Three-tier intrusion detection system using a conditional generative adversarial network

Abdullah Rakib Akand*, Saad Waiez Tanveer†, Sheikh Tonmy‡
*Department of Information System and Security, Bangladesh University of Professionals, Dhaka, Bangladesh
†Department of Information System and Security, Bangladesh University of Professionals, Dhaka, Bangladesh
‡Department of Information System and Security, Bangladesh University of Professionals, Dhaka, Bangladesh
Email: abdullahrakibakand@bup.edu.bd

***Abstract*—Security threat protection is crucial in IoT applications since both devices and data can be compromised. We introduced a three-level intrusion detection system using a conditional generative adversarial network (3LIDS-CGAN), which includes initial, secondary, and tertiary IDS, plus attack type classification. The initial IDS classifies packet features, the secondary IDS further refines suspicious packets using signature and anomaly-based methods, and the tertiary IDS detects adversarial packets with CGAN. Experiments showed that the 3LIDS-CGAN model outperforms existing methods in performance.**

***Index Terms*—Conditional generative adversarial network, Firewall, Intrusion detection system, Proximal policy optimization.**

I. Introduction

In IoT environments, vast amounts of data, both sensitive and non-sensitive, are transmitted, making them vulnerable to attacks due to insufficient security and encryption. Intrusion detection systems (IDS) are essential for monitoring network traffic and detecting malicious activities. IDS can be categorized into signature-based and anomaly-based detection. Signature-based IDS (SIDS) detects known attacks, while anomaly-based IDS (AIDS) identifies unknown attacks. However, traditional IDS often struggle with accurately detecting adversarial intrusions that resemble normal traffic.

To address these challenges, hybrid IDS models combining both SIDS and AIDS have been developed. This paper proposes a three-level IDS using a conditional generative adversarial network (3LIDS-CGAN) to enhance detection accuracy and reduce latency. The model includes initial, secondary, and tertiary IDS stages. In the initial IDS, the firewall extracts packet features and classifies them using a support vector machine (SVM) and golden eagle optimization. Suspicious packets proceed to the secondary IDS for further analysis using signature and anomaly-based methods. The secondary IDS refines the classification, using attack history for signature-based detection and event-based semantic mapping for anomaly-based detection.

The tertiary IDS employs CGAN to detect adversarial packets by learning the adversarial environment, ensuring precise detection. This IDS also integrates proximal policy optimization to classify attack types accurately. Several types of attacks, such as DDoS, phishing, and SQL injection, are mitigated by the proposed IDS. The IDS reduces network complexity by filtering packets and decreasing data dimensionality. Experiments conducted using the NS-3.26 network simulator show that the 3LIDS-CGAN model outperforms existing methods in detecting various attacks.

The proposed IDS offers a robust solution for securing IoT environments, demonstrating superior performance across multiple metrics. The paper concludes with a discussion on future work to further enhance IDS capabilities.

II. Related Works

Intrusion detection with hybrid sampling using deep hierarchical network was proposed in [?], aiming to balance network traffic data through a hybrid method incorporating OSS and SMOTE algorithms. Data preprocessing involves deep hierarchical network usage, followed by classification using hierarchical network comprising CNN and Belts. Evaluation of the proposed system is conducted using NSL-KDD and UNSW-NB15 datasets.

A lightweight intrusion detection system for IoT environment was proposed, consisting of training and evaluation sections with feature extraction and SVM classification [?]. Hybrid neural network-based anomaly detection, proposed in [?], encompasses five phases including flow mapping, sequence packet features extraction, and environmental features extraction.

An ensemble-based intrusion detection system for IoT environment integrates signatures-based and anomaly-based intrusion detection using C5 and one-class SVM [?]. A hybrid intrusion detection method using PCA-GWO and DNN tailored for IoT environments involves preprocessing with one-hot encoding and normalization to reduce dataset dimensionality [?].

Ensemble-based modified adaptive boosting algorithm by Atefi et al. [?] targets imbalance in network intrusion detection, introducing M-Adaboost-A-SMV and M-Adaboost-A-PSO. Wireless intrusion detection employing improved convolution neural network (ICNN) features preprocessing, training, and classification [?]. Intrusion detection using improved genetic algorithm (GA) and Deep Belief Network (DBN) for IoT environments aims at optimal solution detection and network attack classification [?].

Optimization-based hybrid IDS by Rose et al. [?] combines BGWO with statistical algorithms like naïve Bayes for optimal

IoT network intrusion detection. A novel intrusion detection technique combines multi-objective genetic algorithm (NSGA-II) and artificial neural network (ANN) with a decision tree-based random forest classifier for effective anomaly detection [**?**]. Mapping activity of log data using heuristic miner algorithm and subsequent enhancement phase for placement model acquisition [**?**].

Kumar and Harikiran [**?**] propose a privacy preservation approach using a prediction algorithm within a deep neural network to anonymize video content and recognize privacy-preserved actions. Anomaly detection for vibration data in city trains utilizing generative adversarial network with spectral density evaluation and training via long short-term memory algorithm by Kim et al. [**?**].

III. 3LIDS-CGAN MODEL

Our proposed system focuses to detect both signature and anomaly-based intrusions in an IoT environment. It has four consecutive phases such as: i) first level IDS, ii) second level IDS, iii) third level IDS, and iv) attack type classification. The real-time packets entering the IDS model contain several existing and new attacks which are identified with improved accuracy. The first phase classifies incoming packets into three classes namely normal, suspicious, and malicious from which the malicious packets are dropped and suspicious packets are sent to the second phase in which the signature-based and anomaly-based IDS takes place which results in classification of those packets into normal and malicious from which the malicious packets are dropped. The normal packets from first and second phases are processed in the third phase to detect the adversaries to improve the security of the IoT environment.

IV. FIRST TIER IDS(INTRUSION DETECTION SYSTEM)

In first level IDS, first process is packet flow-based feature extraction. The packet features are extracted by using firewall which filters the incoming packets with the features of packet interval time, packet size, packet type, payload length, and timestamp. The extracted features are classified by using SVM and golden eagle optimization which is used to select the kernel function of the SVM like linear, polynomial, radial basis function (RBF), and sigmoid which has four parameters such as cost, gamma, coefficient, and degree. Intrusions are detected based on the extracted features from the firewall. The SVM is already with the normal patterns. Every new packet is matched to the normal patterns if it varies from the threshold then it is marked as intrusion or attack. In this research, we used multiclass SVM for classification. In SVM hyperplane is used to classify the features into three classes, that hyperplane needs to follow the rule in (1),

$$F(y) = (v, y) + a \tag{1}$$

where v represents the normal vector and a represent the bias value and y represents the test sample. In SVM the intrusion detection is performed by selecting optimal kernel for that we proposed golden eagle optimization which selects the optimal kernel from the four kernels (linear, RBF, sigmoid, and polynomial) of SVM. The classification function of SVM is defined as (2).

$$F(y) = \begin{cases} -1, & \text{if } y \in \text{malicious} \\ 0, & \text{if } y \in \text{suspicious} \\ 1, & \text{if } y \in \text{normal} \end{cases} \tag{2}$$

In next stage of SVM, assume $y_2, y_2, \ldots, y_n$ be a training sample. And then, separate the data from the origin for that we need to solve the quadratic programming problems.

$$\min \frac{1}{2}\|W\|^2 + \frac{1}{vn}\sum_{i=1}^{n} E_i - P_n \tag{3}$$

$$W \times \varphi(y_i)) \geq \sigma - Eii = 1, 2, ..., Ei \geq 0 \tag{4}$$

If W and σ solve the quadratic programming problem, then the decision function will be normal for maximum instances in the training set.

$$F(y) = Sign(\,(W \times \varphi(y_i)) - \sigma) \tag{5}$$

This research used RBF kernel function which is selected by golden eagle optimization which optimizes the parameters of c and γ. Every kernel has specific parameters that can be enhanced to get the best performance result which is illustrated in Table 1. SVM identifies the behavior of the normal packets using extracted features. It proposed that SVM classifies the current packets into normal, malicious, or suspicious.

TABLE I
TYPES OF KERNEL FUNCTIONS AND THEIR PARAMETERS

Function of Kernel	**Equation**	**Parameter**
RBF	$k(y_n, y_i) = exp(-\gamma\|y_n - y_i\|^2 + c)$	$c\ and\ \gamma$
Linear	$k(y_n, y_i) = (y_n - y_i)$	$c\ and\ \gamma$
Polynomial	$k(y_n, y_i) = (\gamma(m(y_n - y_i) + s)^b$	$c, \gamma,\ s\ and\ b$
Sigmoid	$k(y_n, y_i) = tanh(\gamma((y_n, y_i) + s)$	$c, \gamma\ and\ s$

Where c represents cost and γ denotes gamma and s represents coefficient and b represents degree. For getting the optimal value from the kernel, the search method is performed using the parameters $c, \gamma,\ s\ and\ b$.

The first process of golden eagle optimization is defined as follows. The attack is modeled through a vector beginning from the current position of the golden eagle; the attack vector is calculated as (6):

$$\overrightarrow{a_i} = \overrightarrow{y_f}^* - \overrightarrow{y_i} \tag{6}$$

Where, $\overrightarrow{a_i}$ is represent the eagle i attack vector and $\overrightarrow{y_f}^*$ represent the best location visited by eagle f, and $\overrightarrow{y_i}$ represent the current location of the eagle i. Next process is to calculate the cruise vector concerning attack vector. Cruise vector is a tangent vector to the circular and that is positioned perpendicular to the attack vector. The tangent hyperplane is calculated as (7),

$$H_1 y_1 + ...H_n y_n = D \rightarrow \sum_{j=1}^{n} H_j y_j = D \tag{7}$$

Where, $\overrightarrow{h}$ = $[H_1, ...H_n]$ represent the normal vector and Y = $[y_1, \dots y_n]$ represent the variable vector, D = $\overrightarrow{h}$. S . S represent the arbitrary point. Therefore, the hyperplane is represented as (8),

$$\sum_{j=1}^{n} A_j y_j = \sum_{j=1}^{n} A_j^t y_j^* \tag{8}$$

Where, $\overrightarrow{a_i}$= $[A_1, ..., A_n]$ represent the attack vector and $Y^* = [y_1^*, ..., y_n^*]$ is represent the location. The new position of the eagle is defined as (9),

$$\triangle y_i = \overrightarrow{R_1} P_A \frac{\overrightarrow{a_i}}{\|\overrightarrow{a_i}\|} + \overrightarrow{R_2} P_b \frac{\overrightarrow{B_i}}{\|\overrightarrow{B_i}\|} \tag{9}$$

where, P_A^t is represent the attack coefficient at iteration t and P_b represent the cruise coefficient and $\overrightarrow{R_1}$ and $\overrightarrow{R_2}$ represent the random vectors between the interval of [0,1]. And $\|\overrightarrow{A_i}\|$ and$\|\overrightarrow{b_i}\|$ represent Euclidean norm of attack that is defined as (10).

$$\|\overrightarrow{A_i}\| = \sqrt{\sum_{j=1}^{n} A_j^2} and \|\overrightarrow{b_i}\| = \sqrt{\sum_{j=1}^{n} b_j^2} \tag{10}$$

The position of the eagle is calculated as (11).

$$y^t + 1 = y^t + \triangle y_i^t \tag{11}$$

The fitness of new position of eagle i is better than the current position; hence the new position is updated. In this algorithm, P_A, P_b is used to shift from exploration to exploitation. Initially, this algorithm starts with minimum P_A and maximum P_b. Starting and finishing parameters are determined by the user and intermediate values are calculated using the linear function, which is defined as (12),

$$\begin{cases} P_A = P_A^0 + \frac{t}{T}|P_A^T - P_A^0| \\ P_b = P_b^0 + \frac{t}{T}|P_b^T - P_b^0| \end{cases} \tag{12}$$

where t represents the current iteration and T represents maximum iteration and P_b^0 and P_b^T represent the initial and final values of attack (P_b) and P_A^0 and P_A^T represent the initial and final values of attack (P_A). Finally, the best kernel function is selected using this algorithm. In our work, RBF is selected as the best kernel for classification. Our proposed system performs accurate classification which improves the accuracy of the process. The firewall ignores the malicious packets and forwards suspicious packets into next-level IDS. The pseudo-code for first-level IDS is provided below in which the selection of kernel is described above in an elaborative manner.

Pseudocode for support vector machine (SVM) with golden eagle optimization

Algorithm 1 Apriori

Input:
D: transaction database;
Min_sup: the minimum support threshold
Output: frequent itemsets

Description:

```
$L_1$= find_frequent_1-itemsets(DB);
for (k=2; $L_{k-1} = \varphi; k++$) {
$C_k$= Apriori_gen($L_{k-1}$);
for each transaction $t \in DB$ {    //scan DB for counts
$C_t$ = subset($C_k, t$); //get the subsets of $t$ that are candidates
for each candidate $c \in C_t$
$c.count++$;
}
$L_k = \{c \in C_k | c.count \geq min\_sup\}$
}
return $L = \bigcup_k L_k$;
Procedure Apriori gen($L_{k-1}$: frequent($k-1$)-itemsets)
```