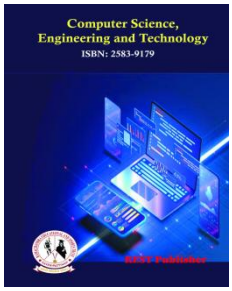
**Computer Science, Engineering and Technology**

**Vol: 1(1), March 2023**

**REST Publisher; ISSN: 2583-9179 (Online)**

**Website: https://restpublisher.com/journals/cset/**

**DOI: https://doi.org/10.46632/cset/1/1/17**




# Sequential Pattern Artificial Neural Network (ANN) Method for Predicting Attacker Transaction in Wireless Sensor Network (WSN)

***A. Neelamadheswari, S. Saranya, S. Yuvaraj, P. Deepanchakravarthi**

*Mahendra Engineering College, Namakkal, Tamil Nadu, India.*

**Corresponding Author Email: neelamadheswaria@mahendra.info*

***Abstract:*** *Nowadays, Wireless Sensor Network (WSN) have created tremendous prominence because this evident benefits while comparing to the conventional cable network are simple in distribution, users mobility over the area of network coverage and ease in new user's connection. With sudden increase of communications in digital technologies are depressed through traffic of network data. The usage demand in an internet has grown daily use for the present cyber world have influences the requirement of network security. The WSN security can be supported through the strategies of attack and defense using Machine Learning (ML) which perform automated in learning from and adapt to the characteristics of wireless communication which are firm in capturing using the features and model of hand-crafted. However, several security attacks have classified as cyber ones and the attack types are interested in the system tentative as well as spoils the data instead of data stealing. Hence, the paper focuses on Intrusion Detection System (IDS) has done through Stacked Autoencoder (SAE) with Artificial Neural Network (ANN) in sequential patterns method is produced improved security in DoS attacks. Moreover, the dataset of security type for DoS in WSN has mainly focused in identifying attacks, malicious node elimination and transmission data security.*

***Keywords:*** *Denial of Service (DoS), Wireless Sensor Network, Intrusion Detection System, Artificial Neural Network, sequential pattern*

## 1. INTRODUCTION

WSN is the most efficient approaches for numerous real-time applications because of its small size, low cost, and simplicity in implementation [1]. The WSN's operation is to track the interest field, data collection, and send accessing point for analyzing post processing. Certain WSN setups have a significant sensor node amounts. Traditional WSN security methods like cryptography, key management and spread spectrum can fail to identify attacks accurately and might require complex hardware and software modification, rendering these options inadequate for addressing WSN security issues as WSN gadgets limit the network's communication, storage, power and computational features [2]. There is an increasing curiosity in new security paradigms including cybersecurity firms spending up to USD 119 billion to deal with these challenges [3]. These has resulted in recently evolving methods for enhancing WSN security against potential intrusions using Machine Learning (ML). The implementation of Artificial Intelligence (AI) in a variety of sectors is currently attracting global attention. Within this broader context, Wireless Sensor Networks (WSNs) emerge as a significant facilitator, exerting their influence in a variety of industries, including medical facilities, industrial ecosystems, precision agriculture and smart infrastructure [4]. In contrast, malicious threats constantly change as well as increasing necessitating the need of a sophisticated network security mechanism. Since, widespread adoption of the Internet, there are increasingly computers interconnected by networks. In recent years, data science approaches are being used to develop effective IDSs that have capable of distinguishing among legitimate as well as unauthorized communications. IDSs are frequently utilized in communication networks for identifying and prevent harmful activities. IDSs are classified into three types are anomaly-based systems, hybrid systems, and systems that use signatures to identify intrusion. Although signature-based can differentiate between existing intrusions with related gathered signatures, it also exhibits a high prevalence of falsified alerts [5]. The network and application layers are executed in devices with high power consumption to ensure data security, whereas the perception layer is executed in less power WSN.

However, WSN is made up of several sensor nodes who communicate with one another over various radio frequencies as well as may perform a variety of sensing, measurement, tracking and surveillance. Those wireless nodes have resource-constrained gadgets, with low computing power, confined battery life, and limited memory capacity [6].

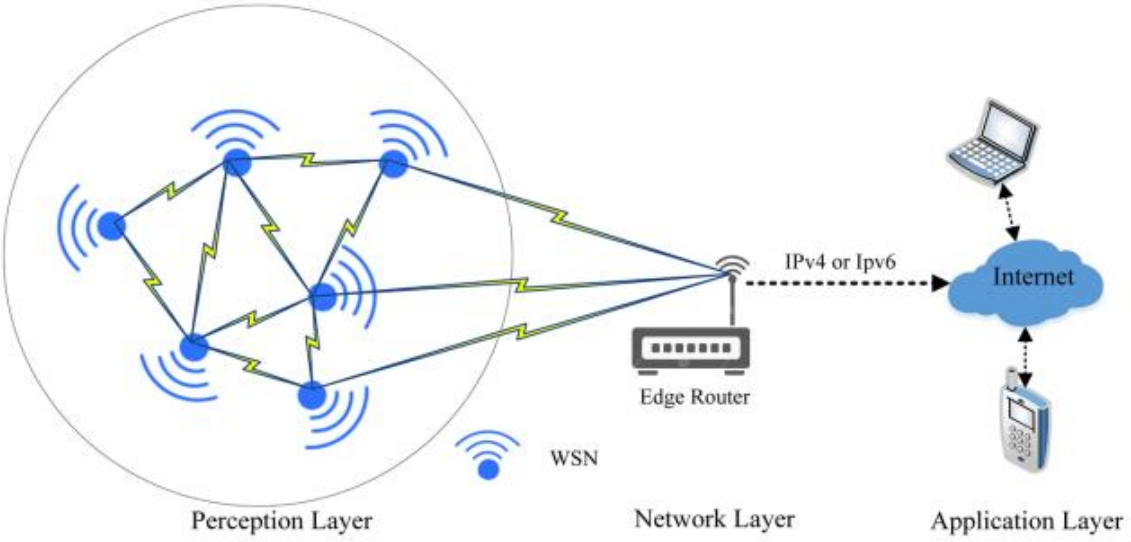



**FIGURE 1**. Communication among the WSN layers

Figure 2 illustrates the communication among WSN layers whereas WSNs has mapped the network topology as well as update the routing database in the perception layer employing different protocols to sustain the infrastructure of the network. The WSN subsequently starts acquiring data from different places and sends it to the edge router as network layer. The WSN nodes are the fundamental components of this layer, while they have specific properties that distinguish them apart from other kinds of networks that are wireless. Within these attributes are as follows

- Network topology of WSN is modified constantly
- Autonomous nodes with no central control
- Mobile or stationary WSN nodes
- WSN transmission range node has been limited
- Limited bandwidth
- Multi-hop connections

Generally, key packet segments are signed with encryption prior to sent from the sending node, then decrypted at the receiving node. In order to maintain integrity, the network needs to be designed such that intruders are unable to change the data being sent. Attackers may utilize interference beams to modify their poles for defense. A rogue routing node additionally has the ability modify crucial data in packets prior sending them. The final prerequisite for achieving the security trinity is access. Availability refers to the functioning of WSN Services at any point in time. In any instance, attackers may initiate attacks which impair network performance or completely disable the network. The most significant risk for network reliability is Denial of Service (DoS) [7]. It occurs when attackers, by delivering wireless interference which is altering network protocols, or draining WSN nodes in various manners prevent the network from establishing services. The User Datagram Protocol (UDP) is a prevalent transport protocol used in 6LoWPAN, and it may be combined with the Datagram Transport Layer Security (DTLS) protocols to assure security of data [8]. At the same time, TLS is implemented using the Transmission Control Protocol (TCP), as well as link-layer encryption and authentication is performed using the AES-128 algorithm. However, the DTLS implementations necessitate the usage of further encryption hardware for maintaining sophisticated encryption processes. Furthermore, it is challenging to incorporate Internet Protocol Security (IPSec), which is usually employed at the network layer and Transport Layer Security (TLS) towards network applications since both protocols include substantial overhead expenses and require significant amounts of resources. In today's world, ML is an important and popular tool whereas assured WSNs will perform tasks related to security with no requiring the use of ML techniques. Autonomous government is frequently referred to as self-governing administration, refers to an organization or entity's capacity to regulate themselves separately despite external intervention or influence. Traditional network security solutions in contrast are unable to properly safeguard WSNs because of restricted resources and computational power. The effectiveness of using

ML approaches to improve the WSNs security is widely documented. The concept of "user authorization" is judged improper in this context. The researcher used ML classification algorithms like RF, NB and KNN to explore the malware networks functional mechanics in IoT. The researchers discovered KNN approach produced the most trustworthy findings and demonstrated the possibility of privacy concerns resolution to permit SVM training using IoT data [9]. The two transactions possess the ability to be successfully completed in a single cycle with no use of an unknown party. Using the standard SVM approach involves more complexity than using this alternate strategy. As an outcome, effective utilization of ML technology may reduce security costs. Anomaly detection may assist in avoiding a variety of negative outcomes, including DoS assaults and monitoring packet analysis. In addition, ML assists to find physical layer methods by allowing network connectivity, minimizing traffic congestion as well as fault detection. This study is divided into two sections namely various intruder types in Sensor Networks utilizing SAE technique with various ML methods and identifying different assaults in WSN utilizing ML methods.

## 2. LIERATURE REVIEW

This paper provides a complete discussion of WSN security techniques with a special focus on WSN security. In this context, several prior surveys have concentrated on delivering various ML strategies for WSN security. S. El Khediri has employed deep Neural Networks (NNs) for creating adaptable IDS. The statistical analysis demonstrated about developing appropriate inferences from different network traffic had become easier as well as more feasible solutions [10]. M. Z. Ghawy have combined particle swarm optimization and backpropagation NNs for creating a solution to WSNs and capable of detecting modest infiltration levels [11]. Similarly E. Mushtaq et al. have advanced improved IDS by combining a two-level classifier using hybrid feature selection method. To determine the malfunctioning WSN nodes, information about traffic was examined employing SVM and MLP method. In contrast, numerous authors have utilized several ML techniques to create a hybrid classifier. D. Wu et al. have provided a full description of a classifier that combines ML and DL approaches. The approaches outperformed a competing technique which employed an MLP model as well as a GA for detecting effects in WSNs. The LTSM and Gaussian Bayes methods were coupled to produce this outcome [13]. A. A. Najar and S. M. Naik have described the methods for training ML model for identifying the IDS goal. This concept is based on a hierarchical structure that includes management positions [14]. Lai et al. have discussed security challenges faced in WSN that may exhaust their restricted resources of energy. Conventional security methods are ineffective owing of communication as well as resource limitations. This paper suggests online learning methods for detecting DoS intruder in WSNs. It presents a FS approach as well as a noise-tolerant, online aggressively passive multi-class classifier. The approach is judged on accuracy,recall, F1-score and precision indicating performance in competition [15]. Ahmad et al. have discussed about WSNs face substantial energy as well as security challenges. When security complexities raise it also influences energy usage whereas the conventional security protocols may ineffective with WSNs because of their limited resources. This research examines the ML methods potential to improve WSN security with minimal costs. It discusses the problems as well as remedies for sensors for identifying threats, intruder as well as malicious nodes using ML. Open questions of adapting ML methods in WSN abilities were also addressed [16]. Salmi et al. have discussed about various security risks particularly in DoS intruders. Conventional IDS are losing effectiveness over sophisticated and complicated attackers. The present paper analyzes previous research on DoS recognition of attacks in WSNs as well as creates DL-based IDS based on a customized dataset. These networks are examined and contrasted with respect to four DoS attack types involve [17]

1. Blackhole attack
2. Grayhole attack
3. Flooding attack
4. Scheduling attack

Gebremariam et al. have focuses on recognizing and localizing malicious nodes in WSNs that may assist in making the network last a while and be more valuable. Anchor nodes having established placements are employed to approximate the position of unidentified nodes. There are several localization approaches for exact node estimate. However, determining appropriate network settings to accurate node localization in network configuration remained difficult. Routing attacks like Sybil, wormhole, blackhole, and replay attacks, may have an influence on the precision of localization as well as quality services in WSNs. This paper provides a safe localization as well as routing threat detection strategy in WSNs that employs optimal hybrid ML approaches to optimize distance, location, and data transfer. This method uses the benchmark datasets CICIDS2017 and UNSW NB15 to determine the average localization accuracy and locate malicious nodes. The method accomplishes 100% average detection accuracy while also greatly improving localization accuracy, having an average error as 0.191 [18]. Mounica et al. have discusses both military and civilian applications using WSN, especially for sensitive environments such as battlefields. Designing security protocols such networks is crucial for improving its reliability and Quality of Service (QoS). However, using WSNs for security exposed it to a variety of malware

as well as hacking attempts. The Sybil Attack is an important threat, in which malicious nodes imitate numerous bogus credentials at the same time, misleading legitimate nodes. To solve this issue, a ML approach has been suggested for identifying Sybil attacks by evaluating raw traffic data along with discriminating between approved and illegitimate Access Points (APs) in a wired as well as wireless context [19]. Gebremariam et al. have developed security improvements in WSNs are critical, especially for preventing routing attacks which introduce rogue nodes. Sybil attacks are frequent routing assaults that create bogus nodes. This research presents detection as well as localization strategy that uses an optimized Multilayer Perceptron ANN (MLPANN) to prevent a variety of hazards including DoS. The method is constructed and tested on datasets used as benchmarks, resulting in excellent accuracy in detection as well as exact localization which makes it appropriate for scaled as well as hierarchical distributed WSNs [20].

## 3. RESEARCH METHODOLOGY

The goal of this research is to propose a sequential ANN method that uses hidden dense layer modifications to detect intruders in WSNs utilizing SAE and the sequential ANN approach. The primary purpose of SAE with sequential ANN is to initialize its parameter in the context of training data as well as correlate it with backpropagation functions. Once the SAE with sequential ANN has performed as pre-training model has assist ANN with sequential pattern for generating fine-tuned ANN model to detect attack type. The sequence pattern assist in identify the pre-trained model through SAE algorithm and detect the attack type precisely. The WSN-DS dataset is collected from open source kaggle in which csv file shown with few transaction is illustrated in figure 1.

| id | Time | Is_CH | who CH | Dist_To_CH | ADV_S | ADV_R | JOIN_S | JOIN_R | SCH_S | SCH_R | Rank | DATA_S | DATA_R | Data_Sent_To_BS | dist_CH_To_BS | send_code | Expaned Energy | Attack type |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 101000 | 50 | 1 | 101000 | 0 | 1 | 0 | 0 | 25 | 1 | 0 | 0 | 0 | 1200 | 48 | 130.08535 | 0 | 2.4694 | Normal |
| 101001 | 50 | 0 | 101044 | 75.32345 | 0 | 4 | 1 | 0 | 0 | 1 | 2 | 38 | 0 | 0 | 0 | 4 | 0.06957 | Normal |
| 101002 | 50 | 0 | 101010 | 46.95453 | 0 | 4 | 1 | 0 | 0 | 1 | 19 | 41 | 0 | 0 | 0 | 3 | 0.06898 | Normal |
| 101003 | 50 | 0 | 101044 | 64.85231 | 0 | 4 | 1 | 0 | 0 | 1 | 16 | 38 | 0 | 0 | 0 | 4 | 0.06673 | Normal |
| 101004 | 50 | 0 | 101010 | 4.83341 | 0 | 4 | 1 | 0 | 0 | 1 | 25 | 41 | 0 | 0 | 0 | 3 | 0.06534 | Normal |
| 101005 | 50 | 0 | 101010 | 31.91198 | 0 | 4 | 1 | 0 | 0 | 1 | 18 | 41 | 0 | 0 | 0 | 3 | 0.06717 | Normal |
| 101006 | 50 | 0 | 101044 | 24.34167 | 0 | 4 | 1 | 0 | 0 | 1 | 5 | 38 | 0 | 0 | 0 | 4 | 0.06214 | Normal |
| 101007 | 50 | 0 | 101010 | 26.75033 | 0 | 4 | 1 | 0 | 0 | 1 | 21 | 41 | 0 | 0 | 0 | 3 | 0.06662 | Normal |
| 101008 | 50 | 0 | 101044 | 63.66485 | 0 | 4 | 1 | 0 | 0 | 1 | 17 | 38 | 0 | 0 | 0 | 4 | 0.06649 | Normal |
| 101009 | 50 | 0 | 101000 | 32.90217 | 0 | 4 | 1 | 0 | 0 | 1 | 12 | 48 | 0 | 0 | 0 | 1 | 0.07903 | Normal |
| 101010 | 50 | 1 | 101010 | 0 | 1 | 0 | 0 | 30 | 1 | 0 | 0 | 0 | 1230 | 41 | 108.77162 | 0 | 2.3611 | Normal |
| 101011 | 50 | 0 | 101044 | 13.17446 | 0 | 4 | 1 | 0 | 0 | 1 | 10 | 38 | 0 | 0 | 0 | 4 | 0.0613 | Normal |
| 101012 | 50 | 0 | 101044 | 48.16567 | 0 | 4 | 1 | 0 | 0 | 1 | 13 | 38 | 0 | 0 | 0 | 4 | 0.06425 | Normal |
| 101013 | 50 | 0 | 101010 | 66.9102 | 0 | 4 | 1 | 0 | 0 | 1 | 16 | 41 | 0 | 0 | 0 | 3 | 0.07263 | Normal |
| 101014 | 50 | 0 | 101010 | 31.69105 | 0 | 4 | 1 | 0 | 0 | 1 | 17 | 41 | 0 | 0 | 0 | 3 | 0.06716 | Normal |
| 101015 | 50 | 0 | 101010 | 21.52629 | 0 | 4 | 1 | 0 | 0 | 1 | 8 | 41 | 0 | 0 | 0 | 3 | 0.06654 | Normal |
| 101016 | 50 | 0 | 101010 | 74.73928 | 0 | 4 | 1 | 0 | 0 | 1 | 4 | 41 | 0 | 0 | 0 | 3 | 0.0749 | Normal |
| 101017 | 50 | 0 | 101044 | 27.78157 | 0 | 4 | 1 | 0 | 0 | 1 | 29 | 38 | 0 | 0 | 0 | 4 | 0.06139 | Normal |
| 101018 | 50 | 0 | 101010 | 25.5197 | 0 | 4 | 1 | 0 | 0 | 1 | 26 | 41 | 0 | 0 | 0 | 3 | 0.06618 | Normal |
| 101019 | 50 | 0 | 101044 | 41.21473 | 0 | 4 | 1 | 0 | 0 | 1 | 28 | 38 | 0 | 0 | 0 | 4 | 0.0628 | Normal |
| 101020 | 50 | 0 | 101044 | 22.54714 | 0 | 4 | 1 | 0 | 0 | 1 | 15 | 38 | 0 | 0 | 0 | 4 | 0.06147 | Normal |
| 101021 | 50 | 0 | 101000 | 30.32029 | 0 | 4 | 1 | 0 | 0 | 1 | 4 | 48 | 0 | 0 | 0 | 1 | 0.07905 | Normal |
| 101022 | 50 | 0 | 101010 | 33.68188 | 0 | 4 | 1 | 0 | 0 | 1 | 14 | 41 | 0 | 0 | 0 | 3 | 0.06743 | Normal |



**FIGURE 1.** WSN dataset from Kaggle source as an input file

The WSN-DS dataset was obtained and confirmed with the missing information utilizing is.null() function as part of the missing imputation procedure followed by data preprocessing. During data preprocess, normalization has done for better defining the feature's information for accomplishing target as attack_type" in the binary format as "Normal" as 1 and "Intruder" as 0 is determined through labelEncoder(). The preprocessing assist to convert all independent variable into integer or float data type in order to maintain continuous variable and the ID number variable is made dropped. The architecture of SAE with ANN sequence pattern model has shown in figure 2.

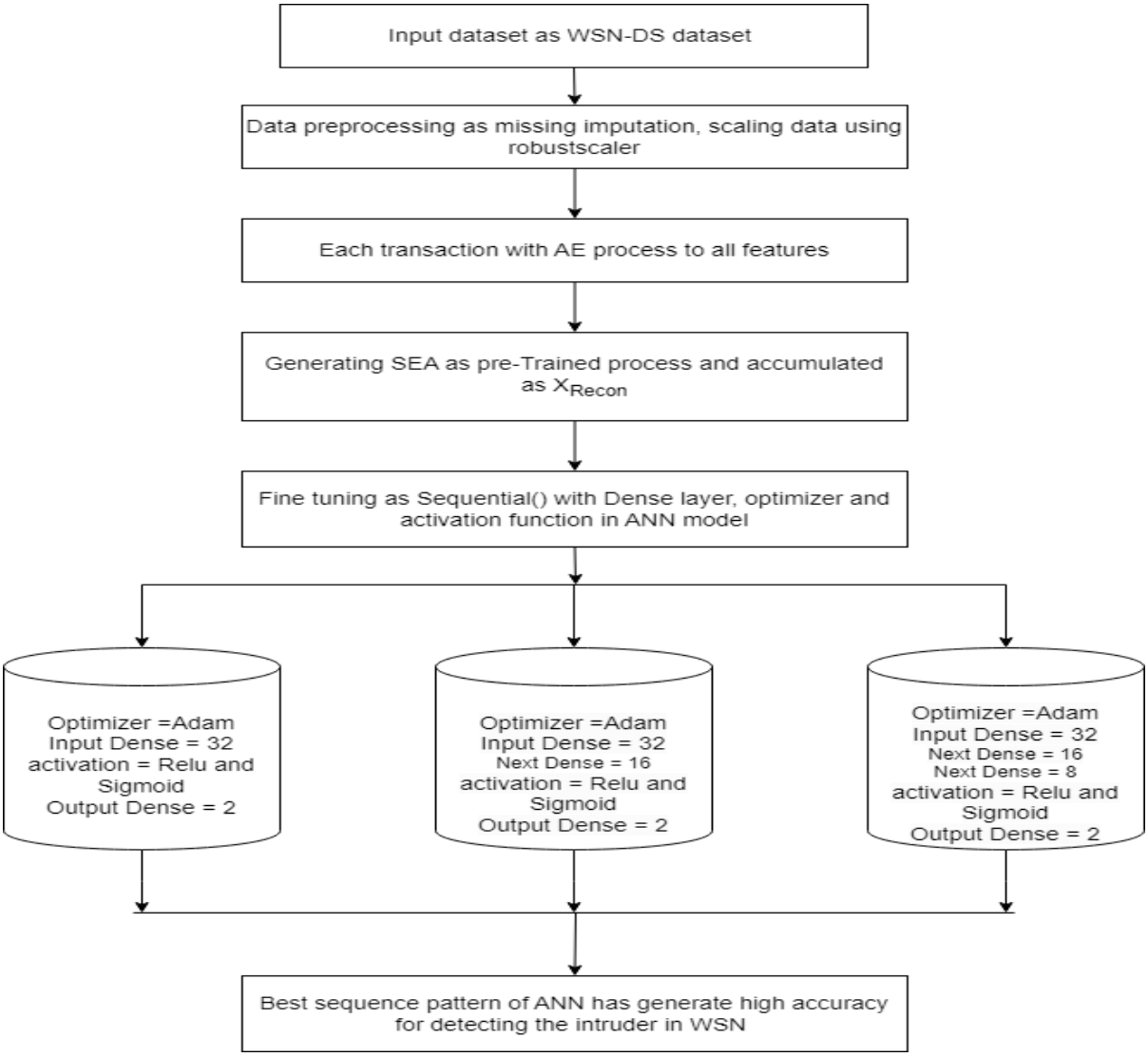



**FIGURE 2.** Architecture of SAE with ANN sequence pattern for IDS

In this study, 16 variables are utilized as an input data preprocessed using missing imputation, normalization process, and scaling all variable units by Robustscaler. The dataset is divided into 70% train dataset and 30% test dataset. The SAE algorithm has play an essential role in securing the transmission of data through encoder and decoder concept using hash key as well as reconstruction of encrypted code into input data which act as a pre-trained process of network layer shown in figure 2.

**Working of Stacked Autoencoder**

One of the unsupervised Neural Network (NN) is Autoencoder (AE) that which help to learn for reducing the difference among input data as well as output data. Encoders and decoders are the two types of AE. The original data gets mapped with encoder code which basically deals with code dimension is lesser than an original data. In the case of decoder, the code has tried to map an original input. The dimensionality reduction is an AE application and consider input as $z \in R^n$, the AE goal is represented as y = z which tried for learning AE function is expressed in equation 1.

$$F_{w,ib}(x) \approx y \tag{1}$$

Where,

W = Weight of the entire neural network

ib = Image bias

The basic reconstruction loss in AE loss function for $L_p$ distance in which Stochastic Gradient Descent (SGD) has been utilized to fine tune the weight and bias in AE module shown in equation 2.

$$\mathbb{L}(w, ib) = min\left\|x - F_{w,\,ib}(x)\right\|_p \quad (2)$$

However, the better results are obtained through Stacked Autoencoder (SAE) that involves various AE in which the output of each AE is assigned to the input of the succeeded AE. The given below steps are basic steps for SAE training.

**Step 1 - Encoder transformation**

The SAE with M number of AE is represented as $m^{th}$ AE's encoder as well as functions of decoder transformation. The encoder transformation function in SAE has been examined using function of encoder transformation for each AE in forward order gets illustrated in equation 3.

$$x_{encoded} = x^m = \alpha^m, \alpha^{m-1}, \ldots, \alpha^2, \alpha^1\,(x) \quad (3)$$

**Step 2 - Decoder transformation**

In the case of SAE decoder transformation function has been evaluated by function of decoder transformation for each AE in reverse order get illustrated in equation 4.

$$x_{Decoded} = x_{reconstrct} = \alpha^1, \alpha^2, \ldots, \alpha^2, \alpha^1(x^w) \quad (4)$$

When one layer is trained, the other layer's parameters get fixed whereas the preceding layer as an output has been utilized as an input for the subsequent layer. Thus, it will continue till the training gets completed. The backpropagation algorithm has been utilized for reducing the reconstruction error once all the layers are trained and all the layer's weights get modified. In order to accomplish high-level features, SAE code is essential for encoding the statistical features. Hence, this research concentrated on SAE code for performing statistical features from network traffic.

Furthermore, this model uses "relu" as an activation signal and was fitted with batch size of 5 as well as 10 epochs using the Adam optimizer. This scenarios SAE-ANN sequence model 1 is the input layer with shape set to 32, while the hidden layer shape is 16 whereas output layer shape is 2. The SAE-ANN sequence model 2 with sequence pattern has the input layer shape is 32, the input layer is 16, the output dense layer is 2, and the activation for the input and output layers as Relu and sigmoid. The SAE-ANN sequence model 3 defined the shape of input layer is 32, the input layer is 16, the succeeding layer is 8, the output dense layer is 2, and the input and output activations are Relu and sigmoid. The best hidden layer was created from an appropriate sequence pattern as well as an improved optimization procedure is proposed to achieve high accuracy in identifying the intrusion as an improved IDS utilizing SAE.

**ANN model working principle**

In order to acquire ANN model with sequence patterns and made by accomplishing possibility with better prediction. Assume, the dataset X with data pairs contain the variables as well as the results ($m^1$, $t^1$), ($m^2$, $t^2$), ….($m^X$, $t^X$) where,

$m^i$ = input value

$t^i$ = target value for i= 1, 2, …, X.

Therefore, this research concentrates on designing NN as F which is ideally specified in equation 5.

$$F(m^i) = t^i \quad (5)$$

Let it allocate for error $\in^i$ normally and n represent the ANN output is formulated in equation 6 and 7.

$$n^i = F(m^i) \quad (6)$$

And

$$t^i = n^i + \in^i \quad (7)$$

Thus, the $n^i$ has represent the parameter regarding weight as well as bias that consider as an optimizer issue. Based on this setting, the ANN in F that minimize the error by considering equation 3.4.

$$E = \frac{1}{N}\sum_{i=1}^{X}\left\|t^i - n^i\right\|^2 (8)$$

Where,

N = Training patterns numbers

If the ANN becomes a two-way categorization N=2. According to the equation, E is a parameter function for F that must be determined in order to find the weight values which minimizes the error by differentiating E. In a single portion of overall method, the investigation focuses is stated in equation 9.

$$\|t - n\|^2 = (t_1 - n_1)^2 + (t_2 - n_2)^2 + \cdots + (t_x - n_x)^2 \quad (9)$$

Thus, the input and output values have been set fixed, but only the output parameter is considered to be determined by weight, which can be distinguished from both sides, as represented in equation 10.

$$\frac{\partial}{\partial w_{ij}}(\|t - n\|^2) = -2(t - n).\frac{\partial n}{\partial W} \quad (10)$$

They are extremely precise as well as validate the fits within the larger context of the NN. The output of NN is defined as $n^i = W_{ij}m^i + b$.

As a result, the output is based on weight as well as differentiation of both sides in accordance with the $W_{ij}$ by chain rule as shown in equation 11.

$$\frac{\partial}{\partial \mathrm{w_{ij}}}(\|\mathrm{t - n}\|^2) = -2(\mathrm{t_i - n_i})\mathrm{m_i} \quad (11)$$

Where,

$m_j$ = $i^{th}$ coordinate position.

The derivative has supplied orientation to the greatest for achieving this lowest point, and then in the contrary direction of the gradient. Furthermore, the derivative itself should be as close to zero as feasible in order to minimize inaccuracy. Thus, the ANN approach is a layer-based network with a number of artificial neurons that typically contain an input layer, a hidden layer, and an output layer.

Relu is a function, and the main advantage of the Relu activation function is that it fails to activate all neurons at the same moment. Additionally, a neuron with a negative value gets reduced to zero or deactivated. As a result, networks grow increasingly sparse while remaining computationally efficient. The gradient values of the graphs on the negative side are zero. It implies that neurons have terminated and cannot be activated in back propagation. For preserving concerns with multi-class, the sigmoid function is utilized that maps the output value from 0 to 1. It performs best in the output layer of classifiers. There are no guidelines for selecting the activation function. Optimization is a method that seeks to reduce network errors. This is critical for improving the model's accuracy. In this study, the Adam method was employed to find an anomaly node by iteration. For each subsequent iteration, these computer methods provide predictions that are subsequently compared with the projected results. The intruder can be described as the discrepancy among the projected as well as actual values. This error message is used for updating both the network's weights and the internal parameters of the methods. The back propagation method corresponds to the SAE-based update mechanism. Adam is an optimizer that combines the benefits of RMSPropas and Adadelta optimizers, and it is considered a good option because it enhances both RMSPropas and Adadelta.

The epoch number visual representation has controls whether the training dataset iteration is transverse. Each epoch gives a training sample capable of updating the internal model's parameters. The epochs begin at 0 and increase incrementally until the epoch number is reached including multiple batches. Hundreds or even thousands of epochs are frequently employed permitting the network to effectively mitigate error. When determining the optimal epoch value, the developer must consider both error and accuracy curves during learning. These graphs are used for determining when a model has acquired knowledge a great deal, poorly, or adequately ready to undergo training. The SAE with a sequential ANN pattern efficiently extracted the multidimensional characteristics among the appropriate variables for categorizing the attacker in the WSN transmission node. The selection of sequence in ANN model is significant for improving the optimum selection of transmission node of the WSN. This may be determined through improving forecast accuracy with the model. Thus, the model performance was increased by improving the best acquired sequence in raising the forecasting IDS accuracy using the SAE approach.

## 4. RESULT AND DISCUSSION

This experiment is carried out with an i5-7400 processor operating at 3.00 GHz as well as a GTX 1050 GPU. The PC features 8.00 GB of RAM and an x64-ased processor. The Python programming language is utilized to train the algorithms in the Keras environment, while Tensorflow is utilized to evaluate the results. The parameters include the hidden layer units, activation function, epoch number, layer count, optimization, and batch size utilized for verifying the models. A critical methodological consideration for each experimental evaluation of SAE with sequential ANN pattern is selecting the sequential search space corresponding to each data transaction. However, this implies the sometimes incorrect assumption that parameters with similar designation operate similarly for IDS in WSN. Figure 6 depicts the epochs of sequence pattern with loss function as binary cross entropy, and the optimum outcomes are reached in the SAE-ANN sequence model 1, that has higher accuracy than other SAE-ANN sequence patterns. Thus, the acquired sequence pattern leads to identify the IDS performance of SAE with ANN sequence pattern in WSN is able to assessed using confusion matrix metrics.

```
model_3 = Sequential()
model_3.add(Dense(32, input_dim=in_dim, activation='relu'))
model_3.add(Dense(16, activation='relu'))
model_3.add(Dense(5, activation='sigmoid'))
# Model Compilation
model_3.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
```

```
model_3.fit(X_train, y_train, validation_data=(X_test, y_test), batch_size=5, epochs=10)

Epoch 1/10
7168/7168 [==============================] - 34s 5ms/step - loss: 0.2604 - accuracy: 0.7285 - val_loss: 0.1748 - val_accuracy: 0.7643
Epoch 2/10
7168/7168 [==============================] - 38s 5ms/step - loss: 0.1657 - accuracy: 0.7945 - val_loss: 0.1540 - val_accuracy: 0.8362
Epoch 3/10
7168/7168 [==============================] - 33s 5ms/step - loss: 0.1284 - accuracy: 0.8682 - val_loss: 0.1117 - val_accuracy: 0.8809
Epoch 4/10
7168/7168 [==============================] - 38s 5ms/step - loss: 0.1055 - accuracy: 0.8879 - val_loss: 0.0993 - val_accuracy: 0.8921
Epoch 5/10
7168/7168 [==============================] - 39s 5ms/step - loss: 0.0901 - accuracy: 0.9173 - val_loss: 0.0798 - val_accuracy: 0.9381
Epoch 6/10
7168/7168 [==============================] - 34s 5ms/step - loss: 0.0695 - accuracy: 0.9511 - val_loss: 0.0597 - val_accuracy: 0.9575
Epoch 7/10
7168/7168 [==============================] - 30s 4ms/step - loss: 0.0536 - accuracy: 0.9649 - val_loss: 0.0558 - val_accuracy: 0.9628
Epoch 8/10
7168/7168 [==============================] - 21s 3ms/step - loss: 0.0482 - accuracy: 0.9664 - val_loss: 0.0473 - val_accuracy: 0.9699
Epoch 9/10
7168/7168 [==============================] - 22s 3ms/step - loss: 0.0453 - accuracy: 0.9685 - val_loss: 0.0476 - val_accuracy: 0.9702
Epoch 10/10
7168/7168 [==============================] - 21s 3ms/step - loss: 0.0473 - accuracy: 0.9688 - val_loss: 0.0443 - val_accuracy: 0.9696
<keras.callbacks.History at 0x195819d3f70>
```

**FIGURE 4**. Epochs for SAE-ANN sequence model 1

**TABLE 1.** Confusion matrix for various SAE-ANN models

| Classification of SAE-ANN model | Confusion matrix class values for IDS in WSN-DS test dataset | | | |
|---|---|---|---|---|
| | True Positive (TP) | True Negative (TN) | False Positive (FP) | False Negative (FN) |
| SAE-ANN sequence model 1 | 56348 | 46983 | 5556 | 3476 |
| SAE-ANN sequence model 2 | 59135 | 49867 | 2073 | 1324 |
| SAE-ANN sequence model 3 | 57388 | 47915 | 4620 | 2476 |

Table 1 illustrates the 30% test sample from the WSN-DS dataset in which the overall test sample is 1,12,399 transaction records are involved. The TP represent the normal in attack type, TN represent the intruder in attack type, FP represent the wrongly predicted the normal type from actual by the respective model and FN represent the wrongly predicted the intruder type from actual by the respective model. This research involves three different sequence pattern of SAE-ANN model.

**TABLE 2**. Confusion matrix metrics for various SAE-ANN model

| **Classification of SAE-ANN model** | **Confusion matrix class values for IDS in WSN-DS test sample** | | | | | |
|---|---|---|---|---|---|---|
| | **Accuracy (%)** | **Precision (%)** | **Recall (%)** | **F1-Score (%)** | **Sensitivity** | **Specificity** |
| SAE-ANN sequence model 1 | 91.96 | 91.03 | 94.19 | 92.58 | 0.942 | 0.894 |
| SAE-ANN sequence model 2 | 96.96 | 96.61 | 97.81 | 97.20 | 0.978 | 0.960 |
| SAE-ANN sequence model 3 | 93.69 | 92.55 | 95.86 | 94.18 | 0.959 | 0.912 |

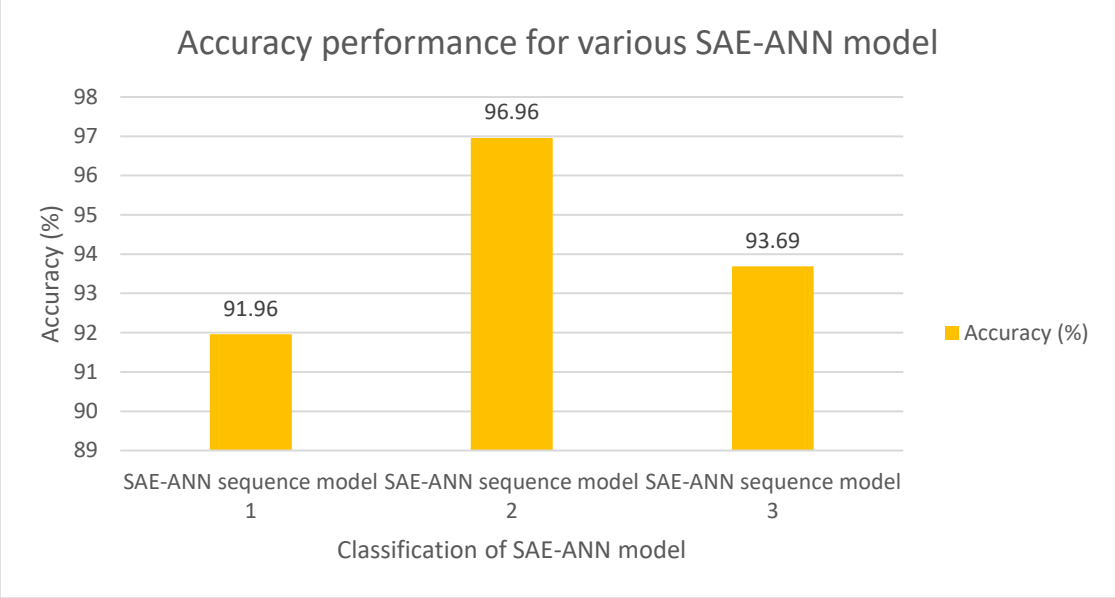



**FIGURE 5**. Accuracy for various ANN models

Figure 5 illustrates the accuracy for three different SAE-ANN sequence models with adam optimizer. SAE-ANN Sequence Model 2 has an accuracy of 96.96%, which is greater than the other two SAE-ANN sequential models. The suggested SAE-ANN sequential pattern model 2 has a better prediction in categorizing IDS from transaction nodes in WSN.

## 5. CONCLUSION

To provide security, unidentified individuals must be prevented from acquiring data or other items, as well as from changing or destroying user data. The classic definition of security includes integrity, availability, and confidentiality. One of the appealing aspects of wireless sensor networks is that they encourage numerous researchers to behave in accordance with various security criteria. To overcome the aforementioned challenges, the SAE-ANN with sequential pattern technique was created, which uses distinct sequential patterns as the classifier in recognizing IDS and determining network shielding from DoS attacks. The pre-trained SAE algorithm plays an important role in this research for securing the transmission data by encryption and decryption through hash key and fine-tuned through various sequence pattern of ANN model. Furthermore, the suggested sequential patterns of ANN model with the optimal sequence patterns improved data interpretation as well as accomplished a high predicted accuracy as 96.96% is greater than other SAE-ANN models. Thus, the evaluation of the suggested SAE-ANN with sequential model produces high recognition in IDS while also sending data safely in WSN.

## REFERENCES


[1]. Wazirali, R.; Ahmad, R.; Al-Amayreh, A.; Al-Madi, M.; Khalifeh, A. Secure Watermarking Schemes and Their Approaches in the IoT Technology: An Overview. Electronics 2021, 10, 1744.

[2]. Hussain, F.; Hussain, R.; Hassan, S.A.; Hossain, E. Machine learning in IoT security: Current solutions and future challenges. IEEE Commun. Surv. Tutor. 2020, 22, 1686–1721. [CrossRef]

[3]. Xu, L.D.; Lu, Y.; Li, L. Embedding Blockchain Technology Into IoT for Security: A Survey. IEEE Internet Things J. 2021, 8, 10452–10473.

[4]. Eljakani, Y.; Boulouz, A.; Ben Salah, M.; El Hachemy, S. Performances prediction in Wireless Sensor Networks: A survey on Deep learning based-approaches. In ITM Web of Conferences; EDP Sciences: Les Ulis, France, 2022; Volume 43, p. 01010

[5]. M. Al-Imran and S. H. Ripon, "Network intrusion detection: an analytical assessment using deep learning and state-of-the-art machine learning models," International Journal of Computational Intelligence Systems, vol. 14, no. 1, p. 200, Dec. 2021, doi: 10.1007/s44196-021-00047-4.

[6]. Khashan, O.A.; Ahmad, R.; Khafajah, N.M. An automated lightweight encryption scheme for secure and energy-efficient communication in wireless sensor networks. Ad. Hoc. Netw. 2021, 115, 102448.

[7]. Wang, M.; Lu, Y.; Qin, J. A dynamic MLP-based DDoS attack detection method using feature selection and feedback. Comput. Secur. 2020, 88, 101645. [CrossRef]

[8]. Kumar, P.M.; Gandhi, U.D. Enhanced DTLS with CoAP-based authentication scheme for the internet of things in healthcare application. J. Supercomput. 2020, 76, 3963–3983.

[9]. M. Masdari, "Energy efficient clustering and congestion control in WSNs with mobile sinks," Wireless Pers. Commun., vol. 111, no. 1, pp. 611–642, Mar. 2020.

[10]. S. El Khediri, "MWLEACH: Low energy adaptive clustering hierarchy approach for WSN," IET Wireless Sensor Syst., vol. 10, no. 3, pp. 126–129, 2020.

[11]. M. Z. Ghawy, G. A. Amran, H. AlSalman, E. Ghaleb, J. Khan, A. A. AL-Bakhrani, A. M. Alziadi, A. Ali, and S. S. Ullah, "An effective wireless sensor network routing protocol based on particle swarm optimization algorithm," Wireless Commun. Mobile Comput., vol. 2022, pp. 1–13, May 2022

[12]. E. Mushtaq, A. Zameer, and A. Khan, "A two-stage stacked ensemble intrusion detection system using five base classifiers and MLP with optimal feature selection," Microprocessors Microsyst., vol. 94, Oct. 2022, Art. no. 104660.

[13]. D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and Gaussian processing for anomaly detection in industrial IoT," IEEE Trans. Ind. Informat., vol. 16, no. 8, pp. 5244–5253, Aug. 2020.

[14]. A. A. Najar and S. M. Naik, "DDoS attack detection using MLP and random forest algorithms," Int. J. Inf. Technol., vol. 14, no. 5, pp. 2317–2327, Aug. 2022.

[15]. Lai, Trinh Thuc, Tuan Phong Tran, Jaehyuk Cho, and Myungsik Yoo. "DoS attack detection using online learning techniques in wireless sensor networks." Alexandria Engineering Journal 85 (2023): 307-319.

[16]. Ahmad, Rami, Raniyah Wazirali, and Tarik Abu-Ain. "Machine learning for wireless sensor networks security: An overview of challenges and issues." Sensors 22, no. 13 (2022): 4730.

[17]. Salmi, Salim, and Lahcen Oughdir. "Performance evaluation of deep learning techniques for DoS attacks detection in wireless sensor network." Journal of Big Data 10, no. 1 (2023): 1-25.

[18]. Gebremariam, Gebrekiros Gebreyesus, J. Panda, and S. Indu. "Secure localization techniques in wireless sensor networks against routing attacks based on hybrid machine learning models." Alexandria Engineering Journal 82 (2023): 82-100.

[19]. Mounica, Mandala, R. Vijayasaraswathi, and R. Vasavi. "RETRACTED: Detecting Sybil Attack In Wireless Sensor Networks Using Machine Learning Algorithms." In IOP Conference Series: Materials Science and Engineering, vol. 1042, no. 1, p. 012029. IOP Publishing, 2021

[20]. Gebremariam, Gebrekiros Gebreyesus, J. Panda, and S. Indu. "Localization and detection of multiple attacks in wireless sensor networks using artificial neural network." Wireless Communications and Mobile Computing 2023 (2023).