Nanotechnology Perceptions
ISSN 1660-6795
www.nano-ntp.com

CrTDL: An Advanced Deep Learning Framework for Intrusion Detection in Wireless Sensor Networks

K. Nitya[1], Dr. S. Venkatesan[2], Dr. B. Nancharaiah[3]

[1]Research Scholar, GIET UNIVERSITY, Gunupur, Odisha, and Assistant professor, Department of ECE, Usha Rama College of Engineering and Technology, India, nitya.kanakamedala@gmail.com
[2]Professor, Department ECE, Department of ECE, GIET UNIVERSITY, Gunupur, Odisha. Vasu.sanjay11@gmail.com
[3]Professor & HOD, Department of ECE, Usha Rama College of Engineering and Technology, Telaprolu 521109, India. nanch_bn@yahoo.com

The Wireless Sensor Network stands out as a highly practical framework for a diverse array of social network interactions. In communication within the WSN network, data transmission distances play a crucial role in facilitating information exchange among network nodes. However, the open and mobile nature of the WSN environment introduces security concerns and energy usage restrictions. Consequently, an effective security system becomes imperative for the WSN network to guarantee secure data transmission between nodes. In a WSN setting, this article suggested - attack security prevention. Three phases are included in the proposed Cryptographic Transductive Deep Learning (CrTDL) scheme. For data encryption and decryption, attributes-based homomorphic cryptography is initially used. Second, the network employs the deep learning approach. Finally, the WSN network uses the Markov process is applied in a transductive deep model to avoid and categorise attacks. A concealed Markov model was employed to assess the produced data in order to detect attacks. The reinforcement learning model is used to compute and process the detected attack. The suggested CrTDL is subjected to simulation analysis using traditional ANN, CNN, and RNN models.

Keywords: Security, WSN, Classifier, Cryptography, Attack, Homomorphic.

1. Introduction

The Internet of Things has gained popularity the past several years, and is now being used in a variety of practical applications. Currently, network privacy and safety are thought to be serious issues. Because of the advancement of data technology, the measure of system safety is dramatically rising in daily life [1]. Many hours have been invested in The advancement of

internet applications and technologies, such as WSN, has considerable effect on networks and computer systems [2]. In its traditional form, WSN comprises interconnected elements where intelligent machines communicate without direct human involvement. In smart WSN, sensors are incorporated for efficient information transfer between nodes, allowing connectivity over the Internet. These intelligent WSN devices find applicability in diverse fields such as transportation, healthcare, and agriculture, making them versatile solutions [3].The parameters of WSN devices are changed according to the application to decrease time and resource usage. WSN may be used information transformation applications using right modification and extension. However, a key element of WSN is space and every WSN network safety concern is found on the internet [4].

The traditional (WSN) encounters security issues arising from constrained capabilities, limited resources, and restricted instructions [5]. As smart devices evolve, the complexity of WSN security challenges grows, making security a prominent concern for network integrity [6]. Detecting attacks in the WSN environment poses a significant challenge, leading to the adoption of contemporary tactics. Addressing this issue requires increasingly effective and active attack detection strategies, given the rising prevalence of network intrusions that heavily rely on massive data volumes [7].

To bolster network security, various security measures have been developed, with machine learning (ML) emerging as a valuable tool for computational models in these scenarios. ML offers a secured intelligence strategy for WSN networks, particularly in bot-net defence methods, network traffic analysis, and intrusion detection. ML leverages its capabilities in smart device management to assess device modifications, features, and performance. It engages in activities essential for classification and regression, contributing to the processing of information generated by machines [8]. Moreover, ML provides several security options for WSN networks, including attack processing and prevention.

In the realm of security, ML plays a pivotal role in identifying assaults with diverse manifestations. While various ML-based methods have been developed for addressing security concerns in applications, their effectiveness within the framework of WSN remains a challenge [10].

When detecting attacks, ML is used for recognition based on two analytical methodologies, such as the misused methodology and the signature or anomaly technique. The signature allocated approach includes the identification of assaults based on signatures, which are characteristics of traffic. Attacks are detected when a technology is misused by identifying the incorrect signal that was generated. With the help of botnet instructions, compromised machines are found using the signature-based approach. The drawback of utilizing signature-based solutions is rooted in the absence of well-defined standards for consistently updating analyses to identify unidentified threats [11].An anomaly assigned system is used for attack recognition to get over the drawbacks of -attack identification. The performance for identification must be enhanced, and anonymous attacks must be handled better. Due to the existence of strange and irregular activity, fake signal rates (FSRs) are noticeably high when

using anomaly detection methods. So, in essence, Anomaly and signature-based techniques are examples of heterogeneous approaches that have been created to reduce false positive (FP) in the detection of assaults [12].

Conversely though, Deep Learning (DL), which has higher accuracy and is able to compete with conventional machine learning, has been receiving attention to be able to withstand attack concerns in the WSN environment. Similar to ML, DL serves as a variety of applications like classification, regression, and so on [13]. In fact, ML using artificial neural networks (ANN) is known as deep learning (DL) in conjunction with it to interpret information. For greater processing capacity using DL architecture, the research community created ANN throughout the past century. This DL architecture combines several advancements for increased computational power based on various layers with various neurons that may effectively be employed in an extensive range of applications. With new threats to WSN security, DL has experienced substantial expansion and improvement. So as to deal with security in WSN, the DL procedure is successfully utilizes the system for intrusion detection (IDS) to identify threats [14].

Two popular algorithms are Support Vector Machine (SVM) and Decision Tree (DT) recent classifiers used in DL-based attack classification methods. Rule-based Inside the IDS, ML is utilized to enhance security in WSN while minimising human engagement. (Zhao, Set al., 2020) [15].Additionally, with the addition of DL to IDS, labor-intensive techniques that minimise error are needed for data production. Accurate data classification is carried out for traffic monitoring and attack identification thanks to improvements in DL within IDS.

In this context, ensuring the desired throughput and minimizing end-to-end delay becomes crucial to enhance WSN security by facilitating effective information transmission between nodes. WSN systems are susceptible to various security threats, encompassing but not limited to wormhole attacks, poisoning, packet replication, and various forms of assaults such as denial of service (DoS), as previously emphasized [16]. The occurrence of these attacks has a negative impact on data transfer, resulting in outcomes such as packet drops or redirection to nearby nodes. Malicious nodes cooperate with packet dropping during collaborative assaults to avoid being detected. The cryptographic method need to be utilized for secure data transmission in WSN in order to achieve the appropriate level of security. It is necessary to have a similar key distribution across the network. In the WSN setting, key distribution is a difficult operation. Key management often involves key setup, key distribution, and key reversals. Additionally, the difficulty of load balancing for handling public keys is faced by this cryptographic system [17].

Hence, this research concentrated on the security attack prevention environment for WSN networks. To enhance security in the WSN environment, this research uses DL based cryptographic process for IDS. The proposed Cryptographic Transductive Deep Learning (CrTDL) model comprises of attributes based homomorphic cryptographic process. Based on developed CrTDL cryptographic keys data were encrypted and decrypted with the update in IDS along with reinforcement. After the cryptographic procedure is finished, a modified strategy is offered to get around the drawbacks of the traditional DL model. For attack detection and classification, the suggested. The DL model employs a transductive approach with a hidden Markov process. Transductive learning is built into the CrTDL paradigm for calculation of data using encryption and decryption. Additionally, the proposed CrTDL evaluates assaults in the WSN environment using the CICIDS dataset. A concealed Markov model was employed to assess the produced data in order to detect attacks. The reinforcement learning model is employed to calculate and process the detected attack. With traditional ANN

and CNN models, simulation analysis of the suggested CrTDL is carried out.

The structure of this essay here it is Section II provided the literature synopsis on attacks-based DL for WSN applications. In section III system model for attacks, computation is presented. In section IV cryptographic process involved in the proposed CrTDL for data transmission is presented within section, V presented the overall process involved in the proposed CrTDL. In section, VI discussed simulation results achieved for the proposed CrTDL is evaluated. The general Synopsis of the suggested CrTDL is as cited in the section VII.

2. Related Works

This section compiles the previous studies conducted on WSN for attack classification and prevention. Gleaning insights from a literature review employing ML analysis, algorithms are being explored for efficient intrusion detection. ML is utilized in WSN to bolster security and implement attack categorization methods, where both supervised and unsupervised techniques can be applied for attack classification. Commonly employed ML techniques, such as ANN, Random Forest (RF), and k-means encoder methods, are instrumental in identifying attacks within networks. However, challenges are encountered by researchers when employing unsupervised methods for attack categorization.

In [18] created an autoencoder for the datasets to detect attacks. K-means clustering appears in the built-in autoencoder for attack detection and classification. Considering this, [19] strategy for detecting unsupervised attacks was presented. The proposed unsupervised method focused on using neural networks (NN) or auto-encoders to detect attacks. The unsupervised neural network plays a significant role in distinguishing between abnormal and typical patterns. Likewise, [20] proposed a concept for quantifying anomalies within a network using a distinctive identification mechanism. This described concept utilizes auto-encoders to assess traffic patterns and identify unusual behaviour in the network. Additionally, the author claimed that the large number of abnormalities present and assaults causes problems for autonomous machine learning. Moreover, it is difficult to understand the network unsupervised techniques' anomalies and uneven success rates. Therefore, the most trustworthy and effective supervised approaches must be devised to get around the drawbacks of an unsupervised method. Furthermore, As per the author, Strategies for supervised machine learning are helpful for identifying assaults by including information that is based on labels and classification.

The primary focus of [21] centered on the detection of non-Tor traffic, utilizing Support Vector Machine (SVM) and ANN techniques. The analysis involved the UNB CIC (Canadian Institute for Cyber security) dataset through various ML methods. Furthermore, [22] examined and elucidated the methodologies of RF for gathering information on network traffic and security.

In a different approach, [23] proposed a metadata-based ML strategy for enhancing WSN security by detecting and classifying attacks. This approach integrates a self-normalized neural network into the ML framework to analyze network assaults. The evaluation of the metadata-based model constructed for the network's performance is conducted within the context of WSN, and it is compared with the Feed forward Neural Network. The outcomes demonstrate that the suggested model surpasses the performance of the conventional feed forward network.

In a novel architectural development, [24] introduced the Deep-Coin architecture format for WSN applications, aiming to elevate security in the WSN environment. The Deep- Coin architecture integrates a deep learning model with a block chain energy infrastructure. The model's efficacy is evaluated utilizing a dataset from the Bot-WSN network, and the author argues that the performance framework of Deep-Coin is significant, as indicated by simulation studies.

To enhance anomaly detection, [25] assesses existing research on the discovery of attacks. The developed model incorporates Recurrent Neural Network (RNN) learning to identify the source of anomalies, aiming for efficient predictive performance characteristics. The suggested RNN learning combines various techniques to forecast attacks on the WSN. Furthermore, [26] demonstrates the identification of attacks in WSN environments using both single and RNN learning methods on the UNSW - NB 15 (network intrusion) dataset. The suggested RNN learning classifier's performance is calculated using accuracy, which is 99%. The proposed model, according to the author, is ineffective in a dynamic attack environment. An RNN learning approach was proposed by Chakraborty, [27] to enhance attack detection performance in an unbalanced dataset. The network stacked autoencoder (SAE) accustomed to extract the deep features. The Model of probabilistic neural networks receives the data retrieved from the SAE as input for the detection of single and multiple outliers. When auto encoders are used, network performance for identifying attacks is stable and improved. Only a little amount of study is done for anomaly identification in unbalanced data processing. For example, [28] created a synthetic neural network (SNN) for identifying attacks in WSN dataset with inequalities. The Industrial WSN dataset is processed employing the Synthetic Minority Over- Sampling Technique (SMOTE) to enhance the assessment of network assaults. In the context of an IDS, the recommended use of SMOTE is integrated with a machine learning approach. Although a number of academics have developed techniques for anomaly identification based on machine learning, each one has limits in some areas. Deep learning has recently proven to have advantages over traditional machine learning. Therefore, a deep autoencoder approach for anomaly detection with compromised network traffic in WSN was built by (Meidan, Y et al., 2018) [29]. Nine commercial WSN devices used by botnets like Mirai and BASHLITE are taken into consideration to validate the network's performance even this approach is inadequate. When it comes to categorising and preventing attacks in various network contexts.[30] Presented an attack detection environment in (KoronWSNis, N et al., 2019) [30] considering multiple networks traffic in WSN. The study is completed utilizing the BoT-WSN dataset as a basis for attack detection. In an effort to assess the dataset in the WSN environment, experimentation and diversity are computed with an efficient attack detection technique. The gathered Bot - WSN dataset is computed for and validated for several datasets. For attack categorization and detection, the performance relies on the application of statistical and machine learning approaches. A WSN network combined with a networked, self-learning, autonomous system for anomaly detection was presented in [31]. The performance is based on the attack detection and classification capabilities of the Gated Recurrent Units (GRU). Without needing any support from humans, the suggested model builds the network for the defining of network profiles for labelled data. According to the suggested model's performance, the attacks are successfully detected by the DWSN in 257 milliseconds. Additionally, the created deep learning model offers a 95.6% average Rate of True Positives (TPR) [32] created a method for detecting attacks for WSN using a deep learning algorithm in

a WSN environment. The analysis is based on taking into account an open-source dataset that classifies network threats. With a random forest classifier, the suggested model's neural network performance displays an accuracy of about 98.2-99.4%. Additionally, A deep learning framework that operates without supervision to identify assaults in the WSN One suggestion was a network by [33]. The suggested paradigm focused on the network's detection of harmful activities. The suggested model employs D-PACK autoencoder and CNN, or Convolutional Neural Network, to find the source of harmful internet activity. The network simulation research showed that, with a minimal incidence of false positives of 1%, attack detection accuracy within the network can reach 100%. Through analysis of the current literature, it is discovered that DL performs superior than customary computer learning techniques with an efficient classifier.

Drawing from a review of ML techniques and literature, encoders are utilized in lieu of models for intrusion detection. The assessment of approaches based on ML and soft computing for parameter estimation is carried out, considering factors like accuracy, false positive rate, and computation time. Despite the widespread use of ML based approaches by numerous researchers, certain limitations have been identified. Specifically, effective profile creation is essential for WSN failure detection to identify vulnerabilities and detect anomalies. Therefore, appropriate methods for signature set creation are necessary to enhance performance.

1. ML cannot generate the wide variety of network traffic that the online programme does on the network with the limited traffic instances.

2. The current techniques for identifying malicious traffic have a higher false-positive rate while having a lower rate of IDS anomaly detection.

3. It is challenging to manage network traffic properly since the increase in threat environments.

Consequently, it is imperative that the right technique to improve IDS attack detection. The current method was unable to give a sufficient profile for the computation of network assaults. Thus, this study's objective is to create the cryptography-based attack detection technique CrTDL. Participating effectively in both data encryption and decryption transmission in WSN is the proposed technique. Additionally, the newly created CrTDL improves attack detection while reducing calculation time and false positives.

3. WSN CrTDL Security scheme

The focus of this research was on improving security in WSN setups with attack detection scenarios. Deep learning is used in the proposed security method, Cryptographic Transductive Deep Learning (CrTDL), to identify and categorise attacks on WSN networks. The suggested CrTDL's general architecture is as shown in Figure 1.

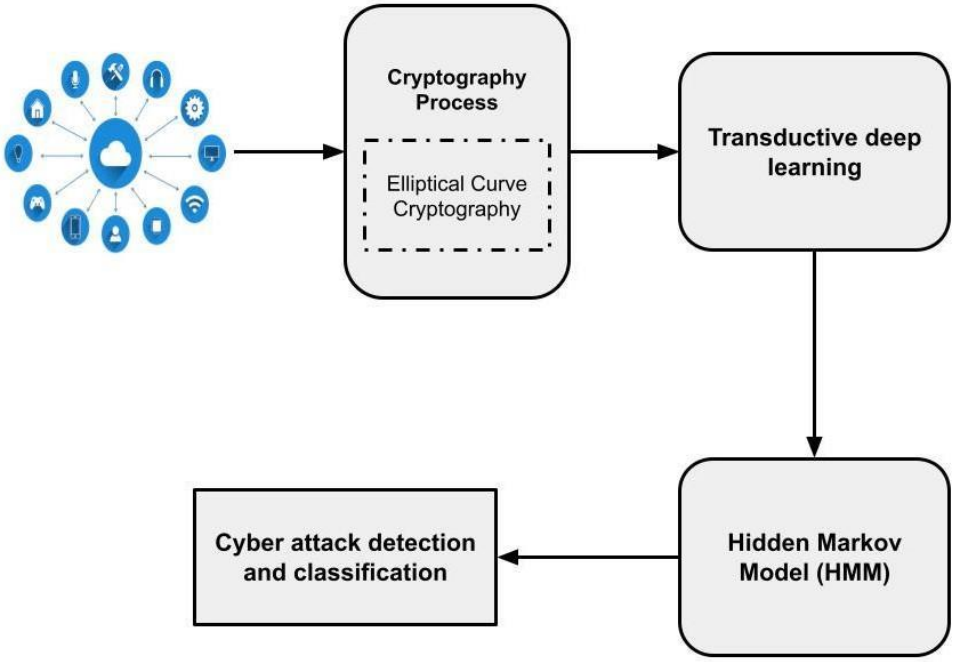


Figure 1: WSN security scheme

The steps are provided as follows, with the general direction of the generated CrTDL shown in figure 1 above:

Step 1: The WSN is initially set up to develop the process of cryptography. Step 2: Using ECC, create keys ensuring WSN security

Step 3: Implementing security using network analysis to thwart assaults

Step 4: Identification and categorization of WSN attack vectors.

Step 5: Applying HMM for Cyber attack prevention in the WSN environment.

Deep learning is used in the built-in security system, which controls how the WSN functions as a whole. The primary allocation or assignment for the nodes in the WSN is carried out using the deep learning building design. The list of the following is the major elements of the CrTDL scheme:

1. Confirmation: This involved determining the reliability of the sources of data that needed to be verified.

2. Secrets and Privacy - The system-protected data needs to be handled or used by specific devices or people.

3. Integrity - Unauthorised users are identified as those who modified or corrupted the material.

4. Availability - The network's ability to provide the essential services.

In light of these factors deep learning must carry out flexible, global, optimized, and comprehensive program operations. However, it is not feasible to overcome the difficulties because of the restricted availability of resources and connectivity suitable strategic planning. Furthermore, deep learning is regarded as a novel networking concept that uses a conventional hierarchical, structured distribution method for the independent data plane and control plane functions. A critical component of deep learning is the transfer of network security administrators via a centralized controller, ensuring ongoing data security across the network. Similar to this, the proposed CrTDL scheme aimed to increase WSN security by utilizing a homomorphic encryption technique in conjunction with Elliptical Curve Cryptography (ECC). The essential procedures in the CrTDL system are enumerated here: The setup stage

1. Phase of key generation
2. Security
3. Decryption
4. Phase five of SDN decryption.

3.1 CrTDL cryptographic security method

The typical homomorphic cryptography procedure and ECC are combined within the suggested plan. Listed below are the traditional steps within the homomorphic framework [20 & 23]:

$I = \{setup, Encrypt, decrypt, SDN\ decrypt\}$

Setup phase

Let p and q be a pair of prime numbers, where k is the bit's size. or 2(k-1).

The sequence of randomly generated numbers is given as g-p g(p-1) mod p2, and The randomly generated numbers are provided as N= p2 q gZ-n. Hashing is completed for advantages. M's integer value using the formula H: 0,1*Z_(2(k-1)) G:Z_n0,1M* 0,1(k-1). Public keys are utilized by produced using the formula PK=(N,g,H,G).

Encryption

The PK (public key) is distributed across nodes within the WSN as sk0,1,.,2(k-1)-1 keys are calculated as pk=gsk modN. Then, sk and pk are employed by indicate nodes within the WSN's public key.

Decryption

According to the secret key sk, the nodes' encrypted information in the WSN network is decrypted by computing mr=B+G(Ask mod N) and then decoding the text encrypted as(A,B).

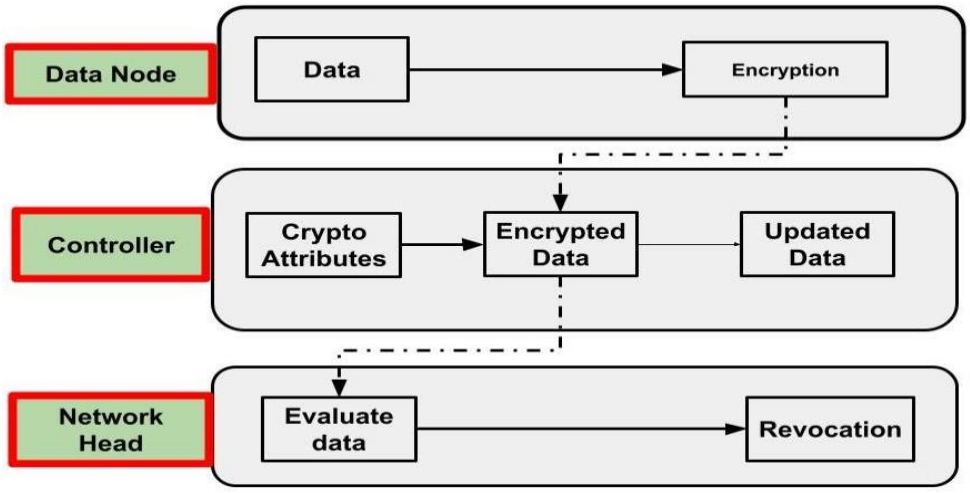


Figure 2: Steps in Cryptographic Process

The Software Defined Network (SDN) architecture handles the encryption function.

Here are the actions in the cryptography process:

Initially, the data node updates and encrypts the data across the WSN head

Step 2: The SDN/controller updates the encrypted data after evaluating the data's properties.

Step 3: In the end, the Cluster Head (CH) evaluates the encrypted information and designates it for revocation.

In figure 2 illustrated the proposed CrTDL combines deep learning and WSN architecture with homomorphic process and ECC integration. As shown in figure 2, the centralized deep learning controller makes up the WSN architecture. In this case, the formation of trust between nodes formats the stable network in which deep learning is serving as a server while maintaining a steady route. A deep learning controller is created for the WSN environment by utilizing the ECC in the homomorphic cryptography process. This used an ECC-integrated, traditional homomorphic cryptography technique. The following lists the standard fundamental steps in the homomorphic cryptography technique.

Input: The coordinates contain ECC points.

Output: H and T generated keys

These images were captured utilizing the WSN SDN.determines the coefficients that will be used the description of the elliptical curve y3=x3+ax+b over the limited domain Fn. which describes the rational numbers or integers as a and b.Chooses an elliptical curve point G=(x_0,y_0) with a big order r, which results in Gr=E (where G is the binary form sequence of points for the ciphertext and r is the positive integer).

1. Chooses a number between m and m<r.
2. Computes $B = Gmmod\ q$

The (G, B) that composed of the public keys is visible by everyone within a clear channel, whereas the (m) that makes up Using the private key of the ECC, represent it.

The following is what SDN performs for any purpose to encrypt messages routed to a specific node in the WSN:

1. Selects at random the unidentified integer k, where k is the secret key produced by the SDN.

2. Determines T=Bk w mod n and H=Gk mod n, where T is established, and H is the secret key that the sender generated. at the together with the message sequence at the recipient end.3. Creates the (H, T) ciphertext.

4. Attribute values have been assigned by SDN to certain nodes A1, A2,...An.

5. Calculates A_n (H,T) = A_n (H,T)

4. Sends (A_H,A_T) to data-bearing nodes.

The centralised the deep learning model is capable of to the encrypted data, but every node that wishes to execute network assessments must first seek authorization from the network, which is done during the revocation phase. A transductive learning model utilizes in the phase of network revocation to classify each node as either a perpetrator or a regular user using deep learning techniques.

Decryption

Input: Encryption points AnH and AnT

Output: Decryption of file W (file)

To decrypt the ciphertext (AnH, An)Alice needs to do the following:

1. Computes $R = AnHmmod\ n$; R – Encrypted message that is transferred to the receiver node

2. Recovers $w = \underline{AnT}\ mod\ n$; w – Original data transmitted from the sender

R

3.2 Attack detection with Concealed Markov Model

In the WSN environment dynamic characteristics are estimated with the probabilistic Bayesian network model with the time-series analysis. The production of hidden sequences from various sequences makes up the Hidden Markov Chain. As stated, the HMM comprises of the vast range of applications in the detection system. The relationship between the sequences are evaluated with the temporal estimation with incorporation of the multiple attack with assessment of the network characteristics to increases the network attack detection rate. The HMM model along with the use of the malicious attack utilizing the calculation of the transfer probabilities to assess the offensive as presented in fig 3.

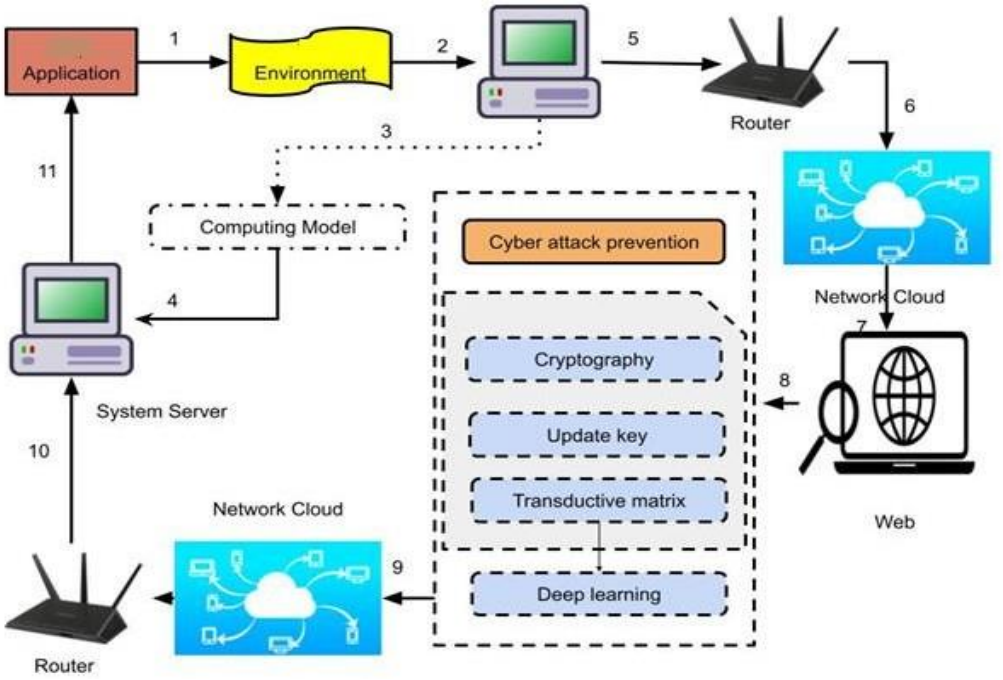


Figure 3: CrTDL Architecture

In the developed model the state is marked as s for the time instances t for the node's numerical value is defined as n_i. The developed model is presented as in equation (1)

$$\lambda = (B, H, C, J, \pi) \qquad (1)$$

As stated in the figure 3 the sequence of observation are measured with the observation layer epresented as $N = (n_1, n_2, \ldots\ldots, n_T)$. With the certain instances of time the attacks are represented as $n_t = v_i$. The attack intents the hidden layer is represented as $H(i_1, i_2, \ldots, i_T)$. Also, the time instances are defined as $i_t = S_i$ the sequences conditional probability is represented as intents denoted as $P(al_i|S_i)$. Similarly, the conditional probability of the attack intents are denoted as S as $P(S_j S_i|)$. Generally, with the HMM model the different probabilities are considered those are probability of calculation, estimation of parameters and decoding value. The incorporation of the transductive model for attack detection with random length $L = (n_1, n_2, \ldots\ldots, n_M)$. The positive integer value is represented as $N = (al_1, al_2, \ldots\ldots, al_n)$. The intents of attacks in hidden layer is defined as H . The model probabilities is represented as in equation (2)

$$P(N \lambda) = \sum_{I} P(N, I \lambda) = \sum P(N I, \lambda) P(I \lambda) \quad (2)$$

The intents of the attacks are computed based on the forward and backward intents. The state in the hidden is denoted as ti with the forward probability of □t (i) denoted as $n_1, n_2, \ldots\ldots\ldots, n_t$. In the same hidden states the time is represented as t+1 in the observed state $n_{t+1}, n_{t+2}, .,$ n_T The probability in backward state $\beta_t(i)$ is stated as in equation (3) and (4):

$$\alpha_t(i) = P(n_1, n_2 \ldots\ldots, n_t, i = q_i \lambda) \quad (3)$$

$$\beta_t(i) = P(n_{t+1}, n_{t+2} \ldots\ldots, n_T \mid i = q_i, \lambda) \quad (4)$$

The state probability is represented as in equation (5)

$$\gamma_t(i) = P(i_t = q_i \mid N \lambda) = \frac{P(i_t = q_i, N \lambda)}{P(N \lambda)} \quad (5)$$

The backward probability is represented as in equation (6),

$$P(i = q_i, N \lambda) = \alpha_t(i) \beta_t(i) \quad (6)$$

Which provides,

$$\gamma_t(i) = \frac{\alpha_t(i) \beta_t(i)}{\sum_{j=1}^{N} \alpha_t(j) \beta_t(j)} \quad (7)$$

In the defined model λ , the observation sequence is stated as N , the probability of state qi and q j with time t and t +1 is given in equation (8) - (10)

$$\delta_t(i,j) = P(i_t = q_i, i_{t+1} = q_j \mid N \lambda) = \frac{P(i_t = q_i, i_{t+1} = q_j N \mid \lambda)}{P(N \mid \lambda)} \quad (8)$$

$$P(i_t = q_i, i_{t+1} = q_j N \mid \lambda) = \alpha_t(i) a_{ij} b_j(N_{t+1}) \beta_{t+1}(j) \quad (9)$$

$$\delta_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(N_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(N_{t+1}) \beta_{t+1}(j)} \quad (10)$$

(11), which presents the condensed equation,

$$P(N|\lambda) = \sum_{i=1}^{t}\sum_{j=1}^{t} \alpha_{t}(i)a_{ij}b_{j}(N_{t+1})\beta_{t+1}(j) \qquad (11)$$

The trained model designates the HMM as $\lambda = (C, J, \pi)$

4. Dataset Description

The proposed CrTDL adopts deep learning for -attack prevention in the WSN network. The dataset considered for the classification of attacks is the CICIDS dataset. However, in a deep learning environment processing the dataset are critical due to high computational time. To withstand those limitations developed CNN regression classifier involved in the estimation of attacks with improved accuracy. However, for effective data processing, the best subset of characteristics must be processed, and extraneous features must be removed without affecting the accuracy or cost of computation. The CICIDS dataset is Employed by created CrTDL to instruct and assess the attack. Table 1 outlines the attributes of both the testing and training datasets.

Table 1: Explanation of CICIDS attributes

Title of the CICIDS dataset's attribute	**Descriptions of the CICIDS dataset's attributes**
Src IP	The IP address of the system
Src Port	Location of the source port
Dest IP	Location of the destination network
Dest Port	destination port's address,
Proto	Layer of Transport Protocol
Date first seen	length of the entire flow
Duration	Time of the whole flow
Bytes	Quantity of messages sent
Packets	Quantity of dispatched packets
Flags	flags TCP concatenation
Class	Identifying the class label as normal, hostile, or suspicious
Attack Type	Attack classification
AttackID	Determining the attack id
Attack Description	Description of the identified attack

To train and test attack sequences in the WSN , attacks are produced using the attributes present in the provided dataset .The division of the dataset into training and testing versions is illustrated in Table 2

Table 2: Distribution of Data Sets

	Distribution of Data Sets	**Training Count**	**Testing Count**
	Routine activity	67,35	9,710
	Unusual occurrence	58,63	12,834
	DoS	45,93	7,458
	Investigative assault	11,66	2,422

Attack Distribution	Elevation of privileges attack	53	67
	Escalation from root to local	995	2,887

Attacks are simulated within the WSN based on the distribution of the training and testing datasets.70% of the data used for training is employed to achieve the required accuracy, and the remaining 30% serves to evaluate the network's performance. Figure 4 presents an overview of the procedure for assault identifying and categorizing.

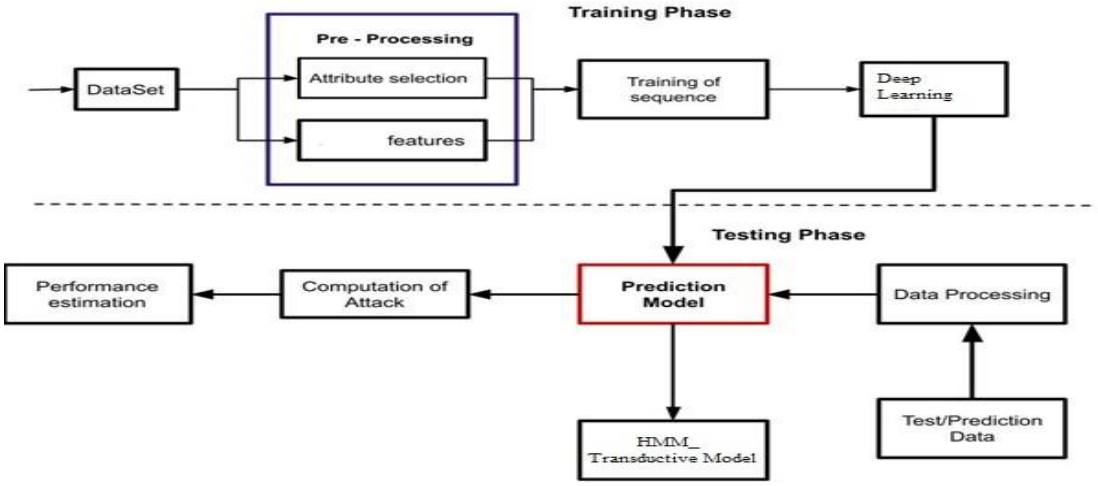


Figure 4: Attack detection and classification process

5. Setting for Simulation

Attack categorization and detection are done using Simulator 3 to evaluate the work of the planned CrTDL Network. DL is employed by the planned CrTDL scheme to centrally manage the WSN environment. Variations in node mobility are employed by estimate the simulation. Attack detection uses the performance metrics listed below and classification:

1. Delay: The length of time required by the network to data from the source should be sent to destination. It is measured in milliseconds (ms)

2. Jitter – Jitter provides the postponement of time for data packets established over the connection in the network. This also estimate the network congestion and change in routes.

3. Total Time Delay – This computed as time taken by the packet to withstand hop count and interference path in the network.

4. Package Handling Rate (PDR) – It is proportion of all packages delivered to the total packet sent from the origin to the destination in the network

5. Throughput – It's calculated as the total data transferred with different scenario for distinct period of time.

A general simulation environment for the suggested classifier for CrTDL cryptography is described in table 3.

Table 3: Environment for Simulation

Variable input	**Configurations and preferences**
Node Density	10,20, 30, 40 & 50
Movement speed (ms/sec)	4 – 10
Models of Movement speed	Stochastic Waypoint
Flow of data	CBR
Routing	AODV
Tx power	1.5MW
Rx Power	1.0MW
Channel type	Wireless Channel
Data Size	512bytes
Physical layer standard	IEEE 802.11

5.1 Simulation Outcomes

The CRTDL scheme's performance is evaluated under varying degrees of node mobility. Node mobility is varied at intervals of 4 ms, 6 ms, and 10ms. Additionally, the number of nodes is adjusted to 10, 20, 30, 40, and 50 to reflect changes in mobility. Table 4 presents the network's performance under various node mobility conditions.

Table 4: Assessment of WSN Performance

Node Mobility = 4ms				
No.of Nodes	**Jitter (ms)**	**PDR %**	**E2E (ms)**	**Throughput %**
10	1.24	0.97	1.04	99
20	1.67	0.96	1.27	97
30	1.93	0.98	1.86	98
40	2.34	0.98	2.07	98
50	2.57	0.96	2.45	98
Node Mobility = 6ms				
No.of Nodes	**Jitter (ms)**	**PDR %**	**E2E (ms)**	**Throughput %**
10	1.3	0.98	1.68	98
20	1.67	0.97	1.94	97
30	1.73	0.98	2.37	98
40	2.93	0.97	2.19	98
50	1.94	0.96	2.09	97
Node Mobility = 10ms				
No.of Nodes	**Jitter (ms)**	**PDR %**	**E2E (ms)**	**Throughput %**
10	1.05	0.98	1.18	99
20	1.36	0.98	1.36	98
30	1.69	0.97	1.84	98
40	1.86	0.96	2.18	97
50	1.96	0.97	2.38	98

Node mobility is defined in table 4 as the ability of a node to move between two points in a network. 10, 20, 30, 40, and 50 nodes with parameter estimate are the quantity of nodes included in consideration for the analysis. The average amount of time a packet waits in the

queue before being sent from one sender to another is known as the delay. Give the amount of time the packet took to travel from one place to another if there was an E2E delay. The ratio of packets to bytes received at the source is known as throughput. PDR specifies how many of the total packets were really delivered. The proposed CrTDL uses SDN to accomplish data encryption and decryption. For different node mobility, the proposed CrTDL system demonstrates a large throughput. However, as the number of nodes increases, the CRTDL's latency, jitter, and E2E all drastically worsen. The simulated WSN in the NS3 environment is shown in figure 5.

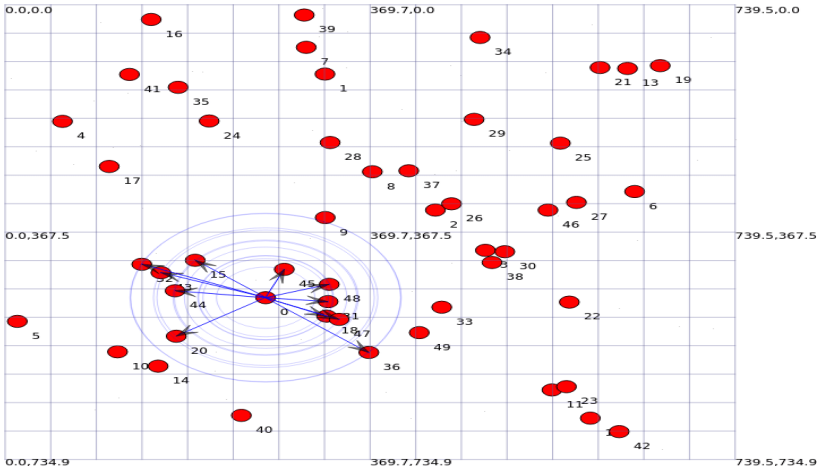
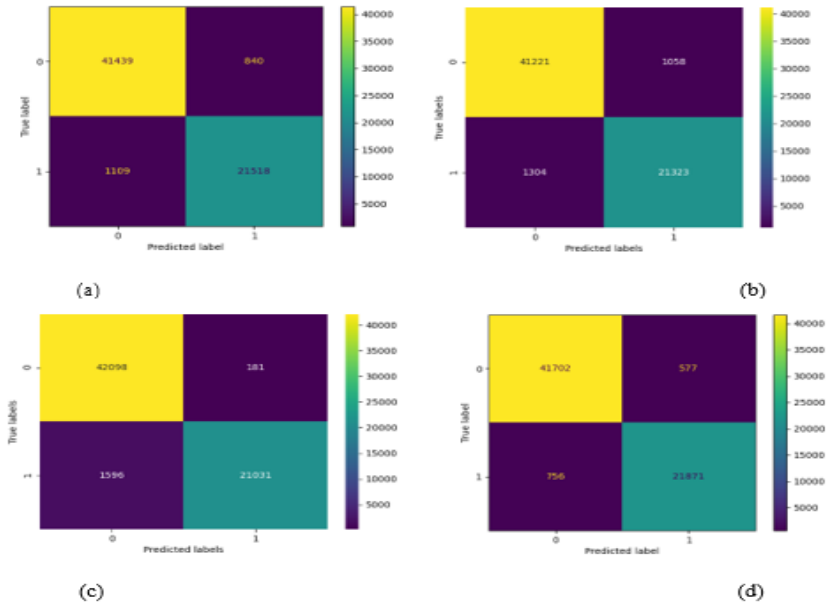


Figure 5: WSN topology

The suggested CrTDL technique for classifying attacks in WSN is shown in figure 6.



Matrix of Confusion for RNN, CNN, ANN, and c, suggested CrTDL can be observed in

Figure 6.

Figure 7 illustrates the ROC curves calculated for the various classifiers. Various classifiers, including the suggested CrTDL with decision tree, ANN, CNN, RNN, and compared in terms of their estimation.

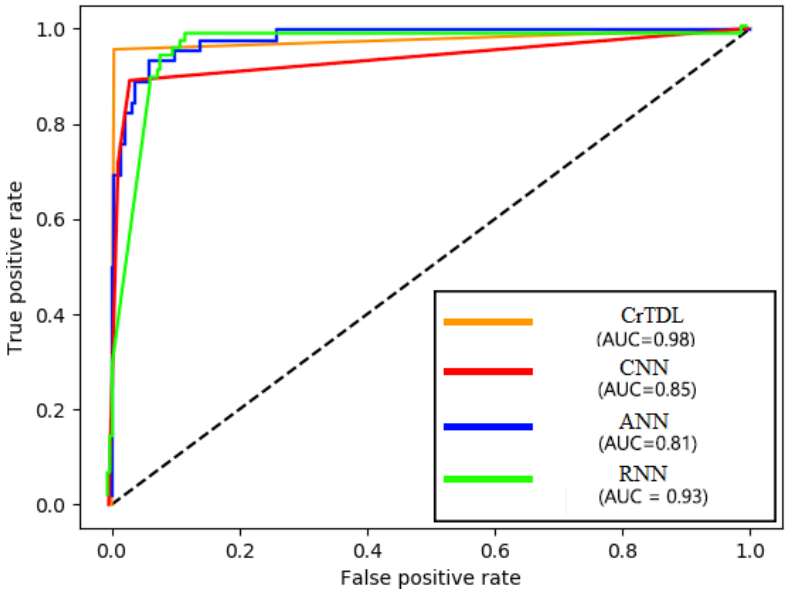


Figure 7: Comparison of ROC

In terms of the assault classification, the classifier's performance is carried out along with the computation of the confusion matrix. CNN regression classifier is created for attack classification in order to accomplish the types of attacks. The attacks performance of the embedded CNN is compared with the effectiveness of the Regression model, as well as traditional CNN, ANN, and RNN classifiers. The classifier results are presented in Table 5 as follows:

Table 5: Effectiveness of the Classifier

Parameters	**CNN**	**ANN**	**RNN**	**Proposed CrTDL**
TN	41440	41222	42099	41703
FN	1109	1305	1597	757
FP	841	1059	182	578
TP	21519	21324	21032	21872

The computation of the confusion matrix is done concurrently with the classifier's performance in classifying attacks. With the intention of finishing assault categorization, CNN regression classifier was developed. The display of classification performance in the CNN integrated Regression model is compared to that of traditional classifiers like CNN, ANN, and RNN. The subsequent results of the classifier are exhibited in Table 5:

Estimation is employed to establish that the proposed CrTDL outperforms other classifiers significantly. Table 7 shows the proposed CrTDL's assault categorization effectiveness using

common classifiers as CNN, ANN, and RNN classifier.

Accuracy: It connected all of the predictions to the number of values which were correctly anticipated. Formula (12) defines it.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} (12)$$

Sensitivity or Recall: This is the percentage of accurately predicted value to all predicted value that is defined. Equation (13) defines it.

$$Recall = \frac{TP}{TP+FN} \qquad (13)$$

Accuracy: The proportion of actual positive values to all anticipated values is shown.

Equation (14) states it.

$$Recall = \frac{TP}{TP+FP} \qquad (14)$$

F1 - Score: Provides the mean recall and accuracy average. in relation to one another.

Equation (15) states the F1-Score.

$$F1_{Score} = 2 * \frac{Precision*Recall}{Precision+Recall} \qquad (15)$$

Confusion Matrix: It offers a performance evaluation of the suggested approach along with a comparison of the actual and expected numbers. The study is based on the estimate of TP, FN, FP, and TN. It is represented in equation (16).

$$Confusin\ Matrix = \begin{bmatrix} TPF \\ FN \quad TN \end{bmatrix} \qquad (16)$$

In contrast, True Positive (TP) is defined as the count of instances that an AI model predicts as positive and are indeed positive.

A number initially predicted as negative, but later identified as positive in an artificial intelligence algorithm, is termed as a false positive (FP).

In the context of the AI model, True Negative (TN) refers to instances where the predicted value is negative and corresponds to the actual negative expectation. Conversely, False

Negative (FN) represents predicted outcomes that were initially anticipated as positive but later forecasted as negative in an artificial intelligence algorithm.

Table 6: Comparative classifier analysis

Parameters %	**CNN**	**ANN**	**RNN**	**CrTDL**
Accuracy	98	97	98	99
Precision	97	96	99	98
Recall	96	93	94	98
F1 – Score	94	95	95	97

The suggested CrTDL scheme obtains 98% accuracy rate, whereas CNN, ANN, and RNN demonstrate 98%, 97%, and 98% accuracy rates, respectively.Based on the performance of attack classification. This suggests that the suggested CrTDL approach performs admirably in the classification of attacks. Similar to how proposed CrTDL outperforms CNN, Classifiers in the recall include ANN and RNN scenario. The suggested CrTDL's recall is 99%, which is almost 3% higher than the value of the traditional CNN, ANN, and RNN classification techniques.

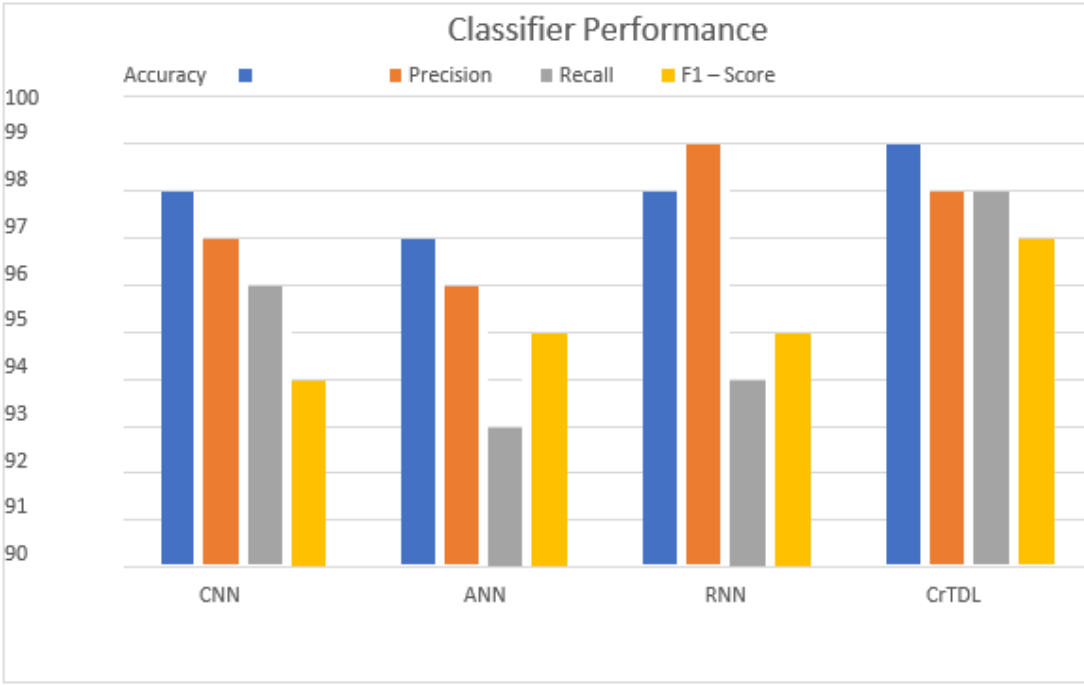


Figure 8: Comparison of performance

The accuracy and recall of the suggested CrTDL's performance show higher performance. The proposed CrTDL performs less well than the RNN classifier in terms of precision. A rise in the quantity of concealed layers may be the cause of the decreased precision.

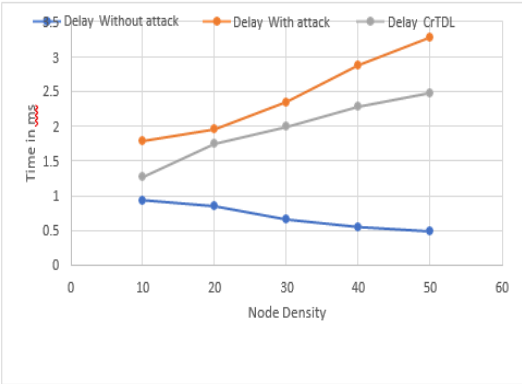
Detection is estimated using the classifier that CrTDL has proposed for the assault. With the suggested CrTDL parameters, the environment's variations for a normal, attacked, and unattacked state are computed. The calculation is done with a mobility of 10 m/s. The suggested CrTDL's performance for both attack and non-attack scenario

Table 7: Comparison of Results

Delay			
No.of Nodes	Without attack	With attack	**CrTDL**
10	0.94	1.79	1.27
20	0.85	1.96	1.75
30	0.66	2.35	1.99
40	0.55	2.88	2.28
50	0.49	3.28	2.48
Jitter			
No.of Nodes	Without attack	With attack	CrTDL
10	0.32	0.97	1.06
20	0.47	0.97	1.37
30	0.49	1.27	1.70
40	0.63	1.48	1.87
50	0.80	1.96	1.97
PDR			
No.of Nodes	Without attack	With attack	CrTDL
10	89.57	54	98
20	90.44	57	98
30	88.55	48	97
40	87.57	52	97
50	90.66	55	97
E2E			
No.of Nodes	Without attack	With attack	CrTDL
10	0.64	1.34	0.64
20	0.59	1.54	0.59
30	0.78	1.97	0.78
40	0.74	2.36	0.74
50	0.73	2.07	0.73
Throughput			
No.of Nodes	Without attack	With attack	CrTDL
10	94	53	99
20	92	57	98
30	93	45	99
40	94	48	96
50	93	43	98

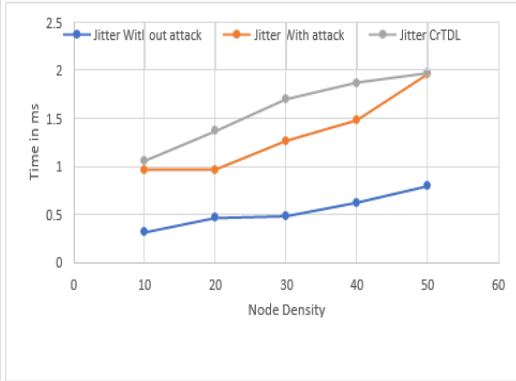
The estimation of the suggested CrTDL's variables was assessed both in the event of an assault and not. The performance study indicates that the recommended CrTDL technique improves network performance as opposed to having an integrated assault. Figure 9 presents the performance of the suggested CrTDL for scenarios with and without attacks for various parameters.

Comparison of Delay



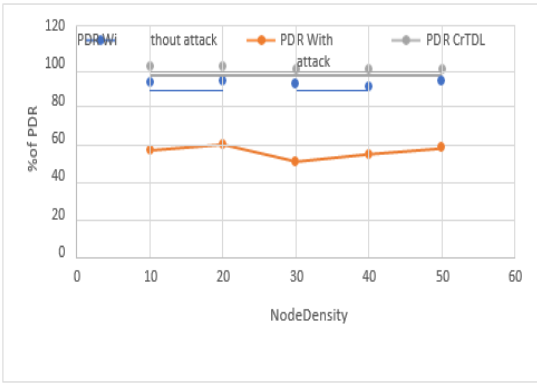
(a)

Comparison Of Jitter



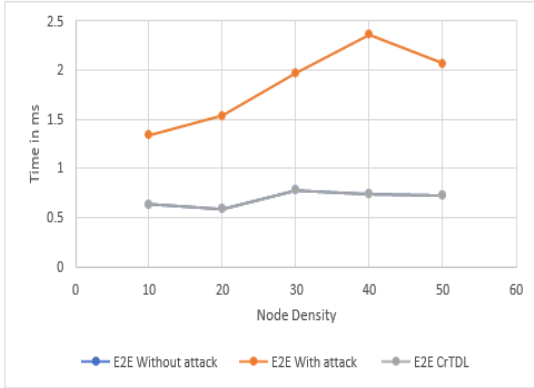
(b)

Comparison Of PDR



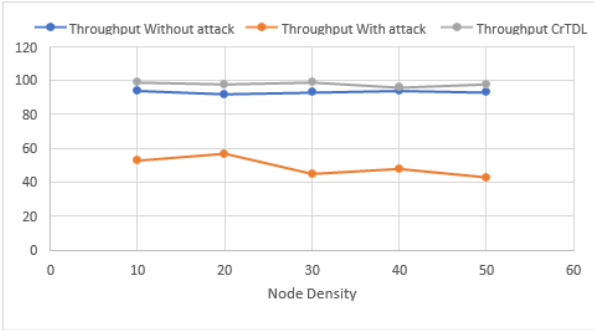
(c)

Comparison of E2E



(d)

Comparison of Throughput



(e)

Figure 9: (a) Delay Comparison (b) Jitter comparison (c) PDR comparison (d) E2E comparison (e) Contrasting throughput

Similar to this, for the suggested CrTDL, the network's rate of assault detection is calculated as shown in table 8.

Table 8: Rate of attack detection for different nodes

Attack Nodes	**Node = 10**	**Node = 20**	**Node = 30**	**Node = 40**	**Node = 50**
2	91	89	90	87	84
4	88	86	84	83	82
6	85	92	86	79	89
8	87	85	82	77	84
10	83	83	88	82	78

Estimates of the detection rates of 10, 20, 30, 40, and 50 nodes are made by taking into account the various attacks on each node. The rate of attack node of node 10 detection is 91%. The rate of node detection is assessed for various node sizes and attack types. The suggested CrTDL's assault detection rate is calculated for different nodes and levels of mobility. Table 10 shows the effectiveness of the suggested CrTDL for categorizing attacks.

Table 9: CrTDL assault performance categorization

	Percentage of Accuracy	**Precision**	**Recall**
Normal activity	98	98	97
Anomaly	97	96	99
DoS	96	97	97
Probe Attack	92	92	93
Gain root access	93	93	92
Root to Local	95	96	94

To analyze, the classifier's performance of the CrTDL technique is calculated for various types of attacks. Based on the analysis, the CrTDL scheme exhibits a higher anomaly detection accuracy, with a value of 98%. The categorization accuracy for a black hole attack is expected to be 96%. Similarly, it is estimated to have been 96% successful in terms of Blackhole and targeted DoS attacks. For anomaly detection, a recall value higher than 98% is estimated.

6. Conclusion

In this paper, a centralised SDN architecture-based security strategy for the WSN network was described. The WSN data saved in SDN and encrypted as part of the proposed CrTDL method. The regression classifier model based on AbaBoost is designed to identify and categorize attacks in order to secure network data. The CrTDL scheme showcases significant performance in types of attacks, as observed in a comparison with conventional classifiers such as RNN, CNN, and ANN. While traditional classifiers like CNN, ANN, and RNN yield classification accuracies of 98%, 97%, and 98%, respectively, the suggested CrTDL technique achieves an accuracy of 99%. This suggests that compared to the standard CNN, ANN, and RNN classifiers, the suggested CrTDL cryptography-based classifier performs about 3% better. The suggested CrTDL system is efficient for safe transfer of data in the WSN network, according to study. This work can be enhanced in the future by including a integrating an IDS with a real-time assault detection system.

References

1. Tsiknas, K., Taketzis, D., Demertzis, K., & Skianis, C. (2021). Threats to Industrial WSN: A Survey on Attacks and Countermeasures. WSN, 2(1), 163-188.
2. Chegini, H., Naha, R. K., Mahanti, A., & Thulasiraman, P. (2021). Process Automation in an WSN–Fog–Cloud Ecosystem: A Survey and Taxonomy. WSN, 2(1), 92-118.
3. Kimani, K., Oduol, V., & Langat, K. (2019). security challenges for WSN-based smart grid networks. International Journal of Critical Infrastructure Protection, 25, 36-49.
4. Saharkhizan, M., Azmoodeh, A., Dehghantanha, A., Choo, K. K. R., & Parizi, R. M. (2020). An RNN of deep recurrent neural networks for detecting WSN attacks using network traffic. IEEE Internet of Things Journal, 7(9), 8852-8859.
5. Kimani, K., Oduol, V., & Langat, K. (2019). security challenges for WSN-based smart grid networks. International Journal of Critical Infrastructure Protection, 25, 36-49.
6. Radanliev, P., De Roure, D., Cannady, S., Montalvo, R. M., Nicolescu, R., & Huth, M. (2018). Economic impact of WSN risk-analysing past and present to predict the future developments in WSN risk analysis and WSN insurance.
7. Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in WSN security: Current solutions and future challenges. IEEE Communications Surveys & Tutorials, 22(3), 1686-1721.
8. Tahsien, S. M., Karimipour, H., & Spachos, P. (2020). Machine learning based solutions for security of Internet of Things (WSN): A survey. Journal of Network and Computer Applications, 161, 102630.
9. Sagduyu, Y. E., Shi, Y., & Erpek, T. (2019, June). WSN network security from the perspective of adversarial deep learning. In 2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON) (pp. 1-9). IEEE.
10. Bagaa, M., Taleb, T., Bernabe, J. B., & Skarmeta, A. (2020). A machine learning security framework for WSN systems. IEEE Access, 8, 114066-114077.
11. Cui, L., Yang, S., Chen, F., Ming, Z., Lu, N., & Qin, J. (2018). A survey on application of machine learning for Internet of Things. International Journal of Machine Learning and netics, 9(8), 1399-1417.
12. Roopak, M., Tian, G. Y., & Chambers, J. (2019, January). Deep learning models for security in WSN networks. In 2019 IEEE 9th annual computing and communication workshop and conference (CCWC) (pp. 0452-0457). IEEE.
13. Ferrag, M. A., Maglaras, L., Moschoyiannis, S., & Janicke, H. (2020). Deep learning for security intrusion detection: Approaches, datasets, and comparative study. Journal of Information Security and Applications, 50, 102419.
14. Berman, D. S., Buczak, A. L., Chavis, J. S., & Corbett, C. L. (2019). A survey of deep learning methods for security. Information, 10(4), 122.
15. Zhao, S., Li, S., Qi, L., & Da Xu, L. (2020). Computational intelligence enabled security for the internet of things. IEEE Transactions on Emerging Topics in Computational Intelligence, 4(5), 666-674.
16. Ullah, F., Naeem, H., Jabbar, S., Khalid, S., Latif, M. A., Al-Turjman, F., & Mostarda, L. (2019). security threats detection in internet of things using deep learning approach. IEEE Access, 7, 124379-124389.
17. Salloum, S. A., Alshurideh, M., Elnagar, A., & Shaalan, K. (2020, March). Machine Learning and Deep Learning Techniques for security: A Review. In AICV (pp. 50-57).
18. Mirsky, Y., Doitshman, T., Elovici, Y., & Shabtai, A. (2018). Kitsune: an RNN of autoencoders for online network intrusion detection. arXiv preprint arXiv:1802.09089.
19. Yuan, X., Li, C., & Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In 2017 IEEE International Conference on Smart Computing (SMARTCOMP) (pp. 1-8). IEEE.

20. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baWSN—network-based detection of WSN botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12-22.
21. Hanif, S., Ilyas, T., & Zeeshan, M. (2019, October). Intrusion detection in WSN using artificial neural networks on UNSW-15 dataset. In 2019 IEEE 16th International Conference on Smart Cities: Improving Quality of Life Using ICT & WSN and AI (HONET-ICT) (pp. 152-156). IEEE.
22. Khraisat, A., Gondal, I., Vamplew, P., Kamruzzaman, J., & Alazab, A. (2019). A novel RNN of hybrid intrusion detection system for detecting internet of things attacks. Electronics, 8(11), 1210.
23. Gu, Z., Nazir, S., Hong, C., & Khan, S. (2020). Convolution neural network- based higher accurate intrusion identification system for the network security and communication. Security and Communication Networks, 2020.
24. Ferrag, M. A., & Maglaras, L. (2019). DeepCoin: A novel deep learning and blockchain-based energy exchange framework for smart grids. IEEE Transactions on Engineering Management, 67(4), 1285-1297.
25. Vangipuram, R., Gunupudi, R. K., Puligadda, V. K., & Vinjamuri, J. (2020). A machine learning approach for imputation and anomaly detection in WSN environment. Expert Systems, 37(5), e12556.
26. Haji, S. H., & Ameen, S. Y. (2021). Attack and anomaly detection in WSN networks using machine learning techniques: A review. Asian Journal of Research in Computer Science, 30-46.
27. Chakraborty, D., Narayanan, V., & Ghosh, A. (2019). Integration of deep feature extraction and RNN learning for outlier detection. Pattern Recognition, 89, 161-171.
28. Zolanvari, M., Teixeira, M. A., & Jain, R. (2018, November). Effect of imbalanced datasets on security of industrial WSN using machine learning. In 2018 IEEE International Conference on Intelligence and Security Informatics (ISI) (pp. 112- 117). IEEE.
29. Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., & Elovici, Y. (2018). N-baWSN—network-based detection of WSN botnet attacks using deep autoencoders. IEEE Pervasive Computing, 17(3), 12-22.
30. KoronWSNis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-WSN dataset. Future Generation Computer Systems, 100, 779-796.
31. Nguyen, T. D., Marchal, S., Miettinen, M., Fereidooni, H., Asokan, N., & Sadeghi, A. R. (2019, July). DÏoT: A federated self-learning anomaly detection system for WSN. In 2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS) (pp. 756-767). IEEE.
32. Hasan, M., Islam, M. M., Zarif, M. I. I., & Hashem, M. M. A. (2019). Attack and anomaly detection in WSN sensors in WSN sites using machine learning approaches. Internet of Things, 7, 100059.
33. Hwang, R. H., Peng, M. C., Huang, C. W., Lin, P. C., & Nguyen, V. L. (2020). An unsupervised deep learning model for early network traffic anomaly detection. IEEE Access, 8, 30387-30399.