International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Network Selection over Heterogeneous Health System Using Deep Reinforcement Learning

Kumaran K [1], Sivasakthi P [2], Pandiyan G [3], Saranya G [4], Kirubakaran S[5], Nithish V[6]

Submitted: 25/01/2024 **Revised:** 03/03/2024 **Accepted:** 11/03/2024

Abstract: The proposed method involves using a combination of techniques such as energy-efficient routing, heterogeneous mobile nodes, region-based grid formation, GWO optimization, LSTM prediction, ESPIRIT estimation, and active zones to minimize the overall energy consumption. It works by first randomly deploying mobile nodes throughout the experimental region. Next, the area is split up into grids according to certain parameters such as remaining energy levels and distances between nodes. Every grid has a base node that is chosen using GWO optimization. Next, a deep learning-based machine learning algorithm known as LSTM is employed to predict the future direction of movement of each mobile node within the grid. Finally, the sender node selects the closest node in the active region to transmit data to. Sensor networks are designed to facilitate efficient information sharing by allowing for the distribution of resources and information across a decentralized network. Ad-hoc networks, in particular, have gained popularity due to their ability to operate in dynamic environments where traditional wired connections may be impractical or impossible. In order to effectively communicate within these networks, various protocols have been developed to manage network traffic and ensure reliable delivery of messages. One example of such a protocol is the Dynamic Source Routing (DSR) algorithm, which uses a combination of flooding and source routing techniques to efficiently route packets through the network.

Keywords: *Heterogeneous mobile nodes, Deep Reinforcement Learning, Random Forest (RF), Decision Tree (DT) Energy efficient routing protocol (EERPS), Grey Wolf Optimizer (GWO)*

1. Introduction

This chapter provides a comprehensive exploration of sensor networks and their inherent characteristics, encompassing various types of ad-hoc networks and examining different sensor network architectures applicable in real-time scenarios.[1] It delves into the fundamental components of sensor nodes, They usually consist of a power unit, a transceiver unit, a processor unit, and a sensor unit. [2] Based on the specific applications, extra parts like a mobilizer, a power generator, and a location detecting system may also be integrated into the sensor node architecture.[3] Sensing units play a crucial role and commonly comprise sensors and analog- to-digital converters (ADCs) responsible for converting analogue communications from sensors into digital ones for subsequent processing.[4] The apparatus for processing, often equipped with a tiny storage container , orchestrates procedures for sensor node collaboration in executing assigned sensing tasks. Meanwhile, the transceiver unit facilitates seamless connectivity of the networking node.

The significance of the power units cannot be overstated, with units that scavenge power like solar cells often supplementing it. Additional subunits may vary based on specific application requirements, with many sensor network routing techniques and sensing tasks relying heavily on accurate location knowledge.Hence, the inclusion of a location finding system in sensor nodes is common place.[6] Furthermore, a mobilizer may be deemed necessary for relocating sensor nodes to fulfill specific tasks, underscoring the dynamic nature of sensor networks.[7] It is imperative that all these subunits are compact and lightweight, sometimes even fitting into a module no bigger than a matchbox, to facilitate suspension in the air. Sensor nodes face stringent constraints, including minimal power consumption, operation in high volumetric densities, low production

1 Department of Information Technology, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
Mail ID:kumaran.me.cse@gmail.com
2 Department of Information Technology, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
Mail ID:sivasakthi.p@eec.srmrmp.edu.in
3 Assistant Professor, Department of Artificial Intelligence and Data Science, Saveetha engineering college, Chennai, Tamil Nadu, India.
Mail ID:pandiyangingee@gmail.com
4 Department of Computer Science and Engineering, SRM Institute of Science and Technology, Chennai, Tamil Nadu, India.
Mail ID:ransa.btech@gmail.com
5 Department of Information Technology, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
Mail ID:kirubakaran1809@gmail.com
6 Department of Information Technology, Easwari Engineering College, Ramapuram, Chennai, Tamil Nadu, India.
Mail ID:nithishvenkate@gmail.com

cost, disposability, autonomy, unattended operation, and adaptability to the environment.[8] These constraints underscore the need for innovative design approaches and efficient utilization of resources in sensor network deployment and management.

1.1.Objective

The project aims to achieve energy-efficient routing, mobile nodes are strategically placed within an experimental zone, ensuring randomness to emulate real-world scenarios. Following this, a region-based grid formation strategy is employed to organize the network topology effectively. The Gray Wolf Optimization (GWO) technique is harnessed to select optimal base nodes within each grid, leveraging the inherent strengths of the algorithm in optimizing complex systems.

To further augment the system's capabilities, Long Short-Term Memory (LSTM) networks are integrated into the methodology. LSTM networks are utilized to predict the Direction of Movement (DOM) for each node within the grid, enhancing the adaptability and predictive power of the system in dynamic environments. Additionally, the Estimation of Signal Parameters via Rotational Invariance Technique (ESPRIT) is adopted to precisely determine active zones from sender nodes towards base nodes, facilitating efficient routing decisions.

This comprehensive approach not only ensures energy efficiency but also enhances overall performance by incorporating advanced prediction, optimization, and signal estimation techniques. By integrating these methodologies seamlessly, the proposed method showcases a sophisticated solution tailored to address the challenges of routing in dynamic wireless networks, promising improved efficiency, reliability, and adaptability in real-world deployment scenarios.

2. Existing System

In the existing system, the detection and mitigation of Distributed Denial of Service (DDoS) attacks constitute a crucial aspect of network security. To combat these threats effectively, a multi-step approach is employed, wherein selected features indicative of anomalous behavior are extracted and subsequently passed through various classifiers. These classifiers encompass a diverse range of algorithms, including Support Vector Machine (SVM), Decision Tree, Naïve Bayes, and Multilayer Perceptron (MLP), each offering unique advantages in identifying the type and nature of attacks.

The experimental study leverages publicly available datasets such as KDD Cup 99, a renowned benchmark dataset in the field of intrusion detection, to evaluate the efficacy of the proposed methodology. Through rigorous simulations and analysis, it becomes evident that the Generic Online Intrusion Detection System (GOIDS) paired with the decision tree classifier exhibits superior performance metrics, particularly in terms of detection accuracy and false-positive rates. This robust combination proves to be highly adept at discerning malicious activities amidst legitimate network traffic, thereby enhancing overall security posture.

Furthermore, the incorporation of denoising techniques as feature extractors emerges as a noteworthy strategy to bolster system performance, especially in environments characterized by high levels of noise or signal interference. By employing denoising algorithms, the system can effectively filter out extraneous noise, thus improving the fidelity and reliability of extracted features. Consequently, this refinement contributes significantly to the accuracy and efficacy of the intrusion detection framework, enabling it to discern subtle patterns indicative of malicious behavior amidst complex network dynamics.

The utilization of feature extraction methodologies, particularly denoising techniques, underscores the importance of preprocessing steps in enhancing the robustness and effectiveness of intrusion detection systems. By systematically filtering and refining input data, the system can extract salient features that are crucial for accurate classification and decision-making. This proactive approach not only mitigates the impact of noise and extraneous variables but also enables the system to operate with heightened precision and efficiency in dynamic network environments.

3. Literature Survey

[1] Adversarial machine learning applied to intrusion and malware scenarios: A systematic review, (2020), N. Martins, J. M. Cruz, T. Cruz, and P. H. Abreu, In this chapter, we describe various service and deployment models of cloud computing and identify major challenges. In particular, we discuss three critical challenges: regulatory, security and privacy issues in cloud computing. Some solutions to mitigate these challenges are also proposed along with a brief presentation on the future trends in cloud computing deployment.

[2] Increasing the performance of machine learning-based IDSs on an imbalanced and up-to-date dataset, (2022), G. Karatas, O. Demir, and O. K. Sahingoz, This paper surveys the state-of-the-art in programmable networks with an emphasis on SDN. We provide a historic perspective of programmable networks from early ideas to recent developments. Then we present the SDN architecture and the Open Flow standard in particular, discuss current alternatives for implementation and testing of SDN-based protocols and services, examine current

and future SDN applications, and explore promising research directions based on the SDN paradigm.

[3] BAT: Deep learning methods on network intrusion detection using NSL-KDD dataset, 2022, T. Su, H. Sun, J. Zhu, S. Wang, and Y. Li, It is an attack designed to render a computer or network incapable of providing normal services. DDoS attack uses many compromised intermediate systems, known as botnets which are remotely controlled by an attacker to launch these attacks. DDOS attack basically results in the situation where an entity cannot perform an action for which it is authenticated.

[4] Network intrusion detection based on PSO-xgboost model, 2022, H. Jiang, Z. He, G. Ye, and H. Zhang, The growing use of this technology has been driven by a desire to increase utilization of resources through server consolidation. Virtualization has also made the dream of such utility computing platforms as cloud computing a reality. Today, virtualization technologies can be found in almost every data centre.

[5] Similarity based feature transformation for network

anomaly detection, 2023, Kumaran, K., & Sasikala, E., Our experiments show that the verification of our scheme requires a small, constant amount of overhead, which minimizes communication complexity. In this work, we focus on the construction of PDP scheme for hybrid clouds, supporting privacy protection and dynamic scalability.

[6] On peak versus average interference power constraints for protecting primary users in cognitive radio networks, 2021, R. Zhang, In proposed work, to regulate the average interference power (AIP) over all the fading states, while

the other is to limit the peak interference power (PIP) at each fading state.

[7] Fundamental limits of spectrum-sharing in fading environments, 2022, A. Ghasemi and E. S. Sousa, In proposed system, the capacity gains offered by this dynamic spectrum sharing approach when channels vary due to fading.

[8] Power, sensing time, and throughput tradeoffs in cognitive radio systems: A cross-layer approach, 2018, K. Hamdi and K. B. Letaief, A cross-layer optimization problem to design the sensing time and optimize the transmit power in order to maximize the cognitive system throughput while keeping the interference to the primary user under a threshold constraint

[9] Optimal wideband spectrum sensing framework for cognitive radio systems, 2022, P. Paysarvi-Hoseini and N. C. Beaulieu, Provided secondary transmission opportunities over multiple non overlapping narrowband channels is presented. An efficient iterative algorithm which solves the optimization problem with much lower complexity.

[10] Sensing-based spectrum sharing in cognitive radio networks, 2022, X. Kang etal, This spectrum sharing model can achieve a higher capacity of SU link and improve the spectrum utilization. Also achieved the ergodic capacity of the SU link considering both transmit and interference power constraints

[11] Energy detection based cooperative spectrum sensing in cognitive radio networks, 2021, S. Atapattu, C. Tellambura,and H. Jiang, the exact detection and false alarm probabilities are derived under the generalized "k-out of-n fusion rule at the fusion center with consideration of errors in the reporting channel due to fading.

4. Proposed System

The proposed Wireless Sensor Network (WSN) model serves as a robust framework designed [12-16] explicitly for the detection of network intrusions, operating by effectively categorizing all packet traffic within the network into either benign or malicious classes. Central to this approach is the implementation of an energy-efficient routing algorithm, meticulously crafted to optimize resource utilization while ensuring reliable and timely data transmission in the network.

Through rigorous experimentation and analysis, the efficacy of the proposed energy-efficient routing algorithm is demonstrated across various performance metrics. Notably, the algorithm showcases remarkable achievements, including a commendable 94% packet delivery ratio, a mere 7% packet loss rate, an average residual energy consumption of 9.5 J, and a throughput of 3.4 Mbps. These performance indicators underscore the algorithm's ability to facilitate efficient data transfer within the Internet of Wireless Sensor Networks (IWSN), effectively balancing the demands of data transmission with the constraints of energy availability.

In addition to its routing capabilities, the proposed approach integrates advanced classification models to further enhance the system's intrusion detection capabilities. Leveraging techniques such as Random Forest [5], Decision Tree, and XGBoost, the model employs sophisticated machine learning algorithms to predict and classify anomalies within the network accurately. By training on a comprehensive testing dataset, the anomaly detection model distinguishes between normal network behavior and malicious attacks, thereby fortifying the network's security posture and mitigating potential threats effectively.

Furthermore, the holistic approach adopted in the performance analysis ensures a comprehensive evaluation of the system's effectiveness. By scrutinizing various

metrics ranging from packet delivery ratio to energy consumption [17-18], the analysis provides valuable insights into the system's overall efficiency and resilience. Moreover, the integration of advanced classification models not only enhances the accuracy of anomaly detection but also contributes to a more nuanced understanding of network behavior and potential vulnerabilities.

Overall, the proposed methodology represents a significant advancement in the field of wireless sensor networks and intrusion detection systems. By combining energy-efficient routing algorithms with sophisticated machine learning techniques, the approach offers a robust and scalable solution for detecting and mitigating network intrusions in real-time. Through meticulous experimentation and performance analysis, the proposed framework demonstrates its efficacy in safeguarding IWSNs against a wide range of security threats, thereby ensuring the integrity and reliability of data transmission in critical environments.

5. Software Components

5.1 Matplotlib:

Matplotlib is a powerful Python library extensively used for creating high-quality visualizations and plots. It provides a wide array of functionalities to generate various types of plots, ranging from simple line graphs to complex 3D plots. One of its key strengths lies in its flexibility and customization options, allowing users to tailor their visualizations to specific requirements. At its core, Matplotlib operates on the premise of a hierarchical structure where figures, axes, and plots are the fundamental building blocks. A "figure" serves as the overarching container that encompasses one or more "axes," which in turn house the actual plots or visualizations. This hierarchical structure enables users to create multiple subplots within a single figure, facilitating the comparison of different datasets or visualizations.

5.2 TensorFlow:

Tensorflow lite is a deep learning framework and is based on the tensorflow framework. It is used to reduce the size of a normally huge tensorflow model so that it can be used in modular devices such as mobile phones. We can use tensorflow lite to access the model with android studio. It is International Research Journal of Engineering and Technology a complex procedure and is used to access a minimal reduction algorithm of the model.

5.3 Tkinter:

Tkinter is a Python library used for creating graphical user interfaces (GUIs) with ease. It provides a set of tools and widgets that enable developers to build interactive applications with graphical elements such as buttons, menus, text boxes, and more.At its core, Tkinter is based on the Tk GUI toolkit, which originated as part of the Tcl scripting language. However, Tkinter seamlessly integrates with Python, making it accessible and convenient for developers to create cross-platform desktop applications.

5.2.1 Flow of System:

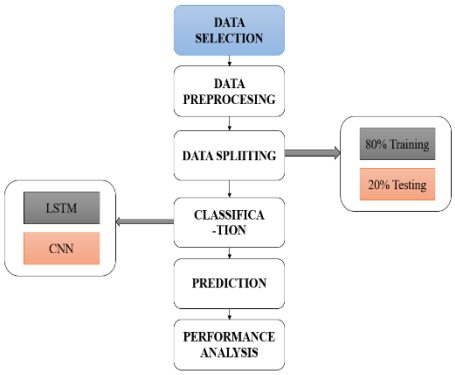

Fig 1. Flow Diagram

5.2.2 Block Diagram:

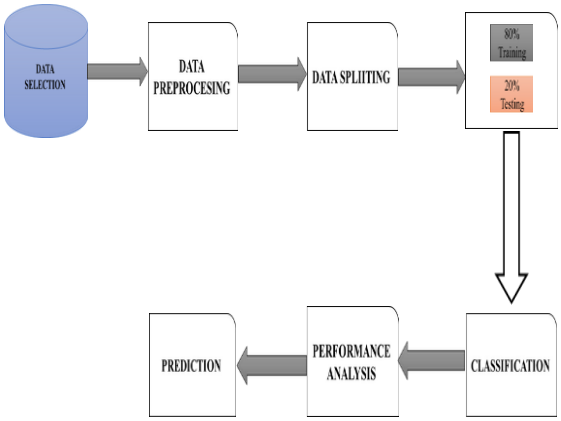

Fig 2. Block Diagram

6. Implementation Methodology

6.1 Data Selection:

In the initial phase, the script imports the dataset labeled as 'VanetDataset.csv', presumably containing pertinent information crucial for the intrusion detection system within Industrial Wireless Sensor Networks (IWSN). Subsequently, it performs an assessment by showcasing the dimensions of the dataset, offering insights into its size and structure. Additionally, to gain a deeper understanding of the interdependencies among variables, the script employs a visualization technique known as a correlation matrix heatmap. This heatmap visually represents the correlations between different variables, enabling the detection of patterns or associations that may influence the intrusion detection process. By utilizing these techniques,

the script sets the stage for further analysis and processing, laying the foundation for subsequent steps in the intrusion detection system development.

TABLE I Sample of input data

Input	Duration	Protocol	Length	Label
51690	0.01339	ICMP	92	Attack
59819	0.04481	ICMP	92	Attack
59584	0.02519	AODV	54	Attack

6.2 Data Preprocessing:

In the preprocessing module, handling missing values is a critical step to ensure the integrity and reliability of the dataset. First, the code checks for missing values in the DataFrame using the isnull() method, which returns a DataFrame of boolean values indicating whether each element is NaN (Not a Number) or None. The sum() function then calculates the total number of missing values for each column by summing the boolean values.

$$\text{Missing\_Values} = \sum_{i=1}^{n} \text{isnull}(x_i)$$

After identifying missing values, they are filled using the fillna() method, where the NaN values are replaced with a specified value, in this case, zero. Filling missing values with zero is a common approach, especially for numerical data, as it allows for the preservation of the dataset's structure and facilitates further analysis without drastically altering the distribution of the data.

DataFramefilled=DataFrameoriginal.fillna(0)

By handling missing values in this manner, the preprocessing module ensures that the dataset is ready for subsequent analysis and model training without being affected by incomplete or inconsistent data.

6.3 Feature Selection:

The Particle Swarm Optimization (PSO) algorithm, employed in the "Feature Selection" module, functions by iteratively exploring the solution space to identify optimal features for classification tasks. Inspired by the collective behavior of birds or fish, PSO adjusts the position of particles based on both their individual best position and the overall best position found within the swarm. The algorithm can be outlined mathematically as follows :

1. Set the particle's initial placements and velocities at randomly.
2. Assess the fitness of each particle according to the objective function.
3. Update the particle's velocity and position using the following equations:

Velocity: $v_i(t+1) = w \cdot c_i(t) + c1 \cdot r1 \cdot (pbesti - x_i(t)) + c2. \quad r2 \cdot (gbest - x_i(t))$ Position: $x_i(t+1) = x_i(t) + v_i(t+1)$

Where:

- The particle's velocity at time t is given by vi(t).
- Particle I's position at time t is given xi(t).
- Particle's optimal position, or pbesti, i.
- gbest is the optimal location that a particle inside the swarm finds itself in.
- W stands for inertia weight, C1 for cognitive coefficient, and C2 for social coefficient.
- Random numbers between 0 and 1 make up r1 and r2

7. Architecture

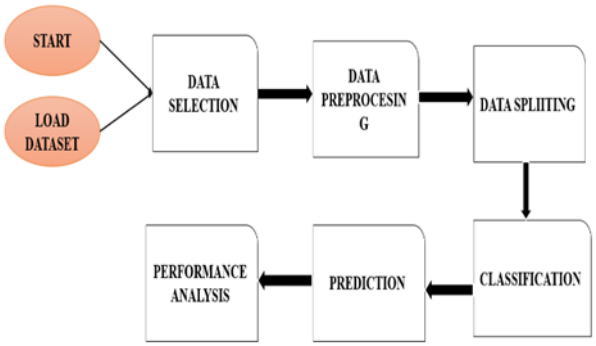

Fig 5. Architecture

8. Acquired Results

It begins with importing necessary libraries such as NumPy, Pandas, Matplotlib, scikit-learn, and others. Then, it loads a dataset ('VanetDataset.csv') and performs exploratory data analysis (EDA) by displaying basic information about the dataset and visualizing the correlation matrix.Next, it handles missing values in the dataset by filling them with zeros. Following this, label encoding is applied to convert categorical variables into numerical format for further processing.

Feature selection is performed using Particle Bee Colony Swarm Algorithm (PSO), and the selected features are scaled using MinMaxScaler. The dataset is then split into training and testing sets.A neural network model is built using TensorFlow's Keras API for classification tasks. Additionally, a hybrid ensemble model consisting of

Adaboost and Random Forest classifiers is created and trained on the training data.

The trained models are then evaluated on the testing data using various metrics such as confusion matrix, classification report, and accuracy score. Visualizations like heatmap and ROC curve are also generated to analyze model performance.Finally, a simple GUI interface using tkinter and easygui libraries allows users to input data related to the ETXPC-RPL Routing algorithm in IoT Network and obtain predictions regarding whether it's an attack or non-attack scenario.

8.1 Correlation Matrix:

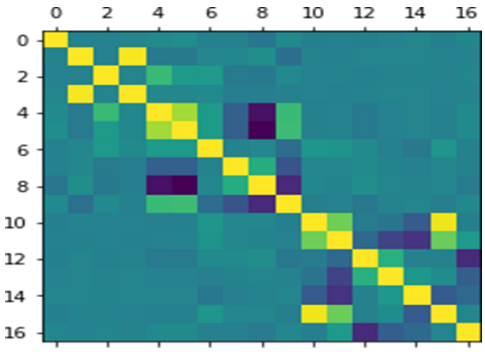

Fig. 6. Correlation Matrix

The correlation matrix shows the correlation coefficients between multiple variable . It helps to understand how the pairs ofstrongly variables are related. The values in the matrix range from -1 to 1 , (-1) Perfect negative correlation when the one variable goes upand the other goes down by the same proportion , (0) No correlation that is no connection b/w the variables, (1)Perfect positive correlation that is both the variables move in the same direction together and the diagonal line from top-left to bottom-right has the value of 1, as eachvariable perfectly correlates with itself. Look for cells with values close to 1 or -1. These signify strong positive or negative relationships. Cells with values close to 0 suggest little to no relationship between those variables.

TABLE II Correlation matric

	Var. A	Var. B	Var. C
Var. A	1	0.8	-0.2
Var. B	0.85	1	0.05
Var. C	-0.2	0.05	1

Strong Positive Correlation: Variable A and Variable B are highly correlated (0.85). As the value of A increases, B is likely to increase as well.

Weak Correlation: Variable B and Variable C have almost no relationship (0.05).

Slight Negative Correlation: Variable A and Variable C have a slight negative correlation (-0.2).

9. Output:

9.1 confusion matrix

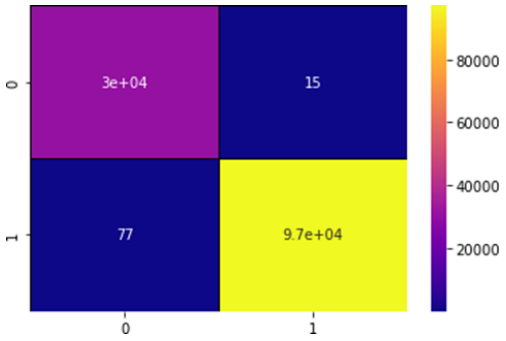

Fig. 7. Confusion Matrix

The Rows: reflect the data points' true classifications in the test set. Row 1: Represents the class of non-attack data points. Row 2: Represents the class of attack data points. Columns : indicate the data points' anticipated classifications in the test set. Column 1: Represents data points predicted as non-attack. Column 2: Represents data points predicted as attack. Values within the matrix: represent the number of data points that fall into each combination of actual and predicted classes. 30,304: Out of the actual non-attack data points (Row 1), 30,304 were correctly predicted as non-attack (Column 1). 11: Out of the actual non- attack data points, 11 were incorrectly predicted as attack (Column 2). These are False Positives (Type I Error), meaning the model mistakenly classified non-attack instances as attacks. 76: Out of the actual attack data points (Row 2), 76 were incorrectly predicted as non-attack (Column 1). These are False Negatives (Type II Error), meaning the model missed actual attacks. 97,182: Out of the actual attack data points, 97,182 were correctly predicted as attack (Column 2)

9.2 Results:

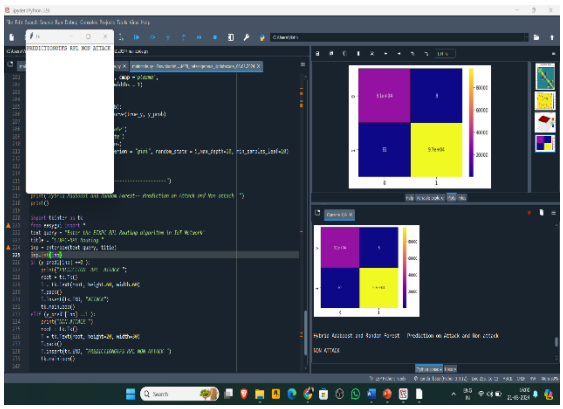

Fig 8. Final Prediction

10. Performance Metrics:

The performance metrics for evaluating the efficiency of the proposed system are defined as follows:

- **True Positive (TP):** Legs correctly categorized as positive.
- **False Positive (FP):** Legs incorrectly categorized as positive.
- **False Negative (FN):** Legs correctly categorized as negative but identified as positive.
- **True Negative (TN):** Legs correctly categorized as negative.

Accuracy: A computation metric reflecting the system's error, calculated as the difference between potential and actual outcomes. Low accuracy arises when the machine consistently evaluates input variables with the same procedure, yielding consistent but incorrect results. The ratio of correct outcomes to the total is known as accuracy.

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

Precision:

It is a measure of random error in algebraic terms.

$$Precision = \frac{TP}{TP+FP}$$

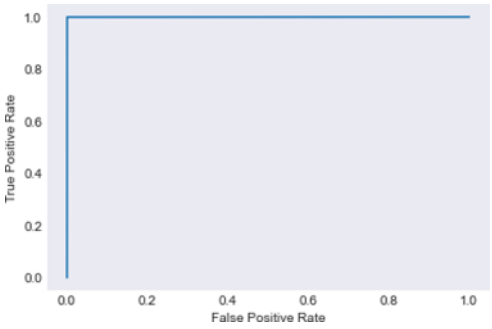

Fig. 9. Receiver Operating Characteristic curve graph

10.1 Dead Nodes Existing & Proposed:

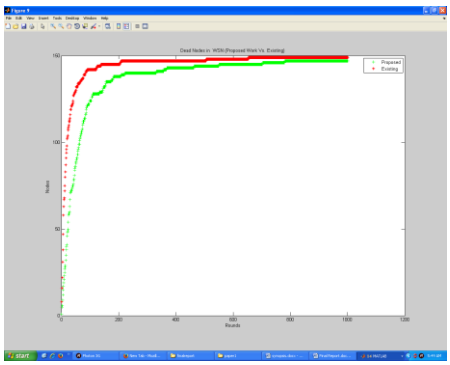

Fig 10. Dead Node Analysis

Here in fig.10 is showing the analysis on the existing and proposed approach absed on the dead nodes. The green li here shows the proposed approach and red line is showing the existing approach. The figure shows that the number of dead nodes after 1000 rounds is high in case of existing approach. The frequency of dead node conversion throughout the communication is high in existing approach, which shows that the presented work has improved the communication.

10.2 Alive Nodes Existing & Proposed:

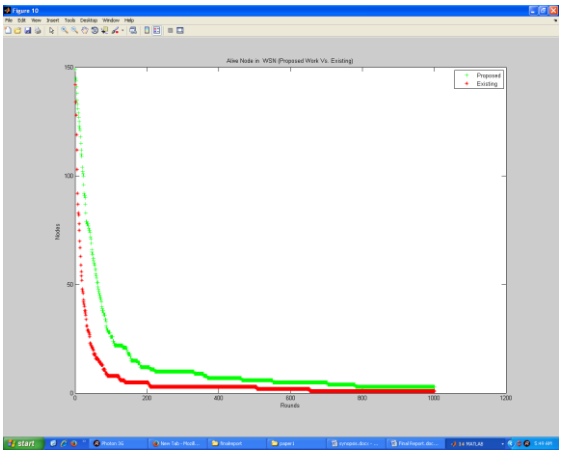

Fig 11. Alive Node Analysis

Here fig.11 is showing the analysis on the existing and proposed approach based on the alive nodes. The green here shows the proposed approach and red line is showing the existing approach. The figure shows that the number of alive nodes after 1000 rounds is less in case of existing approach. The frequency of dead node conversion throughout the communication is high in existing approach, which shows that the presented work has improved the communication. The figure shows that in case of proposed approach after 1000 rounds there are some alive nodes left.

10.3 Network Analysis Existing Vs Proposed:

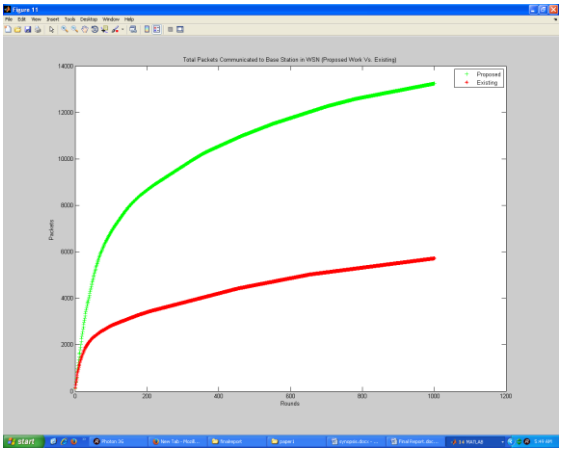

Fig 12. Network Communication Analysis

Here figure 4.12 is showing the analysis on the existing and proposed approach based on the network

communication. The green here shows the proposed approach and red line is showing the existing approach. The figure shows that the communication in case of proposed approach is much higher than existing approach.

11. Conclusion:

This research paper's objective is to introduces a new energy- efficient routing approach tailored for Industrial Wireless Sensor Networks (IWSNs), with the overarching the intention of increasing network lifetime and conserving energy resources. The proposed methodology encompasses a multi-faceted strategy that includes strategic node deployment, optimized base node selection, deep learning-based energy consumption prediction, and advanced routing algorithms. Through a series of systematic experiments and comprehensive evaluations using diverse performance metrics, the Energy-Efficient Routing Protocol (EERP) showcased superior performance when compared to existing approaches. The foundation of the proposed approach lies in the strategic deployment of heterogeneous mobile nodes within the industrial environment. By ensuring randomness and diversity in node placement, the network architecture is inherently conducive to energy-efficient routing. Moreover, the division of the deployment region into square-shaped grids facilitates organized resource management and systematic node organization, laying the groundwork for efficient routing strategies.

One key aspect of the proposed methodology is the meticulous selection of base nodes within each grid. This selection process takes into account various factors such as node proximity and residual energy levels, aiming to minimize energy consumption while maximizing network coverage. The integration of the COOT optimization algorithm, augmented with local search techniques, further enhances the efficiency of base node selection, resulting in optimized energy usage and routing efficiency.

To anticipate future node behavior and energy consumption patterns, an algorithm based deep learning approach utilizing (LSTM) networks is employed. By leveraging historical data, the LSTM networks predict the future Discrete Offload Metric (DOM) of each mobile node within the grid. This predictive capability enables the routing algorithm to dynamically adapt to evolving network conditions and energy demands, thereby optimizing resource allocation and energy consumption.

the combination of systematic node deployment, optimized base node selection, deep learning-based prediction, and advanced routing algorithms represents a significant advancement in energy- efficient routing for IWSNs. The proposed EERP not only extends network longevity but also contributes to energy conservation and improved network performance, making it a compelling choice for deployment in industrial settings.

References:

[1] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, and R. Friedman, ‘‘Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor,’’ in Proc. 11th IEEE Int. Symp. Wearable Comput., Oct. 2007, pp. 37–40.

[2] F. Lau, C. Kuziemsky, M. Price, and J. Gardner, ‘‘A review on systematic reviews of health information system studies,’’ J. Amer. Med. Inform. Assoc., vol. 17, no. 6, pp. 637–645, Nov. 2010.

[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, ‘‘Fog computing and its role in the Internet of Things,’’ in Proc. 1st Ed. MCC Workshop Mobile Cloud Comput. (MCC), pp. 13–16, 2012.

[4] R. Cook, ‘‘Exploring the benefits and challenges of telehealth,’’ Nursing times, vol. 108, no. 24, pp. 16–17, 2012.

[5] Kumaran, K., & Sasikala, E. (2023). An efficient task offloading and resource allocation using dynamic arithmetic optimized double deep Q-network in cloud edge platform. Peer-to-Peer Networking and Applications, 16(2), 958-979.

[6] S. M. R. Islam, D. Kwak, M. Humaun Kabir, M. Hossain, and K.-S. Kwak, ‘‘The Internet of Things for health care: A comprehensive survey,’’ IEEE Access, vol. 3, pp. 678–708, 2015.

[7] M. S. Mahmud, H. Wang, A. M. Esfar-E-Alam, and H. Fang, ‘‘A wireless health monitoring system using mobile phone accessories,’’ IEEE Internet Things J., vol. 4, no. 6, pp. 2009–2018, Dec. 2017.

[8] C. Crema, A. Depari, A. Flammini, E. Sisinni, T. Haslwanter, and S. Salzmann, ‘‘IMU-based solution for automatic detection and classification of exercises in the fitness scenario,’’ in Proc. IEEE Sensors Appl. Symp. (SAS), Mar. 2017, pp. 1–6.

[9] M. Bhatia and S. K. Sood, ‘‘A comprehensive health assessment framework to facilitate IoT-assisted smart workouts: A predictive healthcare perspective,’’ Comput. Ind., vols. 92–93, pp. 50–66, Nov. 2017

[10] Emre Oner Tartan and Cebrail Ciflikli, “An Android Application for Geolocation Based Health Monitoring, Consultancy and Alarm System”, IEEE International Conference on Computer Software & Applications, DOI 10.1109/COMPSAC.2018.10254, pp. 341-344, 2018

[11] W. R. Thompson, ‘‘Worldwide survey of fitness trends for 2019,’’ ACSM’S Health Fitness J., vol. 22, no. 6, pp.10–17, 2018.

[12] C. Shen, B.-J. Ho, and M. Srivastava, ‘‘MiLift:

Efficient smartwatch-based workout tracking using automatic segmentation,'' IEEE Trans. Mobile Comput., vol. 17, no. 7, pp. 1609–1622, Jul. 2018.
[13] ZephyrT Performance Systems | Performance Monitoring Technology. Accessed: Apr. 12, 2020.
[14] Afzaal Hussain, Kashif Zafar and Abdul Rauf Baig, "Fog-Centric IoT Based Framework for Healthcare Monitoring, Management and Early Warning System" IEEE Internet Things, pp. 74168- 74179, Apr. 2021.
[15] "HeDI: Healthcare Device Interoperability for IoT-Based e-Health Nidhi Pathak, Sudip Misra, Anandarup Mukherjee and Neeraj Kumar, Platforms" IEEE Internet Things, VOL. 8, NO. 23, pp. 16845-16852, DECEMBER 1, 2021
[16] Xiaonan Wang and Yajing Song, "Edge-Assisted IoMT-Based Smart-Home Monitoring System for the Elderly with Chronic Diseases" IEEE Sensors letter, VOL. 7, NO. 2, pp. 7500204- 7500204, FEBRUARY 2023.
[17] Kumaran, K., & Sasikala, E. (2023, March). Deep Reinforcement Learning algorithms for Low Latency Edge Computing Systems. In 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP) (pp. 1-5). IEEE.
[18] Saranya, G., & Sasikala, E. (2023). An efficient computational offloading framework using HAA optimization-based deep reinforcement learning in edge-based cloud computing architecture. Knowledge and Information Systems, 65(1), 409-433.