**: Depending upon RNN, 3 gates named “forget gate, input gate, and output gate and memory cell” are integrated with LSTM for controlling the flow of information across LSTM cells. Similarly, GRU deploys special gates, called reset and update gates, for lessening gradient dispersal with slighter computation losses. The update gate $(ut)$ substitutes forget and input gates of LSTM, portraying the retention degree of prior data as revealed in Equation (15).

$$ut = \mu(W_u * [R_{t-1}, Fea_t] + f_u).$$  

(15)

In Equation (15), $\mu$ points out the sigmoid activation function among 0 and 1, $Fea_t$ stands for the input matrix at time step $t$, $R_{t-1}$ stands for the hidden state at the prior time step; $t - 1$ stands for the weight matrix of $ut$ and $f_u$ stands for the bias matrix of $ut$. The reset gate $(rt)$ regulates how much chronological data have to be ignored, which is revealed in Equation (16), wherein, $W_r$ characterize weight matrix of $rt$ and $f_r$ symbolize bias matrix of $rt$.

$$rt = \mu(W_r * [R_{t-1}, Fea_t] + f_r).$$  

(16)
Subsequently, the candidate’s hidden state is revealed in Equation (17), wherein, \( \tanh \) stands for \( \tanh \) activation function. \( f_R \) and \( W_R \) stand for bias matrix and weight matrix of new cell state, \( \ast \) stands for dot multiplication function. Thus, the output \( R_t \) implies linear interruption amid \( \tilde{R}_t \) and \( R_{t-1} \) in Equation (18).

\[
\tilde{R}_t = \tanh(W_R[R_{t-1} \ast r_{t}, \text{Fea}_t] + f_R) \tag{17}
\]

\[
R_t = (1 - u_t) \ast R_{t-1} + u_t \ast \tilde{R}_t. \tag{18}
\]

The forward GRU captures the previous details of input data and the backward GRU obtains the upcoming details of input data. The Bi-GRU is modeled as in Equation (19), wherein, \( \rightarrow R_t \) and \( \leftarrow R_t \) correspond to the hidden state of backward and forward GRU in that order, \( C_t \) corresponds to combining technique of outputs at two directions.

\[
Y_t = C_t(\rightarrow R_t, \leftarrow R_t). \tag{19}
\]

### 5.2. SU-CMO Model

The extant CMBO [26] model gives optimal solutions; still, it tolerates low accuracy. For the aim of overcoming the drawbacks of conventional CMBO, specific improvements were made. The steps in the SU-CMO model and flowchart (Figure 2) are given below.

Step 1: The initial population of \( B \) search agents is initialized.

Step 2: The parameters of \( B, B_c, B_m, T \) are initialized. Here, \( B \) is the count of members in the population matrix \( A \).

Step 3: The initial population is created as per Equation (20).

\[
A = \begin{bmatrix}
A_1 \\
A_2 \\
\vdots \\
A_B
\end{bmatrix}_{B \times m} = \begin{bmatrix}
y_{1,1} & \cdots & y_{1,d} & \cdots & y_{1,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y_{i,1} & \cdots & y_{i,d} & \cdots & y_{i,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y_{B,1} & \cdots & y_{B,d} & \cdots & y_{B,m}
\end{bmatrix}_{B \times m} \tag{20}
\]

Here, \( y_{i,d} \) is the \( d^{th} \) problem variable.

Step 4: The fitness of the search agents is computed as per Equation (21).

\[
Obj = \text{min}(Err) \tag{21}
\]

Step 5: Using Equations (22) and (23), update the sorted population matrix \( A^S \). Here, the \( i^{th} \) population of the sorted population matrix is denoted as \( y_{i,d}^S \). In addition, \( Obj^S \) is the sorted objective function-based vector.

\[
A^S = \begin{bmatrix}
A_1^S \\
A_2^S \\
\vdots \\
A_B^S
\end{bmatrix}_{B \times m} = \begin{bmatrix}
y_{1,1}^S & \cdots & y_{1,d}^S & \cdots & y_{1,m}^S \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y_{i,1}^S & \cdots & y_{i,d}^S & \cdots & y_{i,m}^S \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y_{B,1}^S & \cdots & y_{B,d}^S & \cdots & y_{B,m}^S
\end{bmatrix}_{B \times m} \tag{22}
\]

\[
Obj^S = \begin{bmatrix}
Obj_{1}^S \text{min}(Obj) \\
Obj_{2}^S \text{min}(Obj) \\
\vdots \\
Obj_{B}^S \text{min}(Obj)
\end{bmatrix}_{B \times 1} \tag{23}
\]
Step 6: Using Equation (24), the mice population is chosen.

\[
M = \begin{bmatrix}
M_1 = X^S_1 \\
\vdots \\
M_i = X^S_i \\
M_{B_m} = X^S_{B_m}
\end{bmatrix} = \begin{bmatrix}
y^S_{1,1} & \cdots & y^S_{1,d} & \cdots & y^S_{1,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y^S_{i,1} & \cdots & y^S_{i,d} & \cdots & y^S_{i,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y^S_{B_m,1} & \cdots & y^S_{B_m,d} & \cdots & y^S_{B_m,m}
\end{bmatrix}
\]  

(24)

Step 7: Using Equation (25), the cat population is selected.

\[
C = \begin{bmatrix}
C_1 = X^S_{B_m+1} \\
\vdots \\
C_i = X^S_{B_m+j} \\
C_{B_c} = X^S_{B_m+B_c}
\end{bmatrix} = \begin{bmatrix}
y^S_{B_m+1,1} & \cdots & y^S_{B_m+1,d} & \cdots & y^S_{B_m+1,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y^S_{B_m+j,1} & \cdots & y^S_{B_m+j,d} & \cdots & y^S_{B_m+j,m} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
y^S_{B_m+B_c,1} & \cdots & y^S_{B_m+B_c,d} & \cdots & y^S_{B_m+B_c,m}
\end{bmatrix}
\];  

(25)

Step 8: Here, \(M, B_m, M_i, C, B_c, C_j\) points to the mice population, count of mice, \(j^{th}\) mice, cat population, count of cats and the \(i^{th}\) cat, respectively.

Step 9: The position update of cats is modeled as in Equation (26), where \(C_{j_{\text{new}}}\) new points to the new position of the \(j^{th}\) cat and \(C_{j_{d}}\) is the new value for \(d^{th}\) problem. In addition, the random value \(r\) is estimated randomly within the limit \([0, 1]\). Here, \(I\) is computed as in Equation (27), where \(\text{rand}\) is a random integer.

\[
C_{j_{\text{new}}} = C_{j_{d}} + r \times (M_{k_{d}} - I \times C_{j_{d}})
\]

(26)

Here, \(l = \text{round}(1 + \text{rand})\);  

(27)

Step 10: If \(j = B_c\) if the above condition is satisfied then \(H_i\) is created using Equation (28).

\[
H_i = h_{i,d} = y_{i,d} \& i = 1 : B_m, d = 1 : m, l \in 1 : B
\]

(28)

Step 11: Then, position update of mice takes place based on Equations (29) and (30). Conventionally, \(M_i\) is updated as shown in Equation (30), however, as per the SU-CMO model, \(M_i\) is updated based upon random integers \(r_{a1}\) and \(r_{a2}\) as in Equations (31) and (32). Here, \(r_{a1}\) and \(r_{a2}\) are assigned values of 1.25 and 1.75.

\[
M_{i_{\text{new}}} = m_{i_{d}} + r \times \left(h_{i,d} - l \times m_{i,d}\right) + \text{Sign}(F^m_i - F^H_i)
\]

(29)

\&

\[
i = 1 : B_m, d = 1 : m
\]

\[
M_i = \begin{cases} 
M_{i_{\text{new}}} & F^m_{i_{\text{new}}} < F^m_i \\
M_i & \text{else}
\end{cases}
\]

(30)

\[
M_i = \begin{cases} 
M_{i_{\text{new}}} & F^m_{i_{\text{new}}, r_{a1}} < F^m_i \\
M_i & \text{else}
\end{cases}
\]

(31)

\[
M_i = \begin{cases} 
M_{i_{\text{new}}} & F^m_{i_{\text{new}}, r_{a2}} < F^m_i \\
M_i & \text{else}
\end{cases}
\]

(32)

Step 12:

(a) In case the above condition is not satisfied, then increase \(j\) by 1, and again update \(C_j\).

(b) Terminate the if condition.
Step 13: If \( i = B_m \) then
(a) if the above condition is satisfied, then check if \( t = T \).
(b) if the above condition is not satisfied, then increase \( i \) by 1.
(c) End if.

Step 14: If \( t = T \), then the best solution acquired so far is returned.
Step 15: If \( t \neq T \), then increase \( i \) by 1 and move back to step 8.
Step 16: Terminate.

Figure 2. Flowchart of suggested SU-CMO.

6. Results and Discussion
6.1. Simulation Setup

The presented HC + SU-CMO scheme for the detection of attacks was experimented with in “Python”. The effectiveness of the newly proposed method was evaluated over HC + ALO [27], HC + AO [28], HC + BOA [29], HC + CMBO [26], HC + SSA [30], NN [31], RNN [32], Bi-GRU [25], SVM [33] and KNN [33] concerning popularly used metrics such
as NPV, accuracy and FPR along with other widely used metrics. The convergence study was carried out by experimenting with several iterations, which ranged from 0 to 25 with an interval of 5. The datasets from [34,35] were used for the analysis; they are represented as datasets 1 and 2 in the description part. Dataset 1 has only a single category, which is DDoS, but dataset 2 has multiple attack categories such as backdoor, DDoS, exploit, fuzzes, reconnaissance, shellcode, and worms, however, these attack categories are under single classification output i.e., “Attack”.

6.2. Performance Analysis

The performances of the suggested HC + SU-CMO are computed over extant classification models and optimization models regarding varied metrics. Moreover, evaluations were set up using datasets [34,35] respectively, and relevant outcomes are plotted in Figures 3–7. Analysis of suggested HC + SU-CMO model is analyzed and compared with HC + ALO, HC + AO, HC + BOA, HC + CMBO, HC + SSA, NN, RNN, Bi-GRU, SVM, and KNN for LPs which ranges from 60 to 80 with an interval of 10. Further, the assessment of the adopted scheme over the traditional approaches for accuracy and MCC is signified in Figures 3 and 5 along with an assessment of the adopted scheme over traditional approaches for F-measure and rand index is signified in Figures 4 and 6 for dataset 1 and dataset 2 respectively. In the case of this estimation, the HC + SU-CMO model has grown superior results when compared to other schemes. On looking through Figures 3a and 5a, the accuracy of the HC + SU-CMO model raise compared to other schemes for dataset 1 and dataset 2. Especially, better outputs are attained at the 60th and 70th LP for the proposed scheme regarding accuracy for both datasets. The superiority of the developed model is proven over HC + ALO, HC + AO, HC + BOA, HC + CMBO, HC + SSA, NN, RNN, Bi-GRU, SVM, and KNN models.

The proposed model shows a good accuracy level at the 60th and 70th learning rates; however, at the 80th learning rate, it shows less accuracy level compared to the lower learning rate. Bi-GRU performs poorly on both metrics, accuracy, and MCC.

![Figure 3](image-url)  
Figure 3. Performance of the adopted model over other existing schemes for dataset 1 (a) Accuracy (b) MCC.
Figure 3. Performance of the adopted model over other existing schemes for dataset 1 (a) Accuracy (b) MCC.

Figure 4. Performance of the adopted model over other existing schemes for dataset 1 (a) F-measure (b) Rand Index.

Figure 5. Performance of the adopted model over other existing schemes for dataset 2 (a) Accuracy (b) MCC.

Figure 6. Performance of the adopted model over other existing schemes for dataset 2 (a) F-measure (b) Rand Index.
F-measure is a classification evaluation measure defined as the harmonic mean of recall and precision. Figure 4a depicts the f-measure of the adopted model, producing a positive outcome at learning rate 60th and 70th.

The Rand Index is another widely used measure. It is a measurement of the degree to which two data clusters are like one another. The adopted model’s rand index for dataset 1 is illustrated in Figure 4b. The measure is nearly constant across all learning rates which is preferable to other existing schemes.

The metrics findings for dataset 2 are likewise satisfactory. Figure 5a,b demonstrates the accuracy and MCC of the model used for dataset 2. Accuracy is almost consistent for all given learning rates, whilst MCC also outperforms other known techniques at all learning rates in the graph. SVM, on the other hand, exhibits worse accuracy than other approaches across all learning rates, as well as very poor outcomes in the MCC measure.

After the proposed model, KNN and NN also perform well for dataset 2 in both metrics, i.e., accuracy and MCC.

Figure 6a,b illustrates the F-measure and rand index for the adopted model, as well as other known techniques for dataset 2. The adopted model also performs well in these metrics over the whole learning rate. SVM performs very poorly at the 60th learning rate and improves at the 80th learning rate for the f-measure metric.

All the techniques show outcomes over 60%, the adopted model having results above 80%, outperforming the other known techniques in the paper. Furthermore, we can see that no techniques underperform in the rand index metric.

### 6.3. Convergence Study

The convergence study of the suggested SU-CMO scheme in contrast with traditional schemes such as ALO, AO, BOA, CMBO, and SSA for ranging iterations is illustrated in Figure 7. In this scenario, estimation is computing by varying iterations. While examining the outcomes, the suggested SU-CMO has increased the slightest cost values which is ranging from the 13th–the 25th iteration in the comparison with others. In the same manner, the recommended SU-CMO has grown slightly higher cost from the 0th–12th iterations; although, values are lower than those of those being compared such as ALO, AO, BOA, CMBO, and SSA. While, the existing CMBO has demonstrated the worse outcome i.e., high-cost value in almost all iterations when contrasted with the several alternative schemes such as ALO, AO, BOA, CMBO, and SSA. Therefore, improved convergence is achieved by SU-CMO over alternate schemes.
6.4. Accuracy Analysis

The accuracy of the developed HC+SU-CMO scheme in contrast with conventional approaches such as ALO, AO, BOA, CMBO, SSA, NN, RNN, Bi-GRU, SVM, and KNN for 60, 70, and 80 learning rates is demonstrated in Tables 2 and 3 for dataset 1 and dataset 2 respectively. HC+SU-CMO shows the highest accuracy for learning rate 70 in dataset 1 and dataset 2. The results have summarized that the adopted hybrid classifier + SU-CMO performance is enhanced over the existing models. The phase of feature extraction makes a significant contribution to improving the accuracy rate. A loss of accuracy is possible if features aren’t extracted. Furthermore, hyperparameter tuning plays an essential part in ensuring the validity of the model. Tuning is a method of improving a model’s performance by selecting appropriate parameters without overfitting or producing an excessive amount of variation in the data.

Table 2. Accuracy Analysis of the proposed model over other methods for dataset 1.

<table>
<thead>
<tr>
<th>Learning Rate</th>
<th>HC+ALO</th>
<th>HC+AO</th>
<th>HC+BOA</th>
<th>HC+CMBO</th>
<th>HC+SSOA</th>
<th>NN</th>
<th>RNN</th>
<th>Bi-GRU</th>
<th>SVM</th>
<th>KNN</th>
<th>HC+SU-CMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>0.6625</td>
<td>0.5785</td>
<td>0.63375</td>
<td>0.5875</td>
<td>0.57625</td>
<td>0.7075</td>
<td>0.62625</td>
<td>0.541667</td>
<td>0.71125</td>
<td>0.75</td>
<td>0.8375</td>
</tr>
<tr>
<td>70</td>
<td>0.805</td>
<td>0.788333</td>
<td>0.583333</td>
<td>0.606667</td>
<td>0.625</td>
<td>0.818333</td>
<td>0.58333</td>
<td>0.5</td>
<td>0.585</td>
<td>0.694444</td>
<td>0.848333</td>
</tr>
<tr>
<td>80</td>
<td>0.6425</td>
<td>0.5725</td>
<td>0.65</td>
<td>0.6125</td>
<td>0.6725</td>
<td>0.6875</td>
<td>0.6325</td>
<td>0.5</td>
<td>0.5725</td>
<td>0.541667</td>
<td>0.7725</td>
</tr>
</tbody>
</table>

Table 3. Accuracy Analysis of the proposed model over other methods for dataset 2.

<table>
<thead>
<tr>
<th>Learning Rate</th>
<th>HC+ALO</th>
<th>HC+AO</th>
<th>HC+BOA</th>
<th>HC+CMBO</th>
<th>HC+SSOA</th>
<th>NN</th>
<th>RNN</th>
<th>Bi-GRU</th>
<th>SVM</th>
<th>KNN</th>
<th>HC+SU-CMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>60</td>
<td>0.61625</td>
<td>0.67875</td>
<td>0.72125</td>
<td>0.66625</td>
<td>0.6</td>
<td>0.78625</td>
<td>0.625</td>
<td>0.6375</td>
<td>0.50625</td>
<td>0.8075</td>
<td>0.8365</td>
</tr>
<tr>
<td>70</td>
<td>0.675</td>
<td>0.785</td>
<td>0.763333</td>
<td>0.595</td>
<td>0.658333</td>
<td>0.766667</td>
<td>0.638333</td>
<td>0.6</td>
<td>0.591667</td>
<td>0.793333</td>
<td>0.847333</td>
</tr>
<tr>
<td>80</td>
<td>0.805</td>
<td>0.5675</td>
<td>0.7275</td>
<td>0.755</td>
<td>0.6025</td>
<td>0.7875</td>
<td>0.685</td>
<td>0.6625</td>
<td>0.5</td>
<td>0.7525</td>
<td>0.8175</td>
</tr>
</tbody>
</table>

7. Conclusions

A new hybrid model for the IoT environment is developed using deep learning models such as GRU and Bi-LSTM, where features such as “higher-order statistical features (kurtosis, variance, moments), mutual information, symmetric uncertainty, information gain ratio, and relief-based features” were derived. Further, GRU and BI-LSTM were deployed, where weights of BI-LSTM were optimally chosen via SU-CMO for detecting the attacks. The graph shows that the proposed hybrid classifier + SU-CMO achieves greater accuracy, i.e., ~0.84 for dataset 1 and dataset 2, which is higher than the accuracy of other classic and state-of-the-art models such as hybrid classifier + WOA, hybrid classifier + GWO, hybrid classifier + SLnO, NN, SVM, and DT. The SVM model has accomplished worse outputs than other schemes such as HC + ALO, HC + AO, HC + BOA, HC + CMBO, HC + SSA, NN, RNN, and KNN. Although the suggested model shows a higher accuracy level than other models, there are still improvements to be made in terms of accuracy. It is feasible to enhance the accuracy of the model by using the proposed approach in conjunction with several different kinds of deep learning models. In dataset 2, there are several different kinds of attacks; however, in the present paper, all of these types of attacks are grouped under a single category simply referred to as “attack”. We are considering including the multiclassification of such attacks as part of our future work along with using various kinds of deep learning models.

Author Contributions: Conceptualization, A.S.; methodology, A.S., N.S.G. and P.G.; formal analysis, A.S.; resources, A.S.; writing—original draft preparation, A.S.; writing—review and editing, A.S., N.S.G., P.G. and I.P.; supervision, N.S.G., P.G., J.M.C. and I.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.
Informed Consent Statement: Not applicable.

Data Availability Statement: The data are available on request from the submitting author.

Acknowledgments: The first author acknowledges the university grants commission (UGC) India for providing senior research fellowship (SRF) to carry out this research work.

Conflicts of Interest: The authors declare no conflict of interest.

References


4. Yoon, J. Deep-learning approach to attack handling of IoT devices using IoT-enabled network services. Internet Things 2020, 11, 100241. [CrossRef]


The first author acknowledges the university grants commission (UGC) India for providing senior research fellowship (SRF) to carry out this research work.


32. Kao, L.-J.; Chiu, C.C. Application of integrated recurrent neural network with multivariate adaptive regression splines on SPC-EPG process. J. Manuf. Syst. 2020, 57, 109–118. [CrossRef]

