VOL. 16, NO. 19, OCTOBER 2021 ISSN 1819-6608
ARPN Journal of Engineering and Applied Sciences

www.arpnjournals.com

SECURING AND OPTIMIZING SENSOR NETWORK USING DEEP LEARNING ALGORITHMS

Vimal Kumar Stephen, Robin Rohit Vincent and Mohammed Tauqeer Ullah
Department of Information Technology, University of Technology and Applied Sciences, Ibra, Sultanate of Oman
E-Mail: vimal@ict.edu.om

ABSTRACT

Wireless sensor network (WSN) is a collection of sensor nodes that can sense various physical properties and communicate with one another in various ways. Security is a major concern in many real-world WSN applications. The goal of this work is to improve WSN security by identifying and countering adversarial denial-of-service (DoS) attacks. WSNs are subject to a variety of DoS assaults, depending on the layer they're attacking. This research employs neural network (NN) & support vector machine (SVM) machine learning approaches to detect denial-of-service (DoS) assaults on the MAC layer. After that, it assesses the effectiveness of the two approaches. Securing the MAC layer is critical because it allows sensor nodes to access wireless channels. The results revealed that these algorithms performs well in securing and optimizing the sensor networks.

Keywords: security, WSN, DoS, deep learning, neural network, SVM.

1. INTRODUCTION

A sensor is a device that collects data about a specific physical object or event. The collected data is wirelessly transmitted to a processing unit using a large number of sensors. A WSN is formed when a large number of these sensors work together to monitor a single physical environment. When addressing security concerns, WSNs present a unique set of challenges [1]. WSN has many constraints, such as a lack of available energy. WSN sensor nodes can be powered by batteries or solar energy. The amount of data they can store, how much computing they can do, and how much connection bandwidth they have are all limited. Therefore, security enhancement approaches' computing, communication, and storage needs must be met by sensor nodes with limited resources. A key focus of administration in sensor networks is also unfeasible because of the network's resource limits and dynamism. That's why we needed a decentralised security solution to this problem [2]. Due to their location and difficulty of access, many WSNs go unattended. Continuous monitoring and protection of sensor nodes is therefore difficult. It is common to use WSNs for a wide range of purposes, including emergency response and disaster relief. WSNs are also used in biodiversity mapping, machine surveillance, and precision agriculture. Many attacks can be launched against them. WSN security in such applications must be guaranteed. DoS attacks perpetrated by third parties are detected and countered using two machine learning techniques: NN and SVM.

WSNs will be more secure as a result. NN is an effective data-analysis tool. A multilayer perceptron receives input data sets containing key parameters reflecting the security level of the WSN (MLP). The backpropagation algorithm is used to train the MLP. The BP algorithm is used to reduce the NN's output's total squared error to the bare minimum. After that, an MLP is built on each node using the predefined weights from the training and applied. To detect denial-of-service (DoS) assaults, each node has an MLP. When an assault is detected, the system ceases to function. SVM can be used in place of NN to improve security. SVM utilises statistical learning theory as its foundation. Due to its great precision and versatility in dealing with large amounts of data, it is commonly utilised in industrial settings. The remainder of the document is structured as follows: Section II provides an analysis of Denial-of-Service (DoS) attacks and countermeasures. SVM and NN were employed to enhance security in Section III. As a result of the usage of both approaches, WSN security has improved, as shown in Section IV, and their performance has been examined. Section V includes a conclusion and recommendations for the future.

2. RELATED STUDIES

Attacks in WSN have been classified into various types are vulnerable to denial-of-service (DoS) attacks by their enemies in order to disrupt the service they provide. Such threats can be carried out at various levels of a WSN's architecture. DoS attacks on a typical sensor network were mentioned [3]. They've stated that the resource limitations of WSN's cryptographic authentication mechanisms make them vulnerable to denial-of-service (DoS) attacks [4]. Preventing denial-of-service (DoS) attacks is a critical part of securing adhoc sensor networks, and they propose that identifying malicious nodes can counteract DoS attacks. A fuzzy logic-based DoS detecting MAC protocol has been presented [5]. A hybrid intelligent detection mechanism [6] detects attacks. In order to make judgments, fuzzy interference was employed, and to learn attack definitions, a NN-based technique was used. This hybrid method, on the other hand, has made it easier to spot attacks in part. G. C. Y. Sang *et al.* [7] use SVM or a Radial Basis Function Neural Network to forecast Denial-of-Service (DoS) attacks on web servers (RBFNN). Experiments are carried out by researchers in order to compare the two techniques of machine learning. SVM has been found to be more precise than RBFNN in this study. According to the research, a medium protocol based on generalised neuron (GN) technology could be used to defend against distributed denial of service (DDoS) attacks [8]. An

optimization algorithm known as the particle swarm is used to train the GN. To see how threshold suspicion parameters affect network throughput vs. longevity, we ran simulations. An adversary's DoS assault on WSN's MAC layer is detected and countered using machine learning techniques described in this study. The sensor nodes learn the attack definitions, so when they detect an attack, they will cease to function until the intruder has left their range.

3. DETECTION OF DOS ATTACKS

An adversary's attempt to impair the network's services is called a denial of service (DoS) attack. Malicious nodes can overwhelm legitimate nodes in a denial-of-service attack by flooding them with requests [8]. WSN has a Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) mechanism as one of its features. Ready-to-send (RTS) and clear–to–send (CTS) packets are exchanged in this technique. An RTS packet is sent by the source node when it has data to send. Any node that receives an RTS packet is silenced. Receiving an RTS causes the target node to issue a CTS in response. The CTS packet, like the RTS packet, mutes any nearby nodes. Once the RTS/CTS exchange is complete, the source node can transmit data unhindered by other nodes. The packets of data have been accepted. The protocol stack is divided into several layers, each of which is vulnerable to a different sort of Denial of Service attack. At the MAC layer, three DoS attack types exist: collision, unfairness, and fatigue [9].

- Collision attack

All nodes sense the channel to determine if it is busy or idle before issuing RTS/CTS packets. They only transmit data when the channel is free. As a result, while delivering data packets, no collisions occur. When this is the case, attackers can attack the sensor network by sending a large number of packets at once, causing packets to collide.

- Unfairness attack

To get same channel, all nodes have the same priority. It's first come, first served (FCFS) policy that determines who gets the channel, so the first node to try it gets it. When this occurs, the adversary sends out a huge number of packets quickly or immediately. In this way, malicious nodes are prevented from making use of a public channel.

- Exhaustion attack

Upon receiving an RTS control message, the sensor node responds with a CTS acknowledgement packet. Since attackers are normal nodes, legitimate nodes cannot identify when an RTS packet is being sent by a normal node versus an attacker. As a result, the attackers bombard the regular nodes with RTS messages, which they acknowledge by sending back CTS packets. It has the unfortunate side effect of draining receiver batteries faster than normal.

The research in [10] detects the likelihood of an attack by using the following sensitive parameters:

- Rc is the number of collisions that a node detects in a second.
- With respect to RTS packet arrival rate, it is the number of correctly received RTS messages sent by a node per second. (RTS arrival rate)
- When a packet is waiting in a MAC buffer before transmission, it is known as the average waiting time (Tw).

A range of assault probabilities from 0.1 to 1 are kept track of in the above-mentioned sensitive parameters. When Tw is compared to R_r and R_c, the difference is insignificant. Since DoS attack likelihood is detected using the sensitive parameters, these parameters will be used in this example. The multilayer perceptron receives these parameters as inputs and the appropriate attack probability as targets from the neural network-based technique (MLP) [11]. In order to train the MLP, backpropagation technique is used. A trained MLP is used to train the weights and biases of the MLP applied at each node in the network. Each node sends its R_c and R_r values to its MLP once every minute. The likelihood of an attack on a node can be calculated using MLP as an input. It shuts down automatically if it senses an impending attack. The likelihood of an assault is classified as either low or high using an SVM-based technique. The sensitive parameters R_c or R_r from two classes are used to train the SVM classifier. R_c and R_r parameters are inputs to the trained SVM classifier every minute, and it uses this information to determine whether the attack probability is low or high. If it senses an assault with a high likelihood, the node automatically shuts down.

4. METHODS

A. NN-Based Methodologies

The neural network employed is an MLP. MLP is a neural network (NN) in which neurons are stacked one on top of the other. Input units make up the top layer, while output units make up the bottom layer [12]. In the hidden layer, every other unit is known as a hidden unit. Directed communication linkages connect every neuron to every other neuron. A value is assigned to each link in the chain of communication. The neural network's weights represent the data represented by the weights. The activity of each neuron is a function of all inputs received by that particular neuron. Figure-1 depicts the MLP structure that was employed. There's a layer beneath the surface that's not visible. The input units are denoted by X*1* and X*2*. Biases exist in the Y*1* output and Z*1* hidden unit. W_{01} denotes the output bias on Y*1* (the first output unit). V_{01} denotes the bias on Z*1*'s hidden unit. The hyperbolic tangent sigmoid function is used to activate the hidden layer. Figure-2 depicts this function. It has the same numerical value as (1).

$$y = f(x) = \tanh(x)$$

The output layer's activation function is a linear function. Figure-3 shows you what I'm talking about. This function has the same numerical value as (2).

$$Y = f(x) = x$$

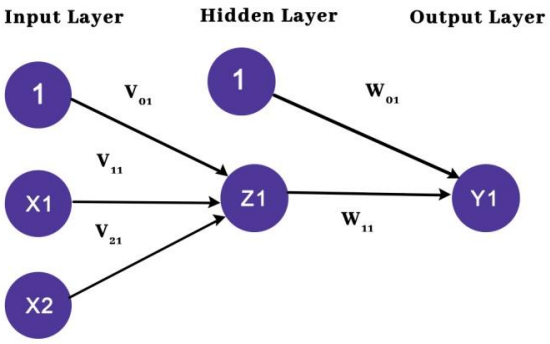

Figure-1. Structure of MLP.

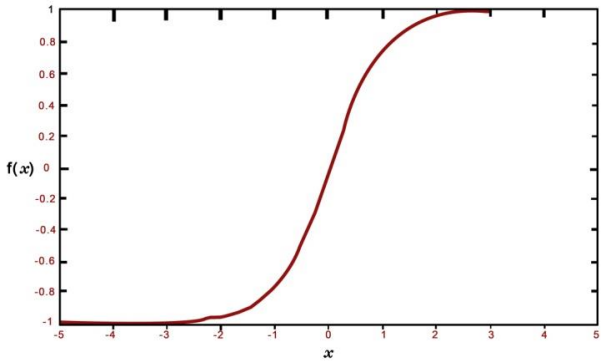

Figure-2. Hyperbolic tan sigma.

a) Training algorithm: The BP method is used to train MLP [11]. Three stages are involved:

- Input pattern feedforward.
- Backpropagation of the associated mistake is calculated and carried out.
- Weights must be adjusted.

Each input unit Xi, (i = 1, 2) receives information during feedforward. xi is the input signal, and Z*1* is the hidden unit that receives it. Z*1*combines its input signals that have been weighted, and it does so thus (3).

$$z_{in} = v_{01} + \sum_{i=1}^{2} x_i v_{i1}$$

After computing its output signal in (4), Z*1* uses its activation function to deliver the signal to the output unit Y*1*.

Z1 = f (Zin)

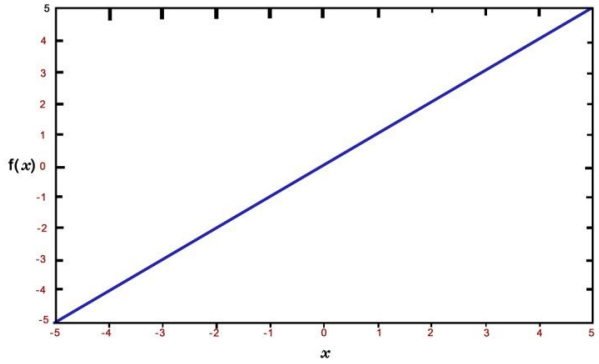

Figure-3. Linear function.

The Y*1* output unit sums its input signals that have been weighted according to (5).

Yin1 =w01 +z1w11

the output signal of Y1 is computed by using its activation function, where (6).

Yin =f (yin)

As part of the learning process, the output unit Y1 assesses its activation y1 against its target value t1. This error is used to calculate the factorδ1. The error at output unit Y1 is propagated back to hidden unit Z1 using the value δ1. It's used to keep the weights in sync between the visible layer and the hidden one. As a result, the weights between the input layer and the hidden layer are also adjusted to reflect the new information. For BP to train a NN, many epochs are required. Gradient descent serves as the algorithm's mathematical foundation.

B. Method Based on Support Vector Machines (SVM)

An SVM is a statistical learning method used for a wide range of tasks, including pattern recognition and denoising. [13] The probability of an assault is calculated using an SVM-based classification algorithm in this paper. A set of data is fed into the learning machine so it can be trained. The data used in the training is binary labelled, with lower and higher values indicating a lower and higher assault likelihood, respectively. For each class, the learning machine seeks to discover a hyperplane and two bounding planes that are as far apart as possible, separating the classes "L" (lower attack probability) and "H" (higher attack probability) (Higher probability of attack) [14]. There are n-dimensional input vectors and m-dimensional target vectors in which di1,+1 specifies the class to which each pair (xi, di) belongs. Consider the training set of these pairings (xi, di), where I = 1 and... and n. Maximum separation hyperplane is denoted by the notation, or the bounding hyperplanes are denoted by, respectively, the notation, and the notation. It is possible to have input vectors that satisfy the constraint w^T x 1 and those that satisfy the constraint w^T x 1 and those that satisfy both.

However, we expect few errors in our situation, thus there is a possibility that some of the input vectors depart from their corresponding bounding plane. To meet the limitations, a positive number, referred to as the slack variable, is added or subtracted from the input vector. As a result, the additional restrictions are written as follows:

$$w^T x - \gamma + \xi \geq 1$$
$$w^T x - \gamma - \xi \leq -1$$

SVM's goal is to minimise the number of input vectors and maximise the margin between bounding planes. Minimizing 1 2wTw gives the best margin, while reducing mP i=1 I yields the lowest error. The following is the formulation in its most basic form:

$$\min_{w,\gamma,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{m} \xi_i$$

Subjects to constraints

$$d_i(w^T x_i - \gamma) + \xi_i - 1 \geq 0, 1 \leq i \leq m$$
$$\xi \geq 0,\ 1 \leq i \leq m$$

where the penalty parameter '*C*' regulates the weighting for the maximum error margin and the sum of the errors in the equations. where The classifier's generalisation power is high because of the value of '*C*.' Quadratic programming is used to solve the problem stated in (8) in its primal form and its dual form. Lagrangian multipliers are used to find the answers [15]. The primal variables ξ, γ and w are calculated using the Lagrangian multipliers. It is best to move the input space to a higher-dimensional space (xi) and then split the hyperplane to the greatest extent in that space.. The following is the formulation in dual form:

$$\min_u L_D(u) = \frac{1}{2} u^T Q u - e^T u$$

With subject to constraints.

$$0 \leq u \leq Ce$$

in where D = diag(d), Q = DKD, Khas been given the kernel matrix, and e^T is the total number of non-negative errors. In this research, we used the GRBF kernel, which is a Gaussian radial basis function.

$$K(x_i, x_j) = \exp\left(\frac{-\|x_i - x_j\|^2}{2\sigma^2}\right)$$

GRBF's '*C*' and 'σ' values have been fine-tuned to reach the highest possible degree of precision. The following is the decision function for the test data:

$$f(x) = \text{sign}\left(\sum_{i=1}^{m} d_i u_i K(x_i, x) - \gamma\right)$$
$$= \text{sign}\left(\sum_{i=1}^{m} d_i u_i (x_i^T x) - \gamma\right)$$

5. RESULTS

A. Simulation Application

The probabilistic wireless network simulator is used to model the WSN scenario (Prowler). Prowler's most notable attributes are as follows:

- Driven by events (event-driven)
- Use deterministic or probabilistic operation to imitate non-deterministic communication channel
- Has the ability to support any number of mites in any topology.
- Optimizing algorithms can easily be integrated with this.
- Runs under MATLAB, therefore it gives you access to prototyping applications quickly and easily
- Ability to see things visually
- Models the communication channel at all levels and the application

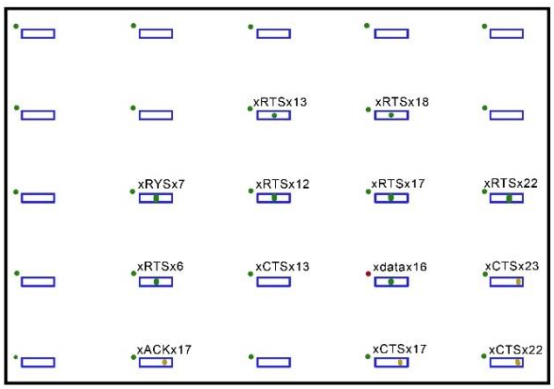

Figure-4. WSN used for significant parameters.

Table-1. Critical parameters averaged over 50 trial runs.

Probability of attack	R_r	R_c
0.1	394.5	111.39
0.2	595.1	125.1
0.3	586.6	136.7
0.4	651.9	164.5
0.5	719.2	178.1
0.6	922.1	221.1
0.7	992.6	240.1
0.8	1057.3	261.2
0.9	1130.33	281.3
1	1191	302.4

B. Results in Numeric Form

Figure-4 depicts a WSN scenario for monitoring the essential parameters. There are 25 sensor nodes in total, each with an ID ranging from 1 to 25. The RTS/CTS control technique is used to exchange data between the nodes. Every 0.25 seconds, each node makes an attempt to send a packet with a probability of success of P. When two nodes communicate at the same time, a collision occurs at the receiver. The request rate R*r* is the number of RTS packets received by a node in a minute. Collision rate R*c* is a metric for determining the average number of collisions per minute. For 50 trials, the values of these parameters are estimated at various probabilities of DoS attack ranging from 0.1 to 1. After that, the numbers are normalised. The likelihood of an assault is a metric for how likely something is to go wrong. At various probabilities of assault, the values shown in Table I correspond to those in Table II. There are graphs showing normalised values of these parameters shown in Figures 5 and 6. Table I shows that R_c or R_r rise in proportion to the risk of attack, as can be shown. In NN-based and SVM-based approaches, training inputs consist of normalised values of these parameters and the accompanying probabilities.

C. Using a NN-Based Strategy for DoS Attack Protection

Figures 5 and 6 show normalised values of crucial parameters that are plotted and sent to the MLP as inputs and targets, respectively. This algorithm uses the MLP's output values (which are actually the chance of attack) to train it to find the appropriate targets. As can be seen in Figure 7, the input values (normalised R_c) are

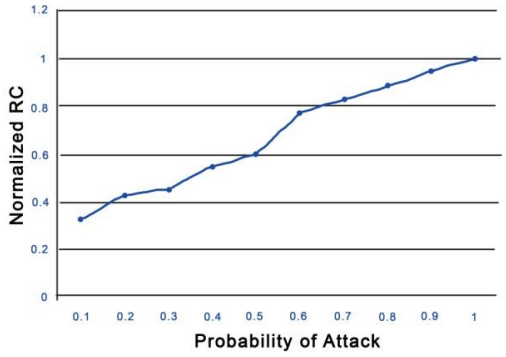

Figure-5. Normalised R_c through fifty trial runs.

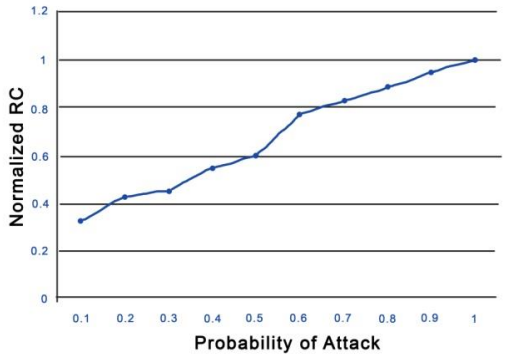

Figure-6. Normalised R_r through fifty trial runs.

and R*r*) versus the desired values (*t*). Figure-8 depicts the relationship between the input values (*x*) and the MLP's outputs (*y*) after it was trained by BP. The best trainable parameters are derived from the trained MLP, and a new MLP is created. Every minute, each node uses its own MLP to check for attack probabilities. In the event that a node detects an attack, it will go offline. Figure-9 depicts the WSN scenario used to detect a denial-of-service attack. Node 1, denoted by the letter 'xx,' is the bad guy in this scenario. A rectangle denotes the location of the opponent. The antagonist is free to move around the scenario as they see fit. In this instance, it results in a constant

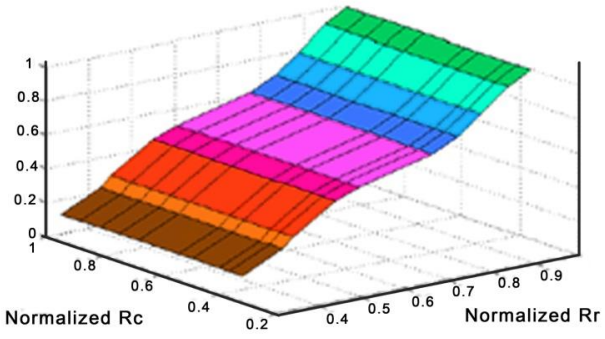

Figure-7. Parameters normalised to values for MLP training (t).

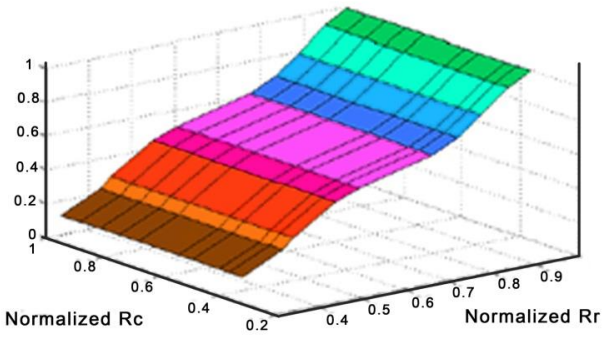

Figure-8. Comparing normalised values with trained MLP's values (y).

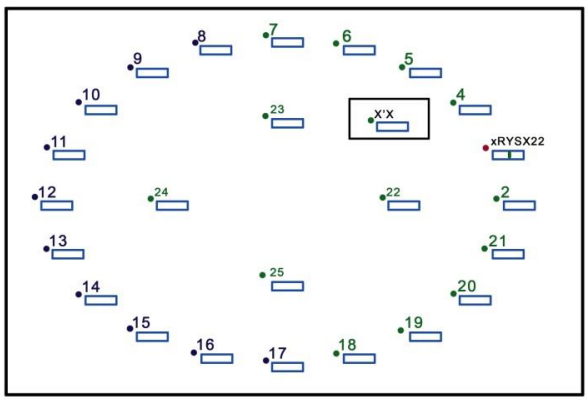

Figure-9. WSN for DoS attack detection.

a strategy based on statistical likelihood Normal nodes, on the other hand, use the CSMA/CA technique to send packets. The MLP senses the parameters R_c & R_r once every minute and then sends the appropriate output to shut off the nodes. The nodes in Figure-10 have all ceased working when they identified the attack (see Figure-10). The red LEDs on these nodes indicate that they are active. They've got little circles on them to indicate where they've been. This node will become active and transmit data when it detects no further attacks for one minute.

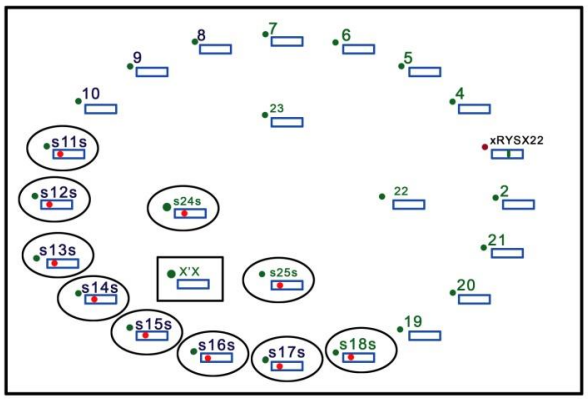

Figure-10. Stopping of nodes on detecting attack using neural network.

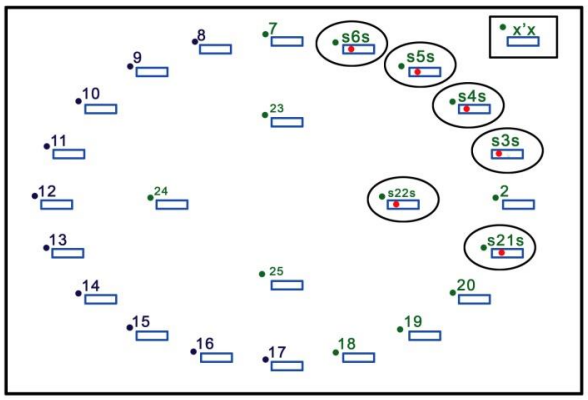

Figure-11. Stopping of nodes on detecting attack using SVM.

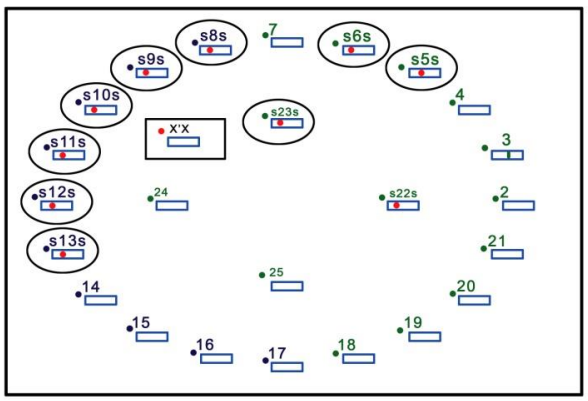

Figure-12. Stopping of nodes on detecting attack using neural network in burst traffic.

D. Use of SVM for DoS Attack Defence

Both low and high attack probabilities exist for the normalised crucial parameters. SVM is trained using these values. There are two classes, and the trained SVM classifies each node every minute based on the values of

important parameters. When a high attack likelihood is recognised, the system ceases to function. Figure-11 shows that all of the nodes that have detected an attack have gone down for the count. The red LEDs on these nodes indicate that they are active. They've got little circles on them to indicate where they've been. This node will become active and transmit data when it detects no further attacks for one minute.

E. Bursty Traffic Security against DENIAL of Service (DoS) Attacks

In bursty traffic, an attack can be detected using a NN-based technique. Previous studies have found that traffic conditions are static, meaning the likelihood of an attack is always the same. Attack likelihood changes with time in bursty traffic, i.e. the attack probability changes with time. The nodes in Figure-12 have all ceased working when they identified the attack (see Figure-12). The red LEDs on these nodes indicate that they are active. They've got little circles on them to indicate where they've been. The nodes that have been assaulted will become inactive if they do not detect an attack within the next minute.

Table-2. Performance Analysis Of Nn And Svm.

Performance metric	NN	SVM
Accuracy (in percentage)	91.43	97.14
Time Taken (in seconds)	0.75	0.25

F. An Examination of How Well Something Works

As shown in Table II, NN outperforms SVM in detecting DoS attacks, both in terms of accuracy and output time. SVM has a 97% accuracy rate, whereas NN has a 91% accuracy rate. While computing the likelihood of an attack, NN requires 0.75 seconds; when computing the probability of an attack, SVM only needs 0.25 seconds.

6. CONCLUSIONS

Two approaches to artificial intelligence (AI) DoS attack detection makes use of SVM and neural networks. No need to reprogram them because they learn by doing. They utilise supervised learning to accomplish their goals. The attack nodes can save power, which extends the life of the network. Narrow-casting neural networks have the ability to solve issues that linear programmes can't. A global minimum is discovered by SVM training using kernel-based approaches. According to the comparison of the two systems' performance, SVM detects DoS assaults with a precision of 97%, while NN does it with a precision of 91%. SVM is faster and more accurate than NN. As a result, instead of using NN to detect DoS attacks, an SVM-based technique is favoured. This work is further developed in the future in a variety of ways. Unsupervised learning can be used in instances when goal outputs aren't available as a technique to get around this problem. The inputs are grouped in this approach. There are a finite number of input patterns that can be classified. Using alternative brain designs like generalised neurons or radial basis networks is the second approach. When employing the radial basis network, you have the advantage of only using local approximators to obtain the input-to-output map. It's quick to learn and doesn't necessitate a lot of practise data.

REFERENCES

[1] C. Shivalingagowda, H. Ahmad, P. V. Y. Jayasree and D. K. Sah. 2021. Wireless Sensor Network Routing Protocols Using Machine Learning. in Architectural Wireless Networks Solutions and Security Issues, Springer Singapore. pp. 99-120.

[2] A. Ghosh, C. C. Ho and R. Bestak. 2020. Secured Energy-Efficient Routing in Wireless Sensor Networks Using Machine Learning Algorithm. in Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks, IGI Global. pp. 23-41.

[3] Vijayakumar. 2020. Application of Machine Learning in Wireless Sensor Network. in Encyclopedia of Wireless Networks, Springer International Publishing. pp. 21-27.

[4] E. B. Priyanka, S. Thangavel and D. V. Prabu. 2020. Fundamentals of Wireless Sensor Networks Using Machine Learning Approaches. in Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks, IGI Global. pp. 233-254.

[5] L. Alsulaiman and S. Al-Ahmadi. 2021. Performance Evaluation of Machine Learning Techniques for DOS Detection in Wireless Sensor Network. International Journal of Network Security & Its Applications, (2): 21-29, doi: 10.5121/ijnsa.2021.13202.

[6] V. Gulyani, T. Dhiman and B. Bhushan. 2020. Introducing Machine Learning to Wireless Sensor Networks. in Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks, IGI Global. pp. 1-22.

[7] M. Uma. 2016. Enhanced Security In Data Transmission In Wireless Sensor Network Using Randomized Path. International Journal of Engineering and Computer Science, doi: 10.18535/ijecs/v5i1.04.

[8] A. G. A. Poornima and B. Paramasivan. 2020. Anomaly detection in wireless sensor network using machine learning algorithm. Computer Communications, pp. 331-337, doi: 10.1016/j.comcom.2020.01.005.

[9] T. Issac, S. Silas and E. B. Rajsingh. 2020. Modelling a Deep Learning-Based Wireless Sensor Network Task Assignment Algorithm. in Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks, IGI Global. pp. 84-109.

[10] B. Sharma. 2020. Wireless Sensor Network Security," in Encyclopedia of Wireless Networks, Springer International Publishing. pp. 1497-1501.

[11] S. M. Mazinani and M. Safari. 2015. Secure Localization Approach in Wireless Sensor Network. International Journal of Machine Learning and Computing, (6): 458-461, doi: 10.18178/ijmlc.2015.5.6.552.

[12] J. Oetjen. 2019. Using artificial intelligence in the fight against spam. Network Security, (7): 17-19, doi: 10.1016/s1353-4858(19)30086-8.

[13] R. Rani. 2018. Distributed Query Processing Optimization in Wireless Sensor Network Using Artificial Immune System. in Computational Intelligence in Sensor Networks, Springer Berlin Heidelberg. pp. 1-23.

[14] M. Gilbert. 2018. The Role of Artificial Intelligence for Network Automation and Security. in Artificial Intelligence for Autonomous Networks, Chapman and Hall/CRC. pp. 1-23.

[15] Prof. A. N. Singh. 2016. Importance of Wireless Sensor Network and Artificial Intelligence for Safety Prerequisite in Mines. International Journal of Engineering and Computer Science, doi: 10.18535/ijecs/v5i11.35.

[16] K. Vimal Kumar Stephen, V. Mathivanan. 2018. An energy aware secure wireless network using particle swarm optimization. Majan International Conference (MIC), 2018/3/19, IEEE.

[17] K. Vimal Kumar Stephen, V. Mathivanan. 2017. Back propagating tree to produce an optimal path to transmit data in a wireless sensor networks. ARPN Journal of Engineering and Applied Sciences, Vol. 12, no. 23, December 2017, ISSN 1819-6608.