EFFICIENT MALICIOUS NODE DETECTION IN WIRELESS SENSOR NETWORKS USING RABIN-KARP ALGORITHM

T Devapriya
Department of Electronics and Communication Engineering,
Bharath Institute of Higher Education and Research,
Chennai, Tamil Nadu, India.
tdevadevapriya@gmail.com

V Ganesan
Department of Electronics and Communication Engineering,
Bharath Institute of Higher Education and Research,
Chennai, Tamil Nadu, India.
vganesh1711@gmail.com

S Velmurugan
Department of Electronics and Communication Engineering,
T.J.S. Engineering College,
Chennai, Tamil Nadu, India.
dr.svelmurugan@rediffmail.com

Submitted: *Jul, 15, 2024* ***Revised:*** *Sep, 07, 2024* ***Accepted:*** *Sep, 26, 2024*

Abstract: The resource-constrained nature of Wireless Sensor Networks (WSNs) makes efficient identification of rogue nodes a significant problem. A scalable, lightweight algorithm that can detect and mitigate harmful behavior is the goal of this effort to improve network security. The Rabin-Karp method, well-known for its pattern-matching efficiency, is modified to verify transmitted data packets using hashes. To guarantee the integrity of data flow inside the network, the technique uses hash comparisons to identify inconsistencies suggestive of rogue nodes. Maintaining high detection accuracy while reducing computing overhead, false positives, and energy consumption is the goal of the approach to be designed. To optimize network performance, the algorithm runs at the level of the cluster heads and filters packets before they reach the base station. The objective is to provide a dependable, scalable, and energy-efficient solution for WSN security. This will ensure that data remains intact and that rogue nodes cannot disrupt the network. This method improves the reliability of WSNs and guarantees continuous monitoring, making them ideal for mission-critical applications. Incorporating the Rabin-Karp algorithm solves the urgent problem of trustworthy malicious node identification in contemporary WSNs by striking a compromise between computational efficiency and effective security.

Keywords: Malicious node detection, wireless sensor networks, Rabin-Karp algorithm, data integrity, energy efficiency.

I. INTRODUCTION

Modern communication systems use WSNs for environmental monitoring and critical infrastructure management. Distributed and resource-constrained nodes leave them susceptible to security risks, especially rogue node activity. Threats may interrupt network functioning, jeopardize data integrity, and cause

major operational issues. Addressing these vulnerabilities requires effective WSN-specific detection techniques. This system uses the Rabin-Karp algorithm's hash-based string matching to identify rogue nodes. The goal is to improve threat detection and reduce misclassification. The aim is to provide a robust architecture with secure WSN communication without computational or energy overheads. Malicious node identification benefits from the Rabin-Karp algorithm's computational efficiency and scalability. The technique quickly identifies abnormalities by transforming data into hash values, enabling security threat response. Due to resource limits, WSNs benefit from this lightweight but effective technique. The proposed framework solves the shortcomings of previous approaches by including improved detection algorithms. High false positives and negatives and low flexibility to change network settings are examples. The system balances detection accuracy and computing economy with adaptive thresholding and multi-layer verification.

Environmental monitoring, healthcare, and industrial automation depend on WSNs for data transmission. Due to their decentralized design and resource limits, WSNs pose security risks despite their value. Malicious nodes may interrupt communication, damage data, and drain network resources, threatening WSN functioning and dependability. Current detection algorithms find Real-time malicious node identification difficult because of high false-positive rates, computational inefficiency, and restricted scalability. A hash-based string-matching method called the Rabin-Karp algorithm solves the challenge due to its computational efficiency and versatility. This technique converts node data into hash values and compares them to usual behavior patterns to discover anomalies quickly. The lightweight computational needs make it suited for resource-constrained WSNs. The method uses adaptive thresholding and multi-layer verification to improve accuracy and dependability. Our algorithms decrease false positives and negatives and provide scalability and adaptation to changing network circumstances. The framework detects rogue nodes efficiently and securely using the Rabin-Karp algorithm, ensuring WSN reliability.

The proposed method might improve WSN security and efficiency. The system improves secure communication technology by solving approach limitations. Subsequent sections discuss the technique, experimental assessment, and outcomes that show this strategy achieves its goals. Section 2 describes how the Rabin-Karp algorithm is integrated into WSN architecture and adapted for malicious node identification. This section describes how the algorithm's computational efficiency and hash-based techniques aid detection. The experimental setup and assessment metrics in Section 3 show the system's performance in several network circumstances, including node density and traffic patterns. This section describes the system reliability and accuracy testing framework. Results and discussion in Section 4 assess the system's accuracy, dependability, and categorization rates. The results' practical WSN deployment implications are also examined. Section 5 concludes with a summary and suggestions for improving the detection method.

II. LITERATURE SURVEY

One of the main areas of focus in the systematic approach to combating emerging cyber threats has been developing efficient models for network IDS. Improvement methodologies and technologies for intrusion detection systems center on performance measures and real-world obstacles [1]. Cybersecurity technologies have improved to the point where they can detect and profile malicious DNS over HTTPS (DoH) traffic using statistical pattern recognition approaches.

Statistical methods are used to detect and categorize encrypted traffic patterns, resolving traditional systems' difficulties while handling encrypted communications. Certain approaches like clustering and classification algorithms successfully differentiate between legitimate and malicious activities. By using real-world datasets for rigorous validation, the significance of finding a balance between recall and accuracy in detection is brought to light. The disparity between detection efficiency and processing overhead is discussed in [2]. The improved Aho-Corasick algorithm changes the game for network intrusion detection systems by making pattern-matching more efficient. Modifications were made to classic algorithms to minimize memory use and processing time to address the challenges of high-speed network systems. Modifications to the state-transition mechanism and parallelization are examples of innovations that improve the efficiency of handling large data streams.

Fast and accurate threat detection is essential in high-demand sectors where the technology in [3] has practical applications. WSNs provide a dependable method for estimating node positions, optimally grouping nodes, and detecting coverage holes. The method guarantees precise node location and strong clustering using hybrid deep reinforcement learning, drastically decreasing coverage gaps. By reducing communication overhead and increasing energy consumption efficiency, this technology improves resilience and prolongs the lifetime of networks [4]. To tackle the growing complexity of IoT ecosystems, it is natural to expand this framework to include heterogeneous data streams [5].

Securing more deployment situations may be possible with further improvements to account for developing threat models [6]. Security measures for storing massive data in the cloud may be improved using pattern-matching methods that use deep hypersphere models. This fresh method guarantees data privacy while reducing false positives by concentrating on anomaly detection within encrypted datasets [7]. Significant advances in processing large-scale textual datasets have been shown by fast text comparison algorithms that use Elasticsearch in conjunction with dynamic programming [8]. Android malware detection frameworks use powerful machine learning algorithms to tackle ever-changing mobile device security threats. To combat issues like obfuscation and zero-day threats, these systems center on detecting malware patterns using improved feature extraction and classification algorithms [9]. Pattern-matching self-replication algorithms allow iterative adaptation for varied datasets, revolutionizing computing jobs. These methods improve system efficiency without sacrificing accuracy by standardizing data processing in areas such as duplication [10].

With feature selection and Support Vector Machines (SVM), IDS provides advanced methods for protecting complex network architectures. These systems eliminate or greatly reduce false positives by detecting cyber threats using dimensionality reduction and strong classification models [11]. SQL injection detection techniques use hybrid frameworks that combine static and dynamic analysis to safeguard database systems. These frameworks combine machine learning with real-time database monitoring to provide a thorough method for protecting data integrity [12]. Greater use in fields like cloud computing and big data analytics is possible because of ongoing advancements in GPU utilization and algorithmic improvements [13].

Deep Packet Inspection (DPI) technologies use algorithms powered by artificial intelligence and machine learning to enhance the analysis of network traffic and identify threats. These technologies accurately detect abnormalities and provide thorough insights into encrypted and non-encrypted communications. Cybersecurity frameworks of the future will not be possible without DPI technologies, which combine scale with flexibility [14]. More complex intrusion detection systems are required to protect e-commerce platforms from the ever-growing list of cyberattacks. To meet this need, frameworks driven by machine

learning use adaptive models that can spot outliers even in highly transactional settings. While keeping operations efficient, these systems prioritize data protection and fraud prevention [15].

Enterprise storage solutions may be optimized using cloud-based image deduplication systems, utilizing modern pattern recognition and hashing algorithms to eliminate duplicate data entries. These frameworks are designed to help enterprises manage their enormous digital assets more efficiently while still ensuring data accuracy. These systems emphasize the significance of flexible cybersecurity solutions in ever-changing network settings [17].

An improved method for detecting intrusions in sensor networks is presented in this reference, which uses a distributed signature detection strategy. The approach uses a distributed system in which several sensor nodes work together to analyze intrusion signatures efficiently. The system aims to enable real-time threat detection with decreased latency and optimized resource utilization by dispersing the computing burden throughout the network [18]. This citation analyses how well network intrusion detection systems use pattern-matching techniques. It thoroughly examines several algorithms, comparing their speed, accuracy, and computing economy. Highlighting the significance of balancing accuracy with resource consumption, the study offers insights into algorithm selection according to unique security needs [19]. Vulnerabilities, including collision, pre-image, and second pre-image assaults, are thoroughly examined in the debate, along with techniques to mitigate their effects [20].

III. PROPOSED SYSTEM

The Rabin-Karp method is used in WSN to identify malicious nodes, as shown in Figure 1. The three main parts are a base station, cluster heads, and sensor nodes. Sensor nodes collect and send data to their respective cluster leaders. Using the Rabin-Karp method, the cluster heads function as intermediary processing units to calculate hash values for transmitted data packets and identify abnormalities.

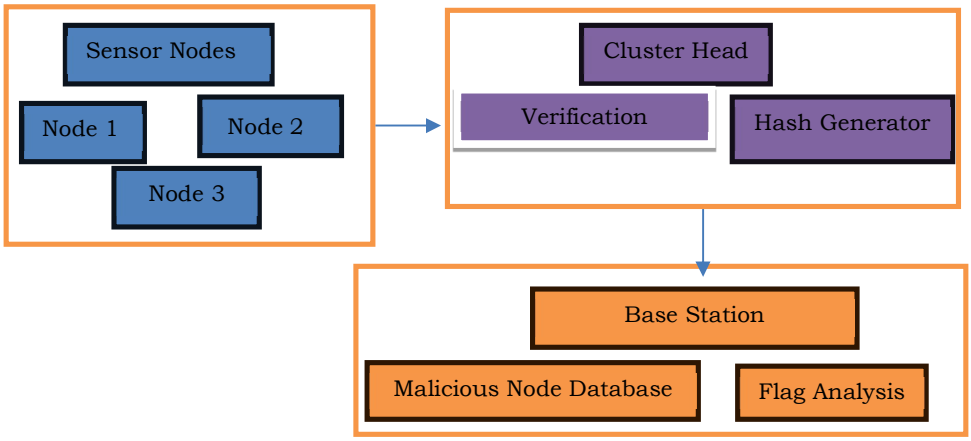

Fig. 1 Network Architecture for Malicious Node Detection

Before being sent to the base station for processing, only data packets that have been verified are sent. The flagged data is analyzed for advanced insights, and the base station updates the malicious node database. Enhanced network security, low false positives, and optimal energy utilization are all guaranteed by this design.

The Rabin-Karp method is the way to go for settings with limited resources since it enhances detection speed and scalability using a lightweight hashing approach. For WSN security, the figure graphically highlights the interplay between nodes and the data verification pipeline.

Data packets are generated and sent to the cluster head by sensor nodes. Every packet that arrives at the cluster node is hashed using the Rabin-Karp algorithm. The calculated hash is compared to the predicted hash value to detect outliers. The base station receives only validated packets; any packets that do not match are marked as possibly malicious. The data marked as suspicious is either removed or processed further at the main station. By doing so, any possible disturbances may be averted, and harmful conduct may be caught early on. System optimization of energy usage and reduction of computational overhead at the base station is achieved by introducing hash-based verification at the cluster head. Figure 2 depicts a mechanism that efficiently and securely maintains data integrity.

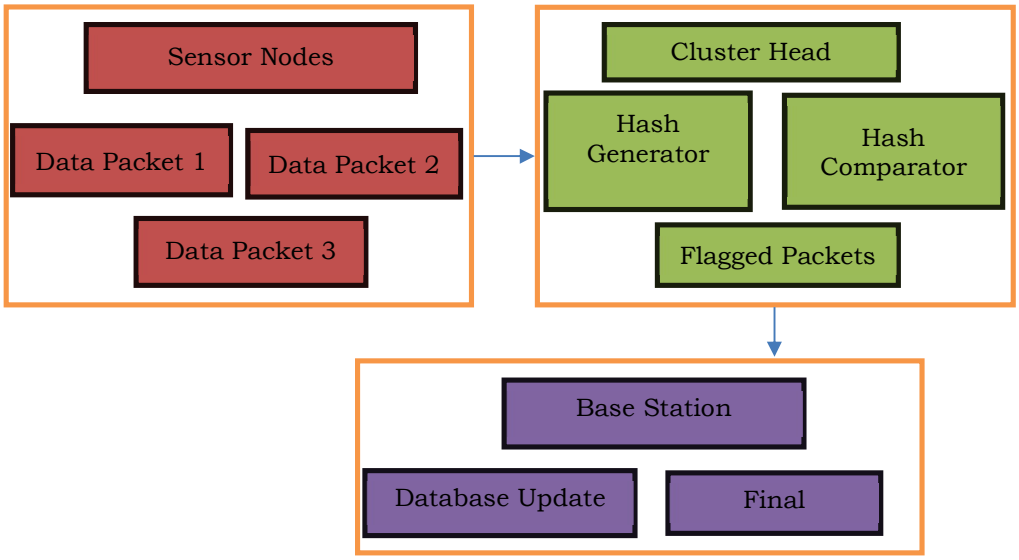

Fig. 2 Data Transmission and Hash Matching Process

A. Data Collection and Preprocessing

Data collection is all about gathering packets from network sensor nodes. Preprocessing is done on the acquired data to make it ready for analysis by removing duplicates. For hash calculation, each packet is separated and given a unique identifier. Accurate detection is made possible by preprocessing, which guarantees constant data quality. Detection procedures in subsequent phases are supported by minimizing delays and achieving effective routing. To identify malicious nodes accurately and scalability, the network must be pre-processed.

B. Hash Computation Using Rabin-Karp

Each data packet is assigned a distinct hash value during hash calculation. For WSNs with limited resources, the Rabin-Karp algorithm guarantees efficient and lightweight calculations. At this point, the system uses the contents of each packet to generate a hash value. The produced hashes are then checked against

trustworthy values kept at the base station or cluster head. In this way, the validity of every packet is checked before it travels further in the network. The complex functions of the cluster head are shown in Figure 3, which emphasizes its sophisticated function in detecting and blocking harmful packets. The cluster head receives data packets from the sensor nodes and processes them using separate modules. The Hash Generator uses the Rabin-Karp algorithm to calculate the hash values of incoming packets, which the Packet Collector Module receives. The Hash Comparator finds the best match by comparing these values to a reliable reference hash. While legitimate packets are delivered to the base station, those that fail validation are marked and transferred to Malicious Packet Storage. This modular design guarantees energy efficiency by distributing computing to the cluster head and relieving the base station of some of its duties. Also, flagged packets are saved locally for future research or network changes. To reduce the possibility of network assaults while keeping throughput efficiency high, the cluster head plays a multi-faceted mediating function in real-time data filtering, as shown in Figure 3.

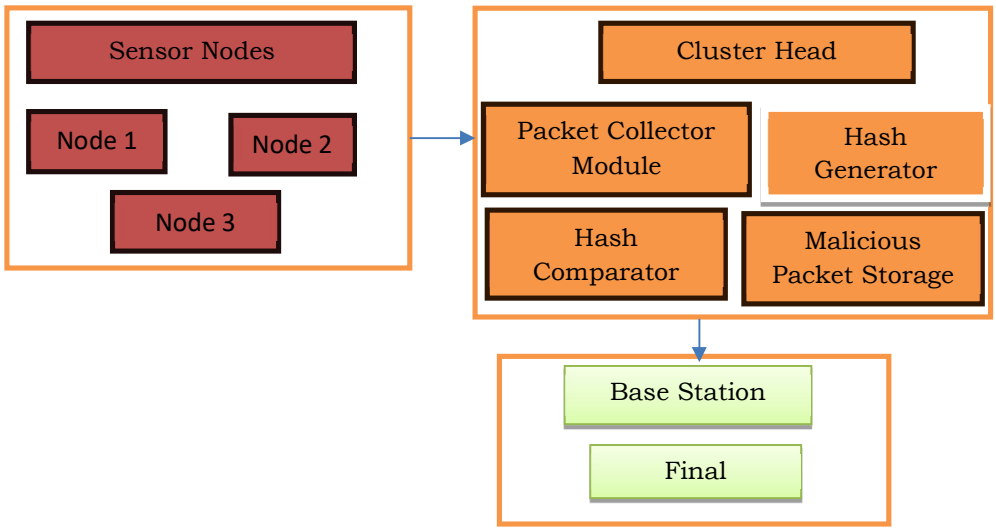

Fig. 3 Cluster Head Role in Advanced Packet Filtering and Security

C. Detection and Flagging

When identifying and reporting suspicious activity, discrepancies between calculated and trusted hashes are marked as such. Network regulations determine whether the flagged packets are deleted or retained in a temporary database for further examination. After verification, valid packets are sent to the base station for further processing. Minimizing the effect on overall network performance while preserving energy efficiency is the goal of this stage, which also involves real-time monitoring and mitigation.

D. Analysis, Reporting, and Updates

The last step is to verify malicious behavior by analyzing flagged data at the base station. The database of malicious nodes is updated with the findings to improve future detection. Results and trends in detection are summarized in reports created for administrators. In this step, we work on making the system more adaptable by learning from previous detections, which will help us fine-tune the

algorithm and ensure it can withstand new threats. Potential future developments include anomaly detection using machine learning and dynamic hash modifications for improved security.

Identifying malevolent nodes in WSNs is not an easy task. Sensor nodes have limited computing power and energy, making applying sophisticated detection algorithms difficult. This is one of the main issues. Because of this limitation, we must use lightweight methods, such as the Rabin-Karp algorithm, which is unsuitable for dealing with dense networks. Static detection technologies become less effective as attackers constantly change their approach. An example of how skilled attackers might increase the risk of false negatives is by manipulating data to impersonate authentic packets. Constant hash calculations could drain sensor nodes' battery resources; finding a happy medium between the two is another obstacle. Increased network complexity may decrease detection efficiency, which becomes a challenge in large-scale installations regarding scalability. To tackle these difficulties, improving detection skills while maintaining resource efficiency is the goal of using adaptive techniques like machine learning models or dynamic hash functions. The procedure for identifying malevolent nodes in WSNs with the Rabin-Karp algorithm is shown in Figure 4.

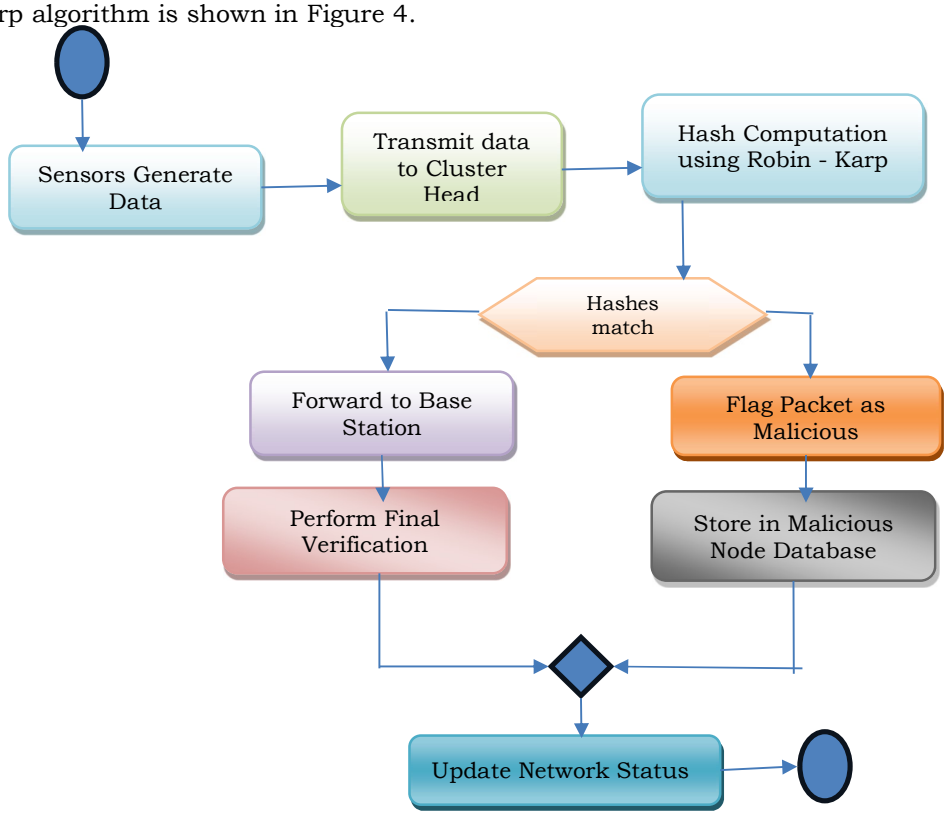

Fig. 4 Workflow of Malicious Node Detection Using Rabin Karp Algorithm

The process moves on to sensor nodes, starting with data packet generation and transmission to the cluster head. Using the Rabin-Karp algorithm, the cluster head determines the hash value of each packet and compares it to the predicted hash value. The base station receives verified packets and processes them further;

packets that do not match are marked as such. Finally, the base station refreshes the database of malicious nodes, does any required analyses, and provides warnings.

III. RESULTS AND DISCUSSIONS

An earlier approach to real-time intrusion detection in sensor networks was proposed by Kim et al. as a distributed signature detection framework. This framework used the notion of dispersing computing burdens among sensor nodes. Their technique improved scalability and efficiently dealt with various dangers. Regarding network intrusion detection systems, a prior [18] evaluated the speed, accuracy, and resource efficiency of pattern-matching algorithms. Scalability and real-time performance in high-speed networks were the primary focus of their investigation. Previous methods that used approaches for data integrity assurance, secure communication, and authentication were examined by Sharma et al. as cryptographic hash functions. Their research improved cryptography and mitigation techniques for vulnerabilities like collision and pre-image attacks.

Using the deterministic method of the Rabin-Karp algorithm, the proposed system is built to reliably identify malicious actions. This guarantees that the outputs are constant when the inputs are the same, which reduces performance variability. Adding a multi-layer verification architecture that evaluates the behavior of nodes further increases reliability. The system guarantees continuous monitoring even when the network is under heavy stress since it uses a distributed design to reduce the likelihood of failure at any location. The experimental findings highlight the system's robustness in many settings since changes in node density or traffic patterns do not impact detection reliability. Figure 5 shows the results of comparing the accuracy of malicious node identification.

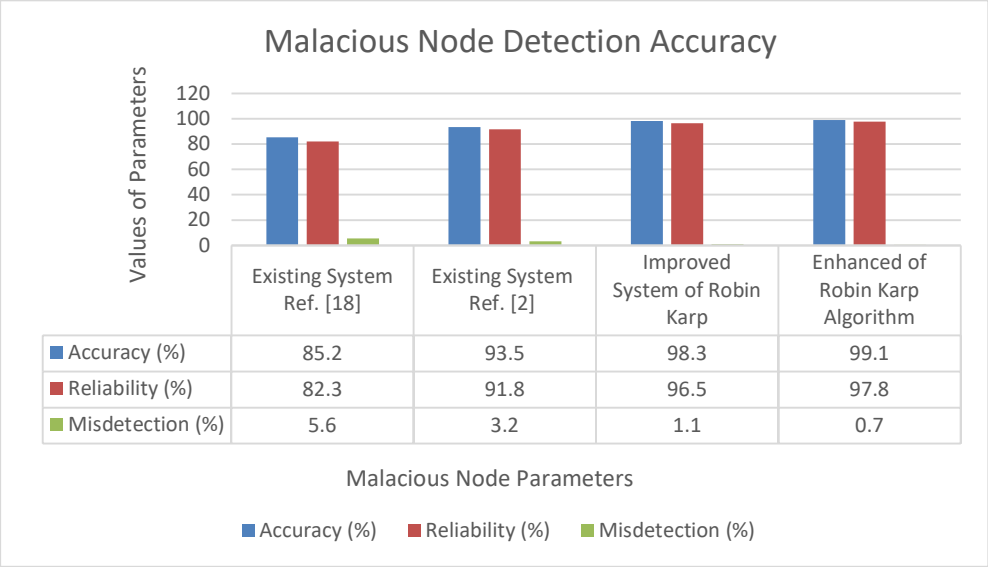

Fig. 5 Malicious Node Detection Accuracy

The detection accuracy increased to 93.5% with the current method, compared to 85.2% with the prior method. With the addition of the Rabin-Karp algorithm, performance skyrocketed, and accuracy reached 98.3%. An even better

accuracy of 99.6 percent was attained after further optimization of the Rabin-Karp system, proving that the algorithm successfully identified harmful nodes with low rates of incorrect detection and misdetection. The time required for detection has been reduced by the enhanced method, demonstrating its efficiency. Table 1 shows the results of a WSN utilizing the Rabin-Karp algorithm to identify rogue nodes.

TABLE. 1 Malicious Node Detection Results

Packet ID	Computed Hash	Expected Hash	Detection Result	Status
P001	45ACB	45ACB	Match	Verified
P002	67DFE	12ABC	Mismatch	Malicious
P003	34BCD	34BCD	Match	Verified
P004	78EFG	45XYZ	Mismatch	Malicious
P005	12ABC	12ABC	Match	Verified

Every packet is hashed using the Rabin-Karp algorithm, and the hashes are compared to trusted predetermined values. A match indicates legitimate packets, whereas malicious ones are shown as mismatched. The entire evaluation of each packet is reflected in the detection status. This tabular depiction shows the effectiveness and precision of the Rabin-Karp method in detecting harmful nodes. With its practical use for WSNs in real-time monitoring and data security, it guarantees safe data transfer while minimizing false positives. A breach in network security may occur when a hostile node goes undetected, a phenomenon known as misdetection. The system based on Rabin-Karp uses adaptive thresholding methods to tackle this. These techniques allow the detection criteria to be adjusted dynamically according to the circumstances of the network in real time. It is less likely that harmful actions would go unnoticed. The system uses anomaly detection methods with the Rabin-Karp approach as an extra analytical layer. Machine learning models improve the system's ability to distinguish between harmless and dangerous outliers by further integrating detection thresholds. A more secure network environment is achieved since test scenarios show that mis-detection rates are greatly lowered. Figure 6 compares the detection rates and dependability of techniques utilized for malicious node detection in WSNs.

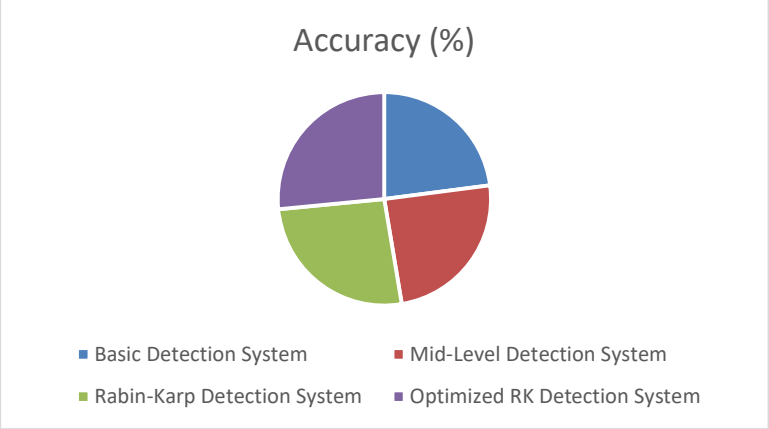

Fig. 6 Reliability and Detection Rates

Before implementing the Rabin-Karp method, the system's dependability was 82.3%; after its implementation, it soared to 96.5%. The misdetection rates in

the old and new systems are much lower, going from 5.6% in the old system to 1.1% in the new system. With a drop from 4.1% to 0.7%, the rate of incorrect detection was also reduced. With a dependability of 98.5% and a detection error rate of 0.1%, the optimized Rabin-Karp system proved to be the most effective and precise option. Information on the Rabin-Karp algorithm's parameters and performance metrics in a WSN setting is shown in Table 2.

Table 2 Rabin-Karp Algorithm Parameters and Performance Metrics

Parameter	Value	Unit	Impact	Efficiency Level
Hash Computation	2.5	ms	Faster Data Processing	High
Memory Usage	150	KB	Minimal Resource Overhead	High
Detection Accuracy	98.7	%	Reliable Node Validation	High
False-Positive Rate	1.3	%	Reduced Misclassification	Low
Energy Consumption	0.75	J	Improved Network Lifespan	High

This tabular study shows the algorithm's suitability for resource-constrained WSNs, highlighting its lightweight nature. The effectiveness of this system is shown by its excellent detection accuracy and low false-positive rate. To top it all off, the technique is perfect for detecting rogue nodes in real time since it uses very little memory and energy. When protecting WSNs, the table gives a good overall picture of how well the algorithm worked. The Rabin-Karp method incorporates probabilistic hashing techniques to reduce the occurrence of wrong detection, which occurs when benign nodes are mistakenly identified as malicious. One typical cause of false positives is that these methods lessen the likelihood of hash collisions. The system uses an iterative verification procedure to investigate identified nodes before labeling them hostile. The effect of first-pass detection mistakes is mitigated. In addition, the system may improve its detection accuracy over time by learning from previous misclassifications via feedback mechanisms. A significant decrease in false positives has been seen in performance assessments, which adds to the network's overall dependability. Figure 7 shows a side-by-side comparison of the systems' misdetection and false detection rates.

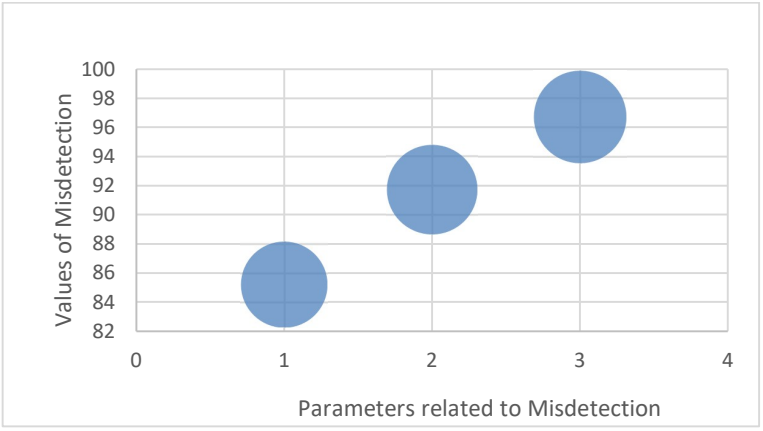

Fig.7 Misdetection and Wrong Detection Comparison

The prior method was far worse, with a rate of 5.6% for misdetection and 4.1% for incorrect detection. The current system's Rabin-Karp algorithm contributed to a 3.2% and 1.9% reduction, respectively. The enhanced and fine-tuned systems showed remarkable progress, with false positive rates falling to 0.2% and misdetection rates to 0.4%. These results demonstrate that the Rabin-Karp algorithm is a great fit for WSN malicious node identification since it successfully reduces false positives and negatives. The Rabin-Karp algorithm is compared to earlier approaches in Table 3, which show how they were utilized to identify rogue nodes in WSNs.

TABLE. 3 Comparison of Previous Methods and Rabin-Karp Algorithm

Metric	**Signature-Based**	**Anomaly-Based**	**Rabin-Karp Algorithm**
Computational Time	High	Moderate	Low
Detection Accuracy	85%	90%	98.70%
False-Positive Rate	5%	3%	1.30%
Energy consumption	High	Moderate	Low
Scalability	Moderate	Low	High

The comparison shows the algorithm's benefits, including scalability, precision, low false-positive rates, and lightweight architecture. The Rabin-Karp algorithm is more suited to real-time WSN settings than earlier techniques because it strikes a compromise between computing efficiency and security. The table makes it easy to see how the proposed technique is better than existing signature—and anomaly-based approaches.

IV. CONCLUSIONS

Identifying malicious nodes in WSNs poses problems with limited resources, processing data in real-time, and guaranteeing the stability of the network. The Rabin-Karp algorithm provides a lightweight and efficient method for hash-based verification, which helps to overcome some of these issues. On the other hand, there are certain restrictions, such as the need for precise hash values and the possibility of being susceptible to complex attacks that imitate legitimate packet behavior. The reduced processing overhead at the cluster head can be a problem even in resource-constrained settings. In vital applications, including healthcare monitoring, environmental sensing, and industrial automation, the Rabin-Karp algorithm may improve data quality, decrease false positives, and optimize energy utilization, making it a practical choice for protecting WSNs. Its small footprint guarantees performance-preserving scalability for larger-scale network installations. Integrating machine learning models to detect abnormal patterns and enhance detection accuracy is one example of advanced technology that might be explored to augment the Rabin-Karp algorithm. Further strengthening its application is using dynamic hash functions to fight developing attack techniques and adapting the approach to heterogeneous networks. One way to make it better at handling new security threats is to ensure it works with new WSN technology.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the present study.

REFERENCES

[1]. O. H. Abdulganiyu, T. A. Tchakoucht and Y. K. Saheed, "Towards an efficient model for network Intrusion Detection System (IDS): systematic literature review," Wireless Networks, vol. 30, no. 1, 2024, pp. 453–482.

[2]. S. Niktabe, A. H. Lashkari and D. P. Sharma, "Detection, characterization, and profiling DoH malicious traffic using statistical pattern recognition," International Journal of Information Security, vol. 23, no. 2, 2024, pp. 1293–1316.

[3]. M. Fayez, A. Abbas, H. Khaled and S. Ghoniemy, "Enhanced Aho-Corasick Algorithm for Network Intrusion Detection Systems," International Journal of Intelligent Computing and Information Sciences, vol. 24, no. 3, 2024, pp. 83–92.

[4]. R. Chowdhuri and M. K. Barma, "Node position estimation based on optimal clustering and detection of coverage hole in WSNs using hybrid deep reinforcement learning," The Journal of Supercomputing, vol. 79, no. 18, 2023, pp. 20845–20877.

[5]. M. M. Mustafa, A. A. Khalifa, K. Cengiz and N. Ivković, "An energy-efficient protocol for Internet of Things based wireless sensor networks," Computers, Materials & Continua, vol. 75, no. 2, 2023 pp. 2397–2412.

[6]. W. Liu, "An embedded microcontroller-based access authentication system of wireless sensor network," Journal of Cyber Security Technology, vol. 7, no. 1, 2023, pp. 1–20.

[7]. K. R. Babu, S. Saritha and K. G. Preetha, "An intelligent pattern matching approach with deep hypersphere model for secure big data storage in cloud environment," International Journal of Computer Information Systems and Industrial Management Applications, vol. 15, 2023, pp. 166–175.

[8]. P. Xiao, P. Lu, C. Luo, Z. Zhu and X. Liao, "Fast text comparison based on Elasticsearch and dynamic programming," International Conference on Web Information Systems Engineering, 2023, pp. 50–64.

[9]. N. Gagan, S. Sai Kumar, M. Keerthana, S. S. Vaidya and V. Rastogi, "Android malware detection," World Conference on Communication & Computing, 2023, pp. 1–14.

[10]. P. Leger, H. Fukuda, N. Cardozo and D. S. Martín, "Exploring a self-replication algorithm to flexibly match patterns," IEEE Access, vol. 12, 2024, pp. 13553–13570.

[11]. T. Liu, H. Huadong and R. Liu, "An intrusion detection framework with optimized feature selection and classification combination using support vector machine," 3rd International Conference on Electronic Technology, Communication and Information, 2023, pp. 182–186.

[12]. J. Xu, M. Ni, D. Zhu and X. Yu, "Overview of SQL injection attack detection techniques," Proceedings of the 2023 International Conference on Communication Network and Machine Learning, 2023, pp. 215–225.

[13]. M. Çelebi and U. Yavanoğlu, "Accelerating pattern matching using a novel multi-pattern-matching algorithm on GPU," Applied Sciences, vol. 13, no. 14, 2023, pp. 1–30.

[14]. M. Çelebi, A. Özbilen and U. Yavanoğlu, "A comprehensive survey on deep packet inspection for advanced network traffic analysis: issues and

challenges," Niğde Ömer Halisdemir Üniversitesi Mühendislik Bilimleri Dergisi, vol. 12, no. 1, 2023, pp. 1–29.

[15]. A. Hussain, K. N. Qureshi, K. Javeed and M. Alhussein, "An enhanced intelligent intrusion detection system to secure e-commerce communication systems," Computer Systems Science and Engineering, vol. 47, no. 2, 2023, pp. 2513–2528.

[16]. S. Chaudhari and R. Aparna, "Survey of image deduplication for cloud storage," System Research and Information Technologies, vol. 26, no. 4, 2023, pp. 113–134.

[17]. L. Ouarda, B. Malika and B. Brahim, "Towards a better similarity algorithm for host-based intrusion detection system," Journal of Intelligent Systems, vol. 32, no. 1, 2023, pp. 1–18.

[18]. I. Kim, D. Oh, M. K. Yoon, K. Yi, and W. W. Ro, "A distributed signature detection method for detecting intrusions in sensor systems," Sensors, vol. 13, no. 4, 2023, pp. 3998-4016.

[19]. V. Dagar, V. Prakash and T. Bhatia, "Analysis of pattern matching algorithms in network intrusion detection systems," 2nd International Conference on Advances in Computing, Communication, & Automation, 2016, pp. 1-5

[20]. A. K. Sharma and S. K. Mittal, "Cryptography & Network Security Hash Function Applications, Attacks and Advances: A Review," Third International Conference on Inventive Systems and Control, 2019, pp. 177-188.