DEVELOPMENT OF WIRELESS SENSOR NETWORK MIDDLEWARE USING MACHINE LEARNING APPROACH

Dr. Sangamesh J. Kalyane
Department of CSE BKIT Bhalki, Karnataka State
E-mail: kalyanesangamesh@gmail.com

Abstract

Security challenges with wireless sensor networks (WSNs) have yet to be entirely resolved, despite their extensive usage in a number of applications. To solve these limits, middleware is often employed as an intermediary layer among WSNs and end users. Most existing middleware, on the other hand, has yet to safeguard data during transmission against malicious or unexpected assaults. This research proposes an intelligent middleware based on the unsupervised learning methodology of Generative Adversarial Networks (GANs). A GAN has two networks: a generator (G) and a detector (D) (D). To fool the attacker, the G creates fake data that looks like actual sampling and blends it with real sensor data. The D is made up of many layers that can distinguish between true and false input. This approach produces a real-world analysis of the data that is safely sent over the WSN. Python is used to develop the platform and keras is used to run the experiments. The findings demonstrate that the suggested strategy not only enhances data accuracy but also data security by avoiding data manipulation. When compared to traditional data transfer, transmitting data from the WSN to the end user becoming significantly better secure and efficient

Keywords: Machine Learning, GAN, Security, WSN

1. INTRODUCTION

Wireless sensor networks (WSNs) are becoming more widely used in scientific and industrial sectors, pushing the frontiers of computer and systems engineering research [1]. To make node-to-node and device-to-device wireless communication easier, WSNs employ a slew of sensor nodes. Distributed data collection and processing systems can be built using resource-constrained sensor nodes. Signal processing, aggregation of data, QoS, and wireless communications are all handled by WSNs. In recent years, safe communication among WSNs has been a difficulty [2]. Due to their low-capacity sensors, WSNs create a huge volume of data, which causes crucial data to be lost during transmission. Sensor nodes are constrained by a variety of factors, including security, data aggregation, middleware needs, power usage, and sensor network diversity. In prior study, using middleware as an intermediary layer among WSNs and end users was found to provide a solution to the aforementioned challenges. The middleware connects the applications with the hardware components of the WSNs. The middleware, which also offers temporary storage, controls the sensor data nodes. [3]. The capability of the middleware to synchronise newer and older nodes enables it to be more effective while supporting a wide range of resources. This guarantees that network performance is not harmed as much as possible when changes are implemented [4]. Because wireless data is sensitive, it is subject to unauthorized access. Secure communication within WSNs is enabled by security features such as allocation of resources and energy availability. Enabling end-to-end security audits could also assist to ensure that nodes are connected securely.

The rest of this paper is organized as follows. In section II describe the related work carried out. Section III describes the motivation of the paper, in section IV describes the propose methodology, the performance analysis and relative simulation are conducted in section V. Finally we draw the conclusion on the proposed scheme in section VI.

2. LITERATURE SURVEY

[1] WSNs have been used in monitoring systems to manage and monitor a variety of interior premises, agricultural areas, and forest monitoring applications during the previous ten years. The most significant issue with WSNs is network security, which is becoming increasingly popular as the number of these devices develops. For base station (BS) safety, cryptography, detection systems, and security placement and routing in WSNs, traditional security measures have been applied. Many researchers have discovered effective solutions to the security issues that WSNs face. These sensors create a lot of data, which needs to be processed and sent to the BS. Due to power constraints, restricted memory (storage density), connectivity, and sensor resource limits, secure authentication methods aren't viable for WSNs.

[2] Due to network energy usage, communication and information transfer among sensors is a major difficulty. This data must be safeguarded against a range of threats. Supporting security qualities like as confidentiality, authenticity, availability, and integrity should be used to protect networks. The bulk of these sensors are not physically shielded, resulting in nodes that have been hacked. When a network's one maybe more nodes are hacked, the adversary might execute a variety of assaults to disrupt network connection. A variety of assaults exist, including adversary, hacked node(s), eavesdropper, and so on. These attacks can cause packets to be lost or altered,

compromising the WSNs' performance. SLPs (source location privacy) are ways for masking sensor data from attackers by generating phoney nodes. The false nodes and packets (dummy message) generate phoney identification and packets, resulting in phoney identities and packets. This strategy has the drawback of requiring more work and overhead.

[3]. v. Dervis Karaboga et al. [15] created an ABC algorithm based on honey bee foraging behaviour and found that it is more effective for node energy optimization. Although the binary detection model is only used to analyse dynamic deployment, this technique is applicable for both dynamic and static deployment.

[4]. Anastasi G et al. [16] explored the issue of rising energy consumption due to node partitioning, as well as the aim of energy conservation in nodes.

3. MOTIVATION

The need for secure data transport has surged as WSNs become more frequently used in industrial, medical, and military applications. The importance of middleware in WSNs has been addressed in a recent research. However, many of these solutions fall short of addressing the safety issue, leaving data and communication at risk. Such information is often sensitive, and it must be protected from attacks and the risk of being disclosed.

The constraints of existing WSN middleware prompted this research, which is based on the following reasons to increase WSN middleware efficiency:

a. The suggested techniques deliver a one-of-a-kind WSN middleware that uses intelligent, unsupervised machine learning to manage and monitor sensor data. By upgrading and eliminating unnecessary data from the sensors, power usage and overhead may be reduced. The proposed unsupervised middleware learning solves this problem.

b. It is offered as a remedy to the problem. They provide a large-scale WSN-capable unsupervised learning approach as well as a complete security policy. GANs are a powerful approach for creating intelligent generator and detector networks that employs game theory. Therefore in study, the advantages of the strong generator model (G) are realised.

The suggested GANs are predicated on a minimax game, in which each agent tries to maximise probability while the other tries to reduce it. G's capacity to produce new data that is comparable to genuine samples has increased as a consequence. The purpose is to make it difficult for the attacker to distinguish between fake information from the generator and actual data from the sensors and database. By raising the likelihood of real data to 1 and reducing the likelihood of bogus data (from the G or an attacker) to 0, the D distinguishes among genuine and attack data.

4. METHODOLOGY

The generator network produces simulated information that is extremely comparable to the input samples' source data. As a result of the false data being mixed in with the actual data, the attackers can't detect the difference (from sensors). With no need to create false packets or information to trick the intruders in this instance, resulting in a significant reduction in power consumption.

MACHINE LEARNING FOR WSNs

Machine learning (ML) approaches were also implemented via middleware platforms. In [13], WSN was investigated using an unsupervised learning approach called self-organizing map (SO-M). This development's goal is to address the issue of spotting network threats in ad hoc networks. SOM has the disadvantage of being unable to identify assaults in a complicated environment and huge datasets, such as those seen in WSNs. Machine learning middleware was developed to solve the issue of ontology heterogeneity (Ma-ML). On the other hand, Ma-overhead ML's may be a worry. Due to system design issues, a WSN's dynamic behaviour has been gradually improved. WSNs employ ML methods to obviate the need for a network overhaul [14]. Machine learning, according to sensor network designers, is an algorithm and a data collecting tool used to build predictive model. To extend the network's life, machine learning enhances resource allocation, utilisation, and delegation.

To do artificial intelligent data sampling, ML employs statistical formulas based on statistical techniques. It develops the ability to respond to a continually changing environment.

In WSN applications, ML interface approaches are critical. The three phases of ML interfacing are data processing, aggregation of data, and interfacing [26]. These phases are used to track and mimic the changing environment that WSNs create.

4.1. Proposed System

1) The proposed techniques result in a one-of-a-kind WSN middleware that uses intelligent, unsupervised machine learning to manage and monitor sensor data. By upgrading and filtering superfluous data from the sensors, energy consumption and overhead may be decreased. Unsupervised learning is the way to solve this issue.

2) The generating network makes false data from the provided sample that is very close to the genuine data. The attackers can't tell the difference since the bogus data is mixed in with the actual data from sensors. There is no need to construct false packets or data to deceive the intruders in this situation, resulting in a huge reduction in power.

3) Multiple analytical models are created: The confusion matrix, visualization, and several CNN layers all validate the correctness of the recommended approach.

4) For verification, we present a complete comparison of the suggested technique with existing approaches such as ML-WSN. The following measurements are used to create a comparison: Average energy consumption, packet delivery ratio, end to end delay and throughput are also important considerations.

4.2 Block Diagram

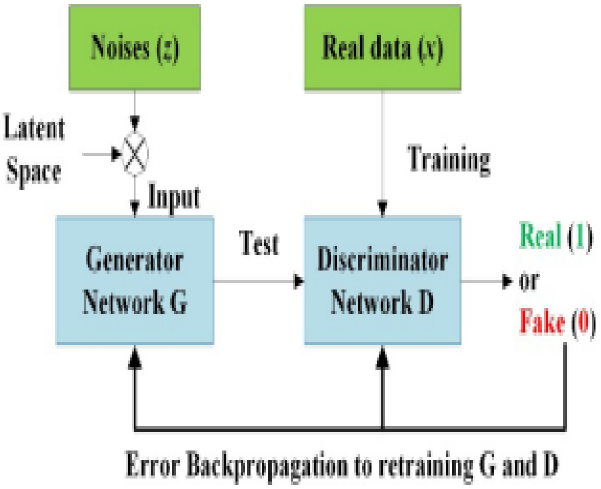

Fig1. Proposed Block Diagram

We provide a GAN-based intelligent security solution for WSN middleware, which enhances typical middleware in terms of security mechanisms, handling of various sensor node characteristics, and filtering and forwarding only real data. This is the first time we've seen the GANs approach used to tackle a security concern in WSN's middleware, to our knowledge. WSNs' middleware also uses a GAN to filter and detect inconsistencies in the suggested data contribution.

5. IMPLEMENTATION

A. Dataset

Several traditional classifiers can't tell the difference among legal and malicious signals. The benchmark N-SL-KDD dataset [12] is used to detect any intrusion in the sensors' data in the system. In both normal and malicious data flow, the N-SL-KDD has an imbalance of classes. The ratio of threats to normal traffic is rather minimal. The Class Imbalance Problem (CIP) is defined as when routine traffic outnumbers attack traffic. When the minority attack class, also known as the attack class, has a substantially lower representation than the mainstream attacking class, or normal assault class, this occurs. The attack traffic benefits from the CIP, and the intrusion detection system misses it. As a result, specialised tactics for repelling such an onslaught, with a focus on minority groups, are urgently needed. Using the suggested generator model, the suggested solution eliminates the imbalance problem. The primary distinction among this model and other strategies is that by feeding only one feature vector from the database to the generator, the generator produces balanced data that is more realistic of real data. The discriminator then uses this characteristic as feedback, allowing it to discriminate among false data (corresponding to 0) and true data (corresponding to 1).

B. Network Generator

From a particular sample, the suggested generator network (G) is utilised to generate numerous assaults data (false) (accessibility to real-time information). Generator, critically, does not have direct access to the real data (dataset); instead, G learns from D. The discriminator gets accessibility to both the original data in the dataset and the data generated from it. The G uses the mistake back propagation discoveries to retrain the generator, allowing it to produce higher-quality fake data.

C. Discriminator Network

To identify which is which, the discriminator D analyses genuine (authentic) and fraudulent data. Both the G and D networks are trained simultaneously and in competition with one another. As a result, the discriminator has access to the dataset's real and fictional data. The D retrains and updates the system based on the findings of error back propagation over 150 iterations, allowing it to distinguish between true and false data.

5.1 Performance Analysis

The performance of secure wireless sensor network middleware (SWSNM), our proposed GAN-based wireless sensor middleware, is evaluated in this section, and it is compared to the ML-WSN and SLEACH technique in terms of energy utilization and throughput, end to end to end delay. The sensing and idle nodes consume 10.2 mW and 0.42 mW of power, correspondingly. It's possible that the simulation will take up to 45 minutes. There are 12 mobile nodes and 138 static nodes in the network. Malicious nodes are supposed to be those that trash all packets passing through them. When the algorithm finds that packets have been lost, it flags those nodes as malicious. Each malicious node's position inside the network is determined, and all those nodes are changed regularly.

5.1.1 Energy Consumption

The energy consumption is defined as the average amount of energy consumed by nodes in the network during Tx, Rx, sensing etc. the proposed work is compared with conventional techniques ML-WSN [11] and SLEACH [17] . The network's energy usage is dramatically decreased when malicious nodes are deployed along with fixed nodes. The proposed technique considers energy consumption during transmission/reception of data, as well as sleeping and inactivity. The energy use is calculated using Equation 1. When node j is active, we assume that the energy expended is used to transmit and receive bits of packets. Furthermore, In both sleep and idle stages, the overall number of nodes within the network is counted, with n denoting the total number of nodes within the network.

$$\sum_{j=1}^{n} \frac{\text{Total energy consumed at node}_j}{n} \qquad (1)$$

The removal of rogue nodes appears to reduce the network's energy consumption. The energy consumption curve of the ML-WSN [11] is rather interesting. While energy consumption is significantly lower in networks with less than 20-30 nodes (a little more than 30), networks with 40 to 100 nodes consume significantly more energy. In terms of energy utilization, this shows that the ML-WSN and SLEACH approach is only reasonable for a tiny number of nodes but proposed work not only provides reliable transmission also provides less energy consumption. Figure 2 depicts the proposed scheme's energy usage.

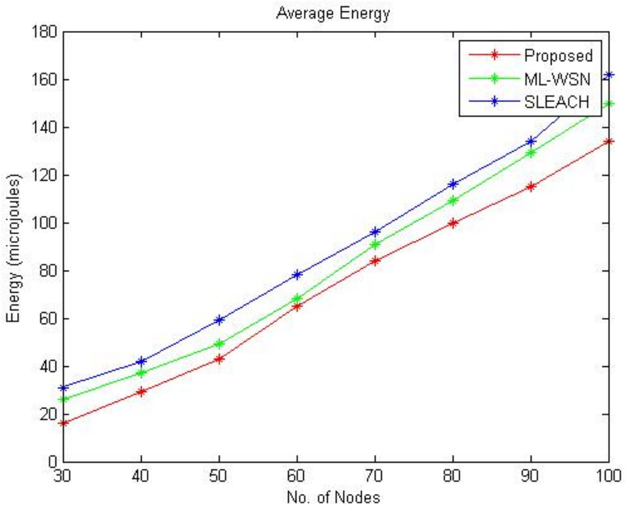

Fig 2.Energy Consumption

5.1.2 Throughput

As indicated in equation 2, throughput is the quantity of data delivered from source nodes to destination or base station in a given length of time. The network without malicious nodes has considerably greater performance than the network with malicious nodes when comparing network throughput for each of the three situations. By a large margin, the proposed work outperform the ML-WSN and SLEACH . Figure 3 depicts the throughput of the proposed work in comparison with conventional methods.

$$\text{Throughput} = \frac{\text{Number of bytes received at BS}}{\text{Total bytes transmitted at source nodes}} \quad (2)$$

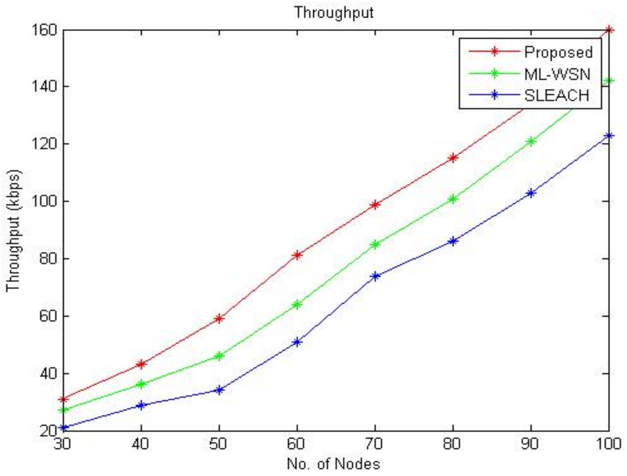

Fig3. Throughput

5.1.3 Packet Delivery Ratio

The packet delivery ratio is shown in Figure 4, and it can be observed that malicious activity in the network causes packet drops. Our technique is capable of preventing fraudulent behaviour and outperforms traditional schemes in terms of packet delivery ratio. We introduce selected forward attack nodes and examine their behaviour in this article. In the presence of malicious nodes, the CH will delete any node transmitting packets that are not authorised until the sensor nodes' keys are authenticated, and the CH will not initiate connection between them until this is validated. Our approach achieves a higher PDR and delivers more authentic packets, as seen in the diagram below.

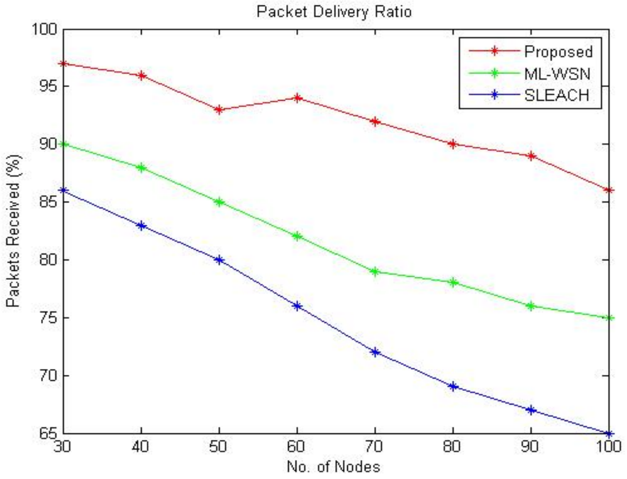

Fig 4.Packet Delivery Ratio

5.1.3 End to End Delay

Due to malicious activity in the network these are possibility of delay in the overall network activity which leads to latency in the network. Figure 5 depicts the average end-to-end delay findings;we see that our system has a shorter delay than ML-WSN and S-LEACH. We look at how long it takes to generate and validate keys in terms of cryptographic operations. In comparison to the other two techniques, the suggested system takes less time to produce keys for independent nodes.

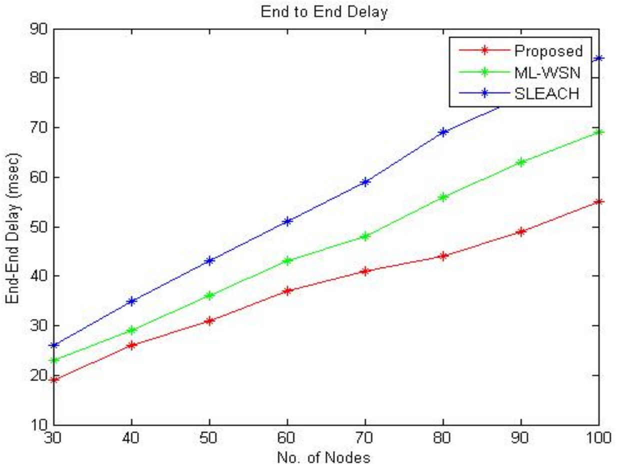

Fig 5 End to End Delay

6. CONCLUSION

The proposed approach SWSNM employs a generator (G) and a discriminator (D). The generator may create fictional data to trick the attacker, as well as resolve data imbalances by producing fresh data to restore balance. We demonstrate that the D is a strong network capable of readily distinguishing between two datasets, even when the false data is highly similar to the actual data. Extensive testing on the NSL-KDD dataset with various supervised learning techniques and comparisons with our proposed work show that when using full (40 features) and reduced (20 features) features, our proposed model provides better accuracy up to 86.5 percent with a low FPR of 21% and 84 percent with a low FPR of 13 percent, respectively. In addition, we used the t-distributed stochastic neighbour embedding (t-SNE) to compare the output of our proposal to the original dataset for both full and reduced features. The recommended proposed excels in data visualisation, packet delivery ratio, end to end delay, energy consumption and throughput when compared with conventional technique that is MLWSN and SLEACH

ACKNOWLEDGEMENT

I would like to thank Visvesvaraya Technological University, Belagavi for providing the required facility to complete my Ph.D within stipulated time and also i would like to thanks Dr. Prashant Sangulgi, Associate Professor BKIT Bhalki, for supporting me to bring this paper successful.

REFERENCES

[1] K. A. Bispo, N. S. Rosa, and P. R. F. Cunha, ‘‘SITRUS: Semantic infrastructure for wireless sensor networks,’’ Sensors, vol. 15, no. 11, pp. 27436–27469, 2015.

[2] G. Xu, W. Shen, and X. Wang, ‘‘Applications of wireless sensor networks in marine environment monitoring: A survey,’’ Sensors, vol. 14, no. 9, pp. 16932–16954, 2014.

[3] S. Hadim and N. Mohamed, ‘‘Middleware for wireless sensor networks: A survey,’’ in Proc. 1st Int. Conf. Commun. Syst. Softw. Middleware, New Delhi, India, Jan. 2006, pp. 1–7.

[4] R. Alshinina and K. Elleithy, ‘‘Performance and challenges of serviceoriented architecture for wireless sensor networks,’’ Sensors, vol. 17, no. 3, p. 536, 2017. [5] J. Al-Jaroodi and A. Al-Dhaheri, ‘‘Security issues of service-oriented middleware,’’ Int. J. Comput. Sci. Netw. Secur., vol. 11, no. 1, pp. 153–160, 2011.

[6] A. Shchzad, H. Q. Ngo, S. Y. Lee, and Y.-K. Lee, ‘‘A comprehensive middleware architecture for context-aware ubiquitous computing systems,’’ in Proc. 4th Annu. ACIS Int.

Conf. Comput. Inf. Sci. (ICIS), Jeju, South Korea, Jul. 2005, pp. 251–256.

[7] Y. Wang, G. Attebury, and B. Ramamurthy, ‘‘A survey of security issues in wireless sensor networks,’’ IEEE Commun. Surveys Tuts., vol. 8, no. 2, pp. 2–23, 2nd Quart., 2006.

[8] Y. W. Law, J. Doumen, and P. Hartel, ‘‘Benchmarking block ciphers for wireless sensor networks,’’ in Proc. IEEE Int. Conf. Mobile Ad-Hoc Sensor Syst., Fort Lauderdale, FL, USA, Oct. 2004, pp. 447–456.

[9] J. Newsome, E. Shi, D. Song, and A. Perrig, ‘‘The sybil attack in sensor networks: Analysis & defenses,’’ presented at the 3rd Int. Symp. Inf. Process. Sensor Netw., Berkeley, CA, USA, 2004.

[10] A. A. Pirzada and C. McDonald, ‘‘Secure routing with the AODV protocol,’’ in Proc. Asia–Pacific Conf. Commun., Perth, WA, Australia, Oct. 2005, pp. 57–61.

[11] K. Lingaraj, R. V. Biradar, and V. C. Patil, ‘‘Eagilla: An enhanced mobile agent middleware for wireless sensor networks,’’ Alexandria Eng. J., to be published, doi: 10.1016/j.aej.2017.03.003

[12] L. M. Ibrahim, D. T. Basheer, M. S. Mahamod. A Comparison Study for Intrusion Database (KDD99, NSL-KDD) Based on Self Organization Map (SOM) Artificial Neural Network. Journal of Engineering Science and Technology, vol. 8, No.1 (2013), pp. 107-11

[13] Lapidot, I., Guterman, H. and Cohen, A., 2002. Unsupervised speaker recognition based on competition between self-organizing maps. IEEE Transactions on Neural Networks, 13(4), pp.877-887.

[14] Kumar, D. Praveen, Tarachand Amgoth, and Chandra Sekhara Rao Annavarapu. "Machine learning algorithms for wireless sensor networks: A survey." Information Fusion 49 (2019): 1-25.

[15] Karaboga, Dervis, and Bahriye Basturk. "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm." Journal of global optimization 39, no. 3 (2007): 459-471.

[16] Anastasi, G., Conti, M., Di Francesco, M., & Passarella, A. (2009). Energy conservation in wireless sensor networks: A survey. Ad hoc networks, 7(3), 537-568.

[17] Masdari, Mohammad, Sadegh Mohammadzadeh Bazarchi, and Moazam Bidaki. "Analysis of secure LEACH-based clustering protocols in wireless sensor networks." Journal of Network and Computer Applications 36, no. 4 (2013): 1243-1260