Malicious Node Detection Using Machine Learning and Distributed Data Storage Using Blockchain in WSNs

Pawar Akash[1], Onkar Ladkat[2], Jadhav Tejas[3], Shendage Prashant[4],

1 Pawar Akash, Department Of Computer Engineering, Maharashtra, India
2 Onkar Ladkat , Department Of Computer Engineering, Maharashtra, India
3 Jadhav Tejas, Department Of Computer Engineering, Maharashtra, India
4 Shendage Prashant, Department Of Computer Engineering, Maharashtra, India
5 Prof. Arati Suryawanshi, , Department Of Computer Engineering, Maharashtra, India

ABSTRACT

Wireless Sensor Networks (WSNs) present unique opportunities for a broad spectrum of applications such as industrial automation, situation awareness, tactical surveillance for military applications, environmental monitoring, chemical or biological detection etc., Wireless Sensor Networks (WSNs) consist of hundreds of tiny nodes having the capability of sensing, computation and wireless communications. Deployed in a hostile environment, individual nodes of a wireless sensor network (WSN) could be easily compromised by the adversary due to the constraints such as limited battery lifetime, memory space and computing capability. It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes. However, it is challenging to secure the flat topology networks efficiently because of the poor scalability and high communication overhead. Since security and performance issues are a big concern in the case of wireless sensor networks, emphasis has been given to the scheme based on Weighted-Trust Evaluation (WTE). Extensive simulation is performed using MATLAB, to verify performance and efficiency of WTE by varying various parameters.

Keywords :- Malicious Node Detection, Machine Learning, Wireless Sensor Networks (WSNs),Blockchain Technology, Data Integrity, Data Security, Trustworthiness, Sensor Node Security, Network Security, Secure Data Retrieval.

INTRODUCTION

Wireless Sensor Networks (WSNs) consist of very small devices, called sensor nodes, that are battery powered and are equipped with integrated sensors, a data processing unit, a small storage memory, and short-range radio communication. Typically, these sensors are randomly deployed in the field. They form an unattended wireless network, collect data from the field, partially aggregate them, and send them to a sink that is responsible for data fusion. It is critical to detect and isolate the compromised nodes in order to avoid being misled by the falsified information injected by the adversary through compromised nodes. Sensor networks have applications in emergency-response networks, energy management, medical monitoring, logistics and inventory management, and battlefield management. In contrast to traditional wireless networks, special security and performance issues have to be carefully considered for sensor networks. For example, due to the unattended nature of sensor networks, an attacker could launch various attacks and even compromise sensor devices without being detected. Therefore, a sensor network should be robust against attacks, and if an attack succeeds, its impact should be minimized. In other words, compromising a single sensor node or few sensor nodes should not crash the entire network. Another concern is about energy efficiency. In a WSN, each sensor node may need to support multiple communication models including unicast, multicast, and broadcast. Therefore, due to the limited battery lifetime, security mechanisms for sensor networks must be energy efficient. Especially the number of message transmissions and the amount of expensive computation should be as few as possible. In fact, there are a numbers of attacks an attacker can launch against a wireless sensor network once a certain number of sensor nodes have been compromised [4]. In literature,

for instance, HELLO flooding attacks, sink hole attacks, Sybil attack, black hole attack, wormhole attacks, or DDoS attacks are options for an attacker. These attacks lead to anomalies in network behaviors that are detectable in general. There are some reported solutions to detect these attacks by monitoring the anomalies.

The purpose of the adversary is to mislead the operator with falsified data. This may lead to more serious consequences; for instance, in the battlefield a false report regarding the operations of the enemy may lead to extra casualties.

LITERATURE SURVEY

1.Ganeriwal et al. proposed a reputation-based framework for data integrity in WSNs. The proposed reputation system takes information collected by each node using a Watchdog mechanism (for direct monitoring and observations) to detect invalid data and uncooperative nodes.

2.Yao et al. proposed a parameterized and localized trust management scheme for WSN security, particularly for secure routing, where each node only maintains highly abstracted parameters to evaluate its neighbours.

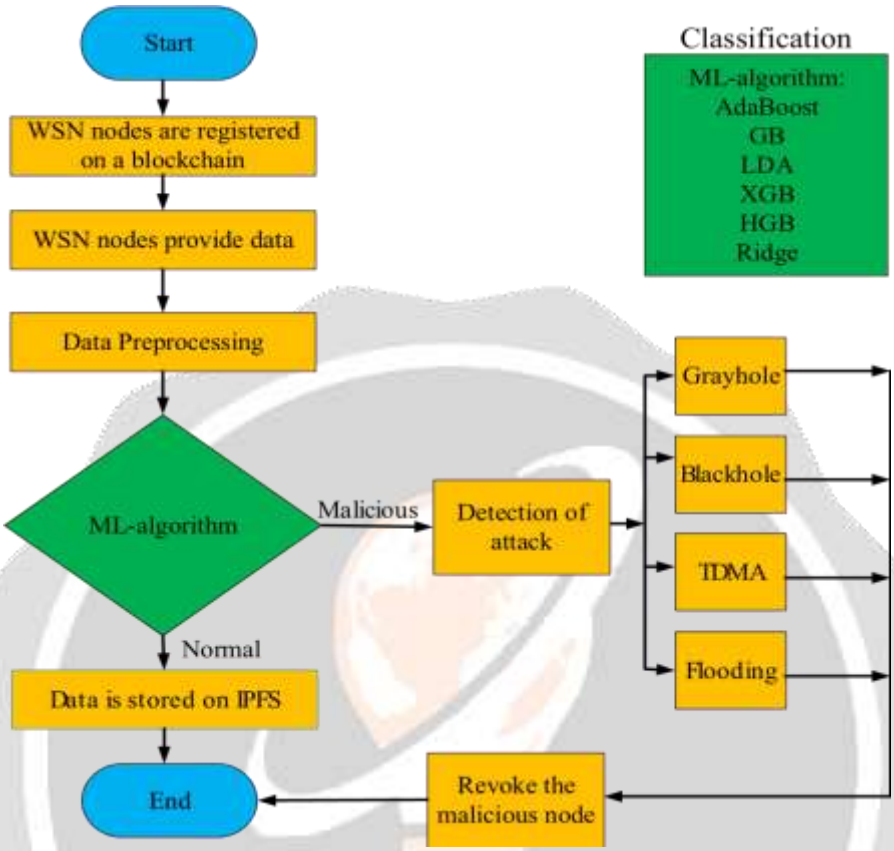
3.Aivaloglou and Gritzalis proposed a hybrid trust and reputation management protocol for WSNs by combining certificate-based and behaviour-based trust evaluations. However, [1-3] cited above only considered a node's QoS property in trust evaluation. Also, the analysis was conducted based on a flat WSN architecture which is not scalable.

4&5Liu et al. [4] and Moraru et al. proposed trust management protocols and applied them to geographic routing in WSNs. However, no hierarchical trust management was considered for managing clustered WSNs. Their work again evaluated trust based on QoS aspects only such as packet dropping and the degree of cooperativeness, while our work considers both QoS and social trust for trust evaluation of a SN.

6. Capra et al. discussed the notion of human trust which could be formed from three sources: direct experiences, credentials, and recommendations. In particular recommendations are trusting information coming from other nodes in the social context. We consider only two sources in our notion of trust, namely, direct experiences and recommendations since it is hard for SNs with limited resources to carry credentials. A significant difference of Capra's work from our work is that we specifically consider individual QoS and social trust property, say, X, and devise specific trust aggregation protocols using direct experiences and recommendations to form trust property X, while Page 22 Capra used the three sources of information to form human trust. Moreover, because different trust properties have their own intrinsic trust nature and react differently to trust decay over time, we identify the best way for each trust property X to take in direct experiences and recommendations information so that the assessment of trust property X would be the most accurate against actual status in trust property X. Another significant difference is that we consider trust formation as the issue of forming the overall "trust" out of individual social and QoS trust properties, while Capra considered it as the issue of forming human trust out of the three sources of trust information.

7.H. Hu et al. proposed weighted trust approach here each SN has a weight associated with it representing the trustworthiness of its sensor reading output. The system periodically calculates the average sensor reading output by taking a weighted summation out of all sensor reading outputs. The weight associated with a SN is dynamically updated according to the deviation of the SN's output from the average output. A larger deviation results in a lower weight. Once the weight of a SN falls below a threshold, the SN is considered a malicious node. The main drawback of this approach is a high false positive probability may result.

URL DIAGRAM



SYSTEM IMPLEMENTATION PLAN

1.Project Initiation: - Define the project's objectives, scope, and success criteria. - Assemble a cross-functional project team with expertise in Machine Learning, Blockchain, and WSNs.

2.Requirements Gathering: - Collaborate with stakeholders to gather detailed requirements for malicious node detection and data storage**.**

3.System Architecture Design: -Design a comprehensive system architecture that integrates Machine Learning and Blockchain components.

- Define the interactions and data flow between these components.

4.Data Collection and Preprocessing: -Set up data collection mechanisms from WSN nodes. Preprocess the collected data, including data cleaning, normalization, and feature extraction.
5.Machine Learning Model Selection: - Choose appropriate Machine Learning algorithms (e.g., Random Forest, SVM, Deep Learning) for malicious node detection.

- Prepare labelled data for training and testing the Machine Learning models

6.Training and Evaluation: - Train the Machine Learning models on historical data.

- Evaluate model performance using metrics like accuracy, precision, recall, and F1-score.

7. Real-time Detection Implementation: - Integrate the trained Machine Learning models into the system for real-time malicious node detection.

- Implement alerting mechanisms for immediate response to threats.

8. Blockchain Integration: - Set up a Blockchain network for secure and distributed data storage.

- Develop and deploy smart contracts for data management on the Blockchain.

9. Sensor Node Integration: - Develop firmware or software for WSN nodes to communicate with the system.

- Ensure secure data transmission from sensor nodes to the system.

10. Data Storage and Retrieval: - Implement mechanisms to securely store sensor data on the Blockchain.

- Enable authorized users to retrieve and verify stored data

11. Security Measures: - Implement encryption and access control to protect data and system integrity.

- Consider zero-trust security principles.

12.User Interface: - Develop a user-friendly interface for system administrators to monitor system status and alerts.

13.Testing and Validation: - Conduct extensive testing, including simulation and real-world testing with sensor nodes.

- Verify the system's accuracy in detecting malicious nodes and data storage integrity using Blockchain.

14.Performance Optimization: - Optimize system performance for resource efficiency and scalability.

15.Documentation: - Create detailed documentation for system architecture, data flows, and operational procedures.

16.Training and Knowledge Transfer: - Provide training to system administrators and end-users.

- Ensure knowledge transfer within the team.

17.Deployment Plan: - Develop a deployment plan that considers hardware, network, and environmental requirements

18.Maintenance and Updates: - Establish a plan for ongoing system maintenance, updates, and security patches.

19.Monitoring and Alerts: - Set up continuous monitoring of the system's health and performance.

- Configure alerts for potential issues and security breaches.

20. Compliance and Regulations: - Ensure compliance with relevant data protection and privacy regulations.

21. Data Backup and Recovery: - Implement a backup and recovery strategy for Blockchain data.

22.User Acceptance Testing: - Conduct user acceptance testing to gather feedback and make necessary improvements.

23.Deployment and Rollout: - Deploy the system in the production environment, following the deployment plan.

24.Post-Deployment Review: - Conduct a post-deployment review to evaluate the system's performance in a real-world scenario.

CONCLUSION

In this paper, we proposed a novel WTE based algorithm for the detection of malicious SNs in WSNs. Here, Weighted Trust Evaluation (WTE) is introduced for solving the Byzantine problem which occurs in Wireless Sensor Networks. The basic idea is that a weight representing the reliability of a node is assigned to each SN in the cluster under a FN. Since malicious nodes usually report falsified information to disrupt the network, if a node sends incorrect information, the FN gradually decreases the weight of the node and detect the node as a malicious node when its weight value becomes lower than a threshold. In addition, a weight recovery mechanism is incorporated in the algorithm to recover the weight of a node whose weight is accidentally decreased. The network area is divided into square grids and malicious nodes are detected locally in a distributed manner. For a relatively small event region located across multiple adjacent grids, inter-grid communication is partially employed to enhance the event detection accuracy. Confidence levels (weights) are used to reflect the behaviour of sensor nodes in reporting their readings in decision- making. Once the weights reach a predefined lower- bound, the corresponding nodes are logically isolated from the rest of the network. Thresholds are properly chosen to achieve high malicious node detection accuracy without sacrificing normal nodes.

REFERENCES

1.Ganeriwal, L. K. Balzano, and M. B. Srivastava, “Reputation-based framework for high integrity sensor networks,” ACM Transactions on Sensor Networks, vol. 4, no. 3, pp. 1–37, May 2008

2.Z. Yao, D. Kim, and Y. Doh, “PLUS: parameterized and localized trust management scheme for sensor networks security,” in Proceedings of IEEE International Conference on Mobile Ad-hoc Sensor networks, pp. 437–446, 2006.

3. E. Aivaloglou and S. Gritzalis, “Hybrid trust and reputation management for sensor networks,” Wireless Networks, vol. 16, no. 5, pp. 1493–1510, July 2010

4&5. Z. Yao, D. Kim, and Y. Doh, “PLUS: parameterized and localized trust management scheme for sensor networks security,” in Proceedings of IEEE International Conference on Mobile Ad-hoc Sensor networks, pp. 437–446, 2006.

6.L. Capra and M. Musolesi, “Autonomic trust prediction for pervasive systems,” in Proceedings of International Conference on Advanced Information and Network Applications, pp. 1–5, 2006

1.

7. H. Wang, P. Tu, P. Wang, and J. Yang, ‘‘A redundant and energyefficient clusterhead selection protocol for wireless sensor network,’’ in *Proc. 2nd Int. Conf. Commun. Softw. Netw.*, 2010, pp. 554–558, doi: 10.1109/ICCSN.2010.46.