# MODELING OF SEEKER OPTIMIZATION ALGORITHM WITH DEEP LEARNING ASSISTED INTRUSION DETECTION IN SECURE IOT ENVIRONMENT

**Abstract**

With the extensive use of IoT devices in several domains comprising healthcare, smart homes, transportation, and industrial automation, the desire for robust intrusion detection system (IDS) is developed paramount to protect against possible cyber-attacks and threats. Intrusion detection is a key feature of making sure the privacy and security of IoT devices and networks. IoT intrusion detection purposes to recognize and respond to unauthorized access, abnormal behavior, and malicious actions in IoT environments. This manuscript offers the design of Seeker Optimization Algorithm based Deep Learning Assisted Intrusion Detection (SOADL-ID) method in IoT environment. The purpose of the SOADL-ID approach lies in the proper detection and grouping of intrusion from the IoT platform. In the presented SOADL-ID algorithm, data pre-processing was executed for converting the input data as suitable format. For intrusion detection and classification, stacked sparse auto encoder (SSAE) mechanism is applied. Lastly, the SOA model was employed for the optimum hyper parameter value of the SSAE and it supports in optimal hyper parameter selection. The simulation validation of the SOADL-ID methodology was tested on benchmark IDS database. The simulation outcome highlighted the better outcome of the SOADL-ID method.

**Keywords:** Intrusion detection systems; Deep learning; Seeker optimization algorithm; Hyper parameter tuning; Security

**Authors**

**Mr. S. Thirumal**
Research Scholar,
Department of Computer Applications,
St. Peter's Institute of Higher Education and Research,
Chennai, India.

**Dr. R. Latha**
Professor and Head,
Department of Computer Applications,
St. Peter's Institute of Higher Education and Research,
Chennai, India.

**Mr. S. Vimal Kumar**
Research Scholar,
Department of Computer Science,
St. Peter's Institute of Higher Education and Research,
Chennai, India.

## I. INTRODUCTION

Recently, Internet of Things (IoTs) has been continuously developing and enabling communications and interconnections between numerous devices by a network; consequently, it is launching novel technology for business processes [1]. Later, many challenges in various aspects like finances, enforcement, proving credibility, and business operations are fore ensuing from the considerable expansion of cyber-security attacks [2]. Cloud computing (CC) is generally utilized as IoT stored information that can be formed as a model to be provided different resources and services for user requests. Generally, CC reduces the human intervention between providers and users [3]. It has received considerable interest from users and organizations because of its impressive features. But transition from the existing to CC platform, numerous struggling problems are faced based on the security and operation mechanism [4]. This security attack builds it a target for several intruders and cyber-criminals; thus, it prevents large number of users from favouring or transferring to the CC platform [5]. There are various reasons why the current cyberattacks rapidly developing. The main reason behind the accessible and existing hacking devices is simple to apply thereby allowing naive hackers to rapidly attack the cloud storage with no certain knowledge or excellent skills [6].

Another method of defense must be designed in IoT networks to secure IoT systems against cyber-attacks [7]. Intrusion Detection Systems (IDSs) satisfy this goal. But classical IDS approaches are lacking or lesser efficient for the security of IoT due to their special features previously described especially, global connectivity, limited energy, limited bandwidth capacity, heterogeneity, and ubiquitous [8]. Currently, artificial intelligence approaches have gained popularity as effective applications for detecting the network attacks comprising IoT networks [9]. Due to this, ML or DL based techniques can obtain normal and anomalous behavior in IoT platforms [10].

This manuscript offers a Seeker Optimization Algorithm with Deep Learning Assisted Intrusion Detection (SOADL-ID) approach in IoT. The purpose of the SOADL-ID method recognizes intrusions from the IoT environment. In the presented SOADL-ID algorithm, data preprocessing was carried out for converting the input data as suitable format. For intrusion detection and classification, stacked sparse autoencoder (SSAE) approach was applied. Lastly, the SOA model was employed for the optimum hyperparameter values of the SSAE and it supports optimal hyperparameter selection. The simulation values of the SOADL-ID methodology were tested on benchmark IDS database.

## II. RELATED WORK

Zhao et al. [11] developed a new NID method for IoT based on the light-weight DNN (LNN). During data preprocessing, the PCA technique is used for attaining feature dimensional reduction to evade higher dimension raw traffic features resulting in great complex model. Moreover, classifier employs the compression and expansion structure, channel shuffle operation, and inverse residual structure. Pampapathi, Guptha, and Hema [12] introduce Filtered DL method for IDS with a data transmission technique. The presented method is comprised of five stages namely cluster formation along with CHS, connectivity, attack detection, data broker, and initialization of sensor networks. The study outperforms the current DL-NN and ANN.

 

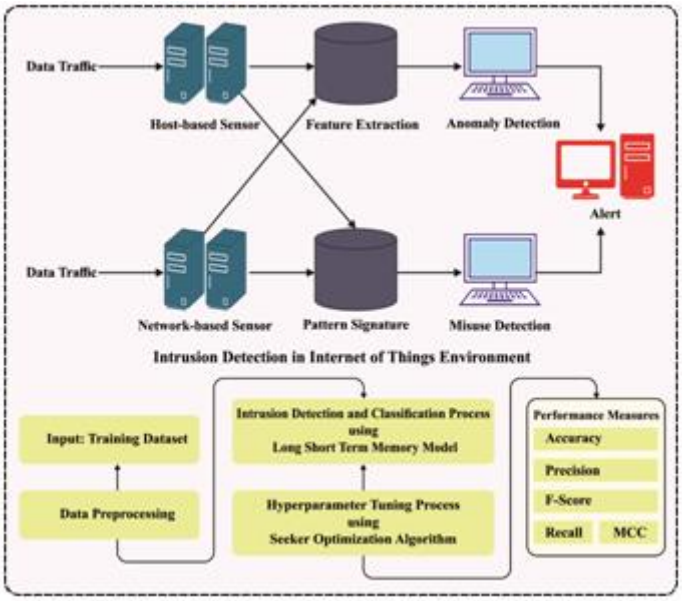


**Figure 1:** Overall flow of SOADL-ID approach

In [13], the authors introduced a method named DL Model Intrusion Detection in Industrial IOTs. The proposed model incorporates AE for selecting optimum features and Cascade Forward BPNN (CFBPNN) for attack and classification recognition. The cascading algorithm utilizes connected links from first to the last layers and identifies the abnormal and normal behaviors and generates an efficient classification.

Vishwakarma and Kesswani [14] introduced an innovative DNN based IDS for detecting malicious packets in real time. The analysis was implemented to recently designed benchmark Netflow based databases for training methods. This technique also developed a packet acquiring and identifying approach for real time attack-detection. Wu et al. [15] recommended an intelligent IDS technique performed by data mining that depends on a fuzzy rough set. Chen et al. [16] used a multi-objective evolutionary CNN (MECNN) as the classifier for identifying intrusions and the MOEA depends on decomposition (MOEA/D) approach is adopted to develop the CNN method. Specifically, a new encoding system was introduced for transforming the topological framework of CNN and later the two different objectives.

## III. THE PROPOSED MODEL

In this manuscript, we have introduced a new SOADL-ID method in the IoT platform. The major aim of the SOADL-ID approach lies in the proper recognition of intrusion from the IoT environment. In the presented SOADL-ID technique, three stages of operations are involved such as data pre-processing, SSAE based detection, and SOA based hyper parameter selection. Fig. 1 exemplifies the working flow of SOADL-ID algorithm.

1. **Preprocessing:** Data normalization is appropriate for classifier issues as it drives to same weighted all the features [17]. In this case, the Min- Max Normalized is employed as it keeps the connection from the actual database and executes a linear conversion on original database. This technique converted the data as present boundaries. The data are normalized within [0,1] by Eq. (1):

$$x' = \frac{x - \min_Y}{\max_Y - \min_Y} \quad (1)$$

In which, x denotes the original values of features, x' implies its normalization value, and $\max_Y$ and $\min_Y$ represents the maximal and minimal feature values Y.

2. **Intrusion Detection using SSAE:** The SSAE model is applied for intrusion detection and classification. The SSAE network contained $Ln$ layers, whereas $n$ labelled the entire layer counts and softmax functioned as a final layer to carry out classification [18]. The network layers are pretrained in an unsupervised manner to deploy an SSAE as a classifier. Fig. 2 depicts the infrastructure of SSAE. Afterward pre- trained the network layers, and supervised fine- tuned has been employed for enhancing solutions by presenting classes to the network. The learning in SSAE is carried out in several stages i.e., initialized parameter; bias $b$ and weight $W$, optimized the cost function, computing the activation maps of $n^{th}$ layers, and fed to $(n+1)^{th}$ layer. At last, the activation map of final hidden state is fed to softmax layer.

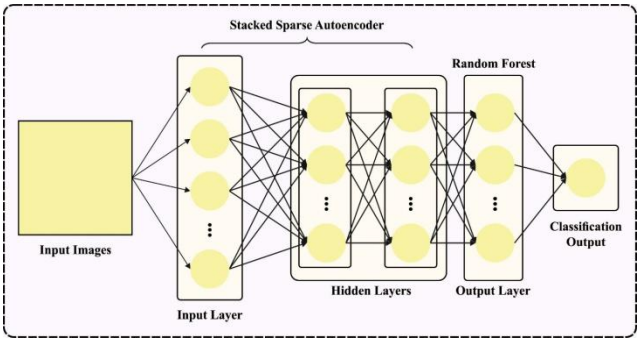


**Figure 2:** Architecture of SSAE

The cost function is deployed for the sparse AE contained in 3 parts such as weight decay, sparsity punishment, and MSE. To provide the database with N trained instances, $(x_1, y_1), (x_2, y_2), \ldots\ldots, (x_n, y_n)$, whereas the data instances are represented by $x_i$ and labeled $y_i$. The cost function of SSAE model can be expressed by $F_{sparse}\ (W, b)$.

$$F_{sparse}\ (W, b) = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{2}||h_{W,b}(x^j) - y^i||^2 + \frac{\lambda}{2}\sum_{n=1}^{L_n - 1}\sum_{i=1}^{s_n}$$

 

$$\times \sum_{j=1}^{S_n+1} (W_{ji}^n)^2 + \beta \sum_{j=1}^{s_n} KL(p \| \hat{p}_j) \quad (2)$$

Eq. (2) defined the MSE over N trained instances. The weighted decay factor that reduced the weight of networks is used for reducing the possibilities of over- fitting. Now, parameter λ used to manage the range of weights that are reduced. During this process, initial hidden state, the average activity of $j^{th}$ unit is provided as:

$$\hat{p}_j = \frac{1}{N} \sum_{i=1}^{N} a_j^2 (x^i) \quad (3)$$

The sparsity of the network has been managed using sparse variable ρ and is set as 0.05. During this case, Kullback- Leibler (KL) divergence has been employed as a sparsity punishment term expressed as Eq. (4):

$$KL(p \| \hat{p}_j) = p \log \frac{p}{\hat{p}_j} + (1-p) log \frac{(1-p)}{(1-p_j} \quad (4)$$

3. **Parameter Tuning using SOA:** Here, the SOA was employed for the optimum hyperparameter values of the SSAE approach and it helps in optimal hyperparameter selection. The SOA technique executes comprehensive analysis on human search performance [19]. Employing "experience gradient" to define the searching way, the searching phase measurement is solved by utilizing uncertain reasoning. The altruistic direction $\vec{f}_{i,a}(t)$, preemptive direction $\vec{f}_{i,p}(t)$, and egoistic direction $\vec{f}_{i,e}(t)$ of $i^{th}$ individual from some dimension is attained.

$$\vec{f}_{i,e}(t) = \vec{p}_{i,best} - \vec{x}_i(t) \quad (5)$$

$$\vec{f}_{i,a}(t) = \vec{g}_{i,best} - \vec{x}_i(t) \quad (6)$$

$$\vec{f}_{i,p}(t) = \vec{x}_i(t_1) - \vec{x}_i(t_2) \quad (7)$$

The searcher employs manner of arbitrary weighted average for attaining the search orientation.

$$\vec{f}_i(t) = sign\left(\omega \vec{f}_{i,e}(t) + \varphi_1 \vec{f}_{i,a}(t) + \varphi_2 \vec{f}_{i,p}(t)\right) \quad (8)$$

with $r_1, r_2 \in \{r, r-1, r-2\}, \vec{x}_i(t_1)$ and $\vec{x}_i(t_2)$: Best benefits of $\{\vec{x}_i(t-2), \vec{x}_i(t-1), \vec{x}_i(t)\}$ individually, $g_{i,best}$: Historical better place in the neighborhood but the$i^{th}$ searching aspects are placed, $p_{i,best}$: Optimal locality in the $i^{th}$ searching aspects of the present locality, $\psi_1$ and $\psi_1$: Random numbers in zero and one; and $\omega$ is: Weight of inertia.

The SOA method, by the computer languages, defines any human natural language, which simulates human intelligence reasoning search performance. Once the method

 

discloses an fuzzy rule, it adjusts to an optimum estimate of main optimizer problems. A superior search step length is more crucial.

$$\mu(\alpha) = e^{\frac{\alpha^2}{2\delta^2}} \quad (9)$$

with parameters $\alpha$ and $\delta$ of membership function.

Based on Eq. (9), the possibility of resultant variable beyond $[-3\delta, 3\delta]$ is below 0.0111. Hence, $\mu_{min} = 0.0111$. In the typical circumstance, an optimum location for individual's take $\mu_{max} = 1.0$ and the worse location is 0.0111.

$$\mu_i = \mu_{max} - \frac{s - I_i}{s - I}(\mu_{max} - \mu_{min}), i = 1,2,\cdots,s \quad (10)$$

$$\mu_{ij} = rand(\mu_i, 1), j = 1,2,\cdots,D \quad (11)$$

with $\mu_{ij}$: Defined by Eqs. (10) and (11), $I_i$: amount of sequences $x_i(t)$ of the present individuals arranged in high-low by function value and $rand(\mu_i, 1)$: Real number in some partition $[\mu_i, 1]$.

It is realized in Eq. (10) simulates the arbitrary search performance of humans. The step size of |- dimension search space is defined by Eq. (12):

$$\alpha_{ij} = \delta_{ij} - \sqrt{-ln(\mu_{ij})} \quad (12)$$

with Parameter $\delta_{ij}$ of Gaussian distribution function that is expressed by Eq. (13):

$$\omega = (itrer_{max} - r)/itrer_{max} \quad (13)$$

$$\delta_{ij} = \omega^* abs(\vec{x}_{min} - \vec{x}_{max}) \quad (14)$$

with $\omega$: Weight of inertia. $\omega$ diminishes linearly from [0.9-0.1] as the evolutionary algebra develops,. $\vec{x}_{min}\, and \vec{x}_{max}$ are correspondingly the variant of minimal value and maximal value of the functions.

$$x_{ij}(t+1) = x_{ij}(t) + \alpha_{ij}(t) f_{ij}(t), i = 1,2,\cdots,s; j = 1,2,\cdots,D \quad (15)$$

with $ith$ searcher individual, $j$: Individual dimensional, $f_{ij}(t)$ and $\alpha_{ij}(t)$ : Seekers' searching direction, correspondingly, and searching step size at time $t$ and $x_{ij}(t)$ along with $x_{ij}(t+1)$: Searchers' position at time $t$ and $(t+1)$, correspondingly.

Fitness choice is a key aspect of the SOA methodology. An encoded performance has been deployed to develop the best solution for candidate performances.

$$Fitness = max(P) \quad (16)$$

$$P = \frac{TP}{TP + FP} \quad (17)$$

In which, $FP$ and $TP$ denote the false and true positive values.

 

## IV. EXPERIMENTAL VALIDATION

In this work, the SOADL-ID technique is tested using three distinct IDS datasets such as UNSW_NB15, KDD_cup99, and NSL-KDD [20]. Table 1 reports the brief IDS outputs of the SOADL-ID technique on the UNSW-NB15 database. The SOADL-ID technique properly identified the normal and attack samples. On 70% of TR set, the SOADL-ID technique attains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and MCC of 97.66%, 97.66%, 97.66%, 97.66%, and 95.31% correspondingly. Afterward, on 30% of TS set, the SOADL-ID approaches realizes average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and MCC of 97.77%, 97.77%, 97.77%, 97.77%, and 95.53% correspondingly.

**Table I**: IDS Outcome Of Soadl-Id Approach On Unsw-Nb15 Dataset

| **UNSW_NB15** | | | | | |
|---|---|---|---|---|---|
| **Class Labels** | $Accu_y$ | $Prec_n$ | $Reca_l$ | $F_{Score}$ | **MCC** |
| **TRP (70%)** | | | | | |
| Normal | 97.68 | 97.62 | 97.68 | 97.65 | 95.31 |
| Attack | 97.64 | 97.69 | 97.64 | 97.66 | 95.31 |
| **Average** | **97.66** | **97.66** | **97.66** | **97.66** | **95.31** |
| **TSP (30%)** | | | | | |
| Normal | 97.95 | 97.63 | 97.95 | 97.79 | 95.53 |
| Attack | 97.58 | 97.91 | 97.58 | 97.74 | 95.53 |
| **Average** | **97.77** | **97.77** | **97.77** | **97.77** | **95.53** |

Table 2 demonstrates an extensive IDS output of the SOADL-ID algorithm on the KDD_cup99database. The SOADL-ID system properly identified the normal and attack samples. On 70% of TR set, the SOADL-ID method attains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and MCC of 95.82%, 95.89%, 95.82%, 95.81%, and 91.70% respectively. Next, on 30% of TS set, the SOADL-ID method obtains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$, and MCC of 95.50%, 95.60%, 95.50%, 95.50%, and 91.10% correspondingly.

Table II: IDS Outcome Of Soadl-Id Approach On Kdd_Cup99 Dataset

| **KDD_cup99** | | | | | |
|---|---|---|---|---|---|
| **Class Labels** | $\mathbf{Accu_y}$ | $\mathbf{Prec_n}$ | $\mathbf{Reca_l}$ | $\mathbf{F_{Score}}$ | **MCC** |
| **TRP (70%)** | | | | | |
| Normal | 97.83 | 94.04 | 97.83 | 95.89 | 91.70 |
| Attack | 93.80 | 97.74 | 93.80 | 95.73 | 91.70 |
| **Average** | **95.82** | **95.89** | **95.82** | **95.81** | **91.70** |
| **TSP (30%)** | | | | | |
| Normal | 97.87 | 93.45 | 97.87 | 95.61 | 91.10 |
| Attack | 93.12 | 97.76 | 93.12 | 95.38 | 91.10 |
| **Average** | **95.50** | **95.60** | **95.50** | **95.50** | **91.10** |

 

**Table III: IDS Outcome Of Soadl-Id Approach On Nsl-Kdd Dataset**

| **NSL-KDD** | | | | | |
|---|---|---|---|---|---|
| **Class** | **$Accu_y$** | **$Prec_n$** | **$Reca_l$** | **$F_{Score}$** | **MCC** |
| **TRP (70%)** | | | | | |
| Normal | 97.92 | 96.35 | 97.92 | 97.13 | 94.21 |
| Attack | 96.28 | 97.87 | 96.28 | 97.07 | 94.21 |
| **Average** | **97.10** | **97.11** | **97.10** | **97.10** | **94.21** |
| **TSP (30%)** | | | | | |
| Normal | 97.59 | 96.42 | 97.59 | 97.00 | 94.01 |
| Attack | 96.42 | 97.58 | 96.42 | 97.00 | 94.01 |
| **Average** | **97.00** | **97.00** | **97.00** | **97.00** | **94.01** |

Table 3 determines entire IDS outcomes of the SOADL-ID method on the NSL-KDD database. The SOADL-ID system properly identified the normal and attack samples. On 70% of TR set, the SOADL-ID algorithm gains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$ , and MCC of 97.10%, 97.11%, 97.10%, 97.10%, and 94.21% correspondingly. Followed by, on 30% of TS set, the SOADL-ID method attains average $accu_y$, $prec_n$, $reca_l$, $F_{score}$ , and MCC of 97%, 97%, 97 %, 97%, and 94.01% correspondingly.

In Table 4, a comprehensive analysis of the SOADL-ID algorithm with other ML approaches take place [21]. The outcomes stated that the SOADL-ID method accomplishes superior outcomes.

**Table IV: Comparative Outcome Of Soadl-Id Technique With Other Ml Approaches**

| **Methods** | **Accuracy (%)** | **Computational Time (sec)** |
|---|---|---|
| SOADL-ID | 97.77 | 1.18 |
| ANN Algorithm | 92.80 | 2.57 |
| C.4.5 Model | 95.72 | 2.52 |
| Bagging | 96.19 | 3.30 |
| KNN Algorithm | 95.97 | 3.20 |
| Ensemble | 94.00 | 3.25 |
| SVM Model | 93.07 | 2.98 |

In Fig. 3, a comparative $accu_y$ results of the SOADL-ID technique with recent approaches are made. The figure showcases that the SVM, ensemble, and ANN models accomplish lower $accu_y$ values of 93.07%, 94%, and 92.80% respectively. Meanwhile, the C.4.5, bagging, and KNN models gain closer $accu_y$ values of 95.72%, 96.19%, and 95.97% respectively. However, the SOADL-ID technique reaches higher $accu_y$ of 97.77%.

 

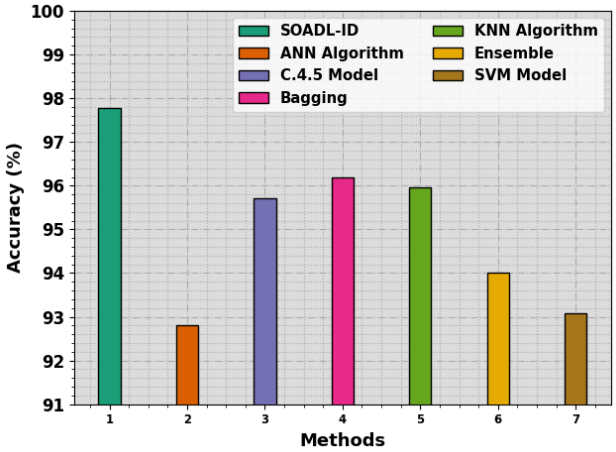


**Figure3:** Accu$_y$ outcome of SOADL-ID methodology with other ML approaches

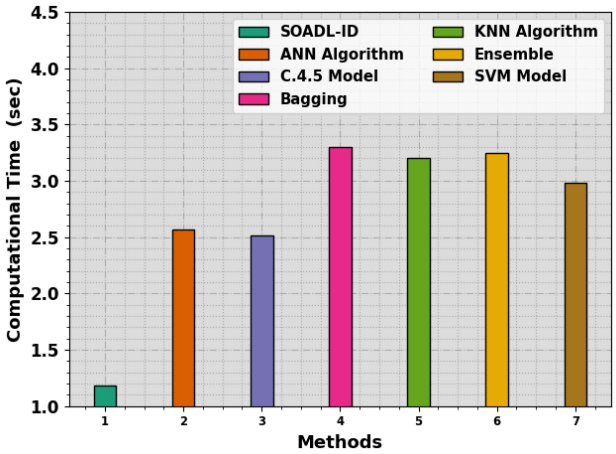


**Figure 4:** CT outcome of SOADL-ID methodology with other ML approaches

In Fig. 4, a comparative CT result of the SOADL-ID technique with recent approaches is made. The figure showcases that the bagging, KNN, and ensemble models achieve higher CT values of 3.30s, 3.20s, and 3.25s respectively. Meanwhile, the ANN, C.4.5, and SVM models gain closer CT values of 2.57s, 2.52s, and 2.98s respectively. However, the SOADL-ID technique reaches higher accu$_y$ of 97.77%. These results inferred the effectual recognition results of the SOADL-ID technique.

## V. CONCLUSIONS

In this manuscript, a new SOADL-ID algorithm has been devisedin the IoTnetwork. The major aim of the SOADL-ID system focused on the detection of intrusion from the IoT environment. In the presented SOADL-ID technique, three stages of operations are involved such as data preprocessing, SSAE based classification, and SOA based hyperparameter selection. For intrusion detection and classification, the SSAE model is applied. Lastly, the SOA was used for the optimal hyper parameter values of the SSAE approach and it helps in optimal hyper parameter selection. The experimentation assessment of the SOADL-ID algorithm was tested on three IDS database. The simulation outcomes demonstrated the better

 

solution of the SOADL-ID methodology with other recent approaches interms of different measures.

## REFERENCES

[1] Ravi, V., Pham, T.D. and Alazab, M., 2023. Deep Learning-Based Network Intrusion Detection System for Internet of Medical Things. IEEE Internet of Things Magazine, 6(2), pp.50-54.
[2] Zhou, X., Liang, W., Li, W., Yan, K., Shimizu, S., Kevin, I. and Wang, K., 2021. Hierarchical adversarial attacks against graph-neural-network-based IoT network intrusion detection system. IEEE Internet of Things Journal, 9(12), pp.9310-9319.
[3] Belhadi, A., Djenouri, Y., Djenouri, D., Srivastava, G. and Lin, J.C.W., 2023. Group intrusion detection in the Internet of Things using a hybrid recurrent neural network. Cluster Computing, 26(2), pp.1147-1158.
[4] Abou El Houda, Z., Brik, B. and Khoukhi, L., 2022. "why should i trust your ids?": An explainable deep learning framework for intrusion detection systems in internet of things networks. IEEE Open Journal of the Communications Society, 3, pp.1164-1176.
[5] Vaiyapuri, T., Sbai, Z., Alaskar, H. and Alaseem, N.A., 2021. Deep learning approaches for intrusion detection in IIoT networks–opportunities and future directions. International Journal of Advanced Computer Science and Applications, 12(4).
[6] Mohamed, H.G., Alrowais, F., Al-Hagery, M.A., Al Duhayyim, M., Hilal, A.M. and Motwakel, A., 2023. Optimal Wavelet Neural Network-Based Intrusion Detection in Internet of Things Environment. Computers, Materials & Continua, 75(2).
[7] Salman, E.H., Taher, M.A., Hammadi, Y.I., Mahmood, O.A., Muthanna, A. and Koucheryavy, A., 2022. An Anomaly Intrusion Detection for High-Density Internet of Things Wireless Communication Network Based Deep Learning Algorithms. Sensors, 23(1), p.206.
[8] Morshedi, R., Matinkhah, S.M. and Sadeghi, M.T., 2023. Intrusion Detection for IoT Network Security with Deep Neural Network.
[9] Tripathy, P.K., Shabaz, M., Zaidi, A., Keshta, I., Sharma, U., Soni, M., Agrawal, A.V., Maaliw III, R.R. and Sharma, D.P., 2023. Policy conflict detection approach for decision-making in intelligent industrial Internet of Things. Computers and Electrical Engineering, 108, p.108671.
[10] Fatani, A., Dahou, A., Abd Elaziz, M., Al-Qaness, M.A., Lu, S., Alfadhli, S.A. and Alresheedi, S.S., 2023. Enhancing Intrusion Detection Systems for IoT and Cloud Environments Using a Growth Optimizer Algorithm and Conventional Neural Networks. Sensors, 23(9), p.4430.
[11] Zhao, R., Gui, G., Xue, Z., Yin, J., Ohtsuki, T., Adebisi, B. and Gacanin, H., 2021. A novel intrusion detection method based on lightweight neural network for internet of things. IEEE Internet of Things Journal, 9(12), pp.9960-9972.
[12] Pampapathi, B.M., Guptha, N. and Hema, M.S., 2022. Towards an effective deep learning-based intrusion detection system in the internet of things. Telematics and Informatics Reports, 7, p.100009.
[13] Jayalaxmi, P.L.S., Saha, R., Kumar, G., Alazab, M., Conti, M. and Cheng, X., 2023. PIGNUS: A Deep Learning Model for IDS in Industrial Internet-of-Things. Computers & Security, p.103315.
[14] Vishwakarma, M. and Kesswani, N., 2022. DIDS: A Deep Neural Network based real-time Intrusion detection system for IoT. Decision Analytics Journal, 5, p.100142.
[15] Wu, Y., Nie, L., Wang, S., Ning, Z. and Li, S., 2021. Intelligent intrusion detection for internet of things security: A deep convolutional generative adversarial network-enabled approach. IEEE Internet of Things Journal.
[16] Chen, Y., Lin, Q., Wei, W., Ji, J., Wong, K.C. and Coello, C.A.C., 2022. Intrusion detection using multi-objective evolutionary convolutional neural network for Internet of Things in Fog computing. Knowledge-Based Systems, 244, p.108505.
[17] Eskandari, A., Aghaei, M., Milimonfared, J. and Nedaei, A., 2023. A weighted ensemble learning-based autonomous fault diagnosis method for photovoltaic systems using genetic algorithm. International Journal of Electrical Power & Energy Systems, 144, p.108591.
[18] Alorf, A. and Khan, M.U.G., 2022. Multi-label classification of Alzheimer's disease stages from resting-state fMRI-based correlation connectivity data and deep learning. Computers in Biology and Medicine, 151, p.106240.
[19] Liu, H., Duan, S. and Luo, H., 2022. A hybrid engineering algorithm of the seeker algorithm and particle swarm optimization. Materials Testing, 64(7), pp.1051-1089.
[20] https://research.unsw.edu.au/projects/unsw-nb15-dataset

 

[21] Albulayhi, K.; Abu Al-Haija, Q.; Alsuhibany, S.A.; Jillepalli, A.A.; Ashrafuzzaman, M.; Sheldon, F.T. IoT Intrusion Detection Using Machine Learning with a Novel High Performing Feature Selection Method. Appl. Sci. 2022, 12, 5015. https://doi.org/10.3390/ app12105015

 