International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 6, June 2024)

# An Intrusion Detection Model for Prediction of Cyberattacks Using ANN Deep Learning Technique

[1]Aman Kumar Yadav, [2]Dr P. K. Sharma
[1]M.Tech Scholar, Department of Computer Science Engineering, NRI Institute of Research & Technology, Bhopal, India
[2]Principal, Department of Computer Science Engineering, NRI Institute of Research & Technology, Bhopal, India

***Abstract*— The rapid advancement of digital technologies has significantly increased the complexity and scale of cyber threats, necessitating robust and adaptive security mechanisms. This paper reviews intrusion detection models that leverage artificial intelligence (AI) techniques for predicting cyberattacks. It examines various AI methodologies, including machine learning (ML), deep learning (DL), and hybrid approaches, analyzing their effectiveness in identifying and mitigating sophisticated cyber threats. The review highlights key challenges in current intrusion detection systems (IDS) and proposes potential enhancements to improve detection accuracy and response times. Emphasis is placed on the integration of AI techniques with traditional IDS frameworks, focusing on how these innovative models can anticipate and thwart emerging cyber threats in real-time. The paper aims to provide a comprehensive understanding of the state-of-the-art AI-driven IDS, their current limitations, and future research directions to bolster cybersecurity defenses.**

***Keywords*—AI, ANN, Cyber, NIDS, HIDS, Security.**

## I. INTRODUCTION

In today's hyper-connected digital landscape, the prevalence of cyber threats poses a significant risk to individuals, organizations, and governments. The increasing sophistication and variety of these threats have rendered traditional security mechanisms, including signature-based and heuristic-based Intrusion Detection Systems (IDS), less effective. Consequently, there is an urgent need for more advanced and adaptive approaches to cybersecurity. This necessity has driven the exploration of Artificial Intelligence (AI), particularly Deep Learning (DL) techniques, for developing more resilient IDS.

Artificial Neural Networks (ANN), a subset of DL, have gained substantial attention due to their remarkable capability in processing and learning from large volumes of data. ANNs mimic the human brain's neural structure, enabling them to recognize complex patterns and make predictions with high accuracy. This ability makes ANNs particularly suitable for intrusion detection, where identifying subtle anomalies in network traffic can be the difference between preventing a cyberattack and falling victim to one.

An intrusion detection system (also known as an intrusion prevention system or IPS) is a piece of hardware or a piece of software that is designed to monitor a network or systems for indications of an intrusion or violations of policy. [1] An IDS is also known as an intrusion prevention system or IPS. If an intrusion is found, the system will either alert an administrator or submit the necessary data to a centralised security information and event management (SIEM) database. Both of these actions will take place if the system finds an intrusion. A SIEM system is able to detect harmful behaviour while avoiding false positives because it correlates data from several sources and uses advanced techniques of alert filtering .[2]

Although there are many more types of intrusion detection systems, two of the most common kinds are network intrusion detection systems (NIDS) and host-based intrusion detection systems (HIDS) (HIDS). Examples of network intrusion detection systems (NIDS) include systems that analyse incoming network traffic, whereas examples of host intrusion detection systems (HIDS) include systems that monitor important operating system files. IDS may also be categorised according to the mechanism that they utilise to detect threats. The most prevalent types are signature-based detection, which identifies hazardous patterns like malware, and anomaly-based detection, which looks for unusual behaviour. Another common kind of detection is the reputation-based variety (recognizing the potential threat according to the reputation scores).

The deployment of ANN-based IDS involves several key components: data preprocessing, model architecture design, training, and evaluation. Data preprocessing includes cleaning

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 6, June 2024)

and transforming raw network data into a format suitable for training the ANN. The architecture design of the ANN, including the number of layers, neurons per layer, and activation functions, significantly impacts the model's performance. Training the ANN requires large, labeled datasets and involves optimizing weights through backpropagation to minimize prediction errors. Finally, evaluating the model's performance involves metrics such as accuracy, precision, recall, and the receiver operating characteristic (ROC) curve to ensure the model's reliability in real-world scenarios.

## II. PROPOSED METHODOLOGY

The proposed methodology is explained using following flow chart-

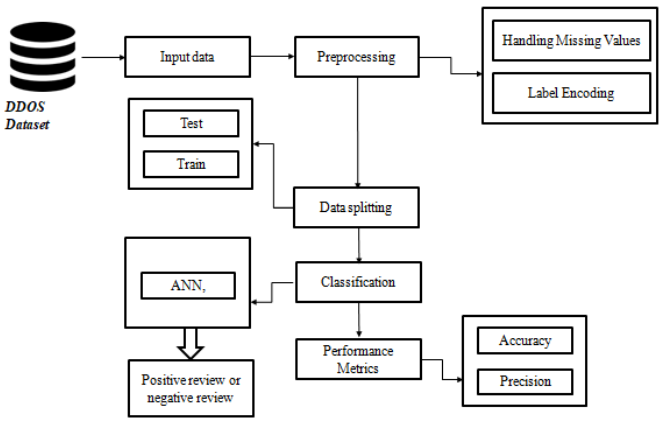



Figure 1: Flow Chart

Steps-

- Firstly, finalize the dataset [13] based on the intrusion detection system, taken from publicly available large dataset repository.
- The data has been preprocessed, and the missing dataset is being sent over right now.
- Now preprocessed data is splitting into the training and testing phase.
- Now artificial neural networks Classification technique is applied
- F-measure, Precision, Accuracy, Recall, and Classification Error are some of the performance criteria you should now evaluate.

These sub-modules form the basis of the proposed research's methodology:

**Data Selection and Loading**

- The process of picking a dataset and loading it into the Python environment is known as data selections.

**Data Pre-processing**

- In data pre-processing, the "noise or unwanted data" in a dataset is filtered out.
- Data deficiency correction and categorical data encoding
- The imputer library is used to get rid of any missing or null values in the data.
- Decomposing a Dataset into Test and Training Sets

**Splitting Dataset into Train and Test Data**

- The term "data splitting" refers to the practise of dividing a dataset into two distinct halves, often for use in a cross-validation setting.
- • The data is split in two; one half is used to build a prediction model, and the other half is used to test how well that model performed.

**Feature Extraction**

Feature extraction is a technique for normalising a set of data's independent variables. Normalization is a procedure that occurs during the pre-processing stage of data processing and goes by another name in the industry.

**Classification**

**ANN-** The artificial neurons of an ANN may be thought of as the vertices in a weighted directed graph. Weighted directed edges represent the connection between neuron outputs and inputs. An external source's signal is received by the Artificial Neural Network as a vector representing a pattern and a

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 6, June 2024)

picture. For each set of n inputs, a mathematical notation x(n) is used to denote the assigned value.

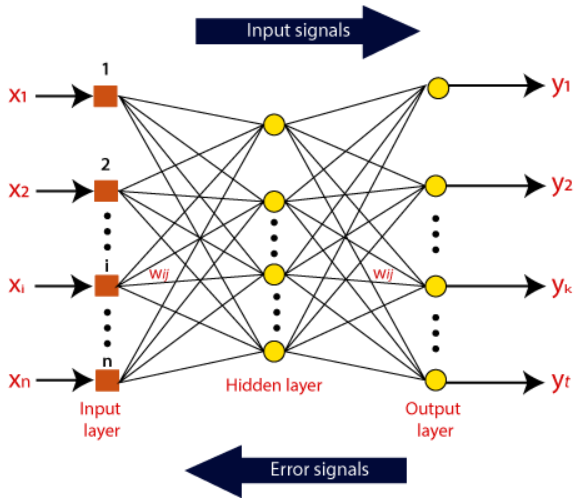



Figure 2: ANN

After that, we multiply each input by its associated weights (these weights are the details utilised by the artificial neural networks to solve a specific problem). These weights often stand for the robustness of the connections between individual neurons in the ANN. The computer device internalises a summary of all the input weights.

With the goal of increasing the system's reaction, bias is introduced if the weighted sum is zero. The input for bias and the value of weight are both 1. Here, the sum of the input weights might be negative or positive infinity. Here, we benchmark a maximum value and run the sum of the weighted inputs through the activation function to constrain the response to acceptable ranges.

**Prediction**

- This study successfully forecasted the data from the dataset by improving the overall performance of the prediction findings, and it does so by using a technique for predicting intrusion detection.

**Algorithm**

**Input:** Intrusion detection Dataset.

Consider the basic information characteristics, such as id with id dur, proto, service, state spkts, dpkts sbytes, dbytes, rate, sttl dttl, sload, etc.

Filtering the null value

Sort the data set according to the characteristics you've chosen.

**Output:** Best values for F-measure, Precision, Accuracy, Recall, and Classification Error

**Step:** 1. now dataset is divided into 2 part train and test dataset like train of y and x and test of y and x

2. Extractions of features, features = {} for intrusion count: features [intrusion count] = True

3. Model selection and split

Y train

Y-test

4. Use a classifier based on deep learning's artificial neural network.

5. Confusion matrix with TP, FP, TN, and FN values shown.

6. Determine the percentage of correct answers, standard error, recall, and f-measure.

7. Create a ROC graph.

**Evaluation**

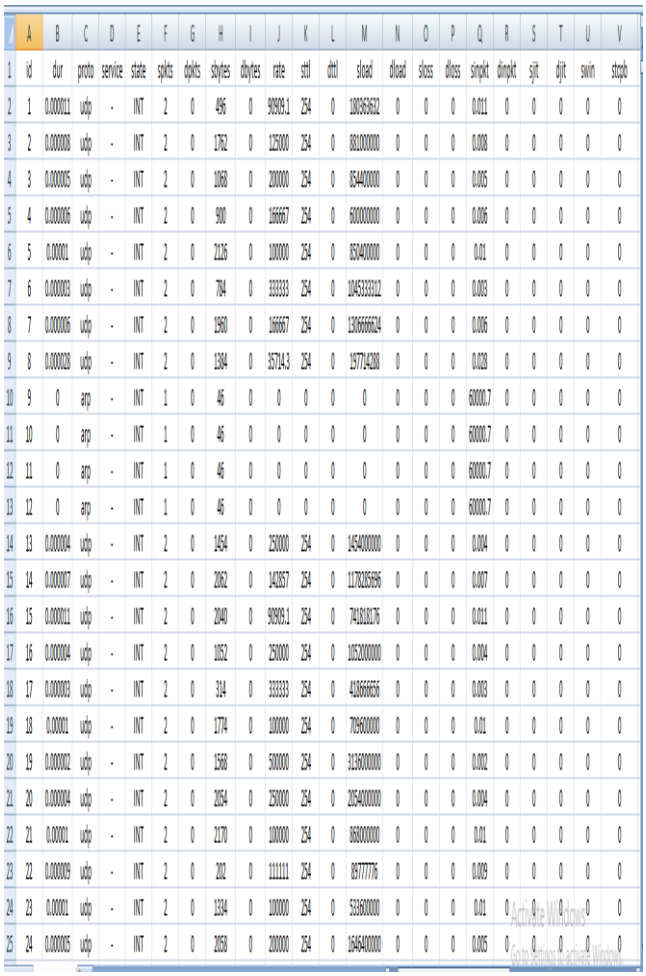
Accuracy, precision, and recall are the main metrics used to assess a classification model.

- Accuracy is defined as the ratio of true positives to total positives, while recall is defined as the ratio of positives to negatives.
- Accuracy = [TP +TN] / [TP+TN+FP+FN]; F1-Score = 2x (Precision x Recall)/(Precision + Recall)
- Classification Error = 100- Accuracy

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 6, June 2024)

## III. SIMULATION RESULTS

To run the simulation, we use the Python Spyder IDE version 3.7.



| | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | id | dur | proto | service | state | spkts | dpkts | sbytes | dbytes | rate | sttl | dttl | sload | dload | sloss | dloss | sinpkt | dinpkt | sjit | djit | swin | stcpb |
| 2 | 1 | 0.000011 | udp | - | INT | 2 | 0 | 496 | 0 | 90909.1 | 254 | 0 | 180363632 | 0 | 0 | 0 | 0.011 | 0 | 0 | 0 | 0 | 0 |
| 3 | 2 | 0.000008 | udp | - | INT | 2 | 0 | 1762 | 0 | 125000 | 254 | 0 | 881000000 | 0 | 0 | 0 | 0.008 | 0 | 0 | 0 | 0 | 0 |
| 4 | 3 | 0.000005 | udp | - | INT | 2 | 0 | 1068 | 0 | 200000 | 254 | 0 | 854400000 | 0 | 0 | 0 | 0.005 | 0 | 0 | 0 | 0 | 0 |
| 5 | 4 | 0.000006 | udp | - | INT | 2 | 0 | 900 | 0 | 166667 | 254 | 0 | 600000000 | 0 | 0 | 0 | 0.006 | 0 | 0 | 0 | 0 | 0 |
| 6 | 5 | 0.00001 | udp | - | INT | 2 | 0 | 2126 | 0 | 100000 | 254 | 0 | 850400000 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 7 | 6 | 0.000003 | udp | - | INT | 2 | 0 | 784 | 0 | 333333 | 254 | 0 | 1045333312 | 0 | 0 | 0 | 0.003 | 0 | 0 | 0 | 0 | 0 |
| 8 | 7 | 0.000006 | udp | - | INT | 2 | 0 | 1960 | 0 | 166667 | 254 | 0 | 1306666624 | 0 | 0 | 0 | 0.006 | 0 | 0 | 0 | 0 | 0 |
| 9 | 8 | 0.000028 | udp | - | INT | 2 | 0 | 1384 | 0 | 35714.3 | 254 | 0 | 197714288 | 0 | 0 | 0 | 0.028 | 0 | 0 | 0 | 0 | 0 |
| 10 | 9 | 0 | arp | - | INT | 1 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60000.7 | 0 | 0 | 0 | 0 | 0 |
| 11 | 10 | 0 | arp | - | INT | 1 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60000.7 | 0 | 0 | 0 | 0 | 0 |
| 12 | 11 | 0 | arp | - | INT | 1 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60000.7 | 0 | 0 | 0 | 0 | 0 |
| 13 | 12 | 0 | arp | - | INT | 1 | 0 | 46 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 60000.7 | 0 | 0 | 0 | 0 | 0 |
| 14 | 13 | 0.000004 | udp | - | INT | 2 | 0 | 1454 | 0 | 250000 | 254 | 0 | 1454000000 | 0 | 0 | 0 | 0.004 | 0 | 0 | 0 | 0 | 0 |
| 15 | 14 | 0.000007 | udp | - | INT | 2 | 0 | 2062 | 0 | 142857 | 254 | 0 | 1178285696 | 0 | 0 | 0 | 0.007 | 0 | 0 | 0 | 0 | 0 |
| 16 | 15 | 0.000011 | udp | - | INT | 2 | 0 | 1940 | 0 | 90909.1 | 254 | 0 | 741010176 | 0 | 0 | 0 | 0.011 | 0 | 0 | 0 | 0 | 0 |
| 17 | 16 | 0.000004 | udp | - | INT | 2 | 0 | 1652 | 0 | 250000 | 254 | 0 | 1652000000 | 0 | 0 | 0 | 0.004 | 0 | 0 | 0 | 0 | 0 |
| 18 | 17 | 0.000003 | udp | - | INT | 2 | 0 | 314 | 0 | 333333 | 254 | 0 | 418666656 | 0 | 0 | 0 | 0.003 | 0 | 0 | 0 | 0 | 0 |
| 19 | 18 | 0.00001 | udp | - | INT | 2 | 0 | 1774 | 0 | 100000 | 254 | 0 | 709600000 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 20 | 19 | 0.000002 | udp | - | INT | 2 | 0 | 1568 | 0 | 500000 | 254 | 0 | 3136000000 | 0 | 0 | 0 | 0.002 | 0 | 0 | 0 | 0 | 0 |
| 21 | 20 | 0.000004 | udp | - | INT | 2 | 0 | 2054 | 0 | 250000 | 254 | 0 | 2054000000 | 0 | 0 | 0 | 0.004 | 0 | 0 | 0 | 0 | 0 |
| 22 | 21 | 0.00001 | udp | - | INT | 2 | 0 | 2170 | 0 | 100000 | 254 | 0 | 868000000 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 23 | 22 | 0.000009 | udp | - | INT | 2 | 0 | 202 | 0 | 111111 | 254 | 0 | 8977776 | 0 | 0 | 0 | 0.009 | 0 | 0 | 0 | 0 | 0 |
| 24 | 23 | 0.00001 | udp | - | INT | 2 | 0 | 1334 | 0 | 100000 | 254 | 0 | 533600000 | 0 | 0 | 0 | 0.01 | 0 | 0 | 0 | 0 | 0 |
| 25 | 24 | 0.000065 | udp | - | INT | 2 | 0 | 2053 | 0 | 15000 | 254 | 0 | 1642400000 | 0 | 0 | 0 | 0.065 | 0 | 0 | 0 | 0 | 0 |

Figure 3: Dataset

The data set is shown in the Python environment (Figure 3). Row and column counts in the dataset might vary widely. Every single column identifies the characteristics by name.

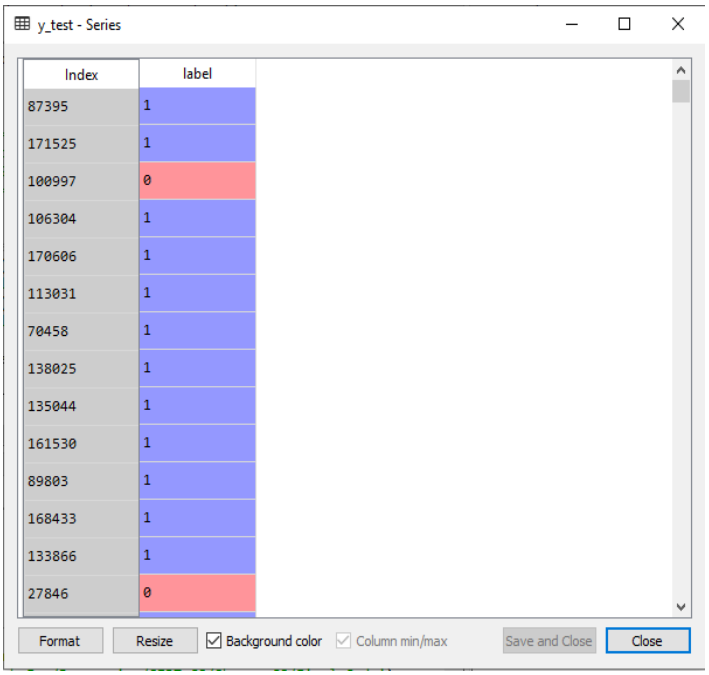



Figure 4: Y test

This dataset's y test is seen in Figure 4. Twenty-thirty percent of the original dataset is used as the train dataset..

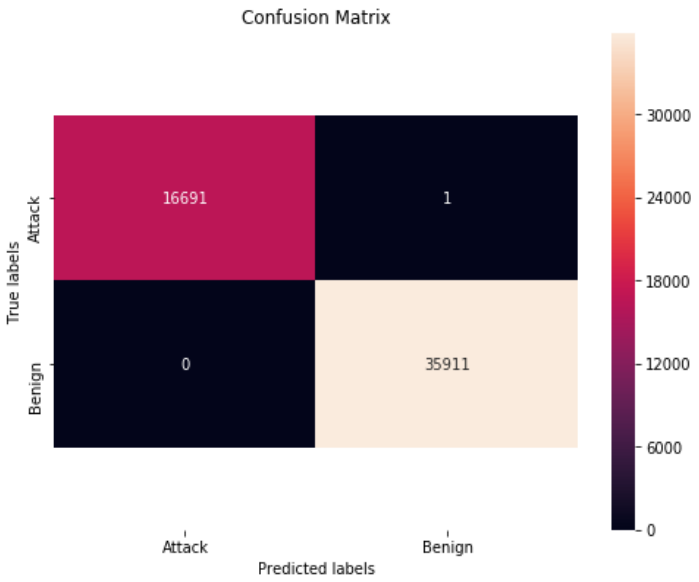



Figure 5: Confusion matrix heat map

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 6, June 2024)

Confusion matrices for heat maps generated by the ANN deep learning classification method are shown in Figure 5. It is a matrix of size N by N that measures how well a categorization model does its job.

Table 1: Simulation Results

| Sr. No. | Parameters | Value (%) |
|---|---|---|
| 1 | Precision | 99.99 |
| 2 | Recall | 99.99 |
| 3 | F_Measure | 99.99 |
| 4 | Accuracy | 99.99 |
| 5 | Error Rate | 0.01 |
| 6 | Sensitivity | 99.99 |
| 7 | Specificity | 99.99 |

Table 2: Result Comparison

| Sr No | Parameter | Previous Work [1] | Proposed Work |
|---|---|---|---|
| 1 | Accuracy | 99.94% | 99.99% |
| 2 | Classification Error | 0.06% | 0.01% |

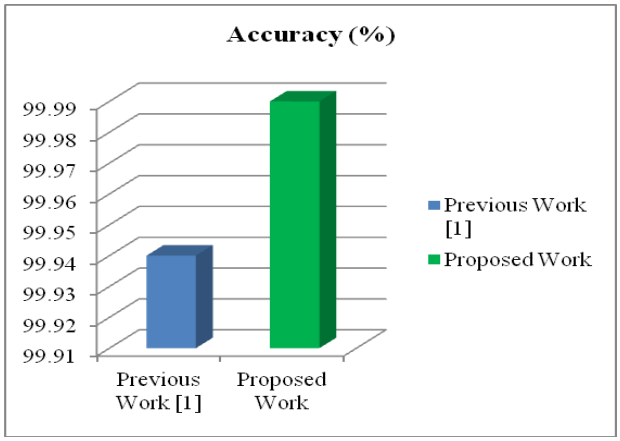



Figure 6: Accuracy Result graph

Figure 6 is presenting the graphical representation of the accuracy. The proposed work achieved better accuracy then existing work.

.

## IV. CONCLUSION

Intrusion detection systems, sometimes referred to as IDSs, are a kind of network security technology that were first developed to identify attacks on specific programs or computers. However, they are no longer used for this purpose. The network intrusion system provides protection for the cyber world against a broad range of different types of efforts to penetrate it. In order to provide attack prediction tactics, artificial intelligence, machine learning, and deep learning are all capable of using their respective capabilities. Through the use of an intrusion detection system, this research provides a strategy that utilizes artificial neural networks for the purpose of predicting cyberattacks. The program known as Python Spyder is used in order to carry out simulation. The ANN method that was suggested achieves an overall accuracy of 99.99% with a classification error of 0.01% during the course of its operation.

## REFERENCES

1. S. Ho, S. A. Jufout, K. Dajani and M. Mozumdar, "A Novel Intrusion Detection Model for Detecting Known and Innovative Cyberattacks Using Convolutional Neural Network," in IEEE Open

Journal of the Computer Society, vol. 2, pp. 14-25, 2021, doi: 10.1109/OJCS.2021.3050917.

2. V. K. Navya, J. Adithi, D. Rudrawal, H. Tailor and N. James, "Intrusion Detection System using Deep Neural Networks (DNN)," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 2021, pp. 1-6, doi: 10.1109/ICAECA52838.2021.9675513.
3. Y. A. Farrukh, Z. Ahmad, I. Khan and R. M. Elavarasan, "A Sequential Supervised Machine Learning Approach for Cyber Attack Detection in a Smart Grid System," 2021 North American Power Symposium (NAPS), 2021, pp. 1-6, doi: 10.1109/NAPS52732.2021.9654767.
4. S. Thirimanne, L. Jayawardana, P. Liyanaarachchi and L. Yasakethu, "Comparative Algorithm Analysis for Machine Learning Based Intrusion Detection System," 2021 10th International Conference on Information and Automation for Sustainability (ICIAfS), 2021, pp. 191-196, doi: 10.1109/ICIAfS52090.2021.9605814.
5. T. T. Nguyen and V. J. Reddi, "Deep Reinforcement Learning for Cyber Security," in IEEE Transactions on Neural Networks and Learning Systems, doi: 10.1109/TNNLS.2021.3121870.
6. W. Xu, J. Jang-Jaccard, A. Singh, Y. Wei and F. Sabrina, "Improving Performance of Autoencoder-Based Network Anomaly Detection on NSL-KDD Dataset," in IEEE Access, vol. 9, pp. 140136-140146, 2021, doi: 10.1109/ACCESS.2021.3116612.
7. K. Cao, J. Zhu, W. Feng, C. Ma, M. Liu and T. Du, "Network Intrusion Detection based on Dense Dilated Convolutions and Attention Mechanism," 2021 International Wireless Communications and Mobile Computing (IWCMC), 2021, pp. 463-468, doi: 10.1109/IWCMC51323.2021.9498652.
8. I. Ullah and Q. H. Mahmoud, "Design and Development of a Deep Learning-Based Model for Anomaly Detection in IoT Networks," in IEEE Access, vol. 9, pp. 103906-103926, 2021, doi: 10.1109/ACCESS.2021.3094024.
9. D. Park, S. Kim, H. Kwon, D. Shin and D. Shin, "Host-Based Intrusion Detection Model Using Siamese Network," in IEEE Access, vol. 9, pp. 76614-76623, 2021, doi: 10.1109/ACCESS.2021.3082160.
10. I. Siniosoglou, P. Radoglou-Grammatikis, G. Efstathopoulos, P. Fouliras and P. Sarigiannidis, "A Unified Deep Learning Anomaly Detection and Classification Approach for Smart Grid Environments," in IEEE Transactions on Network and Service Management, vol. 18, no. 2, pp. 1137-1151, June 2021, doi: 10.1109/TNSM.2021.3078381.
11. https://www.unb.ca/cic/datasets/ids-2017.html