Journal of
Low Power Electronics and Applications

MDPI

Article

Distributed Consensus Gossip-Based Data Fusion for Suppressing Incorrect Sensor Readings in Wireless Sensor Networks

Martin Kenyeres [1,*], Jozef Kenyeres [2] and Sepideh Hassankhani Dolatabadi [1]

[1] Institute of Informatics, Slovak Academy of Sciences, Dúbravská Cesta 9, 845 07 Bratislava, Slovakia; sepideh.ui@savba.sk
[2] Frequentis AG, Innovationsstraße 1, 1100 Vienna, Austria; jozef.kenyeres@frequentis.com
* Correspondence: martin.kenyeres@savba.sk

Abstract: Incorrect sensor readings can cause serious problems in Wireless Sensor Networks (WSNs), potentially disrupting the operation of the entire system. As shown in the literature, they can arise from various reasons; therefore, addressing this issue has been a significant challenge for the scientific community over the past few decades. In this paper, we examine the applicability of seven distributed consensus gossip-based algorithms for sensor fusion (namely, the Randomized Gossip algorithm, the Geographic Gossip algorithm, three initial configurations of the Broadcast Gossip algorithm, the Push-Sum protocol, and the Push-Pull protocol) to compensate for incorrect data in WSNs. More specifically, we consider a scenario where the sensor-measured data (measured by a set of independent sensor nodes) are skewed due to Gaussian noise with a various standard deviation σ, resulting in discrepancies between the measured values and the true value of observed physical quantities. Subsequently, the aforementioned algorithms are employed to mitigate this skewness in order to improve the accuracy of the measured data. In this paper, WSNs are modeled as random geometric graphs with various connectivity, and the performance of the algorithms is evaluated using two metrics (specifically, the mean square error (MSE) and the number of sent messages required for an algorithm to be completed). Based on the presented results, it is identified that all the examined algorithms can significantly suppress incorrect sensor readings (MSE without sensor fusion = -0.42 dB if $\sigma = 1$, and MSE without sensor fusion = 14.05 dB if $\sigma = 5$), and the best performance is achieved by PS in dense graphs and by GG in sparse graphs (both algorithms achieve the maximum precision MSE = -24.87 dB if $\sigma = 1$ and MSE = -21.02 dB if $\sigma = 5$). Additionally, the performance of the analyzed distributed consensus gossip algorithms is compared to the best deterministic consensus algorithm applied for the same purpose.

Keywords: consensus; data aggregation; distributed computing; Gaussian noise; gossip algorithms; incorrect data; information aggregation; information fusion; sensor fusion; sensor networks

check for updates

Academic Editor: Pak Kwong Chan

Received: 11 December 2024
Revised: 5 January 2025
Accepted: 24 January 2025
Published: 26 January 2025

Citation: Kenyeres, M.; Kenyeres, J.; Hassankhani Dolatabadi, S. Distributed Consensus Gossip-Based Data Fusion for Suppressing Incorrect Sensor Readings in Wireless Sensor Networks. *J. Low Power Electron. Appl.* **2025**, *15*, 6. https://doi.org/10.3390/jlpea15010006

Copyright: © 2025 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Incorrect sensor readings are considered a significant threat to Wireless Sensor Networks (WSNs) because they can lead to a range of serious issues, such as increased power-energy consumption, misleading analyses, decreased reliability, reduced data precision, and even disrupt the operation of the entire system [1–3]. Therefore, ensuring high precision in sensor-measured data is one of the most critical aspects for the effective operation

of WSNs [4]. Over the past few decades, numerous researchers have focused on how WSN-based applications can utilize sensor data with greater confidence while minimizing power-energy consumption, despite the presence of incorrect sensor readings. As shown in the literature, there are several approaches to minimizing this issue (e.g., adaptive sampling, environmental compensation, distributed filtering, etc.) [5,6]. However, this remains a challenging and relevant task, as sensors are vulnerable to various environmental factors, collect highly correlated data, and sensor nodes themselves are prone to defects. The application of sensor fusion algorithms is one way to compensate for incorrect sensor readings [4]. Their goal is to generate more meaningful information than a single node can provide, thereby ensuring improved quality of service (QoS) and optimized power-energy consumption. The main objective of this paper is to analyze the applicability of seven distributed consensus gossip algorithms, which were originally designed to converge to a consensus on the average values read by all sensors in WSNs, in suppressing negative factors represented by Gaussian noise. We assume in this work that measurements at each sensor node are affected by Gaussian noise (representing all negative factors affecting sensor readings) with a zero mean and a varying standard deviation (as commonly modeled in related studies). Other negative factors are not considered in our analysis. As demonstrated in the literature review, this model effectively represents many real-world datasets of natural phenomena and events.

1.1. Wireless Sensor Networks

WSNs, a key component of the Internet of Things (IoT), are a contemporary information technology for remotely monitoring and controlling various physical environments [7–9]. They are formed by (potentially) hundreds to thousands of spatially distributed autonomous tiny entities (known as sensor nodes) that communicate with each other via wireless channels and operate in possibly highly noisy environments [10,11]. These sensor nodes are capable of sensing various physical quantities from their surroundings (e.g., temperature, sounds, pollution, wind, blood pressure, light intensity, etc.), whereby WSNs have attracted considerable attention from both academia and industry over the past few decades [12–17]. As discussed in [18–20], WSNs support a variety of standards and protocols, including Zigbee, BLE, LoRa, IEEE 802.15, etc. These networks are typically structured in an architecture where sensor nodes continuously monitor specific physical quantities from their surrounding environment and transmit the collected data to a base station [21]. The base station supervises the other nodes and relays relevant information to end users via the Internet [21].

As shown in the literature [22–29], WSNs have plenty of advantages, thereby finding a wide range of applications across various fields such as health care, security, environmental monitoring, agriculture, industry, forest fire detection, animal tracking, flood detection, etc. Below, we focus on the most notable advantages of WSNs [30–35]:

- ◇ Scalability;
- ◇ Financial affordability;
- ◇ Energy efficiency;
- ◇ Monitoring in real time;
- ◇ Infrastructure-less architecture;
- ◇ Compatibility.

Below, we describe each of the listed advantages. “Scalability” means that sensors can be easily added to and removed from a network. Thus, high scalability allows for an effective increase or decrease in the number of sensor nodes without degrading the performance stability and reliability. “Financial affordability” is one of the most critical design considerations for WSNs, as it is necessary to keep sensor production costs low.

Thus, WSN-based applications are typically cost-effective in spite of the deployment of numerous sensor nodes in real-world applications. "Energy efficiency" is important, since algorithms and sensor nodes are designed to enable long operation of WSNs without any maintenance (or with minimal maintenance). As a result, the network lifetime of WSNs is significantly extended, allowing for their widespread use in vast and hard-to-reach areas. "Monitoring in real time" means that sensor nodes can reliably sense the current status of monitored physical quantities, thereby providing topical information on these quantities and enhancing data collection and decision-making processes. "Infrastructure-less architecture" guarantees a simplified deployment and redeployment of sensor nodes, as well as their seamless reconfiguration. "Compatibility" means that WSNs are compatible with a variety of devices and innovative plug-ins, enhancing their applicability across multiple domains.

As stated above, sensor nodes are typically cheap, tiny, and low-power devices, which limits many of their capabilities [36]. These limitations lead to many drawbacks of WSNs, with the most significant ones outlined below [9,37–42]:

- ◇ Signal interference;
- ◇ Limited power source;
- ◇ Low memory capabilities;
- ◇ Security issues;
- ◇ Constrained transmission range;
- ◇ Narrow bandwidth;
- ◇ Limited sensing capabilities.

Below, we describe each of the drawbacks listed. "Signal interference" means that communication between sensor nodes can be easily disturbed by other wireless-based technologies, leading to a degradation in the quality of communication within WSNs. "Limited power source" is caused by the reliance of sensor nodes on battery power (recharging or replacing batteries can be challenging in many real-world applications). This can prevent sensor nodes from performing demanding computational tasks. "Low memory capabilities" negatively affect the reliability, efficiency, and functioning of WSNs. "Security issues" mean that data confidentiality may be compromised, since securing WSNs is a challenging task due to their energy, memory, and computation constraints. The "Constrained transmission range" of sensor nodes can cause serious problems within large-scale environments and areas with barriers that may interfere with wireless communication (e.g., walls, high buildings, etc.). "Narrow bandwidth" degrades the communication speed in WSN-based applications. "Limited sensing capabilities" mean that sensor nodes may struggle to accurately sense their surrounding environment since external factors are transduced into a measured value due to the use of inexpensive hardware components.

As a result of the previously discussed drawbacks, sensor nodes are susceptible to various environmental factors (e.g., temperature fluctuations, rain, radiation, pressure variations, chemical exposure, dust, electrical interference, etc.) that can significantly impair their performance or even lead to failures [4,43]. As discussed in [44], sensor readings are also affected by other external devices, hardware noise, improper measurement techniques, etc. Furthermore, WSNs are susceptible to attacks by hackers and cybercriminals (e.g., jamming, eavesdropping, spoofing, wormholes, etc.) and must contend with challenges such as high data redundancy, data duplicity, quantized computing, etc. [4,45,46]. Therefore, as indicated in the literature [4,43–46], many researchers have dedicated their efforts to addressing these drawbacks of WSNs over the past decades.

In this paper, we address the issue of how to struggle with incorrect sensor readings caused by negative environmental factors and other factors skewing the measured data. We model these factors as Gaussian noise skewing the data measured by sensor nodes.

Thus, the measured data differ from the true value of the sensed physical quantity due to this noise. We assume that noise is an independent and identically distributed (IID) random variable with the Gaussian distribution that skews the initial inner state at each sensor node in WSNs. More specifically, we focus our attention on the applicability of distributed consensus gossip algorithms for data aggregation to suppress the impact of incorrect sensor readings.

1.2. Sensor Fusion in Wireless Sensor Networks

In general, the term "sensor fusion" is understood as a process of integrating data measured by multiple independent sensor nodes in order to generate detailed and complete information about the monitored environment or process of interest [47]. In other words, an output of sensor fusion allows for a better understanding of the observed quantity and reduces the amount of uncertainty [48]. This means that fused data provide us with more consistent, more accurate, and more dependable information about the observed quantity than a single data source would. The application of sensor fusion can also allow WSNs to execute the same goals even if one or more sensor nodes are removed from WSNs. Moreover, due to sensor fusion, WSNs can fulfill novel functionalities without adding new sensor nodes. As discussed in [49–51], the usage of sensor fusion can be highly beneficial in many real-world scenarios, since it may effectively compensate for failures, noise, data redundancy, etc.

In this paper, we focus on a distributed sensor fusion scheme, where the sensor nodes are not aware of network topology [52]. In addition, no central unit to coordinate the whole process is required in these schemes (in contrast to old-fashioned centralized schemes, where a data fusion center is responsible for data collection). Therefore, distributed sensor fusion is a much more effective solution than centralized schemes, especially in mobile and power-constrained technologies (such as WSNs). In distributed schemes, only the adjacent sensor nodes are required to communicate with each other, and sensor nodes update their states according to local rules. The final goal of each sensor node is to possess at least a precise estimate of the aggregated quantity.

As mentioned earlier in this paper, we focus our attention on distributed consensus gossip-based algorithms for sensor fusion. A consensus is understood as a general agreement achieved among a group of entities [53]. In general, the term 'consensus algorithm' is considered to be an iterative mechanism used to reach agreement (through local interactions) on a single value among involved entities with different initial values. As shown in the literature [54–58], consensus algorithms are applicable for various purposes (leader election, decision processes, cryptocurrency transactions, data replication, data aggregation, load balancing, state estimation, network size estimation, etc.) and find the usage in numerous fields (blockchain, sensor networks, UAVs, big data analysis, etc.). In this paper, we are concerned with gossip-based algorithms, i.e., the selection of the transmitting node/the message receiver is stochastic, which ensures effective information diffusion across WSNs.

1.3. Our Contribution

Our contribution in this paper is to examine the applicability of seven frequently applied distributed consensus gossip algorithms for sensor fusion (namely, the Randomized Gossip algorithm (RG), the Geographic Gossip algorithm (GG), the Broadcast Gossip algorithm (BG) with three initial configurations, the Push-Sum protocol (PS), and the Push-Pull protocol (PP)) to compensate for incorrect data in noisy WSNs. More specifically, we assume that sensor readings are significantly affected by negative environmental factors, worsening the precision of the measurements. As a result, the measured data of every sensor node differs from the true value of the observed physical quantity, which may

affect the operation of WSNs [4]. Our hypothesis is that the application of sensor fusion performed by the mentioned algorithms can ensure a significant increase in the precision of the measured data and improve decision making. The performance of the examined algorithms is tested in random geometric graphs (RGGs) and evaluated by two metrics (the mean square error (MSE) and the number of sent messages required for an algorithm to be completed). In real-world applications, consensus algorithms are required to ensure high estimation precision and a high convergence rate (or a low number of sent messages), thereby optimizing energy consumption and prolonging the network lifetime. Therefore, their application is highly recommended for WSNs and other technologies with constrained computational, communication, memory, and energy capabilities. Also, simple update rules (easy to compute) and low memory demands are necessary, as WSNs are significantly constrained in these aspects—this is fulfilled by every examined algorithm. Thus, our research is primarily focused on identifying algorithms with an acceptable estimation precision and with the minimum number of sent messages.

1.4. Paper Organization

This paper is organized as follows:

- Section 1 (Introduction)—This section, divided into four subsections, provides a theoretical overview of the topics under consideration (WSNs and sensor fusion in WSNs), highlights our contribution and outlines the organization of the paper.
- Section 2 (Related Work)—In this section, we provide a list of the most recent and best-cited papers dealing with how to suppress incorrect sensor readings in sensor networks.
- Section 3 (Theoretical Background)—This section is divided into three subsections and focuses on the theoretical background of the topic. Specifically, we introduce the applied graph model (RGGs), the model of incorrect sensor readings, the used stopping criterion, and the examined algorithms.
- Section 4 (Applied Research Methodology and Metrics)—This section introduces the applied research methodology and the metrics used to evaluate the performance of the examined algorithms.
- Section 5 (Experiments and Discussion)—This section consists of the experimental results obtained using MATLAB 2018b, along with a discussion of observable phenomena.

2. Related Work

In this paragraph, we turn our attention to papers dealing with techniques to suppress incorrect sensor readings in WSNs. Over the past decades, many authors have addressed this issue and proposed various techniques for this purpose. We focus on relevant papers with a high citation rate and papers that have been published recently.

In [59], the authors deal with incorrect readings caused by improper hardware design and calibration. Also, they address how sensor nodes with low remaining power energy can cause incorrect sensor readings. In the paper, the authors discuss that it is not always feasible to ensure a flawlessly configured sensor network; therefore, they recommend automated detection, classification, root cause analysis, and techniques to scrub collected data from the sensor nodes.

The authors of [60] propose a technique to predict future states of the observed physical quantity, which can compensate for previously executed incorrect sensor readings. The authors apply an $O(n^2 \log(n))$ algorithm for the linear prediction problem. This algorithm is capable of calculating the solution from the input sensor readings without determining the planar configuration.

The paper [61] presents a Bayesian approach to reduce the data-associated uncertainty. The presented approach combines prior knowledge of correct sensor readings, noise characteristics, and incorrect sensor readings in order to increase the precision of the measured data. This cleaning is executed by either sensor nodes or the base station.

In [62], a self-validating strategy for multifunctional sensors is proposed. The approach exploits the Gray Bootstrap model, which can generate a prediction model. This bootstrap method ensures that the uncertainty can be estimated without prior information about the distribution of the measured data. In this strategy, a measured value is compared to a predicted value, allowing suppression of incorrect sensor readings.

The authors of [63] propose an in situ calibration method. Here, the benchmark is established based on observation of the monitored environment. The sensor-measured data are subsequently compared to this benchmark and modified.

The paper [64] introduces a method based on the Kalman filter, autoregressive integrated moving average, knowledge-based systems, and smoothing operators. The method is capable of detecting and correcting data with anomalous features.

An easily configurable scheme based on local predictors and listening to correlated adjacent nodes is proposed in [65]. In this scheme, sensor nodes compare the confidence of the sensed value with the output generated by the predictor.

The authors of [66] propose SNRepair, a self-calibrating mechanism based on deep reinforcement learning. The incorrect readings can be calibrated according to historical healthy data in a few seconds. In the experimental section, it is identified that the mechanism can operate with high precision even in highly noisy environments.

The approach in [67] utilizes statistical time series analysis with Pearson's correlation or dynamic time warping coupled with a data-driven algorithm in convolutional neural networks. The authors apply effective statistical methods to identify the most appropriate sensors to predict prior to the computationally intensive training of complex machine learning models.

In [68], the application of average consensus algorithms with the Perron matrix to compensate for incorrect data was investigated in several WSN scenarios. Namely, the graph order weights, the maximum degree weights, and the best constant weights were applied to compensate for Gaussian noise skewing the measured data. From the results, it was seen that all algorithms could significantly suppress Gaussian noise, thereby optimizing the MSE in every examined scenario. The best performance was achieved by the best constant weights, but no difference among the examined algorithms was observed for many configurations of the applied stopping criterion.

The paper [69] focused on the applicability of the Metropolis-Hastings algorithm with various initial configurations to suppress or even overcome Gaussian noise affecting the sensor readings in WSNs. In the paper, it is identified that the algorithm can significantly suppress noise for each scenario. However, it was outperformed by the best-performing algorithm with the Perron matrix (i.e., the best constant weights). In addition, one can see that the initial configuration of the Metropolis–Hastings algorithm had either marginal or no impact on the estimation precision.

Thus, we can see that no other paper deals with the applicability of distributed consensus gossip algorithms for sensor fusion to compensate for incorrect sensor readings. This paper follows on from our previous papers [68,69], where two deterministic consensus algorithms were applied for the same purpose. Accordingly, we also compare the performance outcomes of the top gossip algorithms with those of the best-performing deterministic consensus algorithm.

3. Theoretical Background

This section is divided into three subsections as follows:

- Distributed Consensus Gossip-based Sensor Fusion in Wireless Sensor Networks (Section 3.1)—This subsection provides the model of incorrect sensor readings, insight into distributed consensus gossip sensor fusion, and the applied stopping criterion.
- Random Geometric Graphs (Section 3.2)—In this subsection, we introduce the mathematical used model to model WSNs (RGGs).
- Examined Distributed Consensus Gossip Algorithms (Section 3.3)—Here, we introduce seven algorithms chosen for evaluation in this paper.

3.1. Distributed Consensus Gossip-Based Sensor Fusion in Wireless Sensor Networks

As mentioned above, we consider a scenario where a set of n spatially distributed sensor nodes measure a physical quantity (let us label the true value of this quantity as θ) from their surrounding environment. However, due to factors discussed in Section 1.1, sensor readings deviate from the true value of θ. Thus, the initial inner state $x_i(0)$ of v_i is defined as follows [68,69]:

$$x_i(0) = \theta + z_i, for\, i = 1, 2, ..., n \tag{1}$$

Here, z_i is an IID random variable of Gaussian distribution (representing noise skewing sensor data), which is defined as follows [68,69]:

$$z_i \sim N(\mu, \sigma^2), for\, i = 1, 2, ..., n \tag{2}$$

Here, the symbol μ presents the mean (equal to zero when modeling noise), and the symbol σ denotes the standard deviation. Thus, this means that the sensor reading of every sensor is skewed by Gaussian noise. As shown in the literature, modeling incorrect sensor readings as in (1) is common across many scenarios [70,71].

In order to justify our hypothesis that distributed consensus gossip algorithms for sensor fusion may be applied to suppress Gaussian noise, we show how sensor fusion (RG is applied) can alleviate incorrect sensor readings in a random simple topology formed by five vertices (Figure 1), and the evolution of inner states when data are skewed due to Gaussian noise (Figure 2).

Communication among sensor nodes in WSNs is coordinated—depending on the applied gossip algorithm—by either a single global clock or n local clocks [72]. Each time the clock ticks, specific steps defined by the applied algorithm are executed—referred to as a time instance k. We denote the inner state of v_i at time instance k as $x_i(k)$ and a column vector that gathers the inner states of all sensor nodes at time instance k as $\mathbf{x}(k)$.

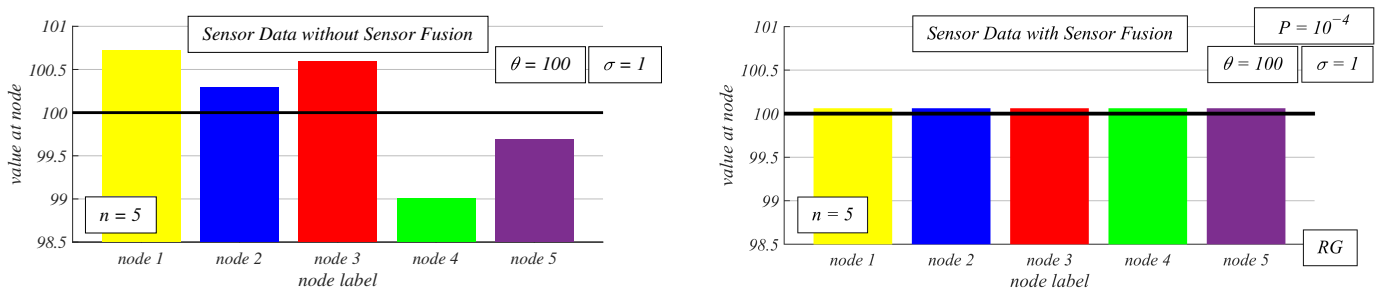


Figure 1. Example of how distributed consensus gossip algorithms can suppress Gaussian noise—**left** figure: no sensor fusion is applied; **right** figure: sensor fusion (randomized gossip) is applied.

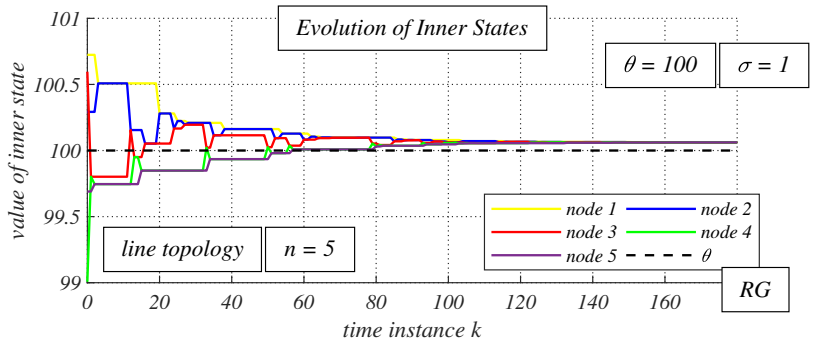


Figure 2. Evolution of inner states when data are skewed due to Gaussian noise (randomized gossip is applied).

As previously mentioned, the operation of the tested algorithm is bounded by a stopping criterion to optimize execution time. In the experimental section, we apply the stopping criterion defined in (3) based on the comparison between the highest and lowest current inner states (from all nodes) at time instance k [72]:

$$| \, max\{\mathbf{x}(k)\} - min\{\mathbf{x}(k)\} \, | < P \tag{3}$$

Once the condition (3) is met for the first time, the algorithm is considered complete and ceases execution. The applied stopping criterion is configurable by setting the value of P; higher values are assumed to ensure higher precision of the final inner states, albeit at the cost of slowing down the algorithms [72]. As one can find in [73–75], the usage of a stopping criterion is necessary to optimize aspects such as power-energy efficiency, resource management, delivery rate, network lifetime, data precision, computational requirements, etc. Therefore, implementing a stopping criterion is essential for real-world applications. In Figure 3, we show how the MSE and the number of sent messages differ in a random simple topology when the value of P is modified (PS is applied).

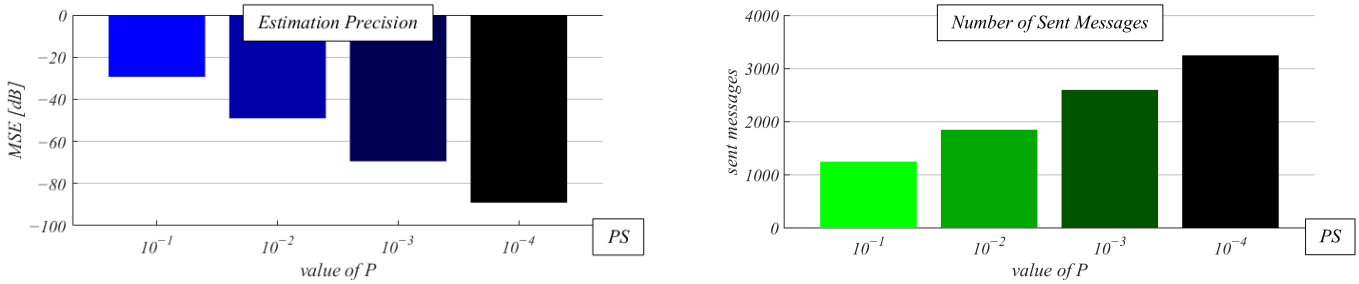


Figure 3. Example of how value of P affects estimation error and number of sent messages.

3.2. Random Geometric Graphs

In this paper, we model WSNs as undirected, simple, finite graphs G with connected topologies determined by the vertex set $\mathbf{V}$ and the edge set $\mathbf{E}$ (i.e., $G = (\mathbf{V},\mathbf{E})$) [76,77]. The vertex set $\mathbf{V}$ comprises all vertices in G, with each representing a sensor node in the WSN. The vertices are distinguishable by a unique index $i = 1,2...,n$ (i.e., $\mathbf{V} = (v_1, v_2,... v_n)$), where n ($n = |\mathbf{V}|$) represents the graph order, or in other words, the number of sensor nodes in WSN. The edge set $\mathbf{E} \subset \mathbf{V}\times\mathbf{V}$ consists of all graph edges in G, which represent a direct (one-hop) link between two vertices (i.e., two corresponding sensor nodes can directly communicate with one another). An edge labeled as e_{ij} links vertices v_i and v_j. The cardinality of this set ($|\mathbf{E}|$) equals the graph size, i.e., the number of direct connections

in WSN. See Figure 4 for an example of an undirected simple finite graph with $n = 14$ and $|\mathbf{E}| = 16$.

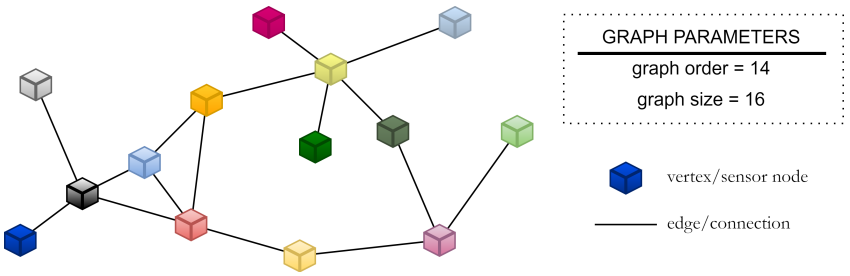


Figure 4. Example of undirected, simple, finite, connected graph.

In this paper, we used RGGs, which are a commonly used graph model for representing systems such as WSNs, IoT, social networks, etc. [78,79]. In RGGs, vertices are randomly placed within a finite square area, and an edge connects each pair of vertices if the distance between them is less than a specified transmission radius, which is uniform across all vertices. To produce graphs with various connectivity, we modified the transmission radius.

3.3. Examined Distributed Consensus Gossip Algorithms

As previously mentioned in this paper, we examine how seven widely used distributed consensus gossip algorithms can mitigate the effects of incorrect sensor readings.

One of these is the RG algorithm [80], which has been identified as a vulnerable solution that requires a significant volume of transmitted messages. Its principles can be described as follows: each sensor node is equipped with its own local clock, which ticks at random intervals. At each time instance k, the sensor node whose clock has ticked (denoted as v_i) selects one of its neighbors, labeled as v_j, uniformly at random for pairwise averaging described below:

$$x_i(k+1) \;=\; x_j(k+1) \;=\; \frac{x_i(k) + x_j(k)}{2} \tag{4}$$

At this time instance, the inner states of all other sensor nodes remain unchanged.

The next tested algorithm is GG [81], which utilizes geographic routing to optimize sensor fusion. This approach increases the diversity of pairwise exchanges; however, sensor nodes must be aware of the geographical positions of other nodes in WSNs. Like in the RG, each sensor node has its own local clock that ticks randomly. At each time instance k, one of the sensor nodes (v_i) is activated and randomly selects, according to a uniform distribution, one of the other nodes in the entire network (v_j) for pairwise averaging, not just from adjacent nodes, as is the case with the RG.

In this paper, we also evaluate the BG [82], an algorithm that does not rely on pairwise communication, in contrast to the two previous algorithms. However, the precision of this algorithm may be significantly lower, potentially limiting its usability in certain scenarios [72]. The principle of this algorithm is as follows: the sensor nodes are equipped with local clocks once again. When the clock of a node ticks (v_i), it sends its current inner state to all adjacent neighbors (v_j), which then apply the following update rule:

$$x_j(k+1) \;=\; \gamma x_j(k) + (1-\gamma) x_i(k) \tag{5}$$

Here, the parameter γ can take the following values:

$$\gamma \in (0,1) \tag{6}$$

The node (v_i) and any nodes that have not received a message from it at this time instance do not update their inner states. This is the only algorithm among the examined

ones that is configurable. The parameter γ, known as the mixing parameter, determines how the current state at v_j and the current state at v_i influence the next inner state of v_j. Lower values of γ ensure that the inner state of v_i has a greater impact on the next inner state of v_j. Its higher values allow for improved estimation precision but at the cost of an increase in the number of sent messages. Therefore, the choice between higher estimation precision and fewer sent messages typically depends on the specific requirements of the application.

Another examined algorithm is the PS, which functions correctly only if the total system mass remains constant, as noted in [83]. Consequently, this approach is significantly vulnerable to message losses. In this algorithm, each node maintains and regularly updates two variables: the state $s_i(k)$ (initialized by local measurements) and the weight $w_i(k)$ (initialized to one). At each time instance k (all sensor nodes are synchronized by a single global clock), every node randomly selects one of its neighbors and sends half of both its current inner state and weight. The same values are also stored in its internal memory. The arithmetic mean estimate can be easily calculated at each node as follows:

$$x_i(k) = \frac{s_i(k)}{w_i(k)} \tag{7}$$

Each sensor node can then compute the value of its inner state for $k + 1$ as the sum of half of its current state and the states received at k. The weights for the next time instance are determined in a similar manner.

The final algorithm to be tested is the PP [84], in which the sensor nodes are once again synchronized by a single global clock. Whenever this clock ticks, each sensor node sends its current inner state (i.e., a Push message) to a randomly chosen neighbor that is selected with a uniform distribution. This neighbor then replies by sending its current state (i.e., a Pull message). Consequently, these two sensor nodes can perform the pairwise averaging defined in (4).

In Figure 5, we present a table outlining the most relevant properties of the examined algorithms to justify their selection. Namely, we provide their number of clocks, connectivity range (meaning whether a node in the WSNs can address its message to only one of its neighbors or an arbitrary node in the network), communication method (meaning whether a message recipient is required to send its message to the message sender (if so, the message of the message recipient is sent at the same iteration)), synchronization, configurability, multifunctionality (meaning whether an algorithm can be applied to estimate other aggregate functions (e.g., the network size)), message transmission types, scalabilities, and algorithmic paradigms.

PARAMETER \ *ALGORITHM*	Randomized Gossip (RG)	Geographic Gossip (GG)	Broadcast Gossip (BG)	Push-Sum (PS)	Push-Pull (PP)
number of clocks	n	n	n	1	1
connectivity range	single-hop	global	single-hop	single-hop	single-hop
communication method	bidirectional	bidirectional	one-way	one-way	bidirectional
synchronization	simple	simple	trivial	complex	very complex
configurable	no	no	yes	no	no
multifunctionality	yes	yes	no	yes	yes
message transmission type	one-hop unicast	global unicast	neighbor broadcast	one-hop unicast	one-hop unicast
scalability	easily scalable	challenging to scale	easily scalable	easily scalable	easily scalable
algorithmic paradigm	stochastic	stochastic	stochastic	stochastic	stochastic
note		needs routing mechanism	high estimation error		

Figure 5. Comparison of properties of examined algorithms.

4. Applied Research Methodology and Metrics

In this section, the applied research methodology and the metrics used to evaluate the algorithms are introduced. As mentioned above, we used the stopping criterion defined in [72] and described in Section 3.1 to bound the execution of the algorithms. This

stopping criterion is determined by the parameter P, which takes the following values in our experiments:

- ◇ $P = 10^{-1}$.
- ◇ $P = 10^{-2}$.
- ◇ $P = 10^{-3}$.
- ◇ $P = 10^{-4}$.

We vary the value of P to analyze the impact of the initial configuration of the applied stopping criterion on both evaluated metrics. The value of P is adjusted from 10^{-1} to 10^{-4}. Using higher values, such as 10^{0} or 10^{1}, is impractical as the initial inner states are not sufficiently dispersed. Similarly, applying values smaller than 10^{-4} does not yield any new insights compared to this setting, and their inclusion would only reduce the paper's clarity from a formal perspective. Four values of P were sufficient to demonstrate how the initial configuration of the stopping criterion affects the algorithms' performance.

As previously mentioned, we modeled incorrect sensor readings as Gaussian noise with $\mu = 0$ and with σ set as follows:

- ◇ $\sigma = 1$.
- ◇ $\sigma = 5$.

As discussed in [85–90], WSNs can be utilized to monitor a wide range of environmental parameters, including air temperature, soil properties (e.g., moisture and roughness), urban noise levels, rainfall, solar radiation, atmospheric pollution, concentrations of gases (e.g., CO_2, O_3, SO_2, NO_2, CO, CH_4), air humidity, flame temperature, and more. As indicated in [91–99], these and other environmental parameters typically follow a Gaussian distribution, making the model applied in this paper a credible representation of real-world WSN applications.

To ensure the high credibility of the presented results, we generated 100 sets of unique initial inner states and 100 graphs (for each connectivity) with the graph order $n = 30$. In the experimental part, we analyzed the algorithms in the RGGs of either dense (DEN) or sparse (SPAR) connectivity (see Figure 6 for an example of both graph types). Only connected graphs were used in our experiments. The connectivity of WSNs influences various factors, including redundancy, reliability, robustness against potential threats, energy consumption, etc. [100]. Generally, densely connected networks are more robust but consume more energy compared to sparsely connected networks.

As mentioned earlier in this paper, we applied two metrics to evaluate the performance of the tested algorithms. Specifically, we measured the number of sent messages for an algorithm to completion and the MSE (see (8)), which is a reasonable metric to quantify the precision of the sensor fusion estimation [101,102]. It is defined in the following:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(x_i(k_l) - \mathbf{1}^{\mathrm{T}}\frac{\mathbf{x}(0)}{n}\right)^2 \tag{8}$$

Here, k_l is the label of the last time instance, i.e., the instance at which an algorithm is completed. The examined algorithms differ in terms of the number of messages sent per each time instance k. The RG requires two messages per time instance, the GG requires two times the number of hops between communicating nodes, the BG requires only one message, the PS needs n messages, and the PP requires two times n messages. In the experimental part, we repeated each algorithm 100 times in each topology (as the algorithms are stochastic), and only the median of a corresponding metric is depicted in the figures.

As discussed in Section 3.3, the BG can be adjusted by setting γ. In our experiments, we set its value to the following:

- ◇ γ = 0.1 (later referred to as BG 1).
- ◇ γ = 0.5 (later referred to as BG 2).
- ◇ γ = 0.9 (later referred to as BG 3).

As mentioned earlier, this parameter determines the ratio between the influence of the current inner state and the received state on the inner state at a message-receiving node. We selected a value close to 0 (lower bound of γ), a value close to 1 (upper bound of γ), and the midpoint value of (6).

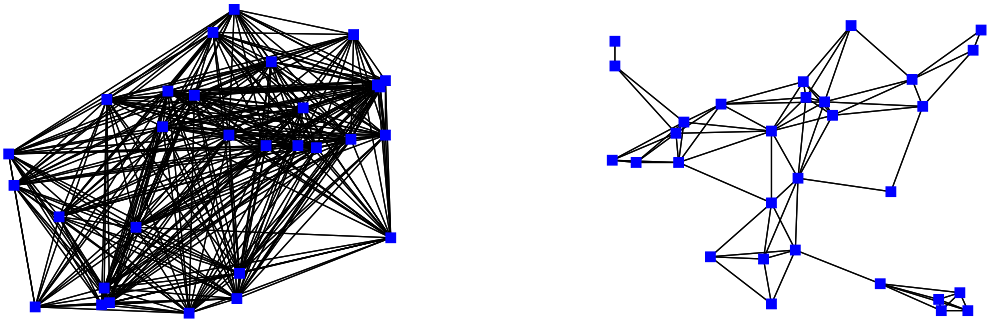


Figure 6. Example of densely connected (**left** figure) and sparsely connected (**right** figure) random geometric graphs, with n = 30.

5. Experiments and Discussion

In this section, we present the experimental results obtained using MATLAB 2018b and discuss the observable phenomena. The performance of the computer running the numerical simulations is follows: six cores, 12 threads, hyper-threading technology, base clock speed = 3.2 GHz, max turbo frequency = 4.6 GHz, RAM capacity = 32 GB, and memory speed = 3600 MHz. As mentioned earlier in this paper, we varied the initial configuration of the applied stopping criterion, the connectivity of RGGs, and σ of the Gaussian noise. The performance of the algorithms was quantified using two metrics (MSE and the number of sent messages), and the best-performing algorithms were compared against the distributed consensus deterministic best constant weights. The best constant weights used for suppressing Gaussian noise have been analyzed in [68,69], where they were identified as the best-performing algorithm for this purpose. In contrast to the other algorithms, the best constant weights are deterministic; therefore, we believe that comparing this algorithm with another paradigm will provide valuable insights.

In the first analysis, we analyzed the MSE in densely connected RGGs for various precisions of the applied stopping criterion if σ = 1. As shown in Figure 7, the value of P had only a marginal effect on the precision of all tested algorithms (only for $P = 10^{-1}$ was the precision a bit lower) except on the BG. Regardless of further decreases in P, the error of all four algorithms did not drop below -24.87 dB. The RG achieved the error in the range $[-24.37, -24.87]$ dB, the GG in $[-24.32, -24.87]$ dB, the PS in $[-24.48, -24.87]$ dB, and the PP in $[-24.61, -24.87]$ dB. Thus, in terms of estimation precision, the choice of any of these four algorithms is almost inconsequential. Unlike in Figure 6, where the MSE was analyzed in an error-free scenario, the value of P had only a minor impact on the MSE (except on the BG). Furthermore, the precision of the BG was lower but improved as γ increased. For $\gamma = 0.9$, the precision was relatively high, making it highly suitable for real-world applications (with a maximum precision of -17.40 dB). A decrease in P led to a lower MSE when the BG was applied. BG 1 achieved the error in the range $[-3.15, -5.32]$ dB, BG 2 in $[-7.62, -9.94]$ dB, and BG 3 in $[-16.58, -17.40]$ dB. The most notable observation is that each algorithm could substantially compensate for incorrect

sensor readings (MSE = −0.42 dB without a sensor fusion mechanism). Consequently, their application proved to be highly effective in suppressing Gaussian noise.

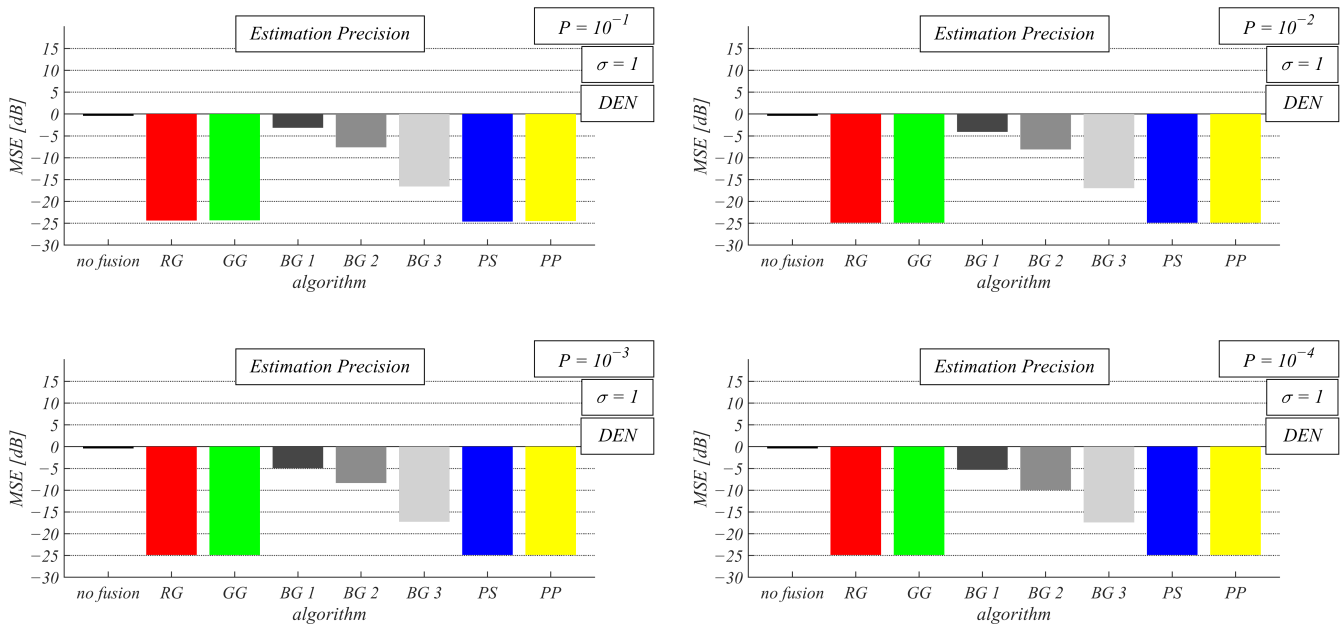


Figure 7. Mean square error for various precisions of applied stopping criterion, in dense random geometric graphs, and for $\sigma = 1$.

In sparsely connected RGGs (Figure 8), the MSEs of the RG, GG, PS, and PP could not drop below −24.87 dB, being consistent with the findings from the previous analysis. Once again, the value of P had little to no impact on MSE when any of these four algorithms are applied. The RG achieved errors in the range of [−23.89, −24.87] dB, the GG in [−24.32, −24.87] dB, the PS in [−23.88, −24.87] dB, and the PP in [−23.71, −24.87] dB (their performances did not differ notably from each other). Thus, the connectivity of the graphs had only a marginal impact on the performance of these algorithms (except for the BG) when used to suppress Gaussian noise. When the BG was applied, the precision of the sensor fusion was consistently lower than that of the other tested algorithms. However, its precision improved compared to the first analysis, with the MSE ranges as follows: BG 1: [−5.85, −6.78] dB, BG 2: [−11.67, −15.38] dB, and BG 3: [−20.95, −23.51] dB. Thus, the BG is also applicable for lower values of γ. Similar to dense RGGs, an increase in γ and a decrease in P led to a lower MSE when the BG was applied. Most importantly, the application of every tested algorithm guaranteed a significant enhancement in estimation precision (MSE = −0.42 dB when no sensor fusion mechanism was used—the same value was observed in dense RGGs).

In the next analysis, we focus on scenarios where $\sigma = 5$, meaning the data are more dispersed compared to the previous two analyses. In Figure 9, we investigated how the algorithms perform in compensating for Gaussian noise in densely connected RGGs. From the results, it is evident that the value of P had only a marginal impact on the MSE for the RG, GG, PS, and PP, which was similar to the previous analysis. The algorithms achieved lower precision (RG: [−20.78, −21.02] dB, GG: [−20.81, −21.02] dB, PS: [−20.84, −21.02] dB, and PP: [−20.89, −21.02] dB) but were more effective in suppressing incorrect sensor readings (MSE = 14.05 dB when no sensor fusion mechanism was used). Therefore, these four algorithms are also applicable for suppressing Gaussian noise when $\sigma = 5$. Once again, the MSE could not drop below the threshold value (−21.02 dB if $\sigma = 5$) when these four algorithms were applied. As further observed in the figures, the BG was significantly

outperformed by the other algorithms and achieved very low precision in every scenario (BG 1: [11.30, 9.42] dB, BG 2: [5.69, 5.03] dB, BG 3: [−1.45, −3.06] dB—this configuration guaranteed the lowest MSE). Therefore, this algorithm is not applicable in dense RGGs if $\sigma = 5$, regardless of the value of γ, even though it improved the MSE compared to the scenario where no sensor fusion mechanism was used. Similar to the case for $\sigma = 1$, increasing γ and decreasing P led to lower MSE values for this algorithm.

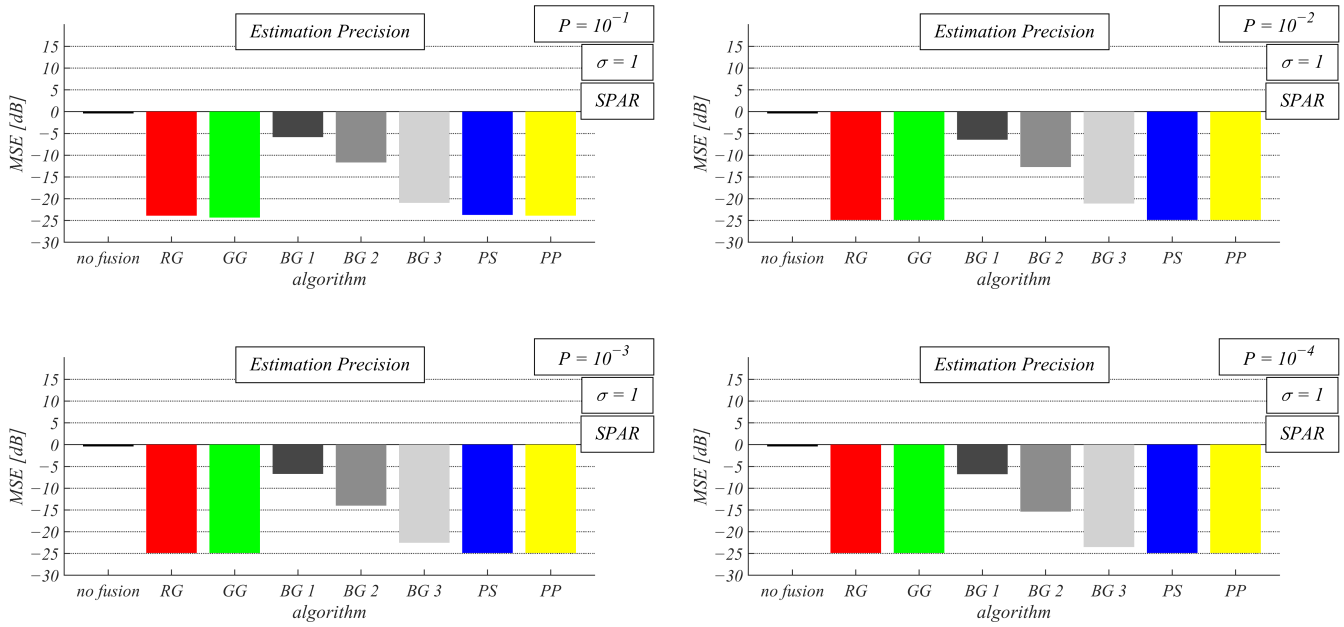


Figure 8. Mean square error for various precisions of applied stopping criterion, in sparse random geometric graphs, and for $\sigma = 1$.

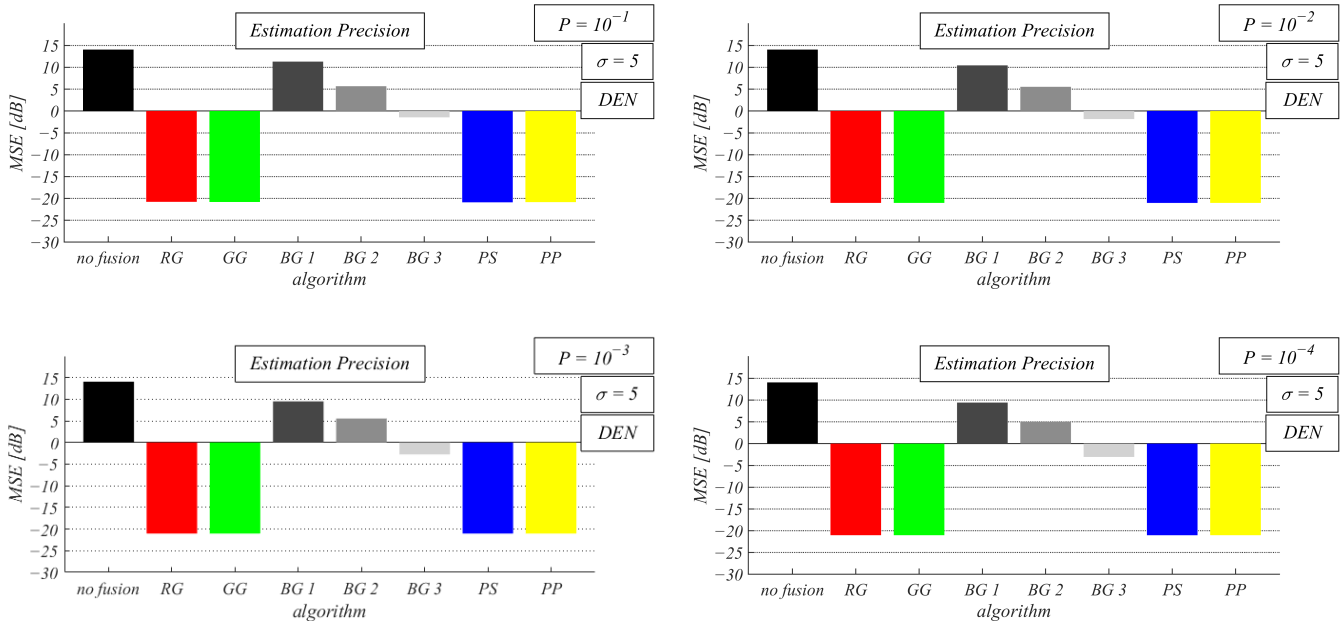


Figure 9. Mean square error for various precisions of applied stopping criterion, in dense random geometric graphs, and for $\sigma = 5$.

In sparse RGGs for $\sigma = 5$ (Figure 10), the precision results of the RG, GG, PS, and PP were similar to the previous analysis (RG: [−20.48, −21.02] dB, GG: [−20.79, −21.02] dB,

PS: [−20.47, −21.02] dB, and PP: [−20.79, −21.02] dB) and were only slightly affected by P again. Thus, the MSE could not drop below −21.02 dB again. However, the BG with $\gamma = 0.9$ achieved usable precision, though it was lower than that of the four previously mentioned algorithms. The precision of the BG configurations was as follows: BG 1: [8.84, 6.35] dB, BG 2: [1.28, −1.04] dB, and BG 3: [−8.59, −11.06] dB. Its precision could be improved by decreasing P. Again, an increase in γ resulted in higher precision for the BG. Except for the BG, the connectivity of the graphs had only a marginal effect on the MSE of the tested algorithms. Similar to the case for $\sigma = 1$, the BG was more effective at suppressing Gaussian noise in sparsely connected graphs than in dense ones. As shown in the figures, the MSE was significantly suppressed by each algorithm, as observed in the previous analysis (MSE = 14.05 dB when no data fusion was used). Additionally, in sparsely connected RGGs, $\sigma = 5$ resulted in higher MSE values compared to σ = 1, just as it did in dense graphs.

In the following paragraphs, we evaluate the other examined metric: the number of sent messages for the algorithm to complete. In Figure 11, we present the results from densely connected RGGs when $\sigma = 1$. From the results, it is evident that the BG with $\gamma = 0.1$ was the best-performing algorithm for each P in terms of the number of sent messages (increasing the value of γ led to a higher number of sent messages). However, this configuration also achieved the lowest estimation precision, as shown earlier. Furthermore, it is observed that the performance of each examined algorithm, in terms of sent messages, degraded as P decreased. The performance of the algorithms in terms of sent messages was as follows: BG 1: [5, 12] messages, BG 2: [10, 32] messages, BG 3: [53, 187] messages, RG: [424, 1183] messages, GG: [486, 1351] messages, PS: [360, 900] messages, and PP: [360, 1020] messages. Therefore, among the four other algorithms (those with high precision in the previous analyses), the PS achieved the best performance in terms of the number of sent messages.

In sparsely connected RGGs (Figure 12), each algorithm required more sent messages than in densely connected RGGs. The lowest number of sent messages was required by the BG with $\gamma = 0.1$. The performance of this algorithm degraded as γ increased, which was consistent with the previous analysis. The performance of the BG in terms of sent messages was as follows: BG 1: [335.5, 1208.5] messages, BG 2: [493.5, 2997.5] messages, and BG 3: [1913.5, 14,458] messages. Unlike in the previous analysis, BG 3 was the worst-performing algorithm in terms of the number of sent messages for each P. As further seen in the figure, among the four other algorithms, the best performance was achieved by the GG. Thus, employing geographical routing is beneficial only in sparsely connected graphs. The performance of the tested algorithms in terms of sent messages was as follows: RG: [1750, 14,385] messages, GG: [1219, 3455] messages, PS: [1320, 11730] messages, and PP: [1560, 13,860] messages. Like in the previous analysis, a decrease in P led to an increase in the number of sent messages for each algorithm.

In Figures 13 and 14, we analyze the number of sent messages when $\sigma = 5$. The results show that the behavior of the functions was similar to the previous analyses. As seen in the figures, the BG with $\gamma = 0.1$ required the fewest sent messages again. Among the algorithms with low estimation error, the PS performed best in dense RGGs, while the GG performed best in sparse RGGs, which was consistent with the previous analysis. As further observed, the algorithms required more sent messages as σ increased. Thus, a higher value of σ not only resulted in higher MSE but also increased the number of sent messages. The performance in dense RGGs was as follows: BG 1: [6, 13] messages, BG 2: [15, 38.5] messages, BG 3: [83, 213] messages, RG: [592, 1384] messages, GG: [687, 1572] messages, PS: [510, 1050] messages, and PP: [540, 1140] messages. The performance in sparse RGGs was as follows: BG 1: [622, 1460] messages, BG 2: [1432.5, 3921] messages,

BG 3: [8236, 21,238] messages, RG: [7986, 23,522] messages, GG: [1762, 4018] messages, PS: [6810, 17,865] messages, and PP: [7480, 22,350] messages.

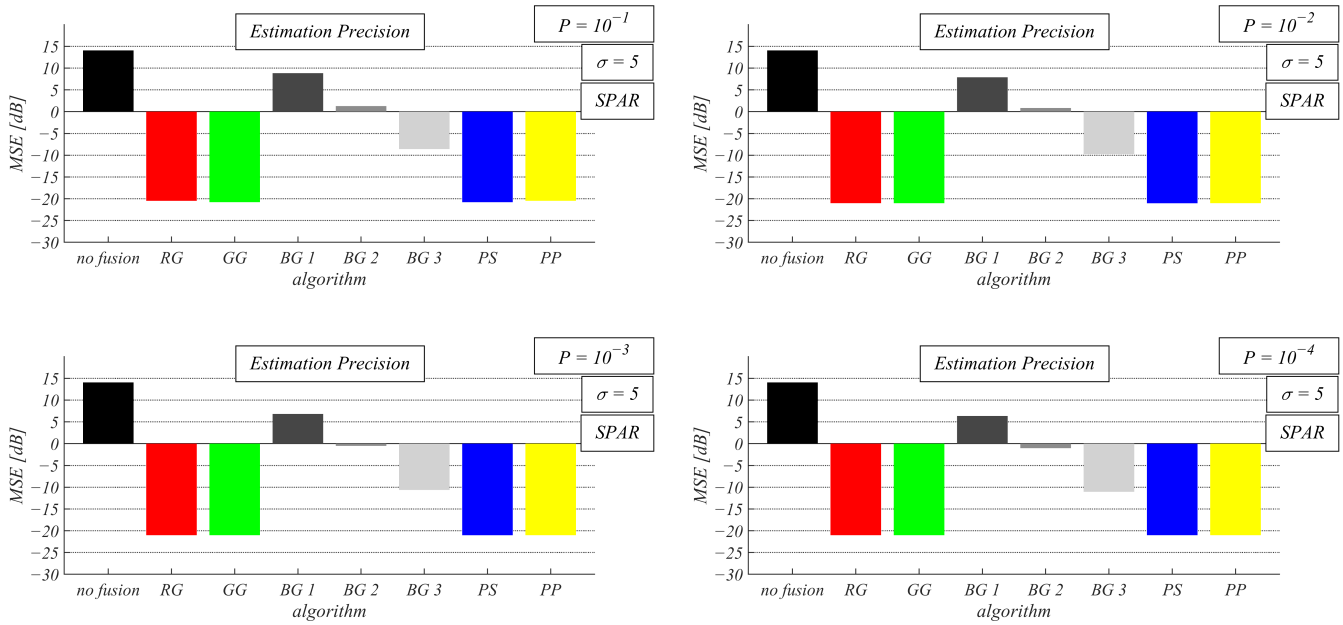


Figure 10. Mean square error for various precisions of applied stopping criterion, in sparse random geometric graphs, and for $\sigma = 5$.

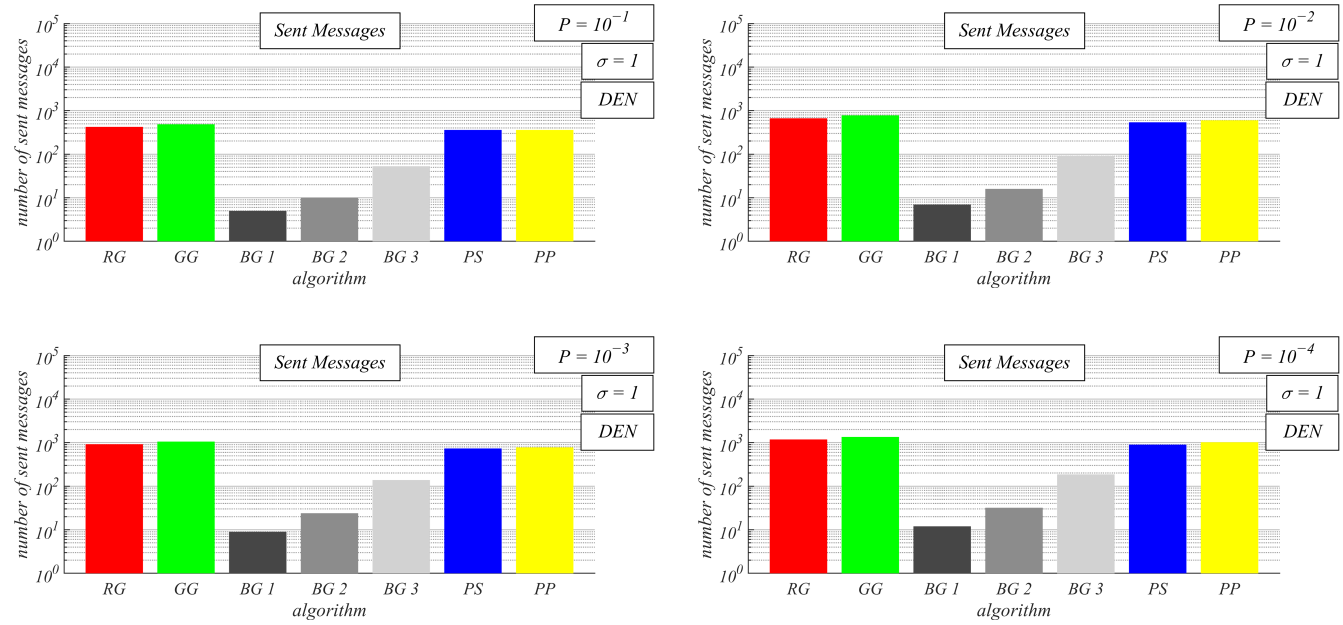


Figure 11. Number of sent messages for various precisions of applied stopping criterion, in dense random geometric graphs, and for $\sigma = 1$.

To ensure the higher credibility of the drawn conclusions, we also conducted experiments on other graph families. Specifically, we tested all algorithms on a dense/sparse Erdős–Rényi graph, a small-world graph, and a scale-free graph, which, along with RGGs, reliably represent real-world systems [103]. The algorithms were tested only for $P = 10^{-4}$ and $\sigma = 1$ to maintain the paper's clarity. As shown in Figure 15, the results are consistent

with those obtained from RGGs—all algorithms, except for the BG, achieved an MSE of −24.87 dB, as observed in the previous analyses.

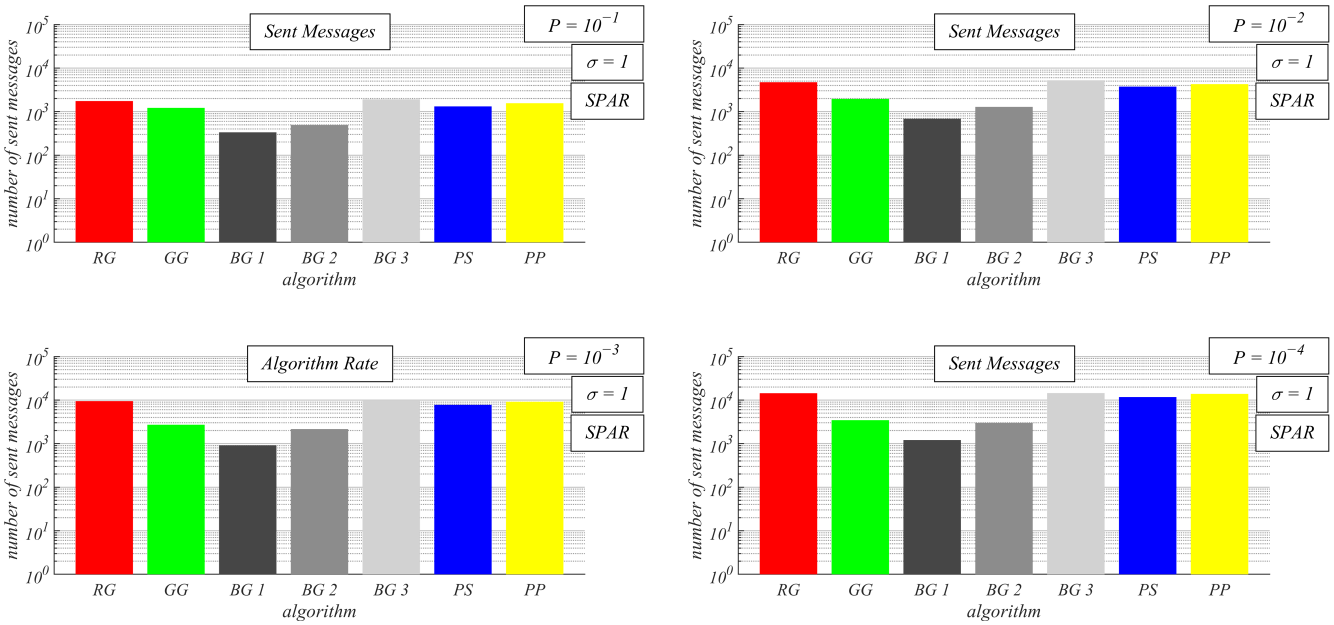


Figure 12. Number of sent messages for various precisions of applied stopping criterion, in sparse random geometric graphs, and for $\sigma = 1$.

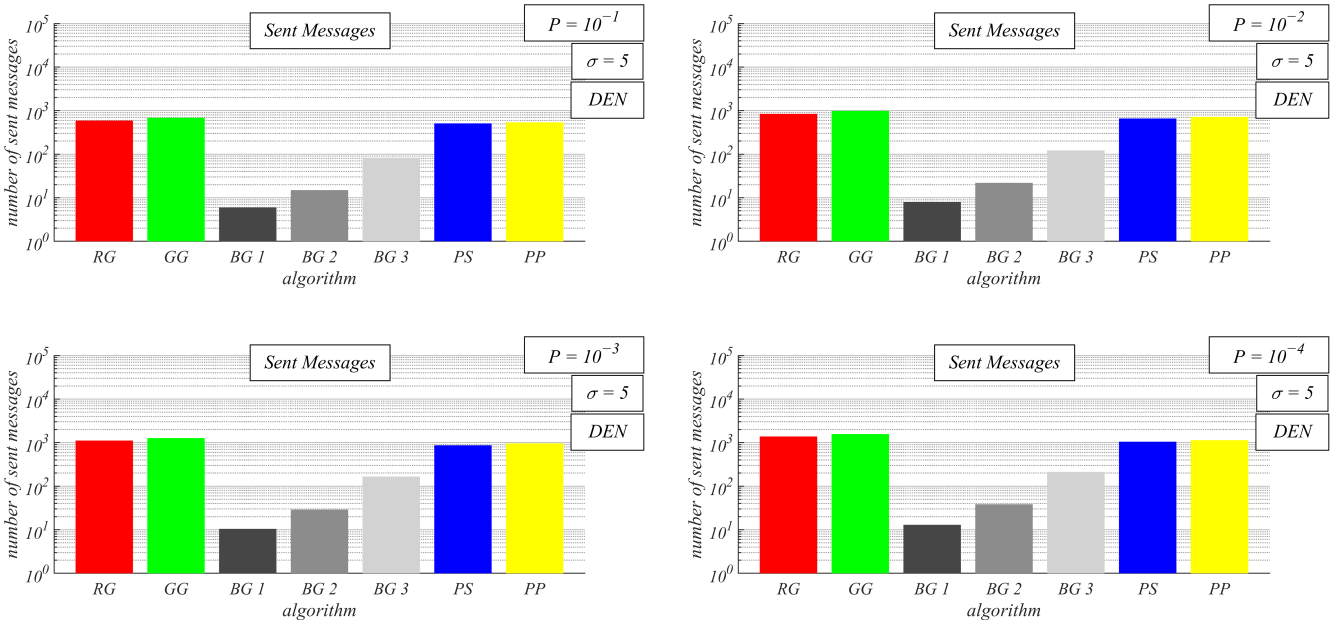


Figure 13. Number of sent messages for various precisions of applied stopping criterion, in dense random geometric graphs, and for $\sigma = 5$.

Based on the presented experimental results, we can conclude that the best algorithm for densely connected graphs is the PS, while the GG performs best for sparsely connected graphs. The precision values of the algorithms did not significantly differ from each other (except for the BG, which could still significantly suppress Gaussian noise despite its higher MSE). However, the algorithms varied in terms of the number of sent messages. The optimal setting for the applied stopping criterion was $P = 10^{-1}$ (paradoxically, this

configuration resulted in the significantly lowest estimation precision in scenarios without errors). Lower values of P only marginally, or not at all, reduced the MSE (except for the BG) but significantly increased the number of sent messages. Therefore, using low values of P is counterproductive compared to error-free scenarios (except for the BG). In Figure A1, a summary table is presented, highlighting the best- and worst-performing algorithms in each scenario.

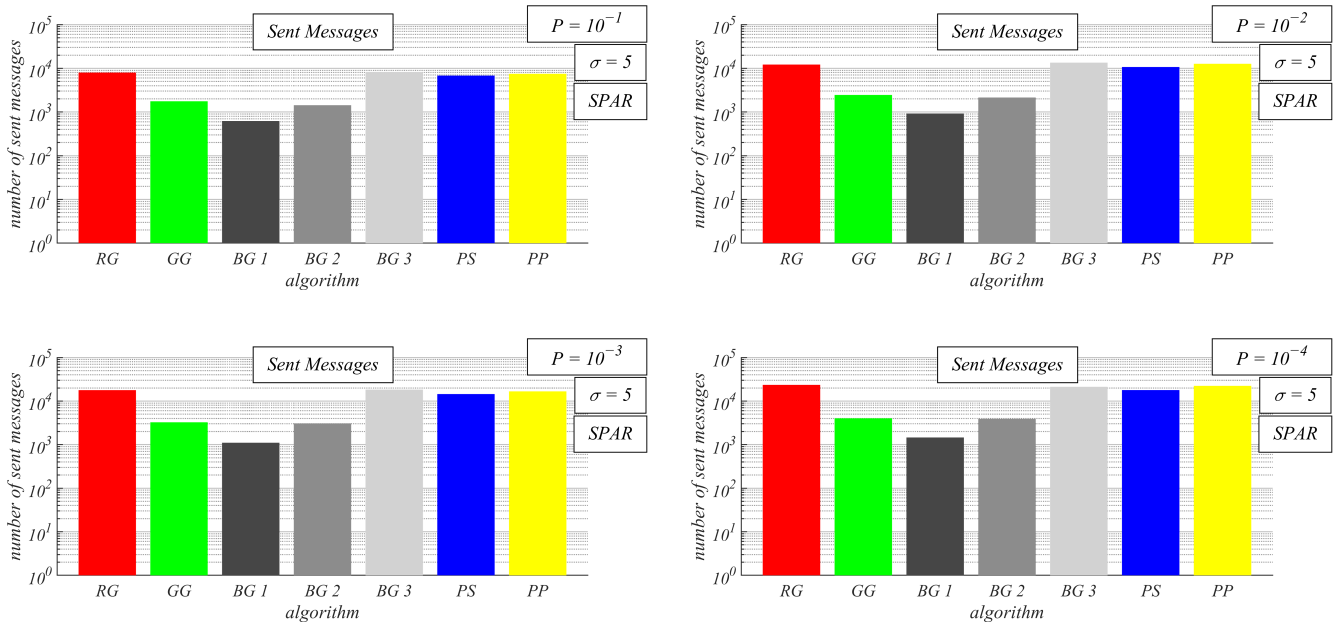


Figure 14. Number of sent messages for various precisions of applied stopping criterion, in sparse random geometric graphs, and for $\sigma = 5$.

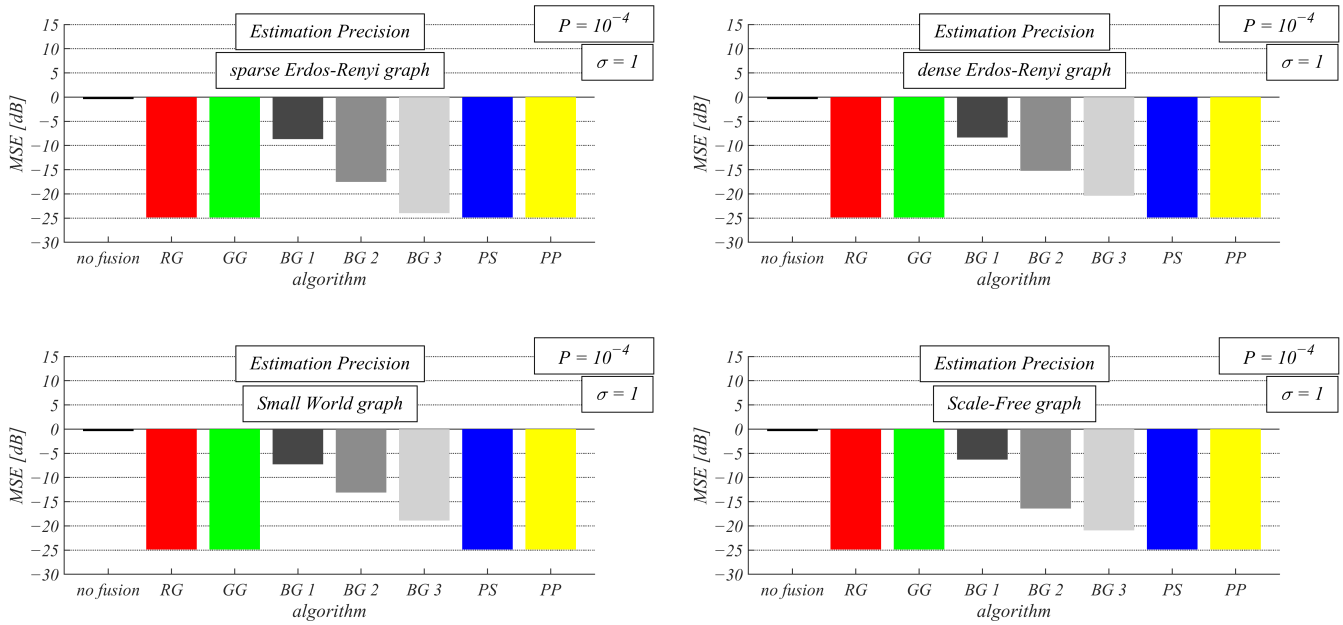


Figure 15. Mean square error of examined algorithms in various graph families for $P = 10^{-4}$ and $\sigma = 1$.

The findings presented in this paper demonstrate that the tested algorithms are highly effective (except on the BG, which is not as effective as the other algorithms) at mitigating the impact of negative factors, represented by Gaussian noise, across all examined scenarios.

Consequently, further optimization of these algorithms in this regard is deemed unnecessary, with the exception of the BG, which exhibited unacceptable precision in certain scenarios. Future research will focus on optimizing these algorithms for noisy WSNs by reducing the number of sent messages and enhancing the estimation precision of the BG.

In Figure 16, we compare the best-performing distributed consensus gossip algorithms with the best constant weights, which is a deterministic consensus algorithm recently identified as the top-performing algorithm for suppressing Gaussian noise within its category [68,69]. In the densely connected RGGs, the comparison was made with the PS, which has been identified as the best gossip algorithm for these graphs. From Figure 16, we can see that the estimation errors of these two algorithms do not notably differ. However, the best constant weights required significantly fewer sent messages than the PS in every examined scenario. In the sparsely connected RGGs, the best constant weights was compared with the GG (the best gossip algorithm for these graphs) and achieved nearly the same precision as the GG. However, the GG significantly outperformed the best constant weights in terms of the number of sent messages.

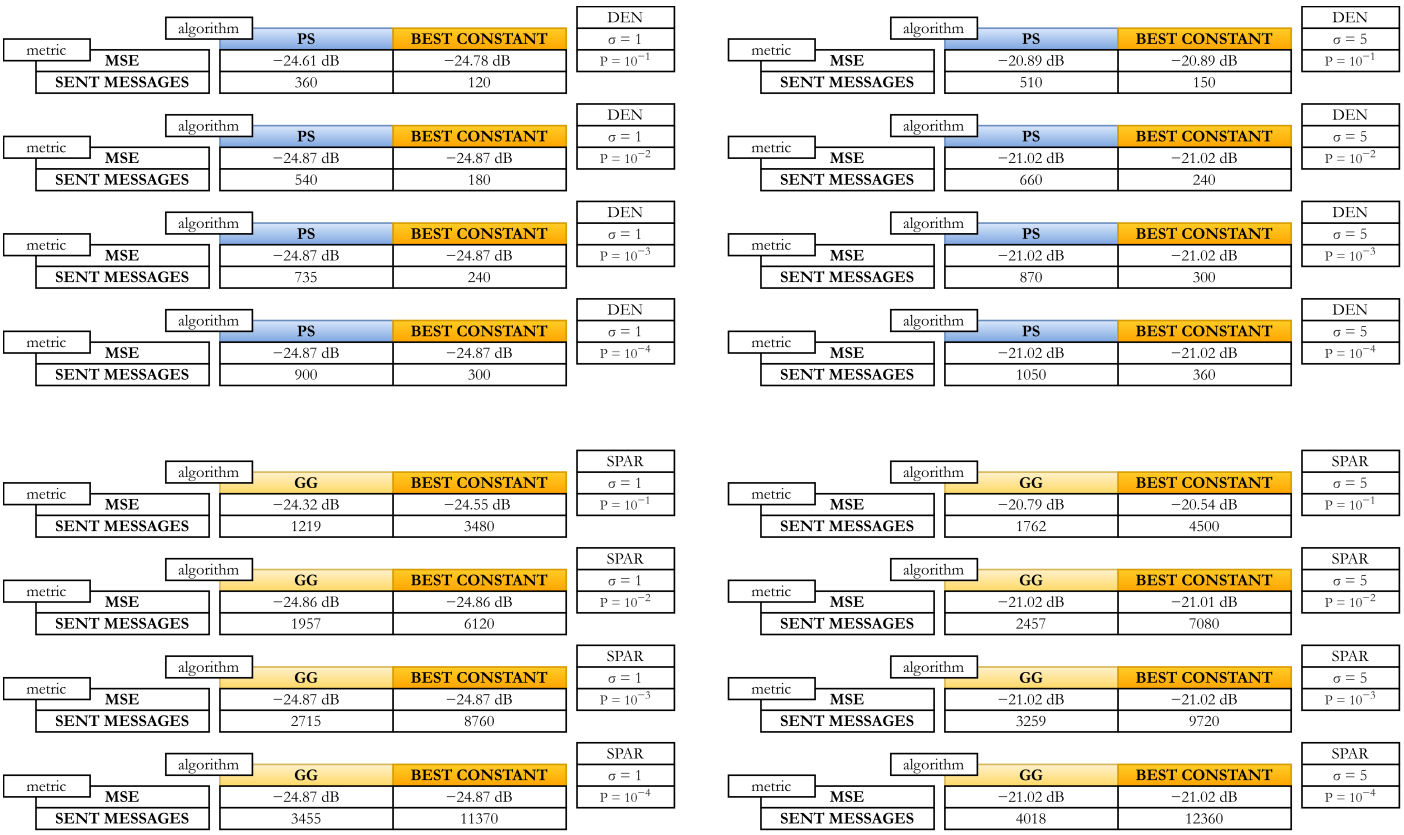


Figure 16. Comparison of best-performing gossip algorithms for sensor fusion with best constant weights (best deterministic algorithm) when Gaussian noise is alleviated.

6. Future Plans

Building on the findings presented in this paper, our future research will focus on testing other sensor fusion mechanisms (e.g., the convex optimized weights, the distributed flooding algorithm, etc.) and comparing them with the results presented here. Additionally, we plan to explore ways to optimize these algorithms in terms of the number of sent messages (e.g., by proposing novel stopping criteria) and investigate methods to improve the precision of the BG.

7. Conclusions

The goal of this paper was to examine the applicability of seven widely used distributed consensus gossip algorithms for sensor fusion, specifically to suppress incorrect sensor readings represented by Gaussian noise. The algorithms were bounded by a stopping criterion with various initial configurations, tested in RGGs with different connectivity levels, affected by noise with varying values of σ, and evaluated using two metrics. The presented results show that the best performance was achieved by the PS in dense RGGs and by the GG in sparse RGGs (indicating that employing geographical routing is beneficial in sparsely connected graphs). All the algorithms, except for the BG, achieved very high precision values and can significantly alleviate Gaussian noise in WSNs. The BG required the fewest sent messages, but its error was much higher compared to the other algorithms. Although the BG can also suppress noise in WSNs, it was not as effective as the other algorithms. The MSE values of all the algorithms, except for the BG, could not drop below the threshold value determined by σ (a higher value of σ caused the value of this threshold to be higher). From the presented results, we can conclude that $P = 10^{-1}$ (the highest examined value of P) is the optimal configuration for the applied stopping criterion, as decreasing P has only a marginal or no impact on the MSE. However, decreasing P results in an increase in the number of sent messages. Therefore, low values of P offer no benefits compared to error-free scenarios. The most important conclusion is that all the algorithms are effective in suppressing Gaussian noise in WSNs. The algorithms (except on the BG) achieved precision values similar to those of the best-performing distributed consensus deterministic algorithm (the best constant weights). In terms of sent messages, the best constant weights algorithm performed better than the PS in dense RGGs but was outperformed by the GG in sparse RGGs.

Author Contributions: Conceptualization, M.K. and J.K.; methodology, M.K.; software, M.K. and S.H.D.; validation, M.K. and J.K.; formal analysis, M.K. and J.K.; investigation, M.K.; resources, M.K. and S.H.D.; data curation, M.K.; writing—original draft preparation, M.K.; writing—review and editing, M.K. and J.K.; visualization, M.K.; supervision, J.K. and S.H.D.; project administration, M.K. and S.H.D.; funding acquisition, M.K. and S.H.D. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Slovak Scientific Grant Agency VEGA under the contract 2/0135/23 "Intelligent sensor systems and data processing" and by the Slovak Research and Development Agency under the contract No. SK-SRB-23-0038 and No. APVV-23-0292.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article and are openly available in https://github.com/kenyeresm/gossipgaussian.

Acknowledgments: We would like to thank the anonymous reviewers of this paper for their supportive and insightful comments.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

RG	Randomized Gossip algorithm
GG	Geographic Gossip algorithm
BG	Broadcast Gossip algorithm

PS	Push-Sum protocol
PP	Push-Pull protocol
WSN	Wireless Sensor Network
RGG	Random Geometric Graph
IoT	Internet of Things
MSE	Mean Square Error
IID	Independent and Identically Distributed
DEN	Densely connected Random Geometric Graph
SPAR	Sparsely connected Random Geometric Graph
BG 1	Broadcast Gossip algorithm with $\gamma = 0.1$
BG 2	Broadcast Gossip algorithm with $\gamma = 0.5$
BG 3	Broadcast Gossip algorithm with $\gamma = 0.9$
Zigbee	Zonal Intercommunication Global Standard
BLE	Bluetooth Low Energy
LoRa	Long Range
IEEE	Institute of Electrical and Electronics Engineers
UAV	Unmanned Aerial Vehicle
QoS	Quality of Service
CO2	Carbon dioxide
O3	Ozone
SO2	Sulfur Dioxide
NO2	Nitrogen Dioxide
CO	Carbon Monoxide
CH4	Methane

Appendix A

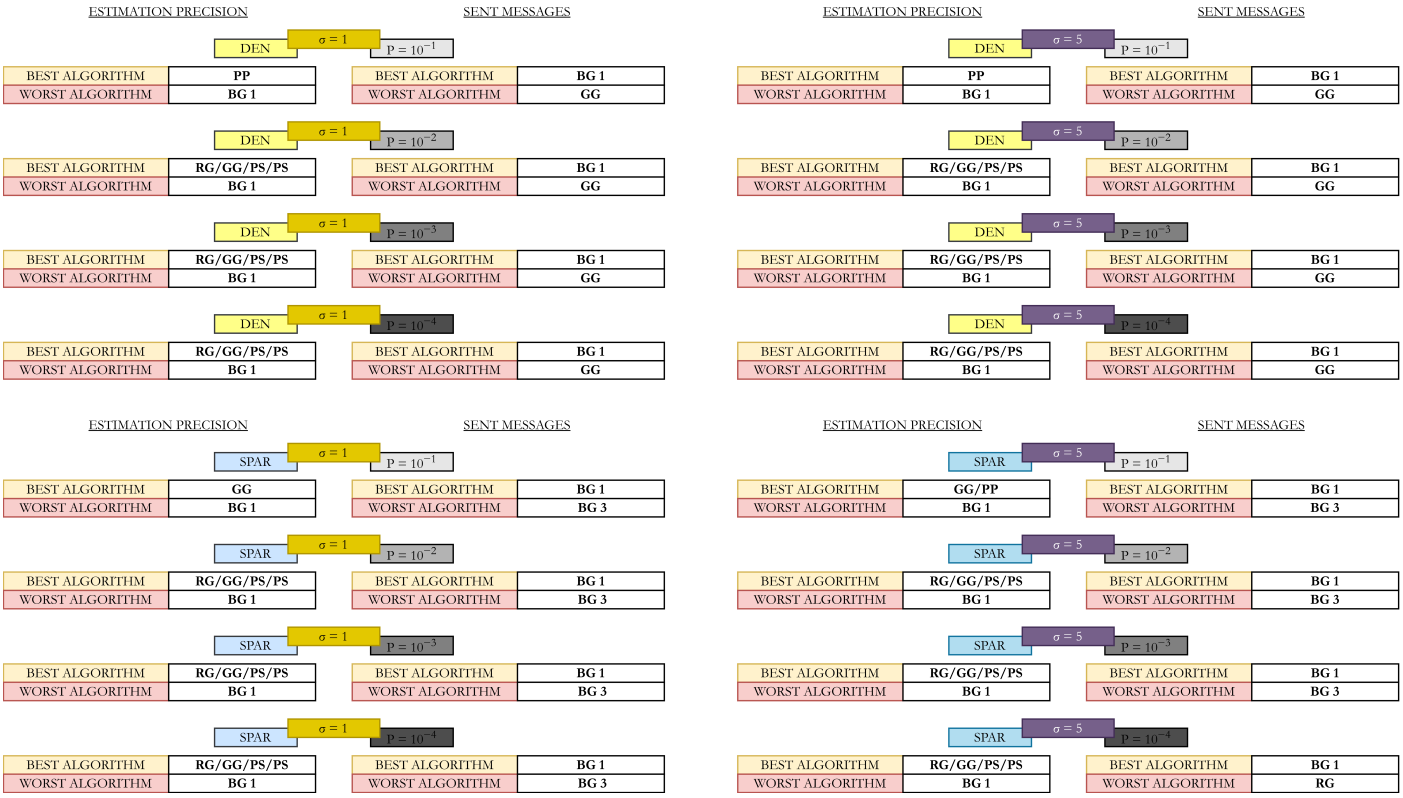


Figure A1. Summarization of simulation results—best and worst algorithm for both metrics, each σ, each connectivity, and each P.

References

1. Li, Y.; Im, P.; Lee, S.; Bae, Y.; Yoon, Y.; Lee, S. Sensor incipient fault impacts on building energy performance: A case study on a multi-zone commercial building. *Buildings* **2023**, *13*, 520. [CrossRef]
2. Kraft, R.; Hofmann, F.; Reichert, M.; Pryss, R. Dealing with inaccurate sensor data in the context of mobile crowdsensing and mhealth. *IEEE J. Biomed. Health Inform.* **2022**, *26*, 5439–5449. [CrossRef] [PubMed]
3. Carvajal, A.; Garcia-Colon, V.R. High capacity motors on-line diagnosis based on ultra wide band partial discharge detection. In Proceedings of the 4th IEEE International Symposium on Diagnostics for Electric Machines, Power Electronics and Drives, Atlanta, GA, USA, 24–26 August 2003.
4. Izadi, D.; Abawajy, J.H.; Ghanavati, S.; Herawan, T. A data fusion method in wireless sensor networks. *Sensors* **2015**, *15*, 2964–297. [CrossRef] [PubMed]
5. Selvaraj, S.K.; Pundikai, V.B. Outlier tolerant adaptive sampling rate approach for wireless sensor node. *Indones. J. Electr. Eng. Comput. Sci.* **2022**, *27*, 281–289. [CrossRef]
6. Shao, T. Distributed filtering in sensor networks based on linear minimum mean square error criterion with limited sensing range. *Int. J. Distrib. Sens. Netw.* **2022**, *18*, 15501329221110810. [CrossRef]
7. Jamshed, M.A.; Ali, K.; Abbasi, Q.H.; Imran, M.A.; Ur-Rehman, M. Challenges, applications, and future of wireless sensors in Internet of Things: A review. *IEEE Sens. J.* **2022**, *22*, 5482–5494. [CrossRef]
8. Ouni, R.; Saleem, K. Framework for sustainable wireless sensor network based environmental monitoring. *Sustainability* **2022**, *14*, 8356. [CrossRef]
9. Ahmad, R.; Wazirali, R.; Abu-Ain, T. Machine learning for wireless sensor networks security: An overview of challenges and issues. *Sensors* **2022**, *22*, 4730. [CrossRef]
10. Nakas, C.; Kandris, D.; Visvardis, G. Energy efficient routing in wireless sensor networks: A comprehensive survey. *Algorithms* **2020**, *13*, 72. [CrossRef]
11. Behera, T.M.; Samal, U.C.; Mohapatra, S.K.; Khan, M.S.; Appasani, B.; Bizon, N.; Thounthong, P. Energy-efficient routing protocols for wireless sensor networks: Architectures, strategies, and performance. *Electronics* **2022**, *11*, 2282. [CrossRef]
12. Chen, S.; Zhang, L.; Tang, Y.; Shen, C.; Kumar, R.; Yu, K.; Tariq, U.; Bashir, A.K. Indoor temperature monitoring using wireless sensor networks: A SMAC application in smart cities. *Sustain. Cities Soc.* **2020**, *61*, 102333. [CrossRef]
13. Vidaña-Vila, E.; Navarro, J.; Borda-Fortuny, C.; Stowell, D.; Alsina-Pagès, R.M. Low-cost distributed acoustic sensor network for real-time urban sound monitoring. *Electronics* **2020**, *9*, 2119. [CrossRef]
14. Metia, S.; Nguyen, H.A.; Ha, Q.P. IoT-enabled wireless sensor networks for air pollution monitoring with extended fractional-order Kalman filtering. *Sensors* **2021**, *21*, 5313. [CrossRef] [PubMed]
15. Mabrouki, J.; Azrour, M.; Dhiba, D.; Farhaoui, Y.; El Hajjaji, S. IoT-based data logger for weather monitoring using arduino-based wireless sensor networks with remote graphical application and alerts. *Big Data Min. Anal.* **2021**, *4*, 25–32. [CrossRef]
16. Aminian, M.; Naji, H.R. A hospital healthcare monitoring system using wireless sensor networks. *J. Health Med. Inf.* **2013**, *4*, 121. [CrossRef]
17. Nourildean, S.W.; Hassib, M.D.; Mohammed, Y.A. Internet of things based wireless sensor network: A review. *Indones. J. Electr. Eng. Comput. Sci* **2022**, *27*, 246–261. [CrossRef]
18. Ragnoli, M.; Leoni, A.; Barile, G.; Ferri, G.; Stornelli, V. LoRa-Based Wireless Sensors Network for Rockfall and Landslide Monitoring: A Case Study in Pantelleria Island with Portable LoRaWAN Access. *J. Low Power Electron. Appl.* **2022**, *12*, 47. [CrossRef]
19. Labib, M.I.; ElGazzar, M.; Ghalwash, A.; AbdulKader, S.N. An efficient networking solution for extending and controlling wireless sensor networks using low-energy technologies. *PeerJ Comput. Sci.* **2021**, *7*, e780. [CrossRef]
20. Suji Prasad, S.J.; Thangatamilan, M.; Suresh, M.; Panchal, H.; Rajan, C.A.; Sagana, C.; Gunapriya, B. Sharmag, A.; Panchal, T.; Sadasivuni, K.K. An efficient LoRa-based smart agriculture management and monitoring system using wireless sensor networks. *Int. J. Ambient Energy* **2022**, *43*, 5447–5450. [CrossRef]
21. Akbar, M.S.; Hussain, Z.; Sheng, M.; Shankaran, R. Wireless body area sensor networks: Survey of mac and routing protocols for patient monitoring under IEEE 802.15.4 and IEEE 802.15.6. *Sensors* **2022**, *22*, 8279. [CrossRef]
22. Malik, N.N.; Alosaimi, W.; Uddin, M.I.; Alouffi, B.; Alyami, H. Wireless sensor network applications in healthcare and precision agriculture. *J. Healthc. Eng.* **2020**, *1*, 8836613. [CrossRef]
23. Zhang, H.; Xing, S.; Wang, J. Security and application of wireless sensor network. *Procedia Comput. Sci.* **2021**, *183*, 486–492.
24. Ullo, S.L.; Sinha, G.R. Advances in smart environment monitoring systems using IoT and sensors. *Sensors* **2020**, *20*, 3113. [CrossRef] [PubMed]
25. Majid, M.; Habib, S.; Javed, A.R.; Rizwan, M.; Srivastava, G.; Gadekallu, T.R.; Lin, J.C.W. Applications of wireless sensor networks and internet of things frameworks in the industry revolution 4.0: A systematic literature review. *Sensors* **2022**, *22*, 2087. [CrossRef] [PubMed]

26. Dampage, U.; Bandaranayake, L.; Wanasinghe, R.; Kottahachchi, K.; Jayasanka, B. Forest fire detection system using wireless sensor networks and machine learning. *Sci. Rep.* **2022**, *12*, 46. [CrossRef]
27. Arshad, J.; Rehman, A.U.; Othman, M.T.B.; Ahmad, M.; Tariq, H.B.; Khalid, M.A.; Moosa, M.A.R.; Shafiq, M.; Hamam, H. Deployment of wireless sensor network and iot platform to implement an intelligent animal monitoring system. *Sustainability* **2022**, *14*, 6249. [CrossRef]
28. Ragnoli, M.; Barile, G.; Leoni, A.; Ferri, G.; Stornelli, V. An autonomous low-power LoRa-based flood-monitoring system. *J. Low Power Electron. Appl.* **2020**, *10*, 15. [CrossRef]
29. Abdollahi, A.; Rejeb, K.; Rejeb, A.; Mostafa, M.M.; Zailani, S. Wireless sensor networks in agriculture: Insights from bibliometric analysis. *Sustainability* **2021**, *13*, 12011. [CrossRef]
30. Haque, M.E.; Asikuzzaman, M.; Khan, I.U.; Ra, I.H.; Hossain, M.S.; Shah, S.B.H. Comparative study of IoT-based topology maintenance protocol in a wireless sensor network for structural health monitoring. *Remote Sens.* **2020**, *12*, 2358. [CrossRef]
31. Ezeigweneme, C.A.; Nwasike, C.N.; Adekoya, O.O.; Biu, P.W.; Gidiagba, J.O. Wireless communication in electro-mechanical systems: Investigating the rise and implications of cordless interfaces for system enhancement. *Eng. Sci. Technol. J.* **2024**, *5*, 21–42. [CrossRef]
32. Sungheetha, A.; Sharma, R. Real time monitoring and fire detection using internet of things and cloud based drones. *J. Soft Comput. Paradig.* **2020**, *2*, 168–174. [CrossRef]
33. Amutha, J.; Sharma, S.; Nagar, J. WSN strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues. *Wirel. Pers. Commun.* **2020**, *111*, 1089–1115. [CrossRef]
34. Sigrist, L.; Ahmed, R.; Gomez, A.; Thiele, L. Harvesting-aware optimal communication scheme for infrastructure-less sensing. *ACM Trans. Internet Things* **2020**, *1*, 1–26. [CrossRef]
35. Alghamdi, T.A. Energy efficient protocol in wireless sensor network: Optimized cluster head selection model. *Telecommun. Syst.* **2020**, *74*, 331–345. [CrossRef]
36. Yun, W.K.; Yoo, S.J. Q-learning-based data-aggregation-aware energy-efficient routing protocol for wireless sensor networks. *IEEE Access* **2021**, *9*, 10737–10750. [CrossRef]
37. Sachan, S.; Sharma, R.; Sehgal, A. Energy efficient scheme for better connectivity in sustainable mobile wireless sensor networks. *Sustain. Comput. Informatics Syst.* **2021**, *30*, 100504. [CrossRef]
38. Landaluce, H.; Arjona, L.; Perallos, A.; Falcone, F.; Angulo, I.; Muralter, F. A review of IoT sensing applications and challenges using RFID and wireless sensor networks. *Sensors* **2020**, *20*, 2495. [CrossRef]
39. Dobrescu, R.; Popescu, D.; Nicolae, M.; Mocanu, S. Hybrid wireless sensor network for homecare monitoring of chronic patients. *Int. J. Biol. Biomed. Eng.* **2009**, *3*, 19–26.
40. Lata, S.; Mehfuz, S.; Urooj, S.; Alrowais, F. Fuzzy clustering algorithm for enhancing reliability and network lifetime of wireless sensor networks. *IEEE Access* **2020**, *8*, 66013–66024. [CrossRef]
41. Chamam, A.; Pierre, S. On the planning of wireless sensor networks: Energy-efficient clustering under the joint routing and coverage constraint. *IEEE Trans. Mob. Comput.* **2009**, *8*, 1077–1086. [CrossRef]
42. Buratti, C.; Conti, A.; Dardari, D.; Verdone, R. An overview on wireless sensor networks technology and evolution. *Sensors* **2009**, *9*, 6869–6896. [CrossRef] [PubMed]
43. Bissadu, K.D.; Hossain, G.; Velagala, L.P. Identifying Sensors Data Integrity Threats of Smart Agriculture: A Collaborative Filtering Approach. *Appl. Eng. Agric.* **2024**, *40*, 565–575. [CrossRef]
44. Bonnet, P.; Gehrke, J.; Seshadri, P. Towards sensor database systems. *Lect. Notes Comput. Sci.* **2001**, *1987*, 3–14.
45. Pirayesh, H.; Zeng, H. Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey. *IEEE Commun. Surv.* **2022**, *24*, 767–809. [CrossRef]
46. Kavousi-Fard, A.; Su, W.; Jin, T. A machine-learning-based cyber attack detection model for wireless sensor networks in microgrids. *IEEE Trans. Ind. Informatics* **2020**, *17*, 650–658. [CrossRef]
47. Krishnamurthi, R.; Kumar, A.; Gopinathan, D.; Nayyar, A.; Qureshi, B. An overview of IoT sensor data processing, fusion, and analysis techniques. *Sensors* **2020**, *20*, 6076. [CrossRef]
48. Yeong, D.J.; Velasco-Hernandez, G.; Barry, J.; Walsh, J.Sensor and sensor fusion technology in autonomous vehicles: A review. *Sensors* **2021**, *21*, 2140. [CrossRef]
49. Wei, Y.; Wu, D.; Terpenny, J. Robust incipient fault detection of complex systems using data fusion. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 9526–9534. [CrossRef]
50. Kong, L.; Peng, X.; Chen, Y.; Wang, P.; Xu, M. Multi-sensor measurement and data fusion technology for manufacturing process monitoring: A literature review. *Int. J. Extrem. Manuf.* **2020**, *2*, 022001. [CrossRef]
51. Li, D.; Wang, Y.; Wang, J.; Wang, C.; Duan, Y. Recent advances in sensor fault diagnosis: A review. *Sens. Actuators A Phys.* **2020**, *309*, 111990. [CrossRef]
52. Xiao, L.; Boyd, S.; Lall, S. A scheme for robust distributed sensor fusion based on average consensus. In Proceedings of the IPSN 2005, Fourth International Symposium on Information Processing in Sensor Networks, Boise, ID, USA, 15 April 2005; pp. 63–70.

53. Shrimali, B.; Patel, H.B. Blockchain state-of-the-art: Architecture, use cases, consensus, challenges and opportunities. *J. King Saud Univ.-Comput. Inf. Sci.* **2022**, *34*, 6793–6807. [CrossRef]
54. Mostefaoui, A.; Raynal, M. Leader-based consensus. *Parallel Process. Lett.* **2001**, *11*, 95–107. [CrossRef]
55. Merezeanu, D.; Nicolae, M. Consensus control of discrete-time multi-agent systems. *Univ. Politeh. Buchar. Ser. A* **2017**, *79*, 167–174.
56. Stamatescu, G.; Stamatescu, I.; Popescu, D. Consensus-based data aggregation for wireless sensor networks. *J. Control Eng. Appl. Informatics* **2017**, *19*, 43–50.
57. Dragana, C.; Mihai, V.; Stamatescu, G.; Popescu, D. In-network stochastic consensus for WSN surveillance applications. In Proceedings of the 9th International Conference on Electronics, Computers and Artificial Intelligence, ECAI 2017, Targoviste, Romania, 29 June–1 July 2017; pp. 1–6.
58. Lizzio, F.F.; Capello, E.; Guglieri, G. A review of consensus-based multi-agent UAV implementations. *J. Intell. Robot. Syst.* **2022**, *106*, 43. [CrossRef]
59. Sharma, A.B.; Golubchik, L.; Govindan, R. Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Trans. Sens. Networks (TOSN)* **2010**, *6*, 1–39. [CrossRef]
60. Clegg, M.; Marzullo, K. Predicting physical processes in the presence of faulty sensor readings. In Proceedings of the IEEE 27th International Symposium on Fault Tolerant Computing, Seattle, WA, USA, 24–27 June 1997; pp. 373–378.
61. Elnahrawy, E.; Nath, B. Cleaning and querying noisy sensors. In Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, New York, NY, USA, 19 September 2003; pp. 78–87.
62. Chen, Y.; Jiang, S.; Yang, J.; Song, K.; Wang, Q. Grey bootstrap method for data validation and dynamic uncertainty estimation of self-validating multifunctional sensors. *Chemom. Intell. Lab. Syst.* **2015**, *146*, 63–76. [CrossRef]
63. Yu, Y.; Li, H. Virtual in-situ calibration method in building systems. *Autom. Constr.* **2015**, *59*, 59–67. [CrossRef]
64. Solomakhina, N.; Hubauer, T.; Lamparter, S.; Roshchin, M.; Grimm, S. Extending statistical data quality improvement with explicit domain models. In Proceedings of the 2014 12th IEEE International Conference on Industrial Informatics (INDIN), Porto Alegre, Brazil, 27–30 July 2014; pp. 720–725.
65. Scheffel, R.M.; Fröhlich, A.A. Increasing sensor reliability through confidence attribution. *J. Braz. Comput. Soc.* **2019**, *25*, 13. [CrossRef]
66. Sinha, A.; Das, D. SNRepair: Systematically addressing sensor faults and self-calibration in IoT networks. *IEEE Sens. J.* **2023**, *23*, 14915–14922. [CrossRef]
67. Ayanoglu, M.B.; Uysal, I. ML Approach to Improve the Costs and Reliability of a Wireless Sensor Network. *Sensors* **2023**, *23*, 4303. [CrossRef] [PubMed]
68. Kenyeres, M.; Kenyeres, J. Average Consensus with Perron Matrix for Alleviating Inaccurate Sensor Readings Caused by Gaussian Noise in Wireless Sensor Networks. *Lect. Notes Netw. Syst.* **2021**, *230*, 391–405.
69. Kenyeres, M.; Kenyeres, J. Using Metropolis-Hastings Algorithm with Stopping Criterion to Suppress Incorrect Sensor Data. *Lect. Notes Netw. Syst.* **2024** *accepted*.
70. Meng, J.; Li, H.; Han, Z. Sparse event detection in wireless sensor networks using compressive sensing. In Proceedings of the 2009 43rd Annual Conference on Information Sciences and Systems, Baltimore, MD, USA, 18–20 March 2009; pp. 181–185.
71. Banos, O.; Damas, M.; Pomares, H.; Rojas, I. On the use of sensor fusion to reduce the impact of rotational and additive noise in human activity recognition. *Sensors* **2012**, *12*, 8039–8054. [CrossRef] [PubMed]
72. Kenyeres, M.; Kenyeres, J. Comparative Study of Distributed Consensus Gossip Algorithms for Network Size Estimation in Multi-Agent Systems. *Future Internet* **2021**, *13*, 134. [CrossRef]
73. Sasikumar, P.; Khara, S. K-means clustering in wireless sensor networks. In Proceedings of the 2012 Fourth International Conference on Computational Intelligence and Communication Networks, Mathura, India, 3–5 November 2012.
74. Park, J.; Moon, K.; Yoo, S.; Lee, S. Optimal stop points for data gathering in sensor networks with mobile sinks. *Wirel. Sens. Netw.* **2011**, *4*, 8–17. [CrossRef]
75. Dong, Q. Maximizing system lifetime in wireless sensor networks. In Proceedings of the IPSN 2005, Fourth International Symposium on Information Processing in Sensor Networks, Boise, ID, USA, 15 April 2005.
76. Fraser, B.; Coyle, A.; Hunjet, R.; Szabo, C. An Analytic Latency Model for a Next-Hop Data-Ferrying Swarm on Random Geometric Graphs. *IEEE Access* **2020**, *8*, 48929–48942. [CrossRef]
77. Gulzar, M.M.; Rizvi, S.T.H.; Javed, M.Y.; Munir, U.; Asif, H. Multi-agent cooperative control consensus: A comparative review. *Electronics* **2018**, *7*, 22. [CrossRef]
78. Bonato, A.; Lozier, M.; Mitsche, D.; Pérez-Giménez, X.; Prałat, P. The domination number of on-line social networks and random geometric graphs. In Proceedings of the Theory and Applications of Models of Computation: 12th Annual Conference, TAMC 2015, Singapore, 18–20 May 2015; pp. 150–163.
79. Kenniche, H.; Ravelomananana, V. Random geometric graphs as model of wireless sensor networks. In Proceedings of the 2010 The 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 26–28 February 2010; pp. 103–107.

80. Gutierrez-Gutierrez, J.; Zarraga-Rodriguez, M.; Insausti, X. Analysis of Known Linear Distributed Average Consensus Algorithms on Cycles and Paths. *Sensors* **2018**, *18*, 968. [CrossRef]
81. Dimakis, A.A.G.; Sarwate, A.D.; Wainwright, M.J.; Scaglione, A. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Process.* **2008**, *56*, 1205–1216. [CrossRef]
82. Aysal, T.C.; Yildiz, M.E.; Sarwate, A.D.; Scaglione, A. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Process.* **2009**, *57*, 2748–2761. [CrossRef]
83. Kempe, D.; Dobra, A.; Gehrke, J. Gossip-based computation of aggregate information. In Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2003, Cambridge, MA, USA, 11–14 October 2003; pp. 482–491.
84. Jelasity, M.; Montresor, A.; Babaoglu, O. Gossip-based aggregation in large dynamic networks. *ACM Trans. Comput. Syst. (TOCS)* **2005**, *23*, 219–252. [CrossRef]
85. Bhandari, S.; Bergmann, N.; Jurdak, R.; Kusy, B. Time series data analysis of wireless sensor network measurements of temperature. *Sensors* **2017**, *17*, 1221. [CrossRef] [PubMed]
86. Ayday, C.; Safak, S. Application of wireless sensor networks with GIS on the soil moisture distribution mapping. In Proceedings of the Symposium GIS Ostrava 2009, Ostrava, Czech Republic, 25–28 January 2009.
87. Peckens, C.; Porter, C.; Rink, T. Wireless sensor networks for long-term monitoring of urban noise. *Sensors* **2018**, *18*, 3161. [CrossRef] [PubMed]
88. Mangundu, E.M.; Mateus, J.N.; Zodi, G.A.L.; Johson, J. A wireless sensor network for rainfall monitoring, using cellular network: A case for namibia. In Proceedings of the 2017 Global Wireless Summit (GWS), Cape Town, South Africa, 15–18 October 2017.
89. Yi, W.Y.; Lo, K.M.; Mak, T.; Leung, K.S.; Leung, Y.; Meng, M.L. A survey of wireless sensor network based air pollution monitoring systems. *Sensors* **2015**, *15*, 31392–31427. [CrossRef]
90. Barroca, N.; Borges, L.M.; Velez, F.J.; Monteiro, F.; Górski, M.; Castro-Gomes, J. Wireless sensor networks for temperature and humidity monitoring within concrete structures. *Constr. Build. Mater.* **2013**, *40*, 1156–1166. [CrossRef]
91. Al-Hemyari, Z.A.; Abbasi, J.N.A. Modelling and analysing the daily temperature of several cities using mixture Gaussian distributions. *Int. J. Comput. Sci. Math.* **2023**, *17*, 320–341. [CrossRef]
92. Mu, H.; Li, Z.; Wang, X.; Liu, S. Temperature distribution measurement using the Gaussian process regression method. *Math. Probl. Eng.* **2017**, *2017*, 2147935. [CrossRef]
93. Pan, S.; Han, Y.; Wei, S.; Wei, Y.; Xia, L.; Xie, L.; Kong, X.; Yu, W. A model based on Gauss Distribution for predicting window behavior in building. *Build. Environ.* **2019**, *149*, 210–219. [CrossRef]
94. Barber, M.; Grings, F.; Perna, P.; Piscitelli, M.; Maas, M.; Bruscantini, C.; Jacobo-Berlles, J.; Karszenbaum, H. Speckle noise and soil heterogeneities as error sources in a Bayesian soil moisture retrieval scheme for SAR data. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens.* **2012**, *5*, 942–951. [CrossRef]
95. De Lannoy, G.J.; Houser, P.R.; Pauwels, V.R.; Verhoest, N.E. Assessment of model uncertainty for soil moisture through ensemble verification. *J. Geophys. Res. Atmos.* **2006**, *111*, 1–18. [CrossRef]
96. Cai, M.; Lan, Z.; Zhang, Z.; Wang, H. Evaluation of road traffic noise exposure based on high-resolution population distribution and grid-level noise data. *Build. Environ.* **2019**, *147*, 211–220. [CrossRef]
97. Wilson, P.S.; Toumi, R. A fundamental probability distribution for heavy rainfall. *Geophys. Res. Lett.* **2005**, *32*, L14812. [CrossRef]
98. Fernandes, A.P.; Rafael, S.; Lopes, D.; Coelho, S.; Borrego, C.; Lopes, M. The air pollution modelling system URBAIR: How to use a Gaussian model to accomplish high spatial and temporal resolutions. *Air Qual. Atmos. Health* **2021**, *14*, 1969–1988. [CrossRef]
99. Spichtinger, P.; Gierens, K.; Smit, H.G.J.; Ovarlez, J.; Gayet, J.F. On the distribution of relative humidity in cirrus clouds. *Atmos. Chem. Phys.* **2004**, *4*, 639–647. [CrossRef]
100. Alemdar, A.; Ibnkahla, M. Wireless sensor networks: Applications and challenges. In Proceedings of the 2007 9th International Symposium on Signal Processing and Its Applications, Sharjah, United Arab Emirates, 12–15 February 2007.
101. Skorpil, V.; Stastny, J. Back-propagation and k-means algorithms comparison. In Proceedings of the 2006 8th International Conference on Signal Processing, ICSP 2006, Guilin, China, 16–20 November 2006; pp. 374–378.
102. Lee, C.S.; Michelusi, N.; Scutari, G. Finite rate distributed weight-balancing and average consensus over digraphs. *IEEE Trans. Autom. Control* **2020**, *66*, 4530–4545. [CrossRef]
103. Sirocchi, C.; Bogliolo, A. Topological network features determine convergence rate of distributed average algorithms. *Sci. Rep.* **2022**, *12*, 21831. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.