International Journal of Computational and Experimental Science and ENgineering (IJCESEN)

Vol. 10-No.4 (2024) pp. 1169-1180
http://www.ijcesen.com

ISSN: 2149-9144

Research Article

A novel optimized deep learning based intrusion detection framework for an IoT networks

P. Jagdish Kumar[1*], S. Neduncheliyan[2]

[1] Research Scholar, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai 600073 Tamil Nadu, India
* **Corresponding Author Email:** pjkcse123@gmail.com - **ORCID:** https://orcid.org/0009-0006-3664-6323

[2] Dean of School Computing, Bharath Institute of Higher Education and Research
Chennai 600073 Tamil Nadu, India
Email: dean.cse@bharathuniv.ac.in - **ORCID:** https://orcid.org/0009-0002-0904-8695

Article Info:

DOI: 10.22399/ijcesen.597
Received : 08 November 2024
Accepted : 16 November 2024

Keywords :

Internet of Things(IoT),
Intrusion Detection Systems(IDS),
Long Short Term Memory(LSTM),
Genetic-Bee.

Abstract:

The burgeoning importance of Internet of Things (IoT) and its diverse applications have sparked significant interest in study circles. The inherent diversity within IoT networks renders them suitable for a myriad of real-time applications, firmly embedding them into the fabric of daily life. While IoT devices streamline various activities, their susceptibility to security threats is a glaring concern. Current inadequacies in security measures render IoT networks vulnerable, presenting an enticing target for attackers. This study suggests a novel dealing to address this challenge through the execution of Intrusion Detection Systems (IDS) leveraging superior deep learning models. Inspired by the benefits of Long Short Term Memory (LSTM), we introduce the Genetic Bee LSTM(GBLSTM) networks for the development of intelligent IDS capable of detecting a wide range of cyber-attacks targeting IoT area. The methodology comprises four key execution: (i) collection of unit for profiling normal IoT device behavior, (ii) Identification of malicious devices during an attack, (iii) Prediction of attack types implemented in the network. Intensive experimentations of the suggested IDS are conducted using various validation methods and prominent metrics across different IoT threat scenarios. Moreover, comprehensive experiments are conducted to evaluate the suggested models alongside existing learning models. The results demonstrate that the GBLSTM-models outperform other intellectual models in terms of accuracy, precision, and recall, underscoring their efficacy in securing IoT networks.

1. Introduction:

In Moder era, the Internet of Things (IoT) is firmly entrenched itself across a spectrum of applications including healthcare, automation, manufacturing, commerce, and residential and commercial sectors. The burgeoning array of IoT applications undeniably offers enhanced comfort across various facets of individuals' paths [1]. IoT represents a fusion of interlinked gadgets or entities interlinked to the web, distinguished by distinct addresses for identification. These items can be remotely controlled and possess the capability to interact with one another, along with gathering ambient data and converting it into valuable insights. Nevertheless, the self-arranging nature and limitations in resources of IoT networks render them vulnerable to an array of risks [2].

The inadequate protocols and minimised of advanced intrusion identification systems in IoT networks expose to various forms of attacks including data breaches, impersonation, Distributed Denial of Service (DDoS), and insecure gateways [3-7]. Such vulnerabilities can result in catastrophic consequences, disrupting system availability and potentially triggering system outages. These issues hinder the widespread adoption of IoT on a global scale, consequently impeding its growth rate [8,9].
There exist certain simplistic approaches for addressing the aforementioned issues. Techniques such as Signature-based Intrusion Detection Systems (IDS) identify attacks and malicious nodes by storing data in databases. Nonetheless, these methods introduce processing overhead and are vulnerable to unidentified hazard. The emergence of

AI Models has enhanced the design of IDS, offering distinct advantages over conventional methodologies, effectively mitigating the challenge of detecting unknown attacks [10-13]. While machine and deep learning algorithm-driven Intrusion Detection Systems (IDS) present numerous advantages over traditional approaches, they encounter challenges with overfitting as the volume of attacks grows [14]. Therefore, it is imperative to carefully choose deep learning algorithms to develop an intelligent IDS capable of scaling with the expanding array of attacks. In summary, there's a necessity for an intelligent and adaptable IDS to anticipate various attack categories within IoT networks. The primary aim of this study is to initiate an innovative IoT secure methodology that is smart, scalable, and reliable for predicting various types of attacks across different scenarios. To achieve this goal, a new DL framework called GBLSTM (Genetic Bee Long Short Term Memory) is introduced. By incorporating the whale optimization algorithm into the LSTM model, the framework presents enhanced scalability, making it well-suited for predicting a wide range of attacks. Moreover, the GBLSTM not only forecasts attacks but also provides recommendations for trust-based countermeasures to safeguard networks against these threats. The research includes extensive experimentation with the suggested learning model using numerous benchmarks and gathering data in live time. Further analysis, experimentation various performance metrics and comparative studies has been briefly described in the forthcoming sections. Section-II discusses about other research works on different IoT attacks with malicious node detection. Section-III delineates the real-time data acquisition process, attack model descriptions, suggested machine learning methodologies, and system architecture. In Section-IV, the experimental configuration and additional benchmark information are expounded. Section-V showcases the outcomes of thorough analysis and juxtaposition with contemporary methodologies. Lastly, Section-V encapsulates the conclusions alongside prospective avenues for further exploration.

2. Related works

Abhishek et al. extensively discuss numerous ML [15]. They analyze and authenticate Gradient Boosting, ensemble classifiers, and random forest algorithms utilizing the Raspberry Pi 3 platform. Misra and others [16] introduced an IDS based on Learning Automata, focusing on specification-based intrusion detection to counter distributed Denial of Service (DoS) attacks targeting IoT. Instead of individual devices, they concentrate on securing the IoT middleware layer. Their envisioned security framework sets a predefined limit for the volume of requests a middleware layer can handle. Upon surpassing this threshold, the system identifies an ongoing attack. Lee et al. introduced a threshold-oriented intrusion identification mechanism tailored for IoT area. Regular power consumption monitoring is employed to identify rogue nodes within the network. Default threshold values are assigned to each network and continuously monitored. If any aberration in energy consumption is detected, specific nodes are flagged as malicious and subsequently expelled [17,18]. Tama et al. introduced an intrusion detection system (IDS) based on anomaly detection, utilizing a gradient boosting machine (GBM) as its core detection mechanism [19]. They optimized the GBM's parameters through a grid search and evaluated the IDS's performance using both retention and cross-folding techniques on 3 distinct datasets: UNSW-NB15, NSL-KDD, and GPRS. Their study demonstrated that the suggested IDS outperformed fuzzy classifiers, GAR forests, and tree populations like precision, specificity, sensitivity, and area under the curve (AUC) of the sub-curve. Primartha et al. conducted an investigation into the efficacy of RF-based Intrusion Detection Systems (IDS) in terms of both accuracy and false alarm rates [20]. They utilized NSL-KDD, UNSW-NB15, and GPRS datasets for both training and model testing purposes. The performance of the suggested IDS was evaluated across various tree configurations, revealing that a set comprising 800 trees yielded the most favourable outcomes, while a set of 20 trees resulted in the least desirable performance. Additionally, through statistical analysis employing the Freedman classification, it was determined that the superior results obtained by the RF-based IDS were comparable to those achieved by hybrid classifiers such as Random + Naive Bayes, as well as distinct classifiers like NBTree and multilayer perceptron.

3. Proposed methodology

This segment delves into the operational framework of the suggested Intrusion Detection System (IDS), comprising data gathering modules, feature abstraction, and the envisioned training algorithms.

3.1 System overview

An outline of the envisaged GBLSTM-NET based Intrusion Detection System (IDS) and Recommendation system is depicted in figure 1. The initial tier of the architecture will replicate the live time emulation of IoT networks utilizing MAC addresses. Subsequently, the second tier

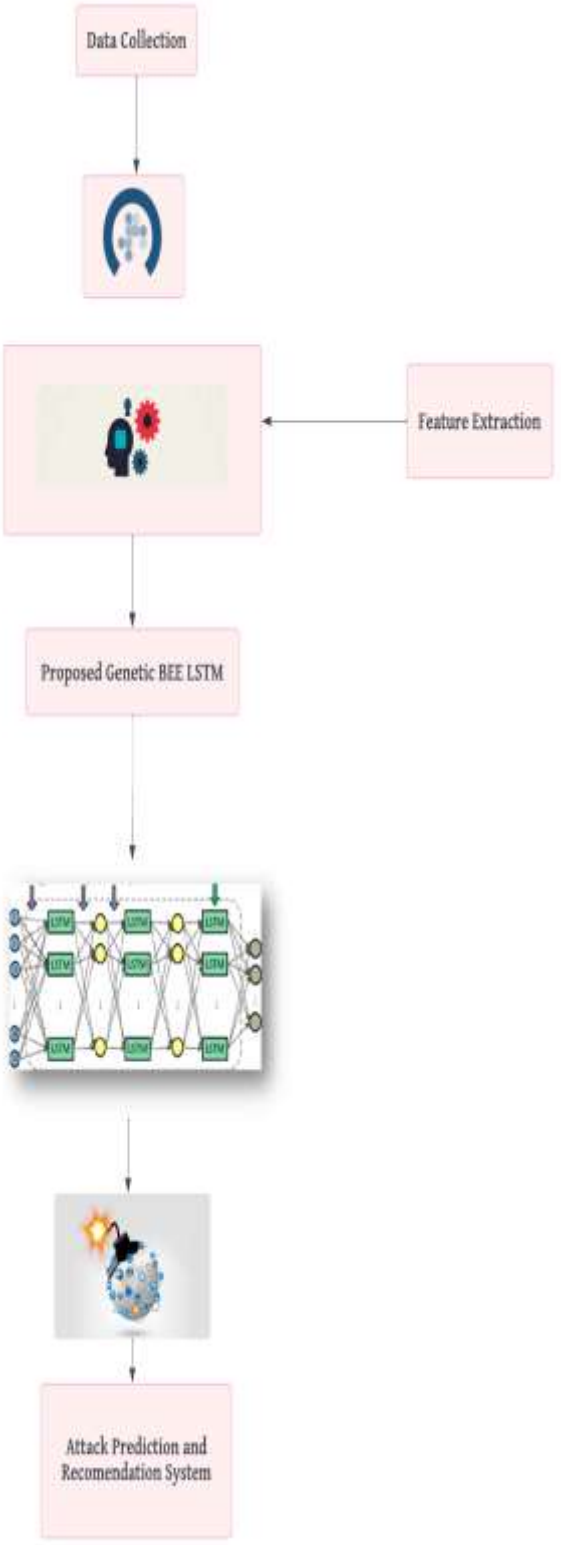


Figure 1 *Overall Architecture for the Proposed Framework*

incorporates a collection of data unit engineered to gather packets during both normal and hazard scenarios. Moving to the third tier, various features are extracted from the preprocessed data, which are then utilized to train the suggested deep learning model aimed at predicting hazards. Lastly, the architecture employs Whale Integrated Long Short-Term Memory (LSTM) networks to forecast malicious nodes pertaining to five major attack types. Upon detecting an attack, the system determines: a) Determine whether the node exhibits malicious behavior or operates within normal parameters, b) the specific the form of assault, and c) The IP address (MAC) of the targeted device during the assault. Following a cybe assault, the suggested framework propagates warning notifications across all network endpoints via a forwarding mechanism.

3.3 Feature extraction:

It's imperative to oversee the nature of characteristics within the amassed collection of data. Subsequently, the initial data undergo preprocessing, yielding computed subsets of features, which are then presented in Table 1.

Table 2 depicts the diverse attributes computed for optimal prognostic maneuvers within network systems. Accordingly, it can be posited that said attributes manifest as numerical values, thereby enhancing the efficacy of classifiers. Consequently, the pivotal endeavor resides in the transformation of numerical attributes into vectors through diverse methodologies. Notably, label encoding and one-hot encoding methodologies emerge as predominant modalities for such conversion processes. In the present study, label encoding was favored owing to its superior simplicity. These encoding methodologies were employed for the anticipation and classification of malevolent nodes as well as discerning the nature of attacks. Additionally, the distribution frequency of hazards across the dataset and ramifications on attributes are illustrated in a corresponding figure. From the figure 2 it's evident that factors like Data Packet Size, IP Address Conflicts, Throughput, and Bandwidth significantly influence various types of attacks. Consequently, these attributes, in addition to addressing features, are employed to train the model with precision.

3.4 Proposed learning model:

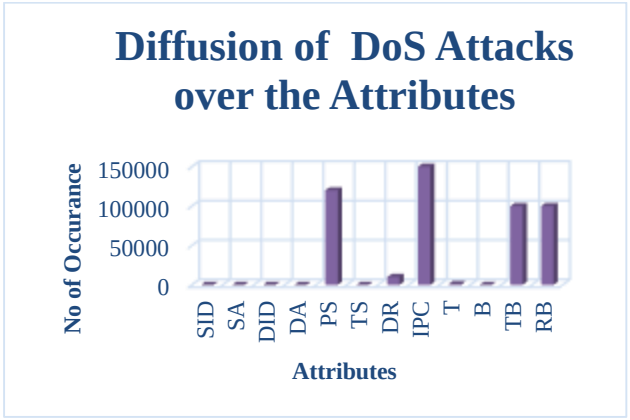
This segment provides an initial glimpse into Long Short-Term Memory (LSTM), the Whale Optimization Algorithm, and the suggested deep learning architectures of GBLSTM.

Genetic bee:

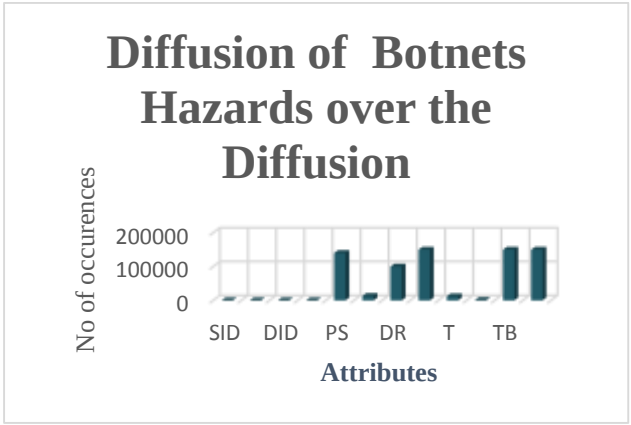
In this segment, the innovative Genetic Bee Colony (GBC) algorithm designed for identifying forward-thinking attributes (figure 3). GBC combines two well-established algorithms, ABC and GA, in a novel hybrid meta-heuristic approachment. The primary mission of this algorithm is to pinpoint the most advantageous attributes to enhance accuracy. Metaheuristic algorithms strive to discover the best

Table 1. *List of attributes retrieved from Datasets*

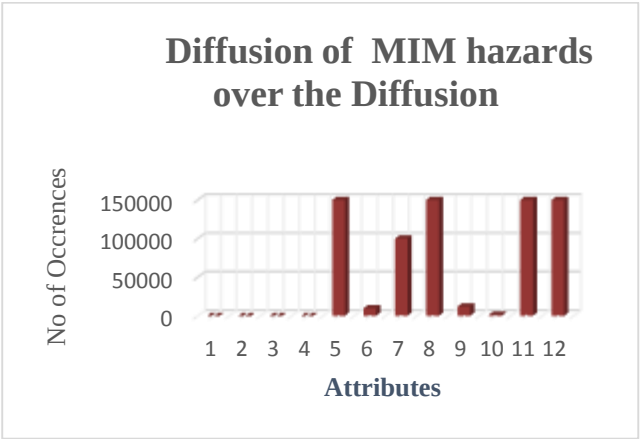
Sl.no	Features Used	Data Types	Illustration
01	Source ID	Numerical	Identifies the IoT nodes that function as the origin for transmitting data.
02	Source Address	Numerical values	Provides the location of the originating IoT nodes.
03	Destination ID	Numerical	Signifies the identification of the IoT nodes functioning as recipients for data.
04	Destination Address	Numerical	Address of the receiver.
05	No of Packets	Numerical (100-1200 bytes)	Represents the count of packets sent/received.
06	Time Stamps	Numerical	Documents the temporal spans for various data transmission instances.
07	TCP/IP Data Rates	Numerical	Highlights differences in protocol distributions between normal and attack data; under normal conditions, UDP packets have superior performance over TCP, whereas this is reversed during attacks, aiding in improved attack classification.
08	IP Address Conflict	Nominal values	Identifies potential terminal access points of an IoT device for analysis of exploitation risks.
09	Throughput	Numerical	Detects assaults by analyzing variations in data flow attributes.
10	Bandwidth	Numerical	Computes the network bandwidth utilized by various IoT devices across different scenarios, including typical and adversarial conditions.
11	No of Bytes Transmitted	Numerical	Quantity of bytes transmitted by individual nodes within the network.
12	No of Bytes Received	Numerical	Volume of bytes received by each node within the network.



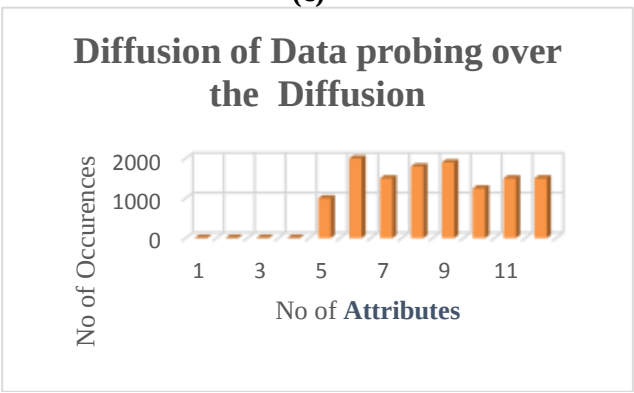
(a)



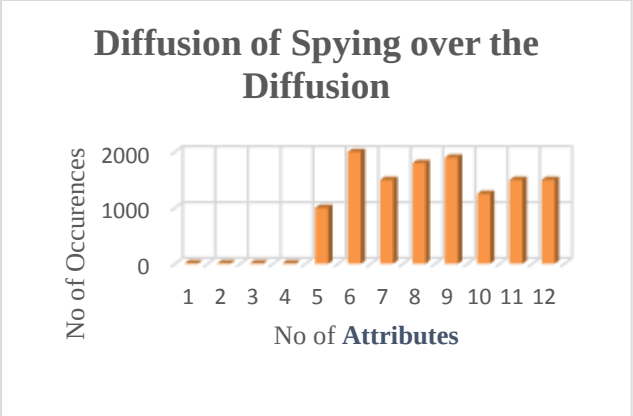
(b)



(c)



(d)



(e)

Figure 2. *The influence of various types of attacks***.**

feasible solution, finding the right equilibrium between exploiting and exploring resources is crucial. While the original ABC algorithm excels in exploration by uncovering new solutions in the optimization search space, it lacks in exploitation, resulting in prolonged computational times required for convergence to the optimal solution. On the other hand, GA demonstrates proficient crossover and mutation operations but falters in effectively exploring the evolutionary search space, leading to premature convergence to local optima. Thus, to achieve a harmonious blend of exploitation and exploration, leveraging the strengths of nature-inspired metaheuristic evolutionary algorithms, and mitigating their weaknesses like premature convergence and computational time, our proposed GBC algorithm integrates GA operators with the ABC algorithm. This fusion yields a modified ABC-based algorithm tailored for constrained optimization. In our adapted approach, GA exploitation operations are embedded into the exploitation phase during the onlooker bee stage to facilitate information exchange among worker bees and onlooker bees for discovering optimal solutions. Similarly, GA exploitation operations are integrated into the scout bee stage to streamline the process of replacing exhausted solutions. The proposed algorithm comprises five stages: preprocessing, representation and initialization, employee bee, onlooker bee, and scout bee phases.

Process mechanism:

Step 1: Set the initial count of bee colonies g_m , establish the pivotal constant Q determine the transfer intensity σ, specify the significance of the transfer factor α□, assign weight to heuristic factors β□, define the minimum number of iterations G_{min} , specify the maximum number of iterations maxG , designate the termination threshold for Genetic Bee algorithm iterations max K_{max}.

Step 2: Place the bees within the starting point, as

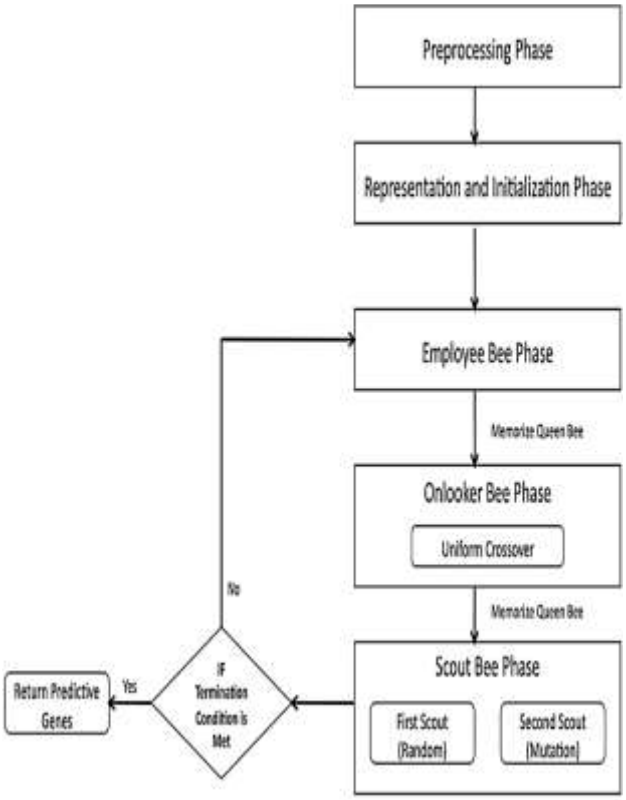


Figure 3 *The main phases of the Genetic Bee Colony (GBC) algorithm.*

per the given formula, to advance the progression of the primary factor and accomplish the selection of all nodes.

$$p_{ij}^{k} = \begin{cases} \frac{\rho_{ij}(NC)^{\alpha}\eta_{ij}(NC)^{\beta}}{\sum_{j\in \text{tabu}_k}\rho_{is}(NC)^{\alpha}\eta_{is}(NC)^{\beta}} \\ 1 \end{cases}$$

j∈tabu$_k$ (1)

Otherwise

In (1), η_{ij}= 1/ $T(j)$ □□,$T(j)$(j □1,2….,ma) represents the processing time of j nodes.

Step 3: Utilize the initial solution discovered by bees as the foundational population for the genetic algorithm. If the ongoing iteration K of the genetic bee algorithm is below the maximum iteration limit K_{max}, commence by assessing the fitness of each individual, identifying the most adept one, and then initiating genetic maneuvers.

Step 4 Determine the resemblance between the most adept member and other individuals within the population. If the resemblance proves significant, prioritize mutation prior to crossbreeding. Conversely, if the resemblance is low, initiate crossbreeding before mutation.

Step 5: Adjust the primary determinant based on the equation mentioned above by the most adept member within the populace. Should the genetic algorithms' iteration count, Gmin, remain less than or equal to G, proceed with genetic operations.

Conversely, if Gmin surpasses G and fulfills the condition (1.1), transition to Step 2 to resume the execution of the colony algorithm.

Step 6: Ensure that if the present count of iterations in genetic algorithms does not exceed the maximum value denoted as G_{max}, the genetic operations persist. However, if the count reaches G_{max}, proceed with the subsequent steps.

Step 7: Revise the worldwide best resolution, and adapt the primary determinant in accordance with the principle outlined in section 1.3.

Step 8: If the present iteration counts of the Genetic Bee algorithm, denoted as Kmax, is less than K, proceed to step 2. However, if Kmax equals K, the algorithm will output the computation results, subsequently concluding

Reconcurrent neural networks:

In Recurrent Neural Networks (RNNs), the hidden layers connect to unseen layers in additional nodes of alternate fresh network, enhancing their ability to remember sequential data. RNNs are particularly suited for time series and big data analysis because of their capacity to encode historical information rapidly. These networks exhibit dynamic behavior in graph shapes, showcasing sequence synchronization. By utilizing internal memory (state), RNNs process input sequences and leverage past data to predict upcoming values. However, in live scenarios with significant time gaps between past and future data points, RNNs struggle to retain meaningful information, resulting in the notorious vanishing gradient problem. To address this, the Long Short-Term Memory (LSTM) network was introduced, significantly enhancing RNN performance.

LSTM – Long short term memory:

An extensively employed learning methodology known as the Long Short-Term Memory (LSTM) network is favoured for its adaptability in retaining information, making it well-suited for handling extensive datasets. Figure 4 illustrates the architecture of the LSTM network, showcasing its efficacy across various applications. The proposed amalgamated learning framework integrates LSTM, a recurrent neural network architecture renowned for its memory retention capabilities, with the Whale optimizer. LSTM is structured around distinct modules: the input gate (I.G), forget gate (F.G), cell input (C.I), and output gate (O.G). Fundamentally, LSTM functions as a memory-centric neural network, adept at retaining information across iterations. In this setup, let's denote the unseen layer output as 'ht', its preceding output as 'ht−1', the cell input state as 'Ct', the cell output state as 'Gt', and their respective former states as 'Gt-1'. The states of the three gates are represented as 'j_t', 'T_f', and 'T0'. The LSTM architecture facilitates the seamless communication of 'Gt' and 'ht' to subsequent layers within the RNN structure.

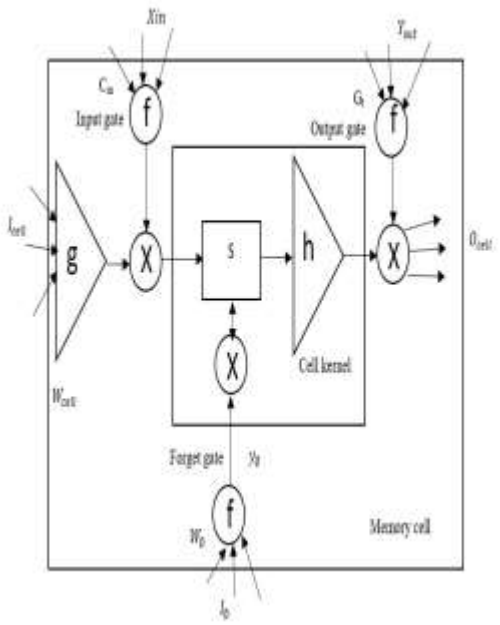


Figure 4 *LSTM Structure*

Central to LSTM's operation is the merging of the previous unit's output with the current input state, leveraging the output and forget gates to upgrade memory. The computation of 'Gt' and 'ht' is governed by the formulation.

$$I.G: \quad j_t = \theta\left(G_l^i.O_t + G_h^i.e_{t-1} + s_i\right) \tag{2}$$

$$F.G: T_f = \theta\left(G_l^f.O_t + G_h^f.e_{t-1} + s_f\right) \tag{3}$$

$$O.G: \ T_o = \theta\left(G_l^0.O_t + G_h^o.e_{t-1} + s_o\right) \tag{4}$$

$$C.I: \widetilde{T_C} = \tanh\left(G_l^C.O_t + G_h^C.e_{t-1} + s_C\right) \tag{5}$$

The weight matrices $G_l^0, G_l^f, \ G_l^i, G_l^C$ are associated with the connections between input gates and output layers, while $G_h^i, G_h^f, G_h^o, G_h^C$ pertain to the weight conditions established between hidden and input layers. The bias vectors "$s_i, s_f, \ s_o, \ s_C$ are also incorporated, and the hyperbolic function tanh is employed. The computation of the cell output state is then formulated as pursues:

$$T_C = k_t * \widetilde{T_C} + T_f * T_{t-1} \tag{6}$$

$$e_t = T_o * \tanh(T_C) \tag{7}$$

The ultimate result score is acquired utilizing the aforementioned equation.

Motivation behind the proposed model:

The utilization of large datasets with LSTM poses several limitations [21-34], primarily due to the increased memory cell requirement, leading to heightened computational complexity. Consequently, this scenario often results in

overfitting issues. To address these challenges, there arises a necessity for a meticulously structured model capable of predicting various categories of both familiar and unfamiliar attacks within IoT networks. In meeting these criteria, an exploration into a low-complexity learning model has been conducted. The primary objective of this hybrid model is to devise a novel algorithm by amalgamating whale algorithms into LSTM networks.

Proposed GBLSTM models:

The utilization of the simple whale algorithms to optimize LSTM network weights is discussed. Here, the various criteria used by whales for searching and capturing prey serve as the primary means to optimize LSTM network weights. The complete mechanism of GBLSTM learning models is depicted in the figure 4. Initially, a random set of weights and biases is assigned to LSTM cells. The accuracy of the proposed model is defined as the fitness function. Each iteration involves the calculation of input biases and weights using mathematical equations (7), (9), and (13). These computed weights are then applied to the LSTM network, where the fitness function is evaluated. If the fitness function reaches the predefined threshold, the iteration halts; otherwise, it continues. This method indicates that while whale optimization may result in slower convergence compared to other meta-heuristic algorithms, it can enhance optimization time and detection efficiency.

In response to the expected onslaught of diverse attacks, GBLSTM-TRS introduces an innovative method to prevent these intrusions within network structures. The proposed system promotes an alternative pathway for data transmission, prioritizing Quality of Service (QoS) awareness, while also blocking paths where malicious activities have occurred. These incidents are then logged in separate repositories of countermeasures, facilitating the prevention of network attacks. Presented below is the detailed pseudocode outlining the operational principles of GBLSTM-TRS.

4. Experimentation setup

The investigation was conducted utilizing Lenovo Thinkpad laptops equipped with Windows 10 operating system, powered by Intel Core i7 processors (8th generation) clocked at 3.6 GHz, 16GB of RAM, and NVIDIA GeForce GTX graphics cards. As delineated previously, a practical environment was simulated through the utilization of the OMNET++-IOT API, while Python programming facilitated the creation of threat models. The envisioned deep learning framework was implemented utilizing TensorFlow version 1.3.5. Data analysis and feature engineering tasks were performed using the pandas and numpy libraries.

Table 2. *The diverse attributes computed for optimal prognostic maneuvers.*

Sl.no	Pseudo code for the GBLSTM -TRS IDS Systems
01	Input F= Characteristics derived from the networks.
02	Output = Forecasting the occurrences of assaults along with suggestions for mitigation.
03	$D_{countermeasure}$ = MongoDb (QoS-Aware Paths)
04	While True:
05	Train the networks utilizing the equations (18) and (19) with the features.
06	Compute the resulting output employing Equations (18) and (19).
07	If ($T_c < T_{t1}$) // T_{t1}
08	Attack No 1 is predicted
09	Else if ($T_c < T_{t21}$)
10	Anticipate a second assault.
11	Advise for the alternative route and obstruct the passage.
12	Else if ($T_c < T_{t3}$)
13	Anticipate a third assault.
14	Suggests an alternative route and obstructs the original pathway.
15	Else if ($T_c < T_{tn}$)
16	Attack N is predicted
17	Suggests an alternative route while obstructing the original course.
18	Else
19	If there is no anticipated assault, proceed to Step 07.
20	End
21	End
22	End
23	End
24	Go to Step 7
25	End

4.1 Benchmark datasets:

In addition to real-time data collection, this research incorporates three distinct benchmarks to assess the proposed model. These benchmarks, namely CIDDS-001, UNSW-NB15, and NSLKDD, are employed to evaluate the model's performance with expanded datasets. CIDDS-001 and UNSW-NB15 datasets, containing real-time traffic, offer substantial advantages in developing an intelligent IDS for monitoring and predicting various categories of attacks in IoT networks. The combined datasets encompass 3.2 million records, including both malicious and normal traffic. CIDDS-001 comprises 12 features with 2 labeling attributes, comprising

80,000 normal and 20,000 attack (DoS) records. Conversely, the UNSW-NB15 benchmark features 49 attributes with a single class attribute. For training, 56,000 instances of normal traffic and 119,341 instances of attacked traffic were utilized, while the testing phase involved 37,000 normal traffic instances and 45,332 attack instances. Similarly, the NSL-KDD benchmark comprises 41 features with a single label attribute.

4.2 Performance evaluation:

To assess and authenticate the proposed intrusion detection system utilizing the Proposed BEL framework, various metrics have been employed including Accuracy, Sensitivity, Selectivity, Specificity, and compared against alternative classifiers such as BLSTM[35-38], CNN+LSTM[39] Ensemble, CART, and Multi-layer Perceptron (MLP15]. The validation criteria have been calculated according to the equations specified below.

$$\text{Accuracy} = \frac{\text{DR}}{\text{TNI}} \text{x100} \quad (8)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+TN}} \text{x100} \quad (9)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TP+TN}} \text{x100} \quad (10)$$

In this context, TP and TN denote the values signifying correct identifications and accurate exclusions, while DR and TNI stand for the count of identified outcomes and the total number of trials, respectively.

5. Results and discussion

The evaluation of the proposed deep learning framework alongside alternative learning methodologies tailored to real-time simulated scenarios has been conducted, complemented by rigorous testing against various benchmarks such as CIDDS-001, UNSWNB15, and NSL-KDD. A comparative analysis has been performed, followed by rigorous statistical scrutiny. The employed deep learning architectures have been deemed suitable for intrusion detection within IoT networks. The methodology involved partitioning the complete datasets into training and testing sets, with a distribution of 70% for training and 30% for testing. In the initial phase, a preliminary set of broad values for the hyperparameters of the model underwent refinement through the utilization of optimized whale algorithms during the training procedure to ascertain optimal outcomes. It was determined that the most favorable outcomes in the tuning phase were achieved with 50 epochs, a learning rate of 0.0001, and a batch size of 80 for output. The figure 4 depicts the precision of detection and the rate of error during training across various scenarios of testing data employed for validation. Additionally, the performance metrics of accuracy and loss were computed for diverse datasets.

The training and testing accuracy of the proposed model across various benchmarks ranges between 98.5% and 99%, with the RMSE error spanning from 0.001 to 0.004. This demonstrates consistent performance characteristics of the model when evaluated against real-time benchmarks, indicating its capability to forecast diverse attack categories beyond those specified. Additionally, sensitivity and specificity metrics have been computed for both benchmark and real-time datasets. Comparison of specificity and sensitivity across various Intrusion Detection System (IDS) datasets is shown in table 3.

Table 3 *Comparison of specificity and sensitivity across various Intrusion Detection System (IDS) datasets.*

Data Sets Description	Attack Types	Performance Metrics(%)	
		Sensitivity	Specificity
Real time datasets	DoS	98.6%	98%
	Botnets	98.7%	98.5%
	MIM	99%	99.4%
	Data Probing	98%	98.5%
	Spying	98.5%	98.0%
CIDDS-001,		98%	98.5%
UNSWNB15		98%	99%
NSL-KDD		98.45%	98.5%

The performance of the deep learning model consistently falls within the 98% to 99% range, demonstrating its adaptability to a wide array of datasets. Furthermore, the effectiveness of the hybrid learning intrusion detection system is contingent upon its integration with established algorithms, which play a significant role in IDS implementation.

Through rigorous evaluation against benchmarks, our proposed deep learning model consistently outshines others. In the realm of IoT security, the efficacy of an Intrusion Detection System (IDS) hinges on its ability to promptly predict attacks. Consequently, we've conducted an analysis of the response time exhibited by our proposed model in anticipating such incursions, juxtaposed with alternative learning models. Figure 4 indicates that the anticipated duration of prediction using the innovative GBLSTM model is under 2 seconds across various attack categories, contrasting with competing algorithms which require between 2 to 3

Table 4. *An assessment comparing various learning frameworks through benchmarking analyses***.**

Sl. No	Algorithm details	Benchmark Used	Performance Metrics		
			Accuracy	Sensitivity	Specificity
01	MLP	CIDDS-001	94.5	93.5	94
		UNSW NB15	95	95	96
		NSL-KDD	90	91	93
02	CART	CIDDS-001	95.3	96	94
		UNSW NB15	94	96.4	94.7
		NSL-KDD	95.0	95	96
03	ENSEMBLE	CIDDS-001	96.8	96.3	95
		UNSW NB15	96	95	96
		NSL-KDD	96.4	96.2	95
04	CNN-LSTM	CIDDS-001	96.5	96	97
		UNSW NB15	97.2	95.6	95
		NSL-KDD	96	95.8	96.2
05	BLSTM	CIDDS-001	96	96.5	96.7
		UNSW NB15	97.2	97	96
		NSL-KDD	97	96.3	96.4
06	PROPOSED MODEL	CIDDS-001	99.3	98.3	99
		UNSW NB15	99.1	98	98.89
		NSL-KDD	99.5	98.7	98.45

seconds for the same task. This showcases the Whale Optimizer's efficiency in terms of convergence time, consequently reducing testing and prediction durations when juxtaposed with alternative learning frameworks. Based on the data presented in Table 4, it is evident that the newly proposed WISL-TRS framework surpasses other established architectures in predicting attacks and implementing countermeasures within IoT networks. Table 5 elucidates this, showcasing how our optimized LSTM model surpasses alternative learning models across diverse benchmarks. This underscores the efficacy of our approach in crafting a robust, intelligent, dependable, and secure Intrusion Detection System (IDS). Several works done on IoT and reported in literature [48-55].

6. Conclusion

This paper introduces a novel optimized deep learning framework called GBLSTM. Extensive datasets were gathered from real-time scenarios and a Python API was developed to simulate various malicious attacks on networks. The performance of the proposed model was evaluated and compared against other learning models using different datasets, including real-time datasets, CIDDC-001, UMSN15, and KDD datasets. The results demonstrate that the proposed Whale Optimized LSTM outperforms other algorithms in terms of accuracy, sensitivity, and specificity in detecting malicious nodes and predicting network attacks. Furthermore, the paper discusses countermeasures to be implemented upon attack prediction. Experimental results indicate that the proposed model maintains consistent performance, while others experience slight degradation. Additionally, the average prediction time is calculated and compared with other algorithms. The simulation findings suggest that the proposed model achieves a better balance between prediction accuracy and response time, making it well-suited for developing secure, intelligent, and scalable IDS for IoT networks. This study highlights the importance of employing lightweight, optimized deep learning architectures to yield enhanced accuracy while minimizing prediction time. Moving forward, the integration of hybrid learning methodologies alongside feature optimization techniques will be imperative for crafting more proficient Intrusion Detection Systems within IoT networks.

Author Statements:

- **Data Availability**: The dataset supporting this study's findings is available from the corresponding author upon request.
- **Conflicts of Interest**: The authors declare no conflicts of interest related to this publication.
- **Funding Statement**: The author declares that no funding was received for this research and publication.
- **Author Contributions**: All authors declare equal contribution to this work.

References

[1] Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: The internet of things architecture, possible applications and key challenges. *In 10th International Conference on Frontiers of Information Technology (FIT)* (pp. 257-260). IEEE.

Table 5 *Evaluation contrasting the suggested GBLSTM-TRS with alternative IDS-recommendation systems is advised*

Slno	**Architecture Details**	**Features Comparision**					
		Intelligent	**Algorithm Used**	**Time Complexity**	**Prediction**	**Performance(Accuracy)**	**Scalable**
01	Fernandez-Gago et al.[40]	No	Bottom up Approaches	High	No	n/A	No
02	Saied et al. [41]	No	Context - Aware/Trust Management System	Very High	No	Low	No
03	Ko et al.[42]	No	Collaborative Filtering methods	Very High	No	Low	No
04	Chen et al.[43]	No	Evidence based Recommendation Systems	Very High	No	Low	No
05	Tormo et al.[44]	No	Fuzzy Logic based Systems	Medium	No	Medium	No
06	Mahmud et al.[45]	Yes	Artificial Neural Network based Recommendation systems	High	Yes	Medium	No
07	Asiri and Miri[46]	Yes	Probalistic Based Neural Networks(PNN) based recommendation systems	High	Yes	Medium	No
08	Al-Turjman [47]	No	Content based IDS	Very High	No	Low	No
09	GBLSTM-TRS	Yes	Deep learning Approach	Low	Yes	High	Yes

[2]Simon, T. (2017). Chapter seven: Critical infrastructure and the internet of things. *In Cyber Security in a Volatile World* (p. 93).

[3]E. Anthi, L. Williams and P. Burnap, (2018) Pulse: An adaptive intrusion detection for the Internet of Things. *Living in the Internet of Things: Cybersecurity of the IoT* - 2018, London, pp. 1-4, doi: 10.1049/cp.2018.0035.

[4] Cybersecurity executive: Medical devices a 'bulls-eye' for cyber-attacks. (2018, February 5). Retrieved from https://www.digitalhealth.net/2017/12/medical-device-functionality-vs-cybersecurity/

[5] Anthi, E., Javed, A., Rana, O., & Theodorakopoulos, G. (2017). Secure data sharing and analysis in cloud-based energy management systems. *In Cloud Infrastructures, Services, and IoT Systems for Smart Cities* (pp. 228-242). Springer.

[6] Cyber hackers can now harm human life through smart meters. (2018, February 5). Retrieved from https://smartgridawareness.org/2014/12/30/hackers-can-now-harm-human-life/

[7] Securing the internet of things: A proposed framework - cisco. (2018, July 13). Retrieved from https://www.cisco.com/c/en/us/about/security-center/secure-iot-proposed-framework.html

[8] Kalpana, P., Srilatha, P., Krishna, G. S., Alkhayyat, A., & Mazumder, D. (2024). Denial of Service (DoS) Attack Detection Using Feed Forward Neural Network in Cloud Environment. *In International Conference on Data Science and Network Security (ICDSNS)* (pp. 1-4). https://doi.org/10.1109/ICDSNS62112.2024.10691181

[9] Nabi, S. A., Kalpana, P., Chandra, N. S., Smitha, L., Naresh, K., Ezugwu, A. E., & Abualigah, L. (2024). Distributed private preserving learning based chaotic encryption framework for cognitive healthcare IoT systems. *Informatics in Medicine Unlocked*, 49, 101547. https://doi.org/10.1016/j.imu.2024.101547

[10] Kalpana, P., & Anandan, R. (2023). A capsule attention network for plant disease classification. *Traitement du Signal*, 40(5), 2051-2062. https://doi.org/10.18280/ts.400523

[11] Shanthamallu, U. S., Spanias, A., & Tepedelenlioglu, C. (2017). A brief survey of machine learning methods and their sensor and IoT applications. In 8th International Conference on Information, *Intelligence, Systems & Applications* (IISA). https://doi.org/10.1109/IISA.2017.8316459

[12] Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning. arXiv preprint arXiv:1801.06275.

[13] Doshi, R., Apthorpe, N., & Feamster, N. (2018). Machine learning ddos detection for consumer internet of things devices. arXiv preprint arXiv:1804.04159.

[14] Shukla, P. (2017). ML-IDS: A machine learning approach to detect wormhole attacks in internet of things. *In Intelligent Systems Conference (IntelliSys)* (pp. 234-240). IEEE.

[15] Verma, A., & Ranga, V. (2019). Machine Learning Based Intrusion Detection Systems for IoT Applications. *Wireless Personal Communications.* 111, 2287–2310 https://doi.org/10.1007/s11277-019-06986-8

[16] Misra, S., Krishna, P. V., Agarwal, H., Saxena, A., & Obaidat, M. S. (2011). A learning automata based solution for preventing distributed denial of service in Internet of Things. *In 4th International Conference on Cyber, Physical and Social Computing* (pp. 114-122).

[17] Kasinathan, P., Costamagna, G., Khaleel, H., Pastrone, C., & Spirito, M. A. (2013). Demo: An IDS framework for Internet of Things empowered by 6lowpan. *In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security* (pp. 1337-1340).

[18] Kalpana, P., Anandan, R., Hussien, A. G., et al. (2024). Plant disease recognition using residual convolutional enlightened Swin transformer networks. *Scientific Reports*, 14, 8660. https://doi.org/10.1038/s41598-024-56393-8

[19] Tama, B. A., & Rhee, K. H. (2019). An in-depth experimental study of anomaly detection using gradient boosted machine. *Neural Computing and Applications*, 31(4), 955-965.

[20] Primartha, R., & Tama, B. A. (2017). Anomaly detection using random forest: A performance revisited. *In International Conference on Data and Software Engineering (ICoDSE)* (pp. 1-6). IEEE.

[21] Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2019). A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment. *Information Sciences.* 513;386-396 https://doi.org/10.1016/j.ins.2019.10.069

[22] Krawczyk, B., Minku, L. L., Gama, J., Stefanowski, J., & Woźniak, M. (2017). Ensemble learning for data stream analysis: A survey. *Information Fusion*, 37, 132-156. https://doi.org/10.1016/j.inffus.2017.02.004

[23] Pongle, P., & Chavan, G. (2015). Real time intrusion and wormhole attack detection in internet of things. *International Journal of Computer Applications*, 121(9).

[24]Sherasiya, T., & Upadhyay, H. (2016). Intrusion Detection System for Internet of Things. *International Journal of Advance Research and Innovative Ideas in Education*, 2, 2344-2349.

[25] Kasinathan, P., Pastrone, C., Spirito, M. A., & Vinkovits, M. (2013). Denial-of-service detection in 6LoWPAN based internet of things. *In IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications* (WiMob) (pp. 600-607).

[26] Andrea, I., Chrysostomou, C., & Hadjichristofi, G. (2015). Internet of things: Security vulnerabilities and challenges. *In IEEE Symposium on Computers and Communication (ISCC)* (pp. 180-187).

[27] Kolias, C., Kambourakis, G., Stavrou, A., & Voas, J. (2017). DDoS in the IoT: Mirai and other botnets. *Computer*, 50(7), 80-84. doi: 10.1109/MC.2017.201.

[28] Xiao, L., Wan, X., Lu, X., Zhang, Y., & Wu, D. (2018). IoT security techniques based on machine learning. arXiv preprint arXiv:1801.06275.

[29] Pahl, M.-O., & Aubet, F.-X. (2018). All eyes on you: Distributed multi-dimensional IoT microservice anomaly detection. *In Proceedings of the 2018 Fourteenth International Conference on Network and Service Management (CNSM)*. Rome, Italy.

[30] Poyner, I., & Sherratt, R. (2018). Privacy and security of consumer IoT devices for the pervasive monitoring of vulnerable people. DOI:10.1049/CP.2018.0043

[31] Apthorpe, N., Reisman, D., & Feamster, N. (2016). A smart home is no castle: Privacy vulnerabilities of encrypted IoT traffic. https://doi.org/10.48550/arXiv.1705.06805

[32] Mukherjee, A., Chakraborty, N., & Das, B. K. (2017). Whale optimization algorithm: An implementation to design low-pass FIR filter. *In Innovations in Power and Advanced Computing Technologies (i-PACT).*

[33] Mukherjee, A., Chakraborty, N., & Das, B. K. (2017). Whale optimization algorithm: An implementation to design low-pass FIR filter. *In Innovations in Power and Advanced Computing Technologies (i-PACT).*

[34] Supreetha, B. S., Shenoy, N., & Nayak, P. (2020). Lion Algorithm-Optimized Long Short-Term Memory Network for Groundwater Level Forecasting in Udupi District, India. *Applied Computational Intelligence and Soft Computing*, 2020, Article ID 8685724. https://doi.org/10.1155/2020/8685724

[35] CIDDS-001 dataset. (2017). Retrieved November 3, 2019, from https://www.hs-coburg.de/forschungkooperation/forschungsprojekte-fentlich/ingenieurwissenschaften/cidds-coburg-intrusion-detectiondata-sets.html

[36] NSL-KDD dataset. (2017). Retrieved November 3, 2019, from http://nsl.cs.unb.ca/nsl-kdd/

[37] UNSW-NB15 dataset. (2017). Retrieved November 3, 2019, from https://www.unsw.adfa.edu.au/australian-centre-for-cyber-security/cybersecurity/ADFA-NB15-Datasets/

[38] Roy, B., & Cheung, H. (2018). A Deep Learning Approach for Intrusion Detection in Internet of

Things using Bi-Directional Long Short-Term Memory Recurrent Neural Networks. *In International Telecommunication Conference. IEEE*.

[39] Hassan, M. M., Gumaei, A., Alsanad, A., Alrubaian, M., & Fortino, G. (2020). A Hybrid Deep Learning Model for Efficient Intrusion Detection in Big Data Environment. *Information Sciences* 513;386-396 https://doi.org/10.1016/j.ins.2019.10.069

[40]Fernandez-Gago, C., Moyano, F., & Lopez, J. (2017). Modelling trust dynamics in the Internet of Things. *Information Sciences*, 396, 72-82. https://doi.org/10.1016/j.ins.2017.02.039

[41]Chen, R., Guo, J., & Bao, F. (2016). Trust management for SOA-based IoT and its application to service composition. *IEEE Transactions on Services Computing*, 9(3), 482-495. DOI: 10.1109/TSC.2014.2365797

[42]Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M. (2013). Trust management system design for the Internet of Things: A context-aware and multi-service approach. *Computer Security*, 39, 351-365. https://doi.org/10.1016/j.cose.2013.09.001

[43] Ko, H. G., Ko, I. Y., & Lee, D. (2018). Multi-criteria matrix localization and integration for personalized collaborative filtering in IoT environments. *Multimedia Tools and Applications*, 77(4), 4697-4730.

[44] Kalpana, P., Malleboina, K., Nikhitha, M., Saikiran, P., & Kumar, S. N. (2024). Predicting Cyberbullying on Social Media in the Big Data Era Using Machine Learning Algorithm. *In International Conference on Data Science and Network Security (ICDSNS) (pp. 1-7).* https://doi.org/10.1109/ICDSNS62112.2024.10691297

[45]Asiri, S., & Miri, A. (2016). An IoT trust and reputation model based on recommender systems. *In 14th Annual Conference on Privacy, Security and Trust (PST)* (pp. 561-568). IEEE.

[46]Mahmud, M., Kaiser, M. S., Rahman, M. M., Rahman, M. A., Shabut, A., Al-Mamun, S., & Hussain, A. (2018). A brain-inspired trust management model to assure security in a cloud based IoT framework for neuroscience applications. *Cognitive Computing*, 10(5), 864-873. https://doi.org/10.48550/arXiv.1801.03984

[47]Zeinali, Y., & Story, B. A. (2017). Competitive probabilistic neural network. *Integrated Computer-Aided Engineering*, 24(2), 105-118. https://doi.org/10.3233/ICA-170540

[48]M, P., B, J., B, B., G, S., & S, P. (2024). Energy-efficient and location-aware IoT and WSN-based precision agricultural frameworks. *International Journal of Computational and Experimental Science and Engineering*, 10(4);585-591. https://doi.org/10.22399/ijcesen.480

[49]S, P., & A, P. (2024). Secured Fog-Body-Torrent : A Hybrid Symmetric Cryptography with Multi-layer Feed Forward Networks Tuned Chaotic Maps for Physiological Data Transmission in Fog-BAN Environment. *International Journal of Computational and Experimental Science and Engineering*, 10(4);671-681. https://doi.org/10.22399/ijcesen.490

[50]D, jayasutha. (2024). Remote Monitoring and Early Detection of Labor Progress Using IoT-Enabled Smart Health Systems for Rural Healthcare Accessibility. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1149-1157. https://doi.org/10.22399/ijcesen.672

[51]S, P. S., N. R., W. B., R, R. K., & S, K. (2024). Performance Evaluation of Predicting IoT Malicious Nodes Using Machine Learning Classification Algorithms. *International Journal of Computational and Experimental Science and Engineering*, 10(3);341-349. https://doi.org/10.22399/ijcesen.395

[52]Achuthankutty, S., M, P., K, D., P, K., & R, prathipa. (2024). Deep Learning Empowered Water Quality Assessment: Leveraging IoT Sensor Data with LSTM Models and Interpretability Techniques. *International Journal of Computational and Experimental Science and Engineering*, 10(4);731-743. https://doi.org/10.22399/ijcesen.512

[53]Alkhatib, A., Albdor , L., Fayyad, S., & Ali, H. (2024). Blockchain-Enhanced Multi-Factor Authentication for Securing IoT Children's Toys: Securing IoT Children's Toys. *International Journal of Computational and Experimental Science and Engineering*, 10(4);1041-1049. https://doi.org/10.22399/ijcesen.417

[54]Radhi, M., & Tahseen, I. (2024). An Enhancement for Wireless Body Area Network Using Adaptive Algorithms. *International Journal of Computational and Experimental Science and Engineering*, 10(3);388-396. https://doi.org/10.22399/ijcesen.409

[55]Nagalapuram, J., & S. Samundeeswari. (2024). Genetic-Based Neural Network for Enhanced Soil Texture Analysis: Integrating Soil Sensor Data for Optimized Agricultural Management. *International Journal of Computational and Experimental Science and Engineering*, 10(4);962-970. https://doi.org/10.22399/ijcesen.572