An Optimized Deep Learning Based Security Enhancement and Attack Detection on IoT Using IDS and KH-AES for Smart Cities

Ayyer DURAISAMY[1]*, Muthusamy SUBRAMANIAM[2], Chinnanadar Ramachandran RENE ROBIN[3]

[1] Department of Computer Science and Engineering, University College of Engineering, Arni, Thatchur - 632326 Tamil Nadu, India
durai.ucea@gmail.com (**Corresponding author*)

[2] Department of Computer Science and Engineering, SRM Institute of Science and Technology, Vadapalani Campus, Chennai - 600026, Tamil Nadu, India
subbu.21074@gmail.com

[3] Department of Computer Science and Engineering, Sri Sairam Engineering College, Chennai - 602109, Tamil Nadu, India
crrenerobin@gmail.com

Abstract: Today, smart cities are being built with the wide deployment of the Internet of Things (IoT). Smart cities (SCs) set out in real time to ameliorate the quality of human life in respect of efficiency and comfort. Security along with privacy are the main issues in most SCs. The IoT-centric frameworks impose certain security threats on smart city applications as they are susceptible to security issues. On this account, an Intrusion Detection System (IDS) is requisite for mitigating the IoT-associated security attacks which take advantage of certain security vulnerabilities. The aim of this paper is to improve the security and attack detection rate as early as possible. In existing works, the accuracy of the attack detection rate and security are the main challenge. To overcome any drawbacks, this work proposes an IDS for detecting the IoT attacks in a city centered on the DLMNN classification. First, the sensor values from a SC are sent to the IDS system (the training phase), which is utilized for testing the respective values. Next, the preprocessing step is performed, and then feature selection (FS) is carried out with the utilization of the Entropy-HOA method. Further on, the classification using DLMNN is performed for detecting the IoT attacks. Then, the results of the classification are analyzed and the attack is identified. Next, a secure data sharing task is performed by using the KH-AES algorithm. Last, the resulting data is forecast. The weights for each layer of the DLMNN have a high impact on the classifier's output. The comparison of the existing technique and of the proposed technique with regard to FS, classification and secure data sharing reveals that the proposed technique obtained the best results.

Keywords: Intrusion Detection System (IDS), Deep Learning Modified Neural Network, Internet of Things (IoT), Krill Herd (KH), AES (Advanced Encryption Standard), Humming Bird Optimization Algorithm, Smart Cities.

1. Introduction

Massive advances in IoT concepts and telecommunication networks bring about incredible developments in the everyday usage of electronics applications and services (King & Awad, 2016). IoT Applications using Machine Learning with selective adversarial samples and techniques are presented in (Khoda et al., 2020). The smart network, IoT, links everything to the Internet for trading data with specific authorized users (Chen et al., 2014). IoT interconnects small devices, real-life sensors, and even systems for acquiring deeper automation, integration and exploration (Zarpelão et al., 2017). Beside planning the long-standing development in the SC, gathering their information for the day-to-day administration of activities is also requisite (Talari et al., 2017).

IoT commences an extent of applications, namely smart homes, health monitoring, smart environment, SC, etc, from this smart city application is developed with Visualized Botnet detection system based Deep Learning for the Internet of Things (Vinayakumar et al., 2020). The merits of Information and Computing Technologies (ICT) in an IoT and the SC are tremendous (Elmaghraby & Losavio, 2014).

In general, SC schemes aim at ameliorating metropolitan infrastructure planning, automating urban processes, reducing the cost of equipment, augmenting city competitiveness, and ameliorating transparency (Garcia-Font, Garrigues & Rifà-Pous, 2017). The joint Traffic control and multi-channel reassignment for core backbone network in IoT using Multi-Agent Deep Reinforcement learning approach (Wu et al., 2020) and signal authentication and security system is developed in (Ferdowsi & Saad, 2019). They regard the IoT networks as the chief target for cyber-attacks. IDSs perceive policy violations or malicious actions by monitoring network traffic. The systems linked through the Internet, are facing an imperative challenge of privacy and security. The privacy –aware offloading in IoT with deep learning technique is

https://doi.org/10.24846/v30i2y202111

presented in (He, Jin & Dai, 2019). IoT systems could be accessed from any place through the untrusted network, that is, Internet. This makes the IoT networks unsecure in case of malicious attacks. The confidential information might be exposed anytime if the security-related issues are unaddressed. On this account, addressing the security issues is highly requisite. 3D Finger Knuckle Identification is proposed in (Cheng & Kumar, 2020). The cognitive privacy middleware for environmental IoT using deep learning technique is presented in (Elmisery, Sertovic & Gupta, 2017). The deep learning (DL) methodology for cyber-security is performed so as to enable the recognition of attacks on social IoT. The DL model's performance was contrasted against the machine learning approach, and also attack detection was performed over the centralized system. Reconfigurable cryptographic Cortex-M0 Processor with memory computing for IoT Security application is analysed in (Zhang et al., 2018). A SC approach encompasses 2 notable security challenges like (i) how to discover the zero-day attacks which take place from disparate protocols of IoT systems in a SC's cloud data center, presuming that the countless attacks are concealed in IoT devices, (ii) how to ascertain an approach for intelligently detecting cyber-attacks (Elmaghraby & Losavio, 2014). Securing IoT Against CPA (Correlation Power Analysis) Attacks using AES with masked approach is presented in (Yu & Köse, 2017). The developers needed to improve disparate techniques for recognizing infected IoT devices. This research contributes to improving the smart city application widely and to enhancing the optimization-based AES for an effective analysis.

In the proposed work, the deep learning-based security enhancement and attack detection are performed. The AES approach is analyzed with a novel strategy called IDS and KH-AES. Krill Herd optimization is used on the feature section of Deep learning neural network model. This is designed for smart city application-based utilities. This paper is organized as follows. Section 2 presents the literature review of various studies and techniques. Section 3 sets forth the proposed algorithm and process flow. Section 4 discusses the experimental results and presents a comparative analysis of attack detection. Section 5 concludes the proposed work efficiency in attack detection of smart cities with suggestions of possible future applications.

2. Literature Review

Various research works are analyzed and reviewed for obtaining best security performance for IoT application. Here various optimization approaches are discussed and reviewed based on performance metrics evaluation.

Deng et al. (2018) detailed the security issues of IoT and also emphasized the necessity for IDS. Disparate sorts of IDSs were analysed, and their IoT application was modelled. The uses of disparate IDSs were contrasted and a plan for the subsequent research phases was made. Network IDS had become a hot problem by utilizing the ML technique and also data mining.

Li et al. (2019) presented a generic approach of collaborative blockchained signature-based IDSs termed CBSigIDS, which could generate and update a trust (key) signature DB in a collaborative environment of IoT. The proposed approach could offer verified distributed structures deprived of the requisite of the trusted intermediaries.

Dawoud et al. (2018) presented a secure IoT approach centered on Software Defined Networks (SDN). Here, the process was generalized for the incorporation of SDNs to IoT. An IDS centered on DL was deployed and a Restricted Boltzmann Machine was employed for the detection approach.

Bu, Isakov & Kinsy (2019) planned and applied a secure and robust strategy for facilitating sensitive information-sharing in IoT. It had 2 noteworthy features that is: i) the scheme utilized Threshold secret sharing for separating the data onto shares to be held by every device in the structure. Here, the data could solely be retrieved by the collaborative groups of devices. ii) The methodology ensured the integrity and privacy of data albeit there were countless collusive and sophisticated attackers who could take control over the device.

Ghosh & Grolinger (2021) have presented the edge-cloud computing for Internet of Things based Data Analytics review. Here the embedded intelligence approach for the Edge cloud is performed based on Deep Learning approach. SDN-centered adaptive approach for defending SC applications against DoS attacks is termed SEAL (SEcures and AgiLe).

Bui et al. (2017) have discussed the AES Datapath Optimization Strategies for Low-Power and Low-Energy Multi-security-Level Internet-of-

Things Applications. The work proposed an IoT framework for addressing the loT cyber-threat in a SC, which was intelligent in many application-specific frameworks.

3. Proposed Methodology

In the proposed work, the sensor values are provided to the IDS for testing. The IDS system is trained to perform testing, which uses the dataset, namely NSL-KDD (Network Security Laboratory- Knowledge Discovery and Data Mining) for training. It has several steps. In the initial stage of preprocessing, the input data is converted into crisp data and it is bifurcated into six attack types, called DoS, U2R, Probing, R2L, normal and unknown. Then Min-Max Normalization is performed and it classifies the attacks into attacked and non-attacked (Normal data). This analyses normal and abnormal rating; if it is abnormal, issue is predicted in SC.

Subsequently, the result is evaluated and analogized to the existing system. The proposed method can be explained by means of the structural design in Figure 1.

3.1 IoT Sensor Values from Smart Cities

Sensing is at the hub of smart infrastructures, which can check themselves and function based on their intelligence. Utilizing sensors to observe public infrastructures, for instance roads and bridges, along with buildings, provides cognizance which facilitates a more efficient utilization of resources, centered on the data amassed by means of these sensors. Here the NSL-KDD dataset is used for this analysis. The values of SC are taken for training state of IDS for detecting the attacks. The IDS is trained for detecting the attacks in IoT.

3.2 Intrusion Detection System

Data collection is the initial and important step for intrusion detection. The data source type and accumulated data location are the two determinate components in the design of IDS. In order to provide the best suitable protection to the focused host or networks, these evaluation metrics are suggested. In the research area of IDS, numerous researchers are using different datasets for their analysis on the intrusions' detection. The KDDCUP, ISCX, and also NSL-KDD are the extensively used datasets.

3.2.1 Attack Data Set

In the proposed work, NSL-KDD dataset is utilized for IDS, which is in the CSV format for model validations. The dataset has normal and attack data, which exhibits the traffic compositions with intrusions. The dataset encompasses various sorts of attacks as follows. But, the actual data-set comprised [125,973] records and [22,544] records of training and testing states, respectively. In this work, eighty (%) of data is employed for training, and twenty (%) data is used for testing reasons.

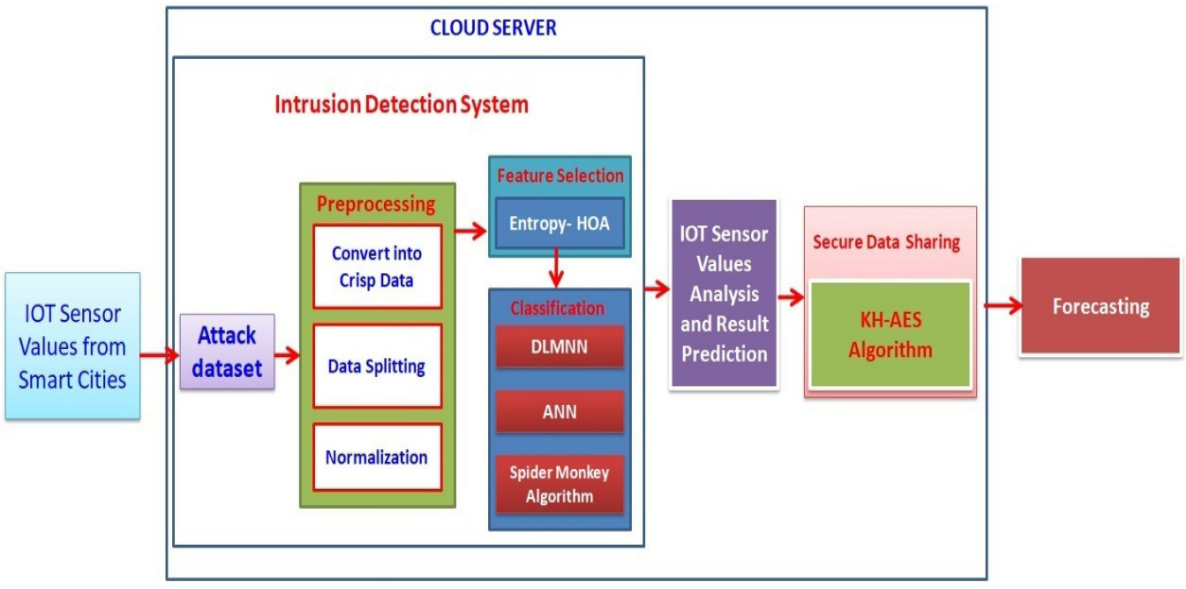

Figure 1. Architecture of the Proposed Scheme

3.2.2 Preprocessing

In the proposed work, the preprocessing step includes three steps. The crisp data conversion, splitting, and normalization are performed in this process. First, the input data is transformed into crisp data, and then converted data is seperated into attack type. Afterwards, the normalization step is carried out.

3.2.2.1 Normalization

This process scales the attribute data to fit them to a particular gamut. Though there are several sorts of normalization techniques, Min-Max Normalization is the one utilized here. This transmutes D to N_{Ra} that fits to the gamut [0.1] and is written as

$$N_{Ra} = \left(\left(\frac{(D - D_{Mn})}{(D_{Mx} - D_{Mn})} \right) * (1-0) + 0 \right) \quad (1)$$

where, N_R denotes the normalization, zero and one denote the range and D_{Mx} and D_{Mn} are the maximum and minimum values of data.

3.2.3 Feature Selection

Eliminating the insignificant features and concentrating only on the important features may increase the execution speed of the process. FS technique simplifies a dataset by reducing its dimensionality and identifying pertinent underlying features without sacrificing prognostic accuracy. In the proposed work, hummingbird optimization algorithm (HOA) is used for FS. The hummingbird algorithm uses entropy model rather than levy flight behavior. So this algorithm is named Entropy – HOA.

3.2.3.1 Entropy-HOA

A stochastic optimization approach that emulates the foraging behavior of hummingbirds is HOA. The qualities of food sources are considered as the fitness function values, whereas the best food source is regarded as the optimal solution. The HOA embraces '2' phases that is i) self-searching ii) guided searching**.** The initialization of HOA is done by the following formula

$$Q_i = U_b - R_a \cdot (U_b - L_b) \quad (2)$$

where Q_i is the position of the *i*th hummingbird in the population ($i \in 1,2,...N$), N is the populace size, and L_b and U_b signify the lower bound and the upper bound of the variables in the search space, respectively. R_a represents a random number betwixt zero and one.

a) Self Searching Space

Consider N the number of hummingbirds with populace size $i = 1,2,3...N$ with D dimensional variables in its search space. Presume that $Q_i^t = Q_{i,1}^t, Q_{i,2}^t, ...Q_{i,D}^t$ denotes the *i*th hummingbird in *t*th generation. Here, hummingbirds, based on their own experience, look for better food sources. When they incessantly explore for a better food source, the position of each hummingbird is updated based on its former gradient information

$$Q_i^{t+1} = Q_i^t + R_a \cdot \left(Q_i^t - Q_i^{t-1} \right) \quad (3)$$

Here, Q_i^t and Q_i^{t-1} represent the positions of the *i*th hummingbird at the *t*th and *t-1*th generations, respectively, and R_a is a random value in the gamut [0, 1]. The new position of the hummingbirds is produced by applying the entropy model as follows

$$E(Q) = \sum_{i=1}^{N} Q_i \, log_2 Q_i \quad (4)$$

Here, $E(Q)$ is the entropy of Q, and N denotes the population size.

b) Guided-Searching Phase

Here, the current best bird is termed as the territory bird (TB), whereas the others are alluded to as following birds (FB). This sort of behavior is formulated as:

$$Q_i^{T,t+1} = Q_i^{T,t} + R_d \cdot \lambda \quad (5)$$

Here, $Q_i^{T,t}$ signifies the TB's position in a *t* generation; R_d denotes an arbitrary value in the gamut [−1, 1], whereas λ stands equal to $0.1 \cdot (U_b - L_b)$. Also, the FBs move in 2 ways.

When the FBs could not spot the TB, they quickly move toward the next TB

$$Q_j^{F,t+1} = Q_j^{F,t} + R_a \cdot \left(Q^{T,t} - MF . Q_j^{F,t} \right) \quad (6)$$

where MF refers to maximum frequency search of whole population.

When the FBs find the TB, the FBs randomly fly around. In the process of escaping, the FB j randomly picks its companion $s(s \neq j)$ to follow.

The bird j moves toward bird s if the position of bird $s(Q_s)$ is better. But, if Q_s is worse, then bird j travels away from bird s. Forbye, this process is evaluated as

$$Q_j^{F,t+1} = Q_j^{F,t} + R_a \cdot \left(Q_s^{F,t} - Q_j^{F,t}\right) \text{ if } \mathrm{F_i}Q_s^{F,t} \le F_i Q_j^{F,t} \tag{7}$$

$$Q_j^{F,t+1} = Q_j^{F,t} - R_a \cdot \left(Q_s^{F,t} - Q_j^{F,t}\right) \text{ if } \mathrm{F_i}Q_s^{F,t} \ge F_i Q_j^{F,t} \tag{8}$$

Combining the '2' states, the movement pattern of the FBs is written as

If $PF^t \ge R_a$

Perform step 1 using equation (6)

Else, perform step 2 using equations (7) and (8) where PF^t indicates the probability that the FBs are noticed by a TB and is evaluated as

$$PF^t = \frac{R_k\left(F_i Q_j^{F,t}\right)}{N-1} \tag{9}$$

Where $R_k\left(F_i Q_j^{F,t}\right)$ signifies the rank of the FB j amongst other FBs in the populace. The border control strategy employed to avert an invalid search could be expressed as:

$$Q_{i,d}^t = L_b - R_a \cdot (U_b - L_b) \text{ if } Q_{i,d}^t < L_b \text{ or } Q_{i,d}^t > U_b \tag{10}$$

3.2.4 Classification using DLMNN

The above-selected features are classified with the utilization of DLMNN. This network utilizes the SMO algorithm for optimizing its weights. The subsequent hidden layer (HL) has countless hidden nodes. Those nodes sum the product values of the input and weight vector linked to each input. The weight functions of DL-Neural Network are well-optimized by employing the SMO algorithm. The activation operation is performed and its output is taken from successive layer. The performance of the DLMNN is further expressed in Figure 2. The DLMNN classification entails the steps below:

Step 1: Consider the Feature values of disparate data and the respective weights, and express them as

$$F_i = \{F_1, F_2, F_3 \ldots\ldots R_n\} \tag{11}$$

$$W_i = \{W_1, W_2, W_3 \ldots\ldots W_n\} \tag{12}$$

Step 2: Find the product values of the inputs and the arbitrarily picked weight vectors and then sum up those product values.

$$S = \sum_{i=1}^{n} F_i W_i \tag{13}$$

Where, S - Sum value, F_i - Input and W_i - Weight.

Step 3: Ascertain the 'activation function' (AF), which is written mathematically as follows.

$$M_i = f\left(\sum_{i=1}^{n} F_i W_i\right) \tag{14}$$

$\mathrm{M_i}$ denotes Motion induced on activation function. The AF (Ai) category has Gaussian function (f(R)), which is expressed as

$$f(R) = e^{-F_i^2} = A_i \tag{15}$$

Step 4: State the next HL's output as

$$Y_i = B_i + \sum C_i W_i \tag{16}$$

where B_i - Bias value, W_i - Weight between the input and the HLs and C_i - Values modified while applying the AF.

Step 5: Perform the aforesaid 4 steps for each layer in the DLMNN and evaluate the output unit by summing up the input signals' weights for obtaining the value of the output layer neurons

$$O_i = B_i + \sum P_i W_j \tag{17}$$

where P_i - Value of the layer that precedes the output one, W_j - Weights of the HL, O_i - Output unit

Step 6: Contrast the network output and the target value and evaluate the value of the difference between them. The resulting difference value is termed as the error signal and is mathematically written as

$$E_r = D_i - O_i \tag{18}$$

Here, E_r - Error signal, D_i - Aimed target output and O_i - Classifier's current output.

Step 7: Contrast the output unit and the targeted value and ascertain the related error. Based on this error, δ_i value is evaluated. δ_i is employed for allocating the error at the output back to other units on the network.

$$\delta_i = E_r[f(O_i)] \tag{19}$$

Step 8: Utilize the BP methodology to assess the weight correction and this relation is rendered as

$$wr_i = \alpha\delta_i(F_i) \tag{20}$$

Here, wr_i - Weight correction, α - Momentum term, (F_i) - Input vector and δ_i - Error disseminated in the network.

3.2.4.1 Spider Monkey Algorithm (SMO)

A populace-centered stochastic algorithm that is defined by the smart food foraging behavior of spider monkey (SM) is the SMO algorithm. The steps in the SMO are as follows.

Step 1: Instigate the populace of N SMs. This population is implied as D-dimensional vector S_i, where $i = 1,2,3...N$. Also, initialize every S_i with the below formula

$$S_{ij} = S_{Mj} + \varphi \times \left(S_{Xj} - S_{Mi}\right) \tag{21}$$

$$\gamma_i = \gamma_i^{best} + B_i^{best} \tag{22}$$

$\phi \in (0,1)$, S_{Mj} and S_{Xj} denote the lower and upper bounds of S_i in jth direction. Here, V_f - Foraging speed, B_i^{best} -Best solution, and ω_f - Inertia weight for foraging.

Step 2: Local Leader Phase (LLP)

This is the next phase. Based on the experience of LL together with that of group members, SMO modernizes its current location. After contrasting the fitness new location with the present location, it implements greedy selection.

Step 3: Global Leader Phase (GLP)

This phase starts after finishing LLP. The SMO modernizes its position based on the GL's experience in addition to that of the local group members. It is evaluated with the equation below

$$S_{Nij} = S_{ij} + rand[0,1] \times \left(GP_i - S_{ij}\right) + rand[-1,1] \times \left(S_{ri} - S_{ij}\right) \tag{23}$$

Here, GP_j implies the GL's location in the 'jth' dimension and $j \in \{1,2,....D\}$ indicates arbitrarily picked value and 'S_i' updates the location. The probability is computed by using its fitness and is expressed as

$$P_i = 0.9 \times \frac{F_i}{F_{max}} + 0.1 \tag{24}$$

Step 4: Global Leader Learning

The GL modifies its location utilizing certain greedy schemes. Best fit solution in the present swarm is picked as a GL. It checks whether the GL's position is modernized or not and accordingly manipulates the global limits.

Step 5: Local Leader Learning (LLL)

The LL utilizes certain greedy approaches to modify its position. Best fit solution existent in the swarm is picked as a Global Leader. It validates whether the LL's position is modernized or not and consequently manipulates the local limits.

Step 6: LL Decision (LLD)

The LLD phase assists in the group re-initialization when its LL does not update its position to a certain Local Leader limit. This re-initialization leads the group members in the feasible region into the infeasible one.

Step 7: GL Decision

When the GL is not updated for a certain Global Leader limit, the entire swarm is bifurcated to groups. Here, Global Leaders are picked for newly created sub-groups with the aid of GLD process.

3.3 Result of Analysis State and Prediction State

In the classification phase the attacks are classified into 2 disparate datasets called attacked and non-attacked. The normal data is utilized for prediction of other kinds of issues like traffic management, structural health monitoring, pollution prevention, parking optimization, waste management, intelligent transportation and smart buildings.

3.4 Secure Data Sharing Using KH-AES

The predicted results are sent to the user using Krill Herd – Advanced Encryption Standard. The AES is optimized using krill herd optimization algorithm as a key.

3.4.1 AES Approach

This algorithm is useful for encrypting secret text into a decryptable format. AES that stands as a block cipher was found to have a block length of 128bits.

Step 1: Perform SubBytes – It is basically a nonlinear substitution which swaps every byte with another one as per the lookup table (S-box).

Step 2: Execute ShiftRows – It signifies transposition which cyclically shifts all rows for a specific number of times.

Step 3: Perform MixColumns – It is a mixing function that functions at the level of the columns by joining the 4-bytes in every column.

Step 4: Execute AddRoundKey – It merges every byte with the Round Keys (RK). Here, each RK is a resultant of the cipher key according to a key schedule.

3.4.1.1 Krill Herd Optimization Algorithm

This algorithm, the current heuristic optimization method, follows the conduct of krill individuals (KI). The chief goals of this algorithm are: (1) elevating krill density, and (2) acquiring food. The aspects like: a) Movement stimulated by other KI, b) Foraging activity as well as, c) Random diffusion, highly influences the KI's location. On this account, the Lagrangian model (differentiations d/dt of krill location (Yi)) is utilized for expressing the krill's location:

$$\frac{dYi}{dt} = M_i + G_i + D_i \tag{25}$$

Here, M_i - Motion of other KI, G_i - Foraging motion and D_i - *i*th KI's physical diffusion. This algorithm has the following steps:

Step 1: During motion, the direction of a KI's movement is evaluated by the density of a) local swarm, b) target swarm, and c) repulsive swarm, which is written as:

$$M_i^{new} = M^m \alpha_i + \omega_n M_i^{old} \tag{26}$$

where M^m signifies the maximal induced speed, and 'M_i^{old}' represents the motion inertia weight that is in the gamut [0, 1] along with the last motion from optimization, respectively. Here, the gauge α_i is given as

$$\alpha_i = \alpha_i^{loc} + \alpha_i^{tar} \tag{27}$$

Here, α_i^{loc} - Local effects of neighborhood of *i*th KI, α_i^{tar} - Best solution direction as of the *i*th KI and α_i^{tar} is evaluated as

$$\alpha_i^{tar} = C_b K_{ib} X_{ib} \tag{28}$$

Here, coefficient C_b renders the effective α_i^{tar} for the *i*th KI and is evaluated as

$$C_b = 2\left(\frac{r+1}{I_{ma}}\right) \tag{29}$$

Where *r* and I_{ma} are the coefficient factors of local leader phase.

Step 2: Evaluate the foraging behavior as

$$G_i = V_f \gamma_i + \omega_f G_i^{old} \tag{30}$$

where $\gamma_i = \gamma_i^{best} + B_i^{best}$ (31)

"Vf", "γi" and "ωf" are the factors used on the global leader phase.

Step 3: In the physical diffusion of the KI which stands as an arbitrary process, the motion is associated to D_i and δ, which is evaluated as

$$D_i = D_m \delta \tag{32}$$

Here, D_i - Maximal diffusion speed, δ - Arbitrary directional vector, whose arrays are in (-1, 1). So the KI position can be given by

$$Y_i(t+\Delta t) = Y_i(t) + \Delta t \ \frac{dY_i}{dt} \tag{33}$$

Here, Δt signifies a scale factor of the speed vector and its value is completely contingent on the concerned search space. Also, it must be adjusted with regard to the optimization issue.

3.5 Forecasting

It points to the advancement of possible scope, directions, intensity, along with speed of the environmental change for emphasizing the evolutionary path of the future alterations. Here, the encryption values are decrypted via reversing encryption. After the process of secure data sharing, the outcomes of the normal data are clearly displayed using the LCD monitor. For displaying these results, other disparate issues in SC are easily detected.

4. Experimental Results and Discussion

The proposed work uses in classification FS, and secure data sharing is analyzed based on its performance. This work utilizes the NSL-KDD dataset for the implementation of IDS. The dataset encompasses 125,973 records for training along with 22,544 records for testing procedures; each with forty-one features. The performance of the proposed technique is analysed based on three phases such as: a) Feature Selection (FS), b)

classification and c) Secure data sharing. The proposed framework is effectively executed and calculated by Java.

4.1 Analysis of Entropy-HOA

The proposed work uses Entropy-HOA method for FS. It is contrasted with various existing techniques for instance, Particle Swarm Optimization Algorithm (PSO), Cuttlefish Optimization Algorithm (CFA) and Genetic Algorithm (GA) in respect of fitness values (FV). The FVs for 5 numbers of iterations are compared. The comparative values obtained by the proposed and existing approaches are illustrated in Table 1.

Table 1 shows the comparative results for different optimization algorithms. For the first 5 iterations, the existing CFA, GA, and PSO obtain a FV of obtain a FV of 6.4178, 7.2584 and 8.1245, respectively. But the proposed Entropy-HOA gives 10.2354 as a FV, which is higher than the values obtained by other techniques. The FV for every technique increases whenever the number of iterations increases, but the Entropy-HOA method gives the highest FV for all 5 to 25 iterations in contrast with the other techniques. Table 1 can be graphically illustrated by means of Figure 2.

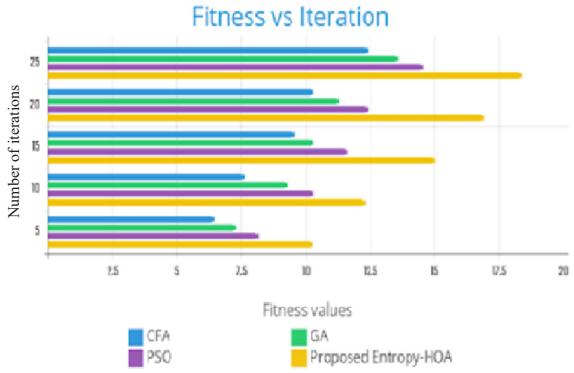

Figure 2. Performance Comparison for the fitness value

4.2 Analysis of DLMNN Classifier

The DLMNN classifier is used for classifying the attacks on IoT. The proposed DLMNN classifier is contrasted with other existing techniques for instance K-Nearest Neighbor (KNN), Naive Bayes (NB), Support Vector Machine (SVM), and Artificial Neural Network (ANN). The metrics that are utilized for the evaluation are F-Measure, sensitivity, and accuracy. The comparisons between the performance of the proposed and existing techniques are outlined in Table 2. Table 2 exhibits the metric values acquired by the proposed DLMNN classifier and the existing SVM, NB, KNN, and ANN.

The graphical portrayal of Table 2 is illustrated in Figure 3. Figure 3 illustrates the performance obtained by the proposed DLMNN and existing classifiers. Figure 3(a) contrasts the proposed DLMNN classifier with the existing techniques with regard to sensitivity. The number of nodes mentioned in Table 2 ranges from 20 to 100 regarding sensitivity. Figure 3(b) contrasts the DLMNN and other existing techniques with regard to accuracy. In this case, for 20 nodes, DLMNN provides an accuracy of 90.3%, but the existing KNN, SVM, NB, and ANN give 88.12%, 87.56%, 84.12% and 86.78%". So it clearly displays that the DLMNN achieves the uppermost accuracy for every node compared with other existing techniques. The F-measure is compared for DLMNN and existing techniques as it is illustrated in Figure 3(c). Based on the Figure 3(c), the proposed classifier gives highest F-measure for every number of nodes when compared with the existing classifiers.

4.3 Performance Analysis of KH-AES

For secure data sharing, the proposed work utilizes KH-AES algorithm. The proposed KH-AES is

Table 1. Comparison Results

No. of iterations	Fitness Value			
	CFA	GA	PSO	Proposed Entropy-HOA
5	6.4178	7.2584	8.1245	10.2354
10	7.5684	9.2354	10.2568	12.2547
15	9.5478	10.2547	11.5647	14.9874
20	10.2547	11.2547	12.3587	16.8745
25	12.3547	13.5418	14.5147	18.3547

Table 2. Performance metrics comparison between the proposed DLMNN classifier and other existing classifiers with respect to (a) sensitivity, (b) accuracy and (c) F-measure

No. of Nodes	Sensitivity (%)					Accuracy (%)				
	SVM	NB	KNN	ANN	Proposed DLMNN	SVM	NB	KNN	ANN	Proposed DLMNN
20	83.87	85.12	86.72	87.65	89.56	84.12	86.78	87.56	88.12	90.3
40	85.55	86.88	87.66	88.9	91.6	86.23	87.24	88.67	89.87	92.1
60	89.12	90.66	91.2	92.6	94.6	88.34	89.45	90.64	92.83	94.6
80	90.77	91.56	92.87	94.8	96.5	89.13	91.45	92.34	94.76	96.4
100	91.6	92.4	93.8	95.9	98.5	91.12	92.34	94.43	95.8	98.6

Number of nodes	F-measure (%)				
	SVM	NB	KNN	ANN	Proposed DLMNN
20	86.12	87.23	88.12	89.58	91.64
40	87.44	89.12	90.65	91.73	93.47
60	90.85	91.78	92.76	93.46	95.13
80	91.87	93.05	94.12	95.56	97.12
100	93.12	94.56	96	97.12	98.74

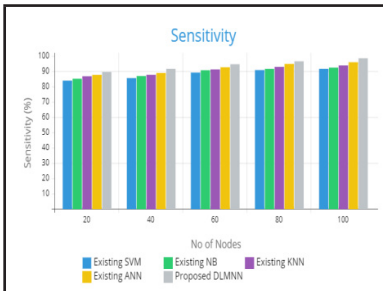

Figure 3(a). sensitivity

Figure 3(b). accuracy

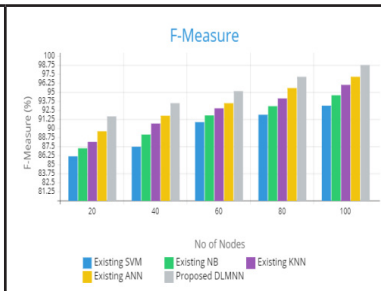

Figure 3(c). F-measure

Figure 3. Comparative Analysis of Proposed and Existing classification Techniques

weighted against existing methods, for instance, Data Encryption Standard (DES), AES and Triple-DES (3DES) in respect of encryption time (ET), security level (SL) and also decryption time (DT), which are illustrated in Table 3.

Table 3 illustrates the performance comparison for KH-AES with the existing techniques for instance DES, 3DES, and AES. Table 3(a) includes the ET and DT obtained by these approaches. The ET and DT for 10 to 50 nodes are compared. For 10

Table 3. Performance Comparison for KH-AES with existing methods

Number of Nodes	Encryption Time (sec)				Decryption Time (sec)			
	DES	3DES	AES	Proposed KH-AES approach	DES	3DES	AES	Proposed KH-AES approach
10	10	12	14	6	12	11.5	13.5	5
20	19	21	22	10	18	20	22	10
30	28	30	28	15	26	28	29	16
40	36	35	36	21	34	36	38	20
50	46	47	44	27	45	48	45.8	25

Techniques	Security Level (%)
DES	85
3DES	87.5
AES	90
Proposed KH-AES	**96**

nodes, the existing DES, 3DES, and AES obtain 10, 12, 14 seconds for ET, whereas the proposed KH-AES takes only 6 seconds to encrypt the data concerned. The ET and DT for KH-AES increase gradually, but compared with existing approaches, the proposed one obtains a lower ET and DT for all nodes. Table 3(b) contrasts the SL of the above-mentioned approaches. The proposed KH-AES obtains highest security (96%) amongst all methods. Table 3 is graphically expressed in Figure 4.

Figure 4 illustrates the performance of the proposed and existing techniques. In Figure 4(a), the ET for the proposed one is weighted against that of existing DES, AES and 3DES for disparate node counts. And in Figure 4(b), the DT is plotted against different node counts. be high. Here, the KH-AES achieves a security level of 96% while the existing DES, 3DES, and AES reach a security level of 85%, 87.5%, and 90% respectively. On this account, the proposed work acquires the highest level of security when analogized with others.

5. Conclusion

It can be concluded that, the security and attack detection model performed well with hybrid techniques and it achieved better results for both proposed approaches. This work performs AD on IoT for SC by proposing an efficient IDS using DLMNN classifier. Entropy-HOA and KH-AES methods are proposed for performing FS phase and safe data sharing phase more efficiently. The performance of the proposed system is

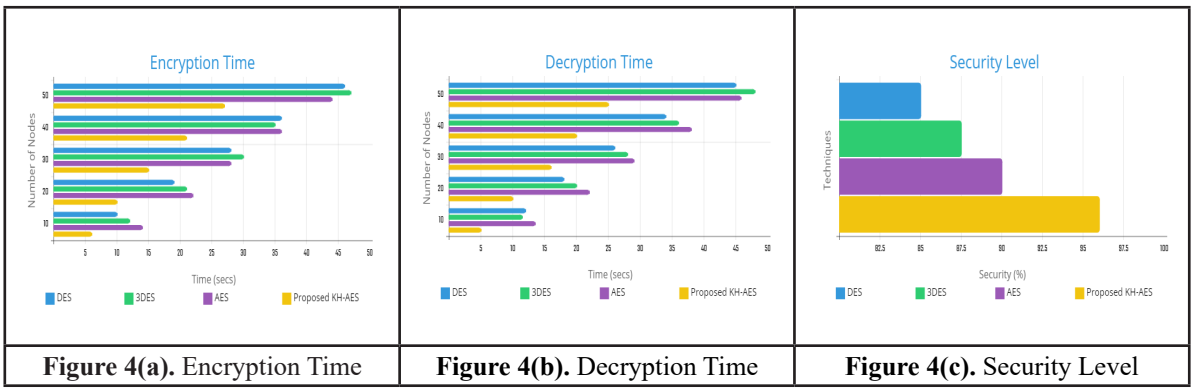

Figure 4(a). Encryption Time | **Figure 4(b).** Decryption Time | **Figure 4(c).** Security Level

Figure 4. Comparative analysis of proposed KH-AES and existing techniques in terms of encryption time, decryption time and security

Figure 4(b) clearly displays that the proposed method obtains a lower DT for every number of nodes when contrasted to the existing ones. The SL of the proposed and existing techniques is plotted as a graph in Figure 4(c). For 10 nodes, the proposed method obtains an ET of 6 seconds but the existing methods for instance DES, AES and 3DES take 10, 14, and 12 seconds for execution. The ET increases gradually when the number of nodes increases, but the proposed technique consumes less ime for every execution in contrast with the other existing methods. For an efficient data sharing, a method's SL should analyzed utilizing the data which is gathered as of the NSL-KDD dataset. In this work, the IDS is employed for training which renders a higher detection and accuracy rate. The IDS tests the sensor values from the smart cities and detects the cyber-attacks. Based on the comparison results, the proposed DLMNN classifier has obtained the highest F-measure, sensitivity and accuracy values when analogized with some existing approaches and the proposed KH-AES achieves a security level of 96% during data transformation. Hence the proposed IDS detects the different types of attacks more efficiently than the existing methods.

REFERENCES

Bu, L., Isakov, M. & Kinsy, M. A. (2019). A secure and robust scheme for sharing confidential information in IoT systems, *Ad Hoc Networks*, *92*, 101762. DOI: 10.1016/j.adhoc.2018.09.007

Bui, D., Puschini, D., Bacles-Min, S., Beigné, E. & Tran, X. (2017). AES Datapath Optimization Strategies for Low-Power Low-Energy Multisecurity-Level Internet-of-Things Applications, *IEEE Transactions on Very Large Scale Integration Systems*, *25*(12), 3281-3290.

Chen, S., Xu, H., Liu, D., Hu, B. & Wang, H. (2014). A Vision of IoT: Applications, Challenges, and Opportunities with China Perspective, *IEEE Internet of Things Journal*, *1*(4), 349-359.

Cheng, K. H. M. & Kumar, A. (2020). Deep feature collaboration for challenging 3D Finger Knuckle Identification, *IEEE Transactions on Information Forsensics and Security*, *16*, 1158-1173.

Dawoud, A, Shahristani, S. & Raun, C. (2018). Deep learning and software-defined networks: towards secure IoT architecture, *Internet of Things, 3*, 82-89.

Deng, L., Li, D., Yao, X., Cox, D. & Wang, H. (2018). Mobile network intrusion detection for IoT system based on transfer learning algorithm, *Cluster Computing, 22,* 9889-9904. DOI: 10.1007/s10586-018-1847-2

Elmaghraby, A. S. & Losavio, M. (2014). Cyber security challenges in Smart Cities: Safety, security and privacy, *Journal of Advanced Research*, *5*(4), 491-497.

Elmisery, A. M., Sertovic, M. & Gupta, B. B. (2017). Cognitive privacy middleware for deep learning mashup in environmental IoT, *IEEE Access, 6*, 8029-8041. DOI: 10.1109/ACCESS.2017.2787422

Ferdowsi, A. & Saad, W. (2019). Deep Learning for signal authentication and security massive internet of things systems, *IEEE Transactions on Communications, 67*(2), 1371-1387. Available at: <https://arxiv.org/abs/1711.01306>.

Garcia-Font, V., Garrigues, C. & Rifà-Pous, H. (2017). Attack classification schema for smart city WSNs, *Sensors*, *17*(4), 771.

Ghosh, A. M. & Grolinger, K. (2021). Edge-Cloud Computing for Internet of Things Data Analytics: Embedding Intelligence in the Edge with Deep Learning, *IEEE Transactions on Industrial Informatics, 17*(3), 2191-2200.

He, X., Jin, R. & Dai, H. (2019). Deep PDS-learning for privacy –aware offloading in MEC-Enhanced IoT, *IEEE Internet of Things Journal*, *6*(3), 4547-4555.

Khoda, M. E., Imam, T., Kamruzzaman, J., Gondal, I. & Rahman, A. (2020). Robust Malware defense in Industrial IoT Applications using Machine Learning with selective adversarial samples, *IEEE Transactions on Industry Applications, 56*(4), 4415 - 4424.

King, J. & Awad, A. I. (2016). A distributed security mechanism for resource-constrained IoT devices, *Informatica*, *40*(1), 133-143.

Li, W., Tug, S., Meng, W. & Wang, Y. (2019). Designing collaborative blockchained signature-based intrusion detection in IoT environments, *Future Generation Computer Systems, 96*, 481-489. DOI: 10.1016/j.future.2019.02.064

Talari, S., Shafie-khah, M., Siano, P., Loia, V., Tommasetti, A. & Catalão, J. (2017). A review of smart cities based on the internet of things concept, *Energies*, *10*(4), 421. DOI:10.3390/en10040421

Vinayakumar, R., Alazab, M., Srinivasan, S., Pham, Q., Padannayil, S. K. & Simran, K. A (2020). Visualized Botnet detection system based Deep Learning for the Internet of Things Networks of Smart Cities, *IEEE Transactions on Industry Applications, 56*(4), 4436-4456. DOI: 10.1109/TIA.2020.2971952

Wu, T., Zhou, P., Wang, B., Li, A., Tang, X., Xu, Zichuan, Chen, K. & Ding, X. (2020). Joint Traffic control and multi-channel reassignment for core backbone network in SDN-IoT: A Multi-Agent Deep Reinforcement learning approach, *IEEE Transactions on Network Science and Engineering, 8*(1), 231-245. DOI: 10.1109/TNSE.2020.3036456

Yu, W. & Köse, S. (2017). A Lightweight Masked AES Implementation for Securing IoT Against CPA Attacks, *IEEE Transactions on Circuits and Systems I: Regular Papers*, *64*(11), 2934-2944. DOI: 10.1109/TCSI.2017.2702098

Zarpelão, B. B., Miani, R. S., Kawakani, C. T. & de Alvarenga, S. C. (2017). A survey of intrusion detection in Internet of Things, *Journal of Network and Computer Applications*, *84*(15), 25-37. DOI: 10.1016/j.jnca.2017.02.009

Zhang, Y., Xu, L., Dong, Q., Wang, J., Blaauw, D. & Sylvester, D. (2018). Recryptor: A Reconfigurable Cryptographic Cortex-M0 Processor With In-Memory and Near-Memory Computing for IoT Security, *IEEE Journal of Solid-State Circuits, 53*(4), 995-1005. DOI: 10.1109/JSSC.2017.2776302