International Journal on Engineering, Science and Technology
Volume 4, Issue 1, 2022 ISSN: 2642-4088

# Intrusion Detection System with a Modified DASO Optimization Algorithm

**Bhushan Deore**
Research Scholar in Electrical Engineering Department, VJTI, Mumbai, India, bsdeore_p18@ee.vjti.ac.in

**Dr. Surendra Bhosale**
Associate Professor and Head of Department in Electrical Engineering Department, VJTI, Mumbai, India

**Abstract:** The Dolphin Atom Search Optimization (DASO) is modified in the proposed work. The Bayesian information gain model has the naïve-bayes classifier centered on the parameters, which include Information Gain (IG), Class-wide Information Gain (CIG), and mutual information. The information is fed to the ID phase for processing at the Deep RNN classifier and the performance is tested with the developed proposed algorithm. An optimization algorithm proposed, is based on the 'Bayesian information gain model' to tune the weights in order to generate effective detection decisions through the fitness measure. The updated information at the selection segment is processed. The performance is outlayed on the general metrics including accuracy, sensitivity, and selectivity and it is better than the results with the existing algorithms. Enhancement performance is observed with the machine learning and Deep Recurrent Neural Network techniques. The proposed DASO is developed by integrating the Dolphin Echolocation (DE) with the Atom Search Optimization (ASO).

**Keywords:** Intrusion detection, Dolphin echolocation, Atom search optimization, Deep recurrent neural network

## Introduction

Internet gained a requisite part in human life. Accordingly, the cyber-attack is a complex and harmful to the information security. Hence, network protection is highly focused by the researchers in the recent years. Various security measures are available to protect the network security, like encryption, ID, authentication, and firewalls. ID is used to find the illegal behaviors based on assumption (Shone *et al.,* 2018). ID plays a key factor in the network protection than other security factors, as ID can aggressively detect the cyber-attacks that exist in the network traffic. The internet allows individuals to change the way they work, learn, and live as both a result of a network's integration into society, but the security threat that people encounter is becoming a major concern in the network domain.

To find different network attacks, like unforeseen attack poses an inescapable technical issue. IDS gained a great accomplishment in the field of information security such that IDS is used to find the invasion that can be an intrusion or ongoing invasion that already occurred. In fact, ID is similar to the classification issues, like multi class classification or binary classification or five-category classification problem. However, the multi class classification issue is to find whether the network behavior is anomalous or normal, while the five-category classification problem is to find whether the network behavior is normal or it includes some of the following attack categories, such as Root to Local (R2L), User to Root (U2R), Probe (Probing), Denial of Service (DoS).The major intention of ID is to increase the accuracy of classifier in finding the anomalous behavior (Yin C, *et al.,* 2017). ID is not a trivial task in the network. However, the ID methods faced various security issues and challenges in the network domain.

Accordingly, provision of effective and robust Network IDS (NIDS) is the major challenge associated in the network security. The NIDS technology uses the signature-based ID method rather than the usage of anomaly detection methods. However, there exist various issues, such as high false error rate, longevity of training data, behavioral dynamics of system, and difficult to obtain reliable training data. However, the current situation reaches a point, where the reliance of these methods leads and inaccurate and ineffective detection. Accordingly, the specifics of this difficulties lead to create an effective anomaly detection method that have the facility to overcome the limitations that are induced by the changes occurred in the modern networks (Fang *et al.,* 2020). The researchers concern three major limitations that exist in the system in command to contribute the challenges of system protection.

The first one is the increasing growth of the amount of network information such that this growth is attributed to the growing level of connectivity, popularity of internet of things, as well as the adoption of cloud services. Effective detection methods are introduced for analyzing the data in an effective and efficient manner. The second issue is resolved by granularity and monitoring the data in-depth that enable to increase the accuracy and effectiveness of detection. However, the analysis of NIDS is required to be more contextually-aware and detailed that means shifting away from high level observations and abstract. The behavioral changes are attributed to the specific elements of network, such as operating system protocols, and individual users. The final cause is the variety of data travelling through modern networks, as well as the number of different protocols. These are the most challenges that introduce high level complexity and difficulty while differentiating the abnormal and the normal behavior (Su *et al.,* 2020).

## Literature Review

The segment examines a variety of existing ID methods. (Shone, N. *et al.* 2018) introduced non-symmetric deep auto encoder (NDAE) model for ID. This method effectively provided dimensionality reduction in the non-symmetric data. This method obtained better classification result and offered high accuracy level with less training time. However, this method was failed to handle the zero-day attacks. (Yin, C. *et al.* 2017) introduced a deep learning approach based on RNN to perform intrusion detection. Here, the performance was evaluated using various learning rate and number of neurons. This method effectively increases detection performance and has the ability to identify the type of intrusion.

However, it failed to reduce training time. (Khan, F.A *et al.* 2019) developed a two-stage deep learning model using the soft-max and the stacked autoencoder classifier for detecting the network intrusions. This method is divided into two parts: the classification phase and the detection phase. The classification phase was responsible to classify the network traffic as abnormal or normal based on the probability score value. This method consumed less execution time, but failed to integrate the deep learning with the reinforcement learning. (Otoum, S. *et al.* 2019) developed a restricted boltzmann machine-based clustered intrusion detection system (RBC-IDS) for detecting network intrusions. This method obtained better accuracy and detection rate. Yang *et al. (*2019) modelled a deep belief network (DBN) using the back propagation (BP) and RBM model. The tentative results indicated that this method improved detection performance, but it failed to improve detection accuracy for small sample sizes. Wu, *et al. (*2018) built a convolutional neural network (CNN) to detect the intrusion, the traffic features were selected from the raw data automatically and the weight coefficient was set using the cost function.

However, the time required to perform the detection process was too high. Zeng *et al.* (2019) developed a light weight approach using the deep learning model to achieve intrusion detection. This model effectively classified the encrypted traffic and detected the malware traffic. Wang *et al.* (2017) developed hierarchical spatial temporal features-based IDS, which effectively learns the high level features. It used the long short term memory (LSTM) classifier for learning the temporal features between various network packets. This method reduced the FAR and increases the overall performance of the system.

## Proposed Dolphin Atom Search Optimization-based Deep RNN

The Pre-processing, feature selection, and the network intrusion detection phase are all included in the proposed optimization model's ID process. The ID data is being used to obtain the input data for the ID procedure at first. The input information is transmitted towards the feature selection segment, wherever the appropriate features stand successfully picked with the help of the newly constructed Bayesian information gain model. The Bayesian information gain model is constructed proceeding the information gain, class-wide information gain, and mutual information is designed using naive bayes classifier.

The specific features are input into the ID phase, wherever the Deep RNN classifier is also used to execute the ID process. A classifier's weights are learned using the proposed DASO method, which was created by combining the DE (Zhao W., *et al* 2019) and ASO (Borkar G., *et al.,* 2017). The proposed DASO-based Deep RNN is depicted schematically in Figure 1.

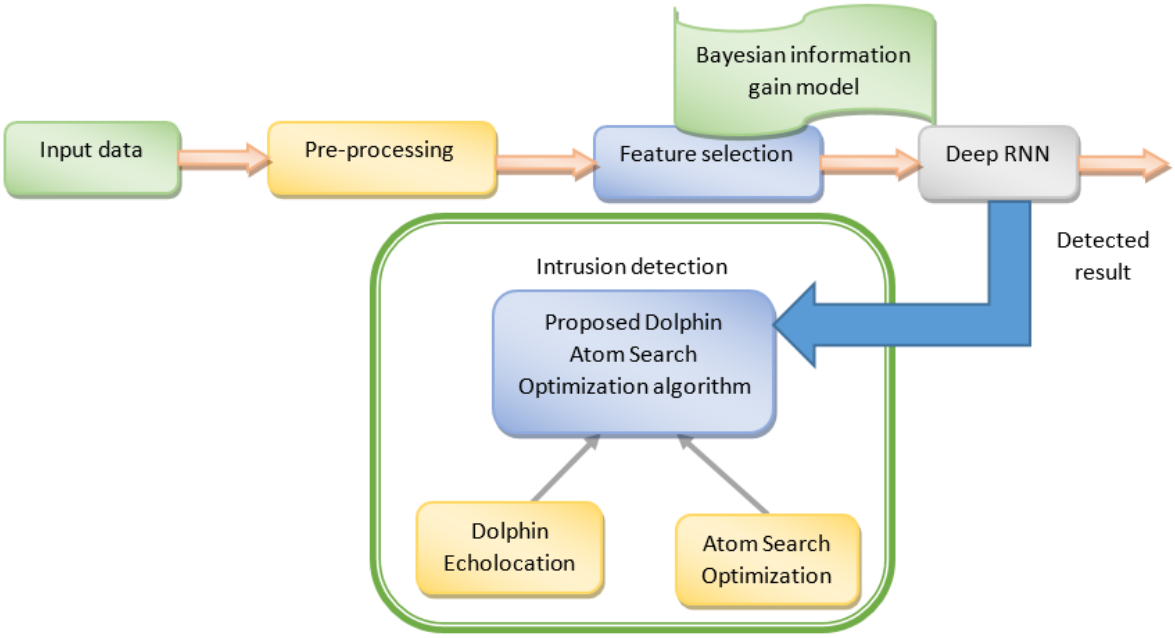



Figure 1. Schematic Illustration of developed Model for ID

**Input Data**

The input data used towards the IDS is composed since the ID dataset that contains system traffic data. Let us look at the database as $H$ with $n$ several network intrusion data $G$, which is displayed as,

$$H = \{G_1, G_2, ...G_i, ...G_n\} \quad (1)$$

where, $G$ denotes the network intrusion data, $H$ represents the database, and $G_i$ represents the network intrusion data located at $i^{th}$ index.

***Pre-processing the Input Data***

The input data $G_i$ stands selected after the database then is entered in the pre-processing Phase. A network traffic data is converted into a sequence of observations. In this phase, the data is cleaned and the noise in the input data is removed. Data pre-processing includes replacing missing values, handling categorical features, and scaling the data. The pre-processed data is signified as $A_i$ with the dimension of $[U \times V]$.

***Feature Selection using Bayesian Information Gain Model***

The pre-processed $A_i$ information is then accepted towards the feature selection section, wherever important items are extracted. Feature selection enables to reduce data size without reducing data quality. The feature selection method is obtained with the recently designed Bayesian information gain model. Similarly, The Bayesian information gain model is advanced using the naive Bayes created on Information Gain, Class-wide Information Gain, and mutual information.

Now, IG, CIG, mutual information are transmitted as input to the Bayesian information gain approach, which processes the data and generates the intrusion behavior as the output such that intrusion behavior can be either abnormal or normal behavior. The output obtained from the Bayesian information gain approach is set as the threshold value, which is optimally used to select the essential features. However, the IG of the traffic data is computed using the entropy that effectively characteristics the behaviors of traffic data. Let us consider the pre-

processed data $A_i$ and say that this data belongs to the class $D$ then and that this data is $p(D)$ and the sum of information it transmits $-\log_2(p(D))$. Therefore, IG of the pre-processed data is considered as,

$$\alpha_i = -p(D)\log_2(p(D)) \tag{2}$$

where, $\alpha_i$ represents the IG. The mutual information is considered using the likelihood values of the pre-processed data and the mutual information that is calculated using $A_i$ is represented as $\beta_i$. In addition to the pre-processed data $A_i$, let us consider additional network traffic data $C$ such that the probability values of the pre-processed data $A_i$ and the network traffic data $C$ is specified as $p(A_i)$ and $p(C)$. The mutual information of the pre-processed data $A_i$ is expressed as,

$$\beta_i = \log_2 \frac{p(A_i, C)}{p(A_i)p(C)} \tag{3}$$

where, $\beta_i$ denotes the mutual information, $p(A_i)$ and $p(C)$ specifies the likelihood of the traffic data $A_i$ and $C$ autonomously, and $p(A_i, C)$ represents the combined likelihood of the traffic data $A_i$ and $C$. The CIG processes traffic behaviors by recognizing the specific class. However, the CIG of the pre-processed data $A_i$ is expressed as,

$$\delta_i = p(a_{A_i} = 1, D)\log \frac{p(a_{A_i} = 1, D)}{p(a_{A_i} = 1)p(D)} + p(a_{A_i} = 0, D)\log \frac{p(a_{A_i} = 0, D)}{p(a_{A_i} = 0)p(D)} \tag{4}$$

where, $A_i$ is the pre-processed data, $D$ denotes the class of the pre-processed, $a_{A_i}$ denotes the value of pre-processed data $A_i$, $p(a_{A_i}, D)$ represents the probability that the data $A_i$ deceits in the class $D$, and $\delta_i$ represents the CIG. Bayesian information gain model takes the input as IG, mutual information, and CIG, which is represented as,

$$M_i = \{\alpha_i, \beta_i, \delta_i\} \tag{5}$$

Here, $\alpha_i$ denotes the IG, $\beta_i$ represents the mutual information, $\delta_i$ indicates the CIG, and $M_i$ is the input of the Bayesian information gain model. With the Bayesian information gain model, the feature selection process is specified as,

$$f = \arg\max_{b=1}^{2} post(M_i / b) \tag{6}$$

$$post(M_i / b) = p(D_b)\prod_{b=1}^{2} p(M_i / D_b) \tag{7}$$

$$p(M_i / D_b) = \frac{1}{\sqrt{2\pi\sigma_b^2}} \times e^{\left(\frac{(M_i - \lambda_b)^2}{2\sigma_b^2}\right)} \tag{8}$$

where, $\sigma_b$ denotes the variance, $\lambda_b$ represents the mean value, $p(D_b)$ indicates the probability of a class $D$, and $f$ indicates the selected features.

***Network Intrusion Detection using Proposed DASO-based Deep RNN***

The proposed DASO-based Deep RNN classifier is used to identify networks. The proposed DASO is created by combining the DE and the ASO, respectively.

*Proposed DASO algorithm:* The proposed DASO is used to execute out Deep RNN classifier training method. The porposed optimization technique employs the fitness function to modify the classifier's weight and bias. The proposed DASO, on the other hand, is created by combining the DE (Zhao W., *et al* 2019) and ASO (Borkar G., *et al.,* 2017). At the initial iteration of DASO, each atom interacts with the other atoms by the attraction or repulsion between them. The repulsion can eliminate the premature convergence of optimization and the over concentration of atoms. At the conclusion of a iteration, every atom in the optimization connects with other atoms via attraction, thereby ensuring the proposed optimization's superior exploitation capacity.

*Solution encoding:* A solution vector format is used to regulate the best solution for network ID based on the fitness metric. The optimal value is acquired with the least amount of error, and the optimal solution is recognized as the finest solution through optimization. The algorithmic stages elaborate in the planned DASO-based Deep RNN, remain as follows:

*Population initialization:* Let us populate DASO with $\tau$ amount of atoms in such a way that the location of each $x^{th}$ atom is defined as follows:

$$P_x = \left[P_x^1, \ldots P_x^J\right]; \quad x = \{1, \ldots c\} \tag{9}$$

where, $P_x^s \left(s = 1, \ldots J\right)$ specifies the $s^{th}$ location element of $x^{th}$ atom in the $J^{th}$ dimensional space.

*Fitness function:* It is intended as the modification between the estimated output and the classifier output value. It is computed built on the error value, with the optimal solution becoming the one with the minimum error value, which is represented as,

$$F_x = \frac{1}{\rho} \sum_{l=1}^{\rho} M_l^{(g,h)} - \eta_l \tag{10}$$

where, $F_x$ indicates the fitness value of $x^{th}$ atom, $M_l^{(g,h)}$ specifies the output from the classifier, and $\eta_l$ denotes the estimated output.

*Calculate the mass:* It is determined at the most basic level using the fitness measure. However, the atomic mass $x^{th}$ at each $r^{th}$ iteration is represented as,

$$X_x(r) = \frac{E_x(r)}{\sum_{x=1}^{c} E_x(r)} \tag{11}$$

where, $X_x(r)$ indicates the mass, and the term $E_x(r)$ is specified as,

$$E_x(r) = \frac{F_x - F_{best}}{e^{F_{worst} - F_{best}}} \tag{12}$$

Here, $F_{best}$ and $F_{worst}$ specifies the best and worst value and is specified as, $F_{best} = \min_{x=1,..c} F_x$, and $F_{worst} = \max_{x=1,..c} F_x$, respectively.

*Determine $R$ neighbours:* Every atom interacts with additional atoms by using the optimal fitness value as the $R$ neighbours for boosting exploration in the first iteration. $R$ progressively decreases with regard to the number of iterations elapsed in such a way that $R$ it is expressed as,

$$R(r) = c - (c-2) \times \sqrt{\frac{r}{\mu}} \tag{13}$$

*Compute the total force:* The total force is defined as the addition of entirely the works that performed continously the $x^{th}$ atom from other atoms, is represented as,

$$Z_x^s(r) = \sum_{t \in R_{bst}} rand_t Z_{xt}^s(r) \tag{14}$$

where, $Z_x^s(r)$ specifies the total force, and $rand_t$ indicate the random number that ranges from 0 to1, respectively.

*Compute acceleration:* With the regular restraint and total force, the acceleration of $x^{th}$ atom at $r^{th}$ time is computed as,

$$K_x^s(r) = \frac{Z_x^s(r)}{X_x^s(r)} + \frac{\vartheta_x^s(r)}{X_x^s(r)} \tag{15}$$

where, $Z_x^s(r)$ is the total force, $\vartheta_x^s(r)$ shows the constraint force, $K_x^s(r)$ indicates the acceleration of $x^{th}$ atom at $r^{th}$ time, and $X_x^s(r)$ represents the mass.

*Update velocity:* Based on $(r+1)$ Repetition, the velocity of $x^{th}$ atom is computed as,

$$Y_x^s(r+1) = rand_x^s\, Y_x^s(r) + K_x^s(r) \tag{16}$$

where, $rand_x^s$ signifies the random number, $K_x^s(r)$ specifies the acceleration.

*Update location of atom:* The proposed DASO is based on the location of $x^{th}$ an atom of ASO. The position of the $x^{th}$ atom $(r+1)^{th}$ is represented by the iteration of its position.

$$P_x(r+1) = P_x(r) + Y_x(r+1) \tag{17}$$

$$Y_x(r+1) = rand_x\, Y_x(r) + K_x(r) \tag{18}$$

where, $X_x(r)$ indicates the mass of $x^{th}$ atom, $Y_x(r)$ is the velocity, $S$ shows the multiplier weight, $\gamma$ represents the depth weight, $\mu$ is the maximum iteration, $V_x$ represents the search space dimension, and $N_x$ indicates the personal best solution. $\chi_{1x}$ and $\chi_{2x}$ are the random number that lies between 0 to 1. Following algorithm states the proposed DASO-based Deep RNN.

Algorithm 1. The proposed DASO-based Deep RNN

Input: $P_x$

Output: $P_x^s(r+1)$ f

Set of number of atoms T , velocity $Y$

Although, the stopping condition is not met

Ensure

Calculate $F$

if $(F_x < F_{best})$ then

$F_{best} = F_x$

$T_{best} = T_x$

End if

Compute $X_x(r)$

Determine $R$ neighbors

Compute $Z_x^s(r)$ and $\vartheta_x^s(r)$

Calculate $K_x^s(r)$

Update $Y_x^s(r+1)$

Update $P_x^s(r+1)$

End for

End while

Return $T_{best}$

## Results and Discussion

This segment talks about the various aspects of the proposed DASO-based Deep RNN algorithm. It is mainly focused on its sensitivity, accuracy and specificity.

### Experimental Setup

The proposed DASO-based Deep RNN is executed in PYTHON using NSL-KDD and BoT-IoT datasets. NSL-KDD dataset carry numerous recorded to solve the optimization issues such that these records are reasonable. Hence, it is affordable to ruin the complete set of data without the requirement to select the small portion of data. It consists of various data files and does not have any redundant records. The Bot-IoT dataset comprises

source files in a variety of formats, including csv files, argus files, and pcap files. These files, however, are split based on the type of attacks.

***Performance Metrics***

The performance revealed by the suggested DASO-based Deep RNN is estimated based on the evaluation metrics, like sensitivity, accuracy, and specificity.

***Accuracy:*** It is a portion that describes the number of properly classified samples, expressed as,

$$\mathrm{A} = \frac{I_p + I_n}{I_p + I_n + J_p + J_n} \tag{19}$$

where, $I_p$ signifies true positive, $I_n$ displays True Negative, $J_p$ represents the False Positive, $J_n$ displays False Negative, $\mathrm{A}$ indicates Accuracy.

***Sensitivity:*** It is the measure of correctly classified positive samples out of total positive samples, which remains represented as,

$$\mathrm{B} = \frac{I_p}{I_p + J_n} \tag{20}$$

where, $\mathrm{B}$ represents sensitivity.

***Specificity:*** It specifies the total number of correct detected negative samples out of total negative samples, which is specified as,

$$\mathrm{X} = \frac{I_n}{I_n + J_p} \tag{21}$$

where, $\mathrm{X}$ denotes specificity.

## Comparative Methods

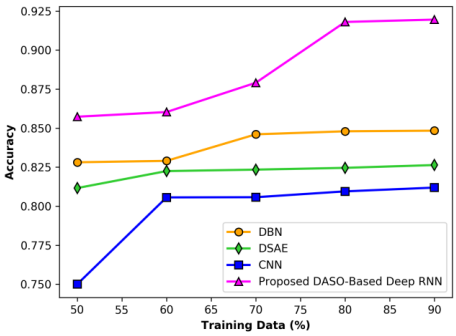
The proposed DASO-based Deep RNN efficiency is evaluated by comparing it to current techniques such as Deep Belief Network (DBN) (Shone N, *et al.,* 2018), Convolutional Neural Network (CNN) (Wu K., *et al.,* 2018), and Deep stacked auto-encoder (DSAE) (Khan F A., *et al.,* 2019). This segment enlarge the comparison study performed using the proposed DASO-based Deep RNN with the NSL-KDD and BoT-IoT datasets.

***Analysis using NSL-KDD Dataset***

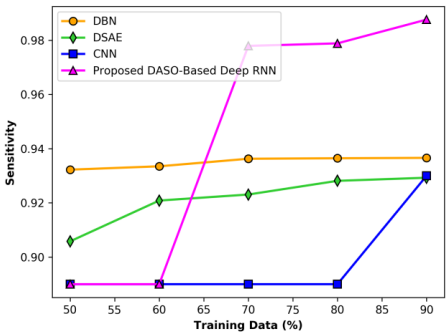
Figure 2 a) For 90% training data, the accuracy of the associated DBN, DSAE, and CNN is 0.8484, 0.8264, and 0.8119, respectively, while the proposed DASO-based Deep RNN gained a higher accuracy is 0.9196. Figure 2 b) For 90% training data, corresponding DBN, DSAE, and CNN obtain sensitivity of 0.9366, 0.9293, 0.9300, separately, while the proposed DASO-based Deep RNN achieves sensitivity is 0.9876. Figure 2 c) For training data is 90%, the specificity of the related DBN, DSAE, and CNN is 0.7400, 0.9008, and 0.9217, respectively, while the proposed DASO-based Deep RNN gained an upper specificity is 0.9642.

***Analysis using BoT-IoT dataset***

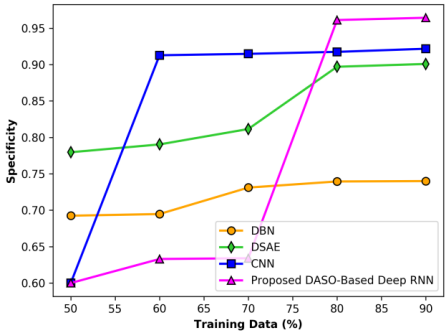
Figure 3 a) For 90% training data, the accuracy of the corresponding DBN, DSAE, and CNN is 0.9642, 0.9737, and 0.9623, respectively, while the proposed DASO-based Deep RNN attained a higher accuracy is 0.9867. Figure 3 b) For 90% training data, the corresponding DBN, DSAE, and CNN obtain sensitivity of 0.9944, 0.9819, 0.9806, respectively, while the proposed DASO-based Deep RNN achieves sensitivity is 0.9988. Figure 3 c) For 90% training data, the specificity of the related DBN, DSAE, and CNN is 0.7920, 0.8398, 0.8466, respectively, while the proposed DASO-based Deep RNN attained a higher specificity is 0.8494.




*(a)*

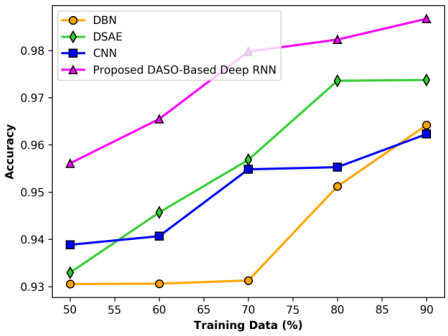



*(b)*

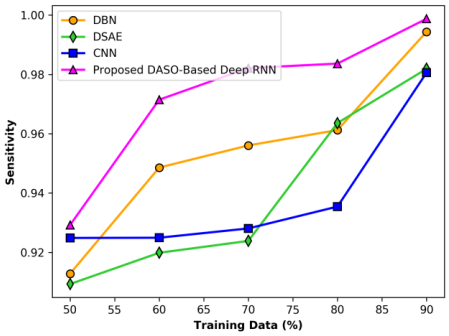



*(c)*

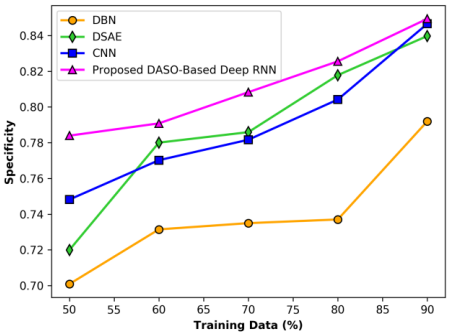
Figure 2. Analysis using NSL-KDD Dataset




*(a)*




*(b)*




*(c)*

Figure 3. Analysis using BoT-IoT dataset

## Discussion on Simlated Work

The comparative discussion is depicted in Table 1. Using the NSL-KDD dataset, the accuracy acquired by the corresponding current DBN, DSAE, and CNN is 0.8484, 0.8264, and 0.8119, respectively, but the planned DASO-based Deep RNN obtained a superior accuracy is 0.9196. The sensitivity of the corresponding DBN, DSAE, and CNN on the NSL-KDD dataset is 0.9366, 0.9293, 0.9300, respectively, whereas the proposed DASO-based Deep RNN attained a higher sensitivity is 0.9876. The accuracy acquired through the equivalent current DBN, DSAE, and CNN with the BoT-IoT dataset is 0.9642, 0.9737, and 0.9623, respectively, while the proposed DASO-based Deep RNN attained a superior accuracy of 0.9867. The specificity achieved through the equivalent current DBN, DSAE, and CNN with the BoT-IoT dataset is 0.7920, 0.8398, 0.8466, respectively, but the proposed DASO-based Deep RNN obtained a higher specificity is 0.8494.

Table 1. Comparison of Different Algorithm with DASO System

| **Metrics/Methods** | | **DBN** | **DSAE** | **CNN** | **Proposed DASO-based Deep RNN** |
|---|---|---|---|---|---|
| **NSL-KDD dataset** | *Accuracy* | 0.8484 | 0.8264 | 0.8119 | **0.9196** |
| | *Sensitivity* | 0.9366 | 0.9293 | 0.9300 | **0.9876** |
| | *Specificity* | 0.7400 | 0.9008 | 0.9217 | **0.9642** |
| **BoT-IoT dataset** | *Accuracy* | 0.9642 | 0.9737 | 0.9623 | **0.9867** |
| | *Sensitivity* | 0.9944 | 0.9819 | 0.9806 | **0.9988** |
| | *Specificity* | 0.7920 | 0.8398 | 0.8466 | **0.8494** |

## Conclusion

An actual and robust network ID mechanism entitled DASO-based Deep RNN is offered here to detect anomalous behavior in the network traffic background. The network traffic statistics is gathered since the database and is delivered to the pre-processing segment, and raw network traffic data is transformed into the sampled data. The pre-processed data is then conceded to the feature selection phase, where the crucial and the appropriate features are effectively designated using the Bayesian information gain model. The Bayesian information gain model developed is using the naive bayes classifier based on the IG, mutual information, and CIG. With the selected features, the Deep RNN classifier detects the traffic behavior as either normal or anomalous. The Deep RNN classifier is trained using the proposed DASO system. This proposed DASO system is designed by adding the DE with the ASO, respectively. The proposed DASO system using the fitness measure improves the weights. The proposed DASO obtained well presentation using the metrics, accuracy, specificity, sensitivity using the values of 0.9867, 0.8494, 0.9988, using the BoT-IoT dataset, respectively. As Forthcoming work, the presentation of the intrusion detection approach can be enhanced using some other optimization algorithm.

## Acknowledgements

This work is carried out at Ramrao Adik Institute of Technology (D. Y. Patil Deemed to be University), where first author, Bhushan Deore is full time employee in the Department of Electonics and Telecommunication Engineering and working part-time research scholar in Veermata Jijabai Technological Institute, Mumbai, India.

## References

Shone, N., Ngoc, T. N., Phai, V. D., & Shi, Q. (2018). A deep learning approach to network intrusion detection. IEEE transactions on emerging topics in computational intelligence, 2(1), 41-50.

Yin, C., Zhu, Y., Fei, J., & He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. IEEE Access, 5, 21954-21961.

Khan, F. A., Gumaei, A., Derhab, A., & Hussain, A. (2019). A novel two-stage deep learning model for efficient network intrusion detection. IEEE Access, 7, 30373-30385.

Otoum, S., Kantarci, B., & Mouftah, H. T. (2019). On the feasibility of deep learning in sensor network intrusion detection. IEEE Networking Letters, 1(2), 68-71.

Yang, H., Qin, G., & Ye, L. (2019). Combined wireless network intrusion detection model based on deep learning. IEEE Access, 7, 82624-82632.

Wu, K., Chen, Z., & Li, W. (2018). A novel intrusion detection model for a massive network using convolutional neural networks. IEEE Access, 6, 50850-50859.

Zeng, Y., Gu, H., Wei, W., & Guo, Y. (2019). "Deep-full-range": A deep learning based network encrypted traffic classification and intrusion detection framework. IEEE Access, 7, 45182-45190.

Wang, W., Sheng, Y., Wang, J., Zeng, X., Ye, X., Huang, Y., & Zhu, M. (2017). HAST-IDS: Learning hierarchical spatial-temporal features using deep neural networks to improve intrusion detection. IEEE access, 6, 1792-1806.

Zhao, W., Wang, L., & Zhang, Z. (2019). A novel atom search optimization for dispersion coefficient estimation in groundwater. Future Generation Computer Systems, 91, 601-610.

Borkar, G. M., & Mahajan, A. R. (2017). A secure and trust based on-demand multipath routing scheme for self-organized mobile ad-hoc networks. Wireless Networks, 23(8), 2455-2472.

Tsai, C. F., Hsu, Y. F., Lin, C. Y., & Lin, W. Y. (2009). Intrusion detection by machine learning: A review. expert systems with applications, 36(10), 11994-12000.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural networks, 61, 85-117.

Liu, A. A., Su, Y. T., Nie, W. Z., & Kankanhalli, M. (2016). Hierarchical clustering multi-task learning for joint human action grouping and recognition. IEEE transactions on pattern analysis and machine intelligence, 39(1), 102-114.

Bamakan, S. M. H., Wang, H., Yingjie, T., & Shi, Y. (2016). An effective intrusion detection framework based on MCLP/SVM optimized by time-varying chaos particle swarm optimization. Neurocomputing, 199, 90-102.

Eesa, A. S., Orman, Z., & Brifcani, A. M. A. (2015). A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems. Expert systems with applications, 42(5), 2670-2679.

Manzoor, I., & Kumar, N. (2017). A feature reduced intrusion detection system using ANN classifier. Expert Systems with Applications, 88, 249-257.

Louvieris, P., Clewley, N., & Liu, X. (2013). Effects-based feature identification for network intrusion detection. Neurocomputing, 121, 265-273.

Varma, P. R. K., Kumari, V. V., & Kumar, S. S. (2016). Feature selection using relative fuzzy entropy and ant colony optimization applied to real-time intrusion detection system. Procedia Computer Science, 85, 503-510.

Kuang, F., Xu, W., & Zhang, S. (2014). A novel hybrid KPCA and SVM with GA model for intrusion detection. Applied Soft Computing, 18, 178-184.

Aburomman, A. A., & Reaz, M. B. I. (2016). A novel SVM-kNN-PSO ensemble method for intrusion detection system. Applied Soft Computing, 38, 360-372.

Inoue, M., Inoue, S., & Nishida, T. (2018). Deep recurrent neural network for mobile human activity recognition with high throughput. Artificial Life and Robotics, 23(2), 173-185.

Liu, W., Ci, L., & Liu, L. (2020). A new method of fuzzy support vector machine algorithm for intrusion detection. Applied Sciences, 10(3), 1065.

Zong, W., Chow, Y. W., & Susilo, W. (2020). Interactive three-dimensional visualization of network intrusion detection data for machine learning. Future Generation Computer Systems, 102, 292-306.

Fang, W., Tan, X., & Wilbur, D. (2020). Application of intrusion detection technology in network safety based on machine learning. Safety Science, 124, 104604.

Jiang, K., Wang, W., Wang, A., & Wu, H. (2020). Network intrusion detection combined hybrid sampling with deep hierarchical network. IEEE Access, 8, 32464-32476.

Lv, L., Wang, W., Zhang, Z., & Liu, X. (2020). A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine. Knowledge-based systems, 195, 105648.

Su, T., Sun, H., Zhu, J., Wang, S., & Li, Y. (2020). BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset. IEEE Access, 8, 29575-29585.