International Journal of Informatics and Communication Technology (IJ-ICT)
Vol. 13, No. 3, December 2024, pp. 453~461
ISSN: 2252-8776, DOI: 10.11591/ijict.v13i3.pp453-461

Fault detection in single-hop and multi-hop wireless sensor networks using a deep learning algorithm

Ramineni Padmasree, Aravalli Sainath Chaithanya
Department of Electronics and Communication Engineering, Rajiv Gandhi University of Knowledge Technologies, Basar, India

Article Info

Article history:

Received Nov 18, 2023
Revised May 24, 2024
Accepted Jun 18, 2024

Keywords:

Decision tree
Gradient boosting
K-nearest neighbours
Malfunctioning node
Random forest
Recurrent neural networks
Wireless sensor network

ABSTRACT

The wireless sensor network (WSN) has received significant recognition for its positive impact on environmental monitoring, yet its reliability remains prone to faults. Common factors contributing to faults include connectivity loss from malfunctioning node interfaces, disruptions caused by obstacles, and increased packet loss due to noise or congestion. This research employs a variety of machine learning and deep learning techniques to identify and address these faults, aiming to enhance the overall lifespan and scalability of the WSN. Classification models such as support vector machine (SVM), gradient boosting clasifer (GBC), K-nearest neighbours (KNN), random forest, and decision tree were employed in model training, with the decision tree emerging as the most accurate at 90.23%. Additionally, a deep learning approach, the recurrent neural network (RNN), effectively identified faults in sensor nodes, achieving an accuracy of 93.19%.

Corresponding Author:

Ramineni Padmasree
Department of Electronics and Communication Engineering
Rajiv Gandhi University of Knowledge Technologies
Basar, India
Email: r.padmasree3@gmail.com

1. INTRODUCTION

A wireless sensor network (WSN) consists of a collection of autonomously distributed sensors that are designed to monitor physical or environmental conditions.These sensors transmit the gathered information wirelessly, facilitating the collection, processing, and transmission of data to a central location for analysis. Communication within a WSN functions in an ad-hoc network setup, where peer-to-peer interactions are involved. These sensors are typically compact and cost-effective, possessing features such as low power consumption, limited computational capacity, memory, battery life, and storage. WSNs find applications across diverse domains such as environmental monitoring, healthcare, industrial automation, and smart cities.

Different categories of WSNs [1] can be identified according to their architectural setups. Among these, single-hop and multi-hop WSNs stand out. In a single-hop WSN, sensor nodes communicate directly with a central station, removing the need for intermediary nodes. Each sensor node forms a direct communication link, denoted by the term “single-hop,” indicating that data is transmitted in a single hop from the source to the central station. Essential features encompass direct communication, simplicity, low latency, and limited range, rendering single-hop WSNs suitable for situations with a small monitored area where direct communication with a central station is feasible.

Multi-hop WSN represents a network architecture where sensor nodes communicate through intermediary nodes, enabling the transmission of data across multiple hops. In contrast to single-hop WSNs, which depend on direct communication with a central station, multi-hop networks [2] employ intermediary nodes to expand coverage, overcome obstacles, and improve reliability. This setup proves beneficial in scenarios with larger areas, obstacles, and a requirement for enhanced network flexibility. Multi-hop WSNs autonomously establish communication routes, offering redundancy and energy efficiency in the transmission of data.

WSNs are typically structured in layered architectures, where each layer has distinct functions contributing to network operations. This organizational approach helps allocate responsibilities among different components. Although the layering model may vary, a common representation involves several layers [3] as shown in Figure 1. In the context of a WSN, the physical layer manages the transmission and reception of raw data wirelessly, addressing aspects like modulation and signal strength. The data link layer supervises dependable communication between directly connected nodes, managing responsibilities like framing, error detection, and media access control (MAC) for channel access. The network layer orchestrates data packet routing, managing path selection, congestion, and network topology, particularly in multi-hop setups. Above this, the transport layer ensures end-to-end communication reliability, covering error recovery, flow control, and data segmentation. The session layer is responsible for establishing, managing, and terminating sessions or connections, adapting to the complexity of communication scenarios. The presentation layer translates, encrypts, and compresses data to enable efficient communication based on specific application requirements. At the topmost layer, the application layer hosts specific functionalities and services, directly interacting with end-users or applications to process, interpret, and utilize collected data.

Each layer [4] serves a unique purpose, contributing to the overall functionality of the WSN. This layered approach aids in the modular design, making it easier to manage, maintain, and upgrade different aspects of the network independently. It’s worth mentioning that the layering structure in WSNs can be customized according to the requirements and limitations of the application. A WSN comprises various cross-layer interactions, incorporating the following:

- Power management plane: the power management aspect in WSNs is vital, emphasizing the optimization and preservation of energy resources in sensor nodes. Effective power management is crucial in WSNs due to the constrained energy capacities of sensor nodes, typically relying on batteries, and the necessity for extended network operation.
- Mobility management plane: it plays a pivotal role in addressing the movement of sensor nodes within the network. This management plane focuses on various challenges associated with dynamic node movement in a WSN, including node tracking, localization, topology maintenance, dynamic routing, energy-efficient movement, handover management, adaptive duty cycling, event-triggered mobility, and optimizing data collection. Its goal is to ensure efficient communication, resource utilization, and adaptation to changing network conditions prompted by node mobility.
- Task management plane: it is a crucial element within a system or network architecture that oversees and coordinates the execution of tasks or activities. In the realm of information technology, this plane involves key functions such as scheduling tasks, allocating resources, managing priorities, balancing loads, controlling concurrency, ensuring fault tolerance, monitoring task progress, handling dependencies, adapting to changing conditions, and facilitating communication and coordination between tasks. Its primary role is to optimize the execution of tasks, promoting efficient resource utilization, maintaining system performance, and achieving operational objectives.

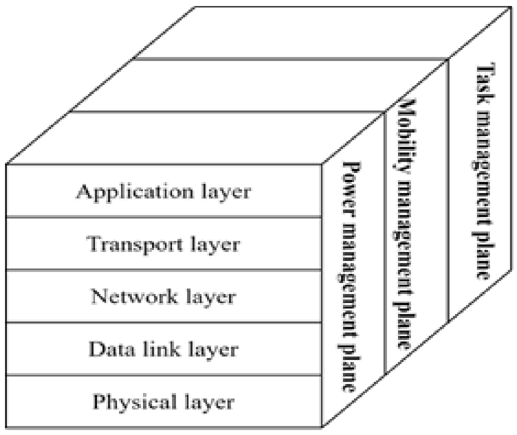


Figure 1. Layered architecture of WSN

These cross-layer interactions within a WSN enhance adaptability, efficiency, and responsiveness. This dynamic approach effectively tackles challenges and fulfills requirements spanning multiple protocol layers, promoting a synergistic and integrated network operation. In many instances, sensors are deployed in unattended or potentially dangerous environments, including forests, highways, and volcanoes. Moreover, given that sensors are electronic devices, they are susceptible to malfunctions. WSNs are susceptible to various types of failures [5], [6], which are listed in Figure 2. This study investigates the detection of faults [7], [8] in both single-hop and multi-hop WSNs utilizing labelled data that includes measurements of humidity and temperature. While earlier studies have explored concerns regarding humidity and temperature in WSNs, generally falling under the category of "data-related faults," they have not explicitly addressed its influence on fault detection [9]-[11], particularly regarding humidity and temperature measurements.

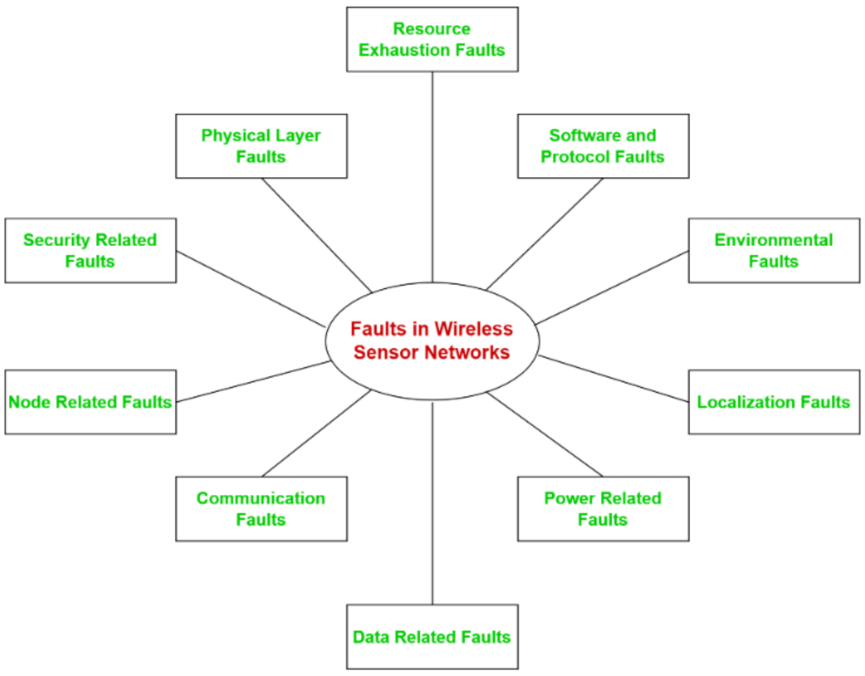


Figure 2. Faults in WSNs

The various types of faults associated with humidity and temperature sensing in WSN are:

- Inaccurate sensing: sensor nodes may produce imprecise or incorrect humidity and temperature readings due to factors like calibration errors, sensor drift, or aging. This compromises the accuracy of environmental data, resulting in unreliable measurements.
- Data corruption: during the transmission of humidity and temperature data, errors or interference in communication may lead to corrupted sensor readings. This results in degraded data quality, potentially affecting the overall reliability of the sensed information.
- Data aggregation errors: in scenarios where data from multiple sensor nodes are aggregated, errors may occur in the summarization process, impacting the accuracy of combined humidity and temperature data. This leads to incorrect aggregation and interpretation of environmental data.

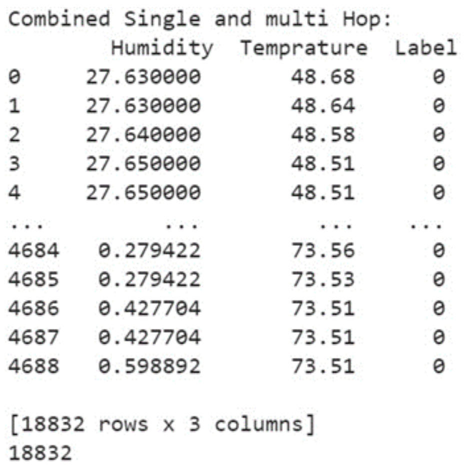
Despite the thorough exploration of fault detection techniques in WSNs, integrating a variety of machine learning and deep learning algorithms, there exists a significant research gap. While prior studies have extensively examined the efficacy of diverse classification models in fault detection, there is insufficient evaluation of their scalability and resource efficiency in real-world WSN deployments. There is an evident need for a systematic assessment of ensemble or hybrid classification approaches, considering factors such as detection accuracy, energy consumption, and adaptability to dynamic network conditions. Bridging this gap can offer valuable insights into optimizing fault detection methods, thereby improving the reliability and lifespan of WSNs while advancing environmental surveillance and network scalability. To tackle these issues, the study aims to implement strategies for detecting and rectifying inaccuracies in sensing. This involves incorporating error-checking mechanisms during data transmission and utilizing data validation and aggregation techniques to enhance the overall quality of humidity and temperature data within the WSN.

2. RESEARCH METHOD

The advent of machine learning has surfaced as an effective approach to address the obstacles within WSNs. has emerged as a powerful solution to confront the challenges within WSNs. Machine learning algorithms [12], [13] empower sensor nodes to make intelligent decisions locally, minimizing the necessity for extensive data transmission to a central server. This localized decision-making approach not only conserves energy but also enhances the overall efficiency of the network. The integration of machine learning into WSNs [14], [15] marks a paradigm shift, elevating their efficiency, adaptability, and overall performance. Leveraging machine learning algorithms enables WSNs to attain heightened intelligence, self-awareness, and the capacity to meet evolving demands across various applications. This synergistic integration holds immense promise for advancing WSN capabilities and unlocking their full potential across diverse fields.

The dataset utilized in this study is derived from an established dataset published in 2010 by researchers affiliated with the University of North Carolina at Greensboro [16]. Employing TelosB motes, the researchers conducted data collection in both a straightforward single-hop and a multi-hop WSN. The dataset encompasses humidity and temperature measurements recorded every 5 seconds over a period of 6 hours. The sample dataset for the combined single-hop and multi-hop scenario is presented in Figure 3.

On the collected dataset, several machine learning techniques are employed for the classification of clean and faulty sensed data. The sensed data are utilized as input for the data preparation module, where it generates a new observation Z from each measurement. Z, in this context, acts as a dependent variable that relies on two measurements: humidity (X) and temperature (Y). The purpose of Z is to determine whether the data is classified as faulty or clean based on the characteristics of the humidity and temperature measurements. Essentially, Z is a binary variable indicating the classification outcome. The various machine learning and deep learning algorithms used are [17], [18].



```
Combined Single and multi Hop:
        Humidity  Temprature  Label
0     27.630000       48.68      0
1     27.630000       48.64      0
2     27.640000       48.58      0
3     27.650000       48.51      0
4     27.650000       48.51      0
...         ...         ...    ...
4684   0.279422       73.56      0
4685   0.279422       73.53      0
4686   0.427704       73.51      0
4687   0.427704       73.51      0
4688   0.598892       73.51      0

[18832 rows x 3 columns]
18832
```

Figure 3. Sample dataset

2.1. Support vector machine (SVM)

The SVM is a technique used for dividing regions. Its objective is to identify the ideal hyperplane for segregating data into two categories. SVM functions by establishing a decision function, which assigns the correct category to each new data point. In linear classification, SVM calculates the most suitable hyperplane for distinguishing samples from two categories. However, when the labeled data presents non-linear separability, the hyperplane derived from the linear scenario becomes inadequate. To overcome this, a Gaussian kernel function is utilized to handle this non-linearity.

2.2. Gradient boosting classifier (GBC)

Implementing the GBC in the context of WSNs entails utilizing this machine learning algorithm for tasks such as fault detection or classification within the WSN environment. The application of GBC in WSN is advantageous due to its capability to manage intricate relationships in the data and offer precise predictions, rendering it suitable for a range of tasks within the network. The effectiveness of the GBC model in WSN is contingent on factors such as the quality of the labeled training data, the appropriateness of chosen features, and the effective tuning of hyperparameters.

2.3. Random forest classifier

The random forest technique is an ensemble learning method primarily based on decision trees, which, when combined as an ensemble, serve as weak learners. In the random forest approach, numerous deep decision trees are employed, each trained on the same dataset but on different subsets. The primary objective is to decrease variance, thereby preventing overfitting on training sets. The procedure involves introducing a new input into the system, which then traverses through each tree in the ensemble. The outcome is determined by either an average or a weighted average of all the reached leaf nodes. For categorical variables, the result might be decided by a majority vote. It’s noteworthy that a higher correlation between trees leads to a higher error rate in the random forest. Hence, the model is deliberately structured to ensure minimal correlation between trees, aiming to enhance overall performance.

2.4. K-nearest neighbours (KNN)

The KNN algorithm can be used for both classification and regression tasks. It operates on the principle that similar data points exist in close proximity to each other, assigning a new data point to the category most similar to the existing ones. The algorithm works as follows:

– Choose the number K of neighbours.
– Calculate the Euclidean distance for each of the K-neighbours.
– Identify the KNN based on the computed Euclidean distance.
– Count the data points in each category among these K-neighbours.
– Assign the new data point to the category with the highest count among the neighbors.
– The model is now ready for use.

2.5. Decision tree

A decision tree functions as a tree-based classifier. It’s structured with internal nodes representing dataset features, branches indicating decision rules, and each leaf node representing an outcome. The tree consists of two primary types of nodes: decision nodes and leaf nodes. Decision nodes aid in decision-making, containing multiple branches, while leaf nodes represent the final outcomes of those decisions and do not lead to further branches.

2.6. Recurrent neural networks (RNN)

The rise in popularity of deep learning techniques [19], [20], utilizing deep neural networks, can be attributed to the advancement of high-performance computing facilities. Deep learning stands out for its ability to process a large number of features, particularly beneficial for handling unstructured data, thus achieving increased power and flexibility. Deep neural networks have proven successful across various learning paradigms, including supervised learning, unsupervised learning, reinforcement learning, and hybrid learning. In RNN [21], the outputs from previous states serve as input for the current state. The hidden layers in RNN have the capability to retain information, with the hidden state being updated based on the output from the previous state. RNN's ability to remember previous inputs, known as long-short term memory (LSTM), makes it well-suited for tasks like time series prediction.

2.7. Machine learning-enabled fault detection starategies for WSN

The process of implementing machine learning algorithms for fault detection is depicted in the flowchart shown in Figure 4. Implementing fault detection in WSN through diverse machine learning approaches involves a set of common procedures. Here is a comprehensive guide:

– Collect data from WSN sensors, covering both normal and faulty conditions, ensuring the dataset accurately reflects real-world scenarios encountered by the WSN.
– Recognize and extract relevant features from the acquired sensor data. These features should encompass crucial characteristics distinguishing between normal and faulty conditions.
– Assign labels to gathered data instances based on their classification as normal or faulty. This labelled dataset is crucial for training and evaluating machine learning models.
– Execute pre-processing tasks, including managing missing data, normalizing or scaling features, and addressing outliers. This ensures the data is appropriately formatted for training machine learning models.
– Choose machine learning algorithms suitable for WSN fault detection. Popular options include SVM [21], GBC, random forest, decision trees, RNN [22], and KNN.
– Utilize a segment of the labelled dataset to train selected machine learning models. During training, models assimilate patterns and relationships between features that distinguish normal and faulty conditions.
– Fine-tune the hyper parameters of the selected algorithms to enhance performance, involving adjustments to parameters controlling the learning process for superior outcomes.

– Validate model performance using cross-validation techniques to ensure robust generalization to new, unseen data.
– Assess models on a distinct testing dataset not used during training, providing an impartial evaluation of their ability to detect faults in WSN.
– Evaluate the performance of each model using metrics such as accuracy, precision, recall, and F1-score. This evaluation helps in choosing the most effective model for fault detection [23]-[25].
– Deploy the chosen model or ensemble of models for real-time fault detection in the WSN. Continuously monitor its performance in the live environment and make updates as needed.
– Regularly monitor the performance of model in the WSN and adjust it as needed to accommodate environmental changes or enhance fault detection capabilities.

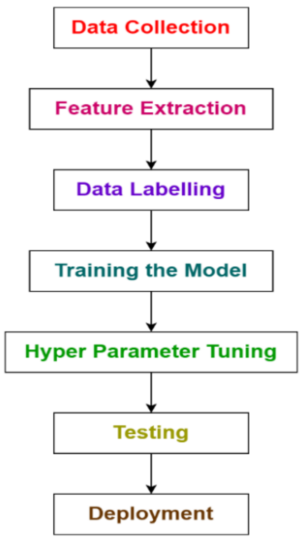


Figure 4. Flowchart of machine learning implementation for fault detection in WSN

3. RESULTS AND DISCUSSION

The aggregated dataset, encompassing humidity and temperature measurements for both single hop and multi hop scenarios, undergoes pre-processing for optimal formatting. After this step, the dataset is partitioned into training and testing sets, with the models being trained using the specified training data. The results showcase the accuracy of fault detection in WSN. In Figure 5, the accuracy graphs of SVM and RFC are displayed, and Figure 6 exhibits the accuracy graphs of KNN, GBC, and decision tree. Additionally, Figure 7 visually represents the layers of RNNs along with accuracy. The fault detection accuracies obtained through the utilization of diverse machine learning and deep learning algorithms are outlined in Table 1.

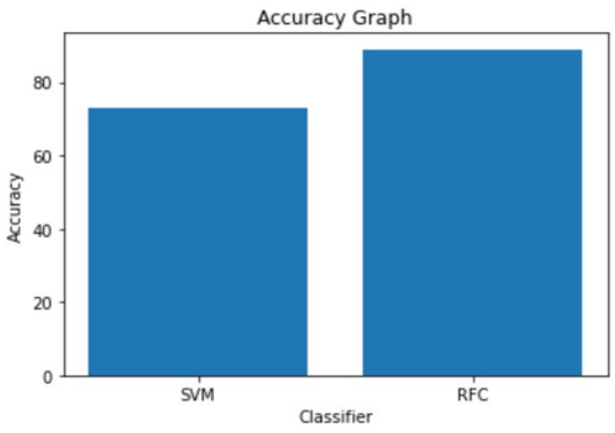


Figure 5. Fault detection accuracy of SVM and RFC

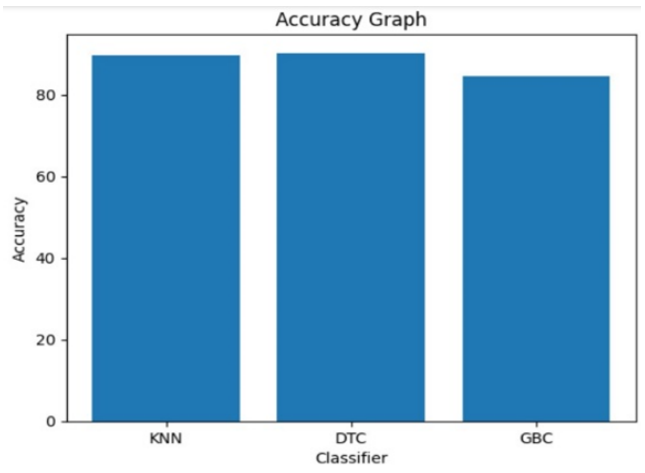


Figure 6. Fault detection accuracy of KNN, DTC, and GBC

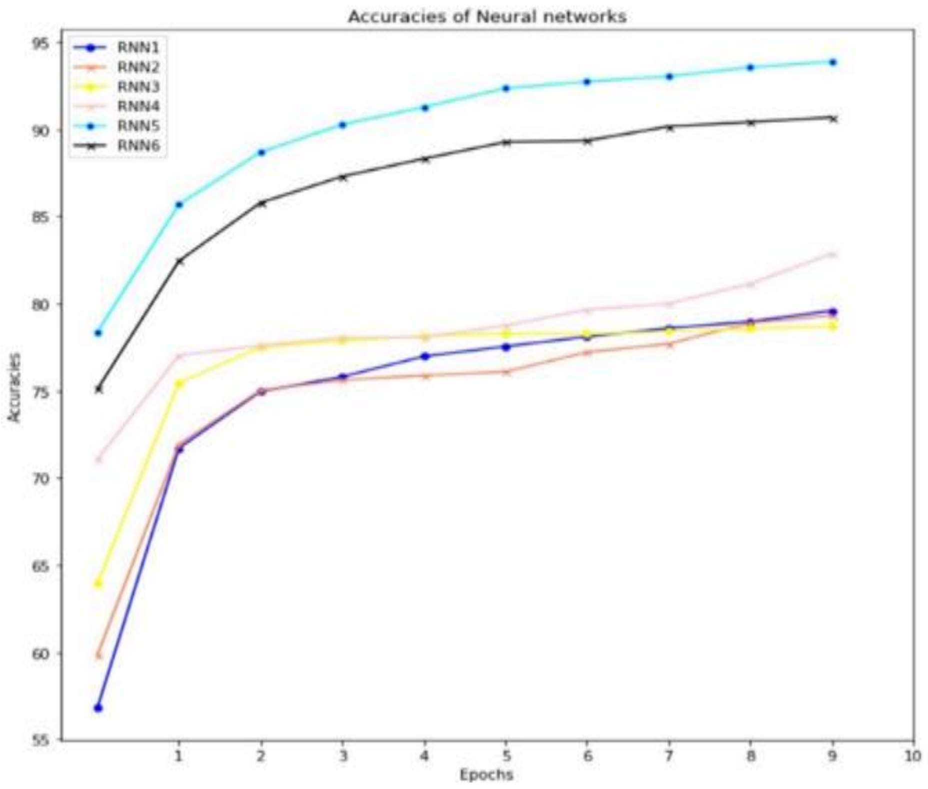


Figure 7. Fault detection accuracy of RNNs

Table 1. Fault detection accuracy of various algorithms

ML and DL algorithms	Accuracy
RNN	93.19%
Decision tree	90.23%
KNN	89.59%
Random forest	89.08%
GBC	84.55%
SVM	73.42%

We discovered that employing various machine learning and deep learning algorithms in our investigation is associated with enhanced identification and resolution of faults within WSNs. The approach proposed in this study exhibited notably superior accuracy in fault detection, notably with the decision tree achieving 90.23% precision and the RNN achieving 93.19% accuracy.

Our research suggests that employing diverse machine learning and deep learning algorithms to address faults in WSN does not compromise their reliability. The proposed approach holds potential for improving the WSN's longevity and scalability without adversely affecting performance. However, further comprehensive studies may be necessary to validate its effectiveness, particularly regarding specific fault scenarios.

Future research endeavors may concentrate on improving fault detection algorithms, creating adaptable mechanisms for dynamic adjustment, investigating energy-conserving fault management tactics, evaluating resilience against particular fault scenarios, validating methods via real-world implementations, scrutinizing scalability and operational efficiency, merging with IoT frameworks, integrating human-in-the-loop methodologies, conducting extensive reliability assessments over time, and undertaking comprehensive cost-benefit analyses.

4. CONCLUSION

Recent observations indicate that the fault detection methodology for WSNs, as delineated in this paper, encompassed several classifiers, including SVM, random forest, KNN, GBC, and decision tree. Our findings conclusively demonstrate that among these classifiers, the decision tree exhibited superior performance, achieving an accuracy of 90.23%. With its structure resembling a tree, featuring internal nodes representing dataset features, branches indicating decision rules, and leaf nodes denoting outcomes, the decision tree proved to be highly effective. Furthermore, our study investigated the RNN, characterized by cyclic connections enabling output from certain nodes to influence subsequent input to the same nodes. The results highlight that the RNN outperformed other algorithms, boasting an impressive accuracy of 93.19% for fault detection.

REFERENCES

[1] J. Yick, B. Mukherjee, and D. Ghosal, "Wireless sensor network survey," *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, Aug. 2008, doi: 10.1016/j.comnet.2008.04.002.

[2] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on wireless sensor networks technology and evolution," *Sensors*, vol. 9, no. 9, pp. 6869–6896, Aug. 2009, doi: 10.3390/s90906869.

[3] A. A. A. Alkhatib and G. S. Baicher, "Wireless sensor network architecture," in *2012 International Conference on Computer Networks and Communication Systems (CNCS 2012)*, 2012, vol. 35, no. Cncs, pp. 11–15, [Online]. Available: https://pdfs.semanticscholar.org/ba9d/7436c266a8b309485fcaa126f661f2665dd3.pdf%0Ahttp://www.ipcsit.com/vol35/003-CNCS2012-N010.pdf.

[4] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Communications Surveys and Tutorials*, vol. 12, no. 2, pp. 222–248, 2010, doi: 10.1109/SURV.2010.020510.00058.

[5] T. Muhammed and R. A. Shaikh, "An analysis of fault detection strategies in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 78, pp. 267–287, Jan. 2017, doi: 10.1016/j.jnca.2016.10.019.

[6] A. Mehmood, N. Alrajeh, M. Mukherjee, S. Abdullah, and H. Song, "A survey on proactive, active and passive fault diagnosis protocols for WSNs: Network operation perspective," *Sensors (Switzerland)*, vol. 18, no. 6, p. 1787, Jun. 2018, doi: 10.3390/s18061787.

[7] R. B. Shaikh, A. H. Sayed, and Z. H. Agusbal, "An algorithm for sensor node failure detection in WSNs," in *International Conference on Electrical, Electronics, and Optimization Techniques, ICEEOT 2016*, Mar. 2016, pp. 1391–1396, doi: 10.1109/ICEEOT.2016.7754912.

[8] R. Mitchell and I. R. Chen, "A survey of intrusion detection in wireless network applications," *Computer Communications*, vol. 42, pp. 1–23, Apr. 2014, doi: 10.1016/j.comcom.2014.01.012.

[9] I. Banerjee, P. Chanak, H. Rahaman, and T. Samanta, "Effective fault detection and routing scheme for wireless sensor networks," *Computers and Electrical Engineering*, vol. 40, no. 2, pp. 291–306, Feb. 2014, doi: 10.1016/j.compeleceng.2013.04.027.

[10] K. Cho, M. Jo, T. Kwon, H. H. Chen, and D. H. Lee, "Classification and experimental analysis for clone detection approaches in wireless sensor networks," *IEEE Systems Journal*, vol. 7, no. 1, pp. 26–35, Mar. 2013, doi: 10.1109/JSYST.2012.2188689.

[11] A. Bari, A. Jaekel, J. Jiang, and Y. Xu, "Design of fault tolerant wireless sensor networks satisfying survivability and lifetime requirements," *Computer Communications*, vol. 35, no. 3, pp. 320–333, Feb. 2012, doi: 10.1016/j.comcom.2011.10.006.

[12] S. Zidi, T. Moulahi, and B. Alaya, "Fault detection in wireless sensor networks through SVM classifier," *IEEE Sensors Journal*, vol. 18, no. 1, pp. 340–347, Jan. 2018, doi: 10.1109/JSEN.2017.2771226.

[13] Z. Noshad *et al.*, "Fault detection in wireless sensor networks through the random forest classifier," *Sensors (Switzerland)*, vol. 19, no. 7, p. 1568, Apr. 2019, doi: 10.3390/s19071568.

[14] R. Regin, S. S. Rajest, and B. Singh, "Fault detection in wireless sensor network based on deep learning algorithms," *EAI Endorsed Transactions on Scalable Information Systems*, vol. 8, no. 32, pp. 1–7, Jul. 2021, doi: 10.4108/eai.3-5-2021.169578.

[15] Aniyan and A. N. Netto, "A machine learning based hybrid approach for fault detection in WSN," *Journal of Emerging Technologies and Innovative Research*, vol. 6, no. 5, 2019, [Online]. Available: https://www.jetir.org/papers/JETIRCV06039.pdf.

[16] S. Suthaharan, M. Alzahrani, S. Rajasegarar, C. Leckie, and M. Palaniswami, "Labelled data collection for anomaly detection in wireless sensor networks," in *Proceedings of the 2010 6th International Conference on Intelligent Sensors, Sensor Networks and Information Processing, ISSNIP 2010*, Dec. 2010, pp. 269–274, doi: 10.1109/ISSNIP.2010.5706782.

[17] G. Pachauri and S. Sharma, "Anomaly detection in medical wireless sensor networks using machine learning algorithms," *Procedia Computer Science*, vol. 70, pp. 325–333, 2015, doi: 10.1016/j.procs.2015.10.026.

[18] X. Zhang, J. Zou, K. He, and J. Sun, “Accelerating very deep convolutional networks for classification and detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 10, pp. 1943–1955, 2016, doi: 10.1109/TPAMI.2015.2502579.
[19] M. Panda, B. S. Gouda, and T. Panigrahi, “Fault diagnosis in wireless sensor networks using a neural network constructed by deep learning technique,” 2020, pp. 77–101.
[20] O. Obst, “Poster abstract: Distributed fault detection using a recurrent neural network,” *2009 International Conference on Information Processing in Sensor Networks, IPSN 2009*, pp. 373–374, 2009.
[21] Y. Yuan, S. Li, X. Zhang, and J. Sun, “A comparative analysis of SVM, Naive Bayes and GBDT for data faults detection in WSNs,” in *Proceedings - 2018 IEEE 18th International Conference on Software Quality, Reliability, and Security Companion, QRS-C 2018*, Jul. 2018, pp. 394–399, doi: 10.1109/QRS-C.2018.00075.
[22] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi, and R. Jenssen, “An overview and comparative analysis of recurrent neural networks for short term load forecasting,” *arXiv preprint*, 2017, doi: 10.1007/978-3-319-70338-1.
[23] M. A. Abdullah, B. M. Alsolami, H. M. Alyahya, and M. H. Alotibi, “Intrusion detection of DoS attacks in WSNs using classification techniuqes,” *Journal of Fundamental Applied Sciences*, vol. 10, no. 4S, pp. 298–303, 2018, [Online]. Available: http://dx.doi.org/10.4314/jfas.v10i4s.94.
[24] S. Osken, E. N. Yildirim, G. Karatas, and L. Cuhaci, “Intrusion detection systems with deep learning: a systematic mapping study,” in *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*, Apr. 2019, pp. 1–4, doi: 10.1109/EBBT.2019.8742081.
[25] R. Singh, J. Singh, and R. Singh, “Fuzzy based advanced hybrid intrusion detection system to detect malicious nodes in wireless sensor networks,” *Wireless Communications and Mobile Computing*, vol. 2017, pp. 1–14, 2017, doi: 10.1155/2017/3548607.

BIOGRAPHIES OF AUTHORS

Ramineni Padmasree holds the position of Assistant Professor in the Department of Electronics and Communication Engineering (ECE) at Rajiv Gandhi University of Knowledge Technologies, Basar. Concurrently, she is pursuing a part-time Ph.D. in Wireless Communications at Osmania University, Hyderabad. She earned her M. Tech degree in Digital Electronics and Communication Systems (DECS) and her B.Tech. degree in ECE from JNTU Hyderabad. Her research interests encompass wireless communication, advanced microcontrollers-embedded systems, wireless sensor networks, antenna designs, and machine learning. She can be contacted at email: r.padmasree3@gmail.com.

Aravalli Sainath Chaithanya is working as an Assistant Professor in the Department of Electronics and Communication Engineering (ECE) at Rajiv Gandhi University of Knowledge Technologies, Basar. Alongside, he is pursuing a part-time Ph.D. in computer vision at Osmania University, Hyderabad. He holds an M.Tech degree in VLSI System Design and a B.Tech degree in ECE from JNTU Hyderabad. His research interests include image processing, computer vision, machine learning, and WSN, with additional interests in VLSI-SoCs and high-performance pipeline processors. He can be contacted at email: chaitanya.aravalli@gmail.com.