International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 03, March 2024)

An Efficient Deep Learning Technique for Intrusion Detection in Wireless Sensor Network

[1]Mousham Kumari Sharma, [2]Dr. Anshuj Jain

[1]Research Scholar, Dept. of Electronics and Communication Engineering, SCOPE College of Engineering, Bhopal, India,
[2]Associate Professor & HOD, Dept. of Electronics and Communication Engineering, SCOPE College of Engineering, Bhopal, India

***Abstract*— Predicting intrusion detection in a wireless sensor network (WSN) involves using machine learning algorithms to analyze network traffic data and predict whether an intrusion or security breach is likely to occur. The goal of intrusion prediction is to provide advanced warning of potential attacks and to allow for proactive measures to be taken to prevent or mitigate the impact of the attack. Several machine and deep learning algorithms can be used for intrusion prediction in WSNs. This paper presents the efficient deep learning technique based on artificial neural network for Intrusion Detection in Wireless Sensor Network. The simulation is performed using python spyder 3.7 software. Simulated results shows that the 99.99% accuracy in the predicted model.**

Keywords*— *WSN, NIDS, Machine, Deep, Python, Accuracy.

I. INTRODUCTION

A wireless sensor network (WSN) is a type of network that consists of a large number of small sensor nodes that are distributed over a geographic area. These nodes are equipped with sensing, computation, and communication capabilities that allow them to gather and transmit data from their environment to a central location.

WSNs are used in a wide range of applications, including environmental monitoring, industrial control and automation, healthcare, and military applications. They are particularly useful in scenarios where it is not feasible or cost-effective to install wired sensors.

Intrusion detection is the process of monitoring a system or network for malicious activities or policy violations and detecting, alerting, or preventing unauthorized access, misuse, or modification of the system. The goal of intrusion detection is to identify security breaches and respond to them in a timely manner to minimize their impact.

The Intrusion detection is an important aspect of wireless sensor networks (WSNs) as these networks are vulnerable to various security threats due to their distributed and resource-constrained nature. Intrusion detection systems (IDSs) can be used to detect and respond to security breaches in WSNs.

There are two types of intrusion detection systems: anomaly-based and signature-based. Anomaly-based IDSs detect deviations from normal network behavior, while signature-based IDSs match the observed traffic against a database of known attack signatures.

In a wireless sensor network, intrusion detection can be challenging due to the limited resources available in the network nodes. Some common techniques used for intrusion detection in WSNs include:

1. Energy-based detection: This technique measures the energy consumption of the nodes and detects anomalies in the energy consumption pattern.
2. Data fusion: This technique aggregates data from multiple nodes to detect anomalies that cannot be detected by individual nodes.
3. Distributed detection: This technique distributes the detection mechanism across multiple nodes to reduce the computational load on individual nodes.
4. Machine learning-based detection: This technique uses machine learning algorithms to detect anomalies in network behavior.

Intrusion detection is an important aspect of wireless sensor network security, and careful consideration should be given to the selection and implementation of appropriate IDS to protect the network against potential security threats.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 03, March 2024)

II. METHODOLOGY

The proposed methodology is explained using following flow chart-

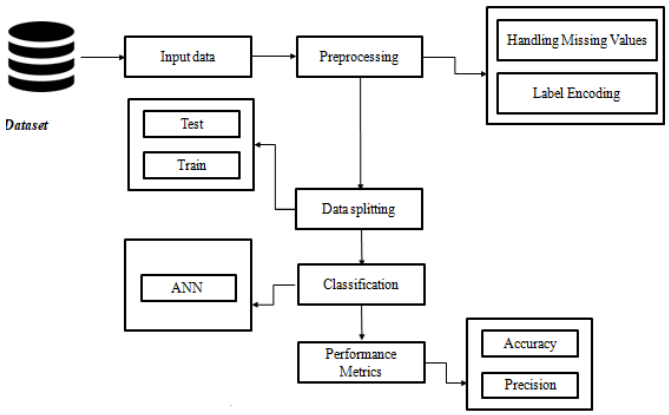


Figure 1: Flow Chart

Steps-

- First, using data from a publicly accessible, big dataset repository, create a final version of the dataset [13] based on the intrusion detection system.
- The missing dataset is currently being sent and preprocessing has been completed.
- The data that has already been processed has been divided into training and testing set.
- Recently, a method based on the classification power of artificial neural networks has been put into use.
- You should now assess several measures of performance such as F-measure, Precision, Accuracy, Recall, and Classification Error.

Methodologically, the suggested study is built on the following sub-modules::

Data Selection and Loading

- A data selection refers to the process of selecting a dataset and loading it into the Python environment.

Data Pre-processing

- "Noise or undesirable data" is removed from a dataset during data pre-processing.
- Data deficiency correction and categorical encoding
- Any blanks or nulls in the da-ta may be remedied with the help of the imputer library.
- Dissecting a Dataset into Test and Training Sets.

Splitting Dataset into Train and Test Data

- Separating Train and Test Sets "Data splitting" refers to the process of separating a dataset into two separate sets, often for the purpose of cross-validation.
- One portion is used to construct a prediction model, while the other is used to evaluate the efficacy of that model.

Feature Extraction

Normalizing the independent variables in a dataset is what feature extraction is all about. Data normalisation, also known as data cleansing, is a process that takes place before any actual data processing is done..

Classification

ANN- The artificial neurons of an ANN may be compared to nodes in a directed graph with weights. The link between a neuron's output and input is shown as a set of weighted directed edges. The Artificial Neural Network takes in data from an external source as a vector that represents a pattern and an image. The assigned value is represented by the mathematical expression x(n), where n is the number of inputs.

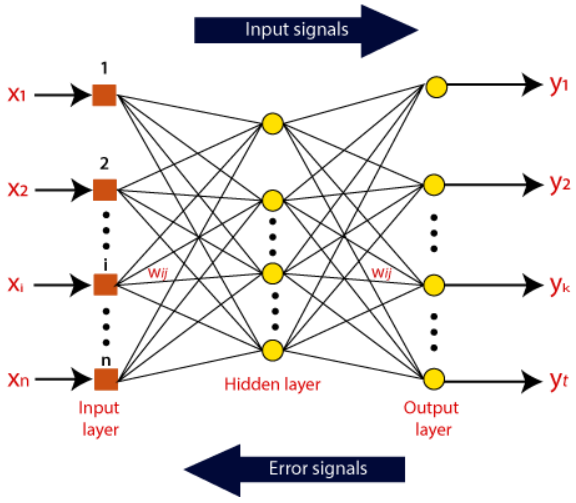


Figure 2: ANN

Following that, we multiply each input by its weight (these weights are the details utilised by the artificial neural networks to solve a specific problem). These weights are a common representation of the stability of the ANN's inter-neuron connections. A compilation of the input weights is stored inside the computing device.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 03, March 2024)

If the weighted sum is zero, bias is applied with the intention of boosting the system's response. Both the bias and weight parameters are set to 1. In this case, the total of the input weights might be zero or infinite. To limit the response to a usable range, we first establish a maximum value and then pass the combined weighted inputs through the activation function.

Prediction

- Using a strategy for predicting intrusion detection, this research was able to correctly anticipate the data from the dataset by enhancing the overall performance of the prediction results..

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.

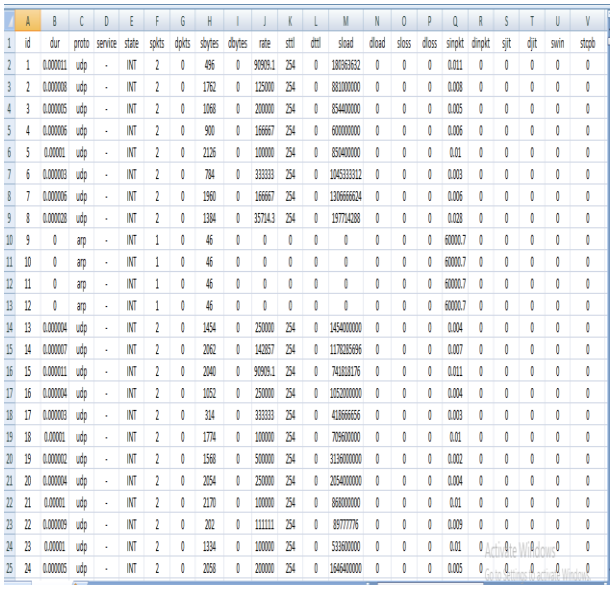
- Accuracy is defined as the ratio of true positives to total positives, while recall is defined as the ratio of positives to negatives.
- Accuracy = [TP +TN] / [TP+TN+FP+FN]; F1-Score = 2x (Precision x Recall)/(Precision + Recall)
- Classification Error = 100- Accuracy

Result Generation

We'll utilise all of the classifications and forecasts we made to produce the final product. The efficacy of the proposed approach is measured by accuracy, error rate, and comparable metrics.

III. SIMULATION RESULTS

The Python Spyder 3.7 is used as the integrated development environment for the simulation.



	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	157714288	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	1170285696	0	0	0	0.007	0	0	0	0	0
16	15	0.000011	udp	-	INT	2	0	2040	0	90909.1	254	0	741018176	0	0	0	0.011	0	0	0	0	0
17	16	0.000004	udp	-	INT	2	0	1052	0	250000	254	0	1052000000	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	89777776	0	0	0	0.009	0	0	0	0	0
24	23	0.00001	udp	-	INT	2	0	1334	0	100000	254	0	533800000	0	0	0	0.01	0	0	0	0	0
25	24	0.000005	udp	-	INT	2	0	2058	0	200000	254	0	1646400000	0	0	0	0.005	0	0	0	0	0

Figure 3: Dataset

Python is used to demonstrate the data set (Figure 3). The number of rows and columns in the dataset may fluctuate greatly. Each column contains the actual names of the characteristics being tracked.

International Journal of Recent Development in Engineering and Technology
Website: www.ijrdet.com (ISSN 2347 - 6435 (Online) Volume 13, Issue 03, March 2024)

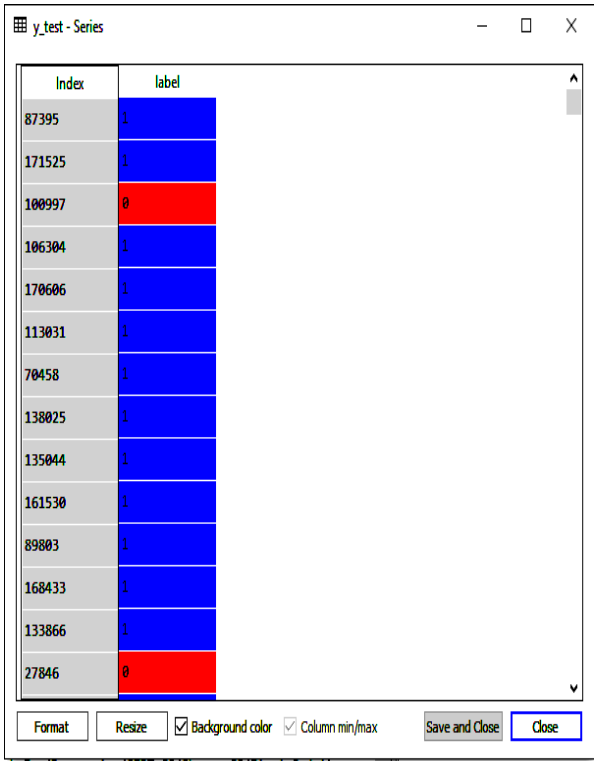


Figure 4: Y test

Figure 4 displays the results of a y test performed on this data set. Only 23% of the whole dataset is utilised for training purposes.

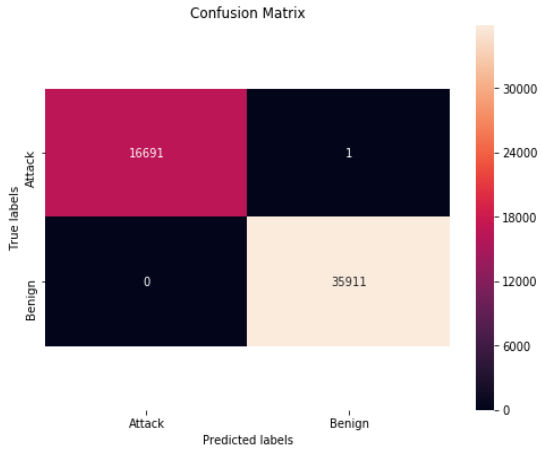


Figure 5: Confusion matrix heat map

Figure 5 displays confusion matrices for heat maps created using the ANN deep learning classification approach. It is a NN matrix used to evaluate the efficacy of a classification system.

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	Classification Error	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%	99.99%
2	Classification Error	1%	0.01%

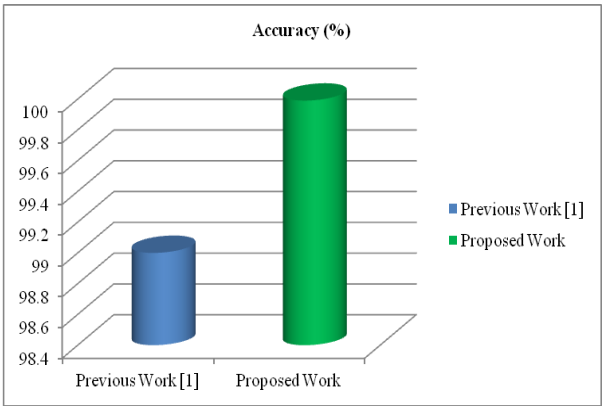


Figure 6: Accuracy Result graph

The precision is graphically shown in Figure 6. The planned work was more precise than previous efforts.

IV. CONCLUSION

In order to detect attacks on specific applications or computers, network administrators developed intrusion detection systems, often known as IDSs. A network intrusion system guards the cyber environment from potential threats. An attack prediction strategy may be obtained by the use of AI, ML, or deep learning. This research introduces an artificial neural network method for employing an intrusion detection system to foretell cyberattacks. The Python Spyder programme is used to carry out the simulation. By using the suggested ANN approach, we are able to reduce the average classification error to 0.01%, for a total accuracy of 99.99%.

REFERENCES

1. S. Subbiah, K. S. M. Anbananthen, S. Thangaraj, S. Kannan and D. Chelliah, "Intrusion detection technique in wireless sensor network using grid search random forest with Boruta feature selection algorithm," in Journal of Communications and Networks, vol. 24, no. 2, pp. 264-273, April 2022, doi: 10.23919/JCN.2022.000002.
2. H. W. Oleiwi, D. N. Mhawi and H. Al-Raweshidy, "MLTs-ADCNs: Machine Learning Techniques for Anomaly Detection in Communication Networks," in IEEE Access, 2022, doi: 10.1109/ACCESS.2022.3201869.
3. I. Mbona and J. H. P. Eloff, "Detecting Zero-Day Intrusion Attacks Using Semi-Supervised Machine Learning Approaches," in IEEE Access, vol. 10, pp. 69822-69838, 2022, doi: 10.1109/ACCESS.2022.3187116.
4. S. Otoum, N. Guizani and H. Mouftah, "On the Feasibility of Split Learning, Transfer Learning and Federated Learning for Preserving Security in ITS Systems," in IEEE Transactions on Intelligent Transportation Systems, doi: 10.1109/TITS.2022.3159092.
5. H. Siddharthan, T. Deepa and P. Chandhar, "SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features," in IEEE Access, vol. 10, pp. 33095-33110, 2022, doi: 10.1109/ACCESS.2022.3161566.
6. M. S. A. Muthanna, R. Alkanhel, A. Muthanna, A. Rafiq and W. A. M. Abdullah, "Towards SDN-Enabled, Intelligent Intrusion Detection System for Internet of Things (IoT)," in IEEE Access, vol. 10, pp. 22756-22768, 2022, doi: 10.1109/ACCESS.2022.3153716.
7. P. Freitas De Araujo-Filho, A. J. Pinheiro, G. Kaddoum, D. R. Campelo and F. L. Soares, "An Efficient Intrusion Prevention System for CAN: Hindering Cyber-Attacks With a Low-Cost Platform," in IEEE Access, vol. 9, pp. 166855-166869, 2021, doi: 10.1109/ACCESS.2021.3136147.
8. M. Ozkan-Okay, Ö. Aslan, R. Eryigit and R. Samet, "SABADT: Hybrid Intrusion Detection Approach for Cyber Attacks Identification in WLAN," in IEEE Access, vol. 9, pp. 157639-157653, 2021, doi: 10.1109/ACCESS.2021.3129600.
9. Y. K. Saheed and M. O. Arowolo, "Efficient Cyber Attack Detection on the Internet of Medical Things-Smart Environment Based on Deep Recurrent Neural Network and Machine Learning Algorithms," in IEEE Access, vol. 9, pp. 161546-161554, 2021, doi: 10.1109/ACCESS.2021.3128837.
10. A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021, doi: 10.1109/ACCESS.2021.3120626.