IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 24, 2022, accepted March 9, 2022, date of publication March 25, 2022, date of current version March 31, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3162288

Robustness Enhanced Sensor Assisted Monte Carlo Localization for Wireless Sensor Networks and the Internet of Things

ARNE BOCHEM AND HANG ZHANG
Telematics Group, Institute of Computer Science, University of Goettingen, 37073 Göttingen, Germany
Corresponding author: Arne Bochem (bochem@informatik.uni-goettingen.de)

ABSTRACT With distributed sensor systems commonly found in Wireless Sensor Networks or the Internet of Things, knowing the location sensor data was acquired from is very important, especially in scenarios with mobile sensors. Range-free Monte Carlo Localization based approaches are very energy efficient and do not require additional hardware beyond a radio, which is found on sensor nodes anyways. However, the use of motion sensor data based dead reckoning greatly improves the accuracy of location estimates and increases robustness against faulty or malicious actors within the network. In this work, we propose Robustness Enhanced Sensor Assisted Monte Carlo Localization (RESA-MCL). We show RESA-MCL's effectiveness with respect to both general localization accuracy and robustness against malicious attacks or malfunctioning nodes. To evaluate and compare our scheme against existing approaches, we introduce three attack models based on malicious anchor nodes. The performance of RESA-MCL is evaluated under these attack models and our approach outperforms existing schemes in both very low and higher anchor node density environments, achieving a localization error of 0.5 with an anchor density of 0.33. Overall, RESA-MCL outperforms comparable approaches at lower anchor densities with up to 48 % lower localization error and demonstrates strongly increased robustness against attacks with minimal computational overhead.

INDEX TERMS Localization, wireless sensor networks, security, Internet of Things, Monte Carlo localization, range-free localization.

I. INTRODUCTION

In today's world, more and more Internet of Things (IoT) devices with various types of sensors, as well as Wireless Sensor Networks (WSN), are getting deployed to cover a wide range of scenarios, from smart homes [9], decentralized initiatives by volunteers for measuring air quality [2], [17], over industrial uses [10] to wildlife monitoring [16]. To make sense of the gathered data, it is important to know where it was measured. In the case of, for example, a WSN with fixed nodes, the installation points of each sensor can be noted, but many applications rely on mobile sensors, which makes it necessary for sensor nodes to be able to determine their locations dynamically.

The most common approach to this is the use of the Global Positioning System (GPS). However, the use of GPS has a number of disadvantages. The sensors are relatively costly and consume high amounts of power. They also rely on being able to receive satellite signals, which makes indoor operations impossible and also leads to reduced accuracy in certain outdoor environments. To mitigate the first two points, a solution is to equip only a small subset of nodes with GPS sensors. These nodes then act as so-called seed or anchor nodes, which assist other nodes in localizing themselves. Instead of using mobile anchor nodes equipped with GPS sensors, the use of static anchors with preset locations is also a common approach.

Different types of localization algorithms exist. They can be mainly divided into range-based [4], [12], [18], [19], [21] and range-free approaches [6]–[8], [15]. In range-based approaches, unknown nodes (non-anchor nodes trying to localize themselves) must actively determine the distance to anchor nodes or the angles of incoming radio signals. Common measurements used in such approaches include

The associate editor coordinating the review of this manuscript and approving it for publication was Lei Shu.

Time of Arrival (ToA) [21], Time Difference of Arrival (TDoA) [22], Angle of Arrival (AoA) [4], and Received Signal Strength (RSS) [20].

A popular technique is to use RSS along with an appropriate propagation model to estimate the distance between an unknown node and an anchor node. This solution is based on the assumption that the RSS decreases proportionally with increasing distance from the transmitter. However, certain limitations need to be considered. For example, in TDoA-based solutions highly precise clock synchronization between nodes must be guaranteed; it's important to consider the multipath, Non-Line-of-Sight (NLoS) conditions and array calibration in AoA-based solutions; radio noise levels, multipathing and measurement errors can affect the performance of RSS-based solutions [21]. Generally speaking, range-based approaches typically require additional, specialized hardware, clock synchronization and have higher power consumption, which enables the active measurements that have to be performed by unknown nodes. In addition, the inherent limitations of each type of measurement can affect localization accuracy in certain situations.

In order to reduce complexity, hardware dependency and energy costs, research is conducted on range-free solutions which are usually based on connectivity alone. Since no active measurements are required from the unknown nodes, these approaches are easier to implement and also have lower deployment costs. A well-known representative approach in this category is the Monte Carlo Localization (MCL) algorithm, which was adapted for localization in mobile WSNs by Hu and Evans in 2004 [8]. Unlike approaches designed for partially or fully static networks, MCL allows all nodes in the network to move arbitrarily over time and uses the movement to improve localization performance. The probability distribution of each node's current position is represented as a set of weighted samples (particles) in MCL. Impossible samples which are outside the communication range of anchor nodes are eliminated by a Bayesian filtering process. The estimated node's location is the average of all remaining samples after the filtering process. MCL requires no additional hardware and is suitable for both mobile and static scenarios. However, MCL's performance quickly degrades in situations where connectivity between unknown nodes and anchor nodes only occurs rarely. Therefore, guaranteeing a high anchor node density is very important when applying MCL.

The various approaches presented so far have not considered the security aspects of localization. If malicious nodes that publish erroneous locations are present in the network, these approaches may not work as well as shown in their experiments. Even in the absence of malicious nodes, performance degradation or even collapse of the entire system may occur if a subset of the anchor nodes does not function properly. For example, in approaches where anchor nodes are statically distributed in the environment, if one or multiple of the anchor nodes do not work, normal nodes in the vicinity of these anchor nodes will not be able to obtain the information required for localization such as RSS, TDoA, etc., which will eventually lead to localization failure. Therefore, to make the proposed approaches more suitable for real-world scenarios, anchor node malfunctions need to be considered in the development of the localization algorithms.

In this work, we propose Robustness Enhanced Sensor Assisted Monte Carlo Localization (RESA-MCL), which both achieves a higher localization accuracy compared to previous schemes and also improves the localization scheme's robustness against incorrect information being broadcast by malicious anchor nodes. To achieve this, RESA-MCL continuously employs dead reckoning, as described by Hartung *et al.* [7], instead of only using it when out of anchor range like the original SA-MCL scheme. RESA-MCL detects malicious anchor nodes through motion-based plausibility checks and limits the influence that malicious nodes can exert on the location estimate through a novel particle subsetting technique. We thoroughly analyze the performance of RESA-MCL under different attack models with up to 90 % of anchor nodes acting maliciously and motivate parameter choices in data-driven manner.

The contributions of our paper can be summarized as follows: (1) We propose the RESA-MCL scheme, which achieves higher localization accuracy (up to 48 % lower error than comparable recent approach) in fully mobile low anchor density situations than comparable schemes with low computational complexity while also including robustness enhancements to mitigate attacks. (2) Specific attack models for scenarios with malicious or malfunctioning anchor nodes are defined. (3) RESA-MCL and previous approaches are evaluated and compared under three different attack scenarios. (4) Robustness enhancements in RESA-MCL greatly improve its performance in all attack models compared to approaches without robustness enhancements. (5) Thorough experimental evaluation including ablation experiments are used to verify the favorable properties of the scheme, such as low localization error at low anchor densities and resistance against attacks. (6) An optimized version of the original MCL [8] simulator—including implementations of SA-MCL [7] and RESA-MCL—is provided to ease future evaluation and comparison with other approaches.

The structure of this paper is organized as follows: Section II reviews related works. In Section III, we describe our proposed scheme in detail. Our evaluation methodology and results are shown in Section IV. Finally, we present conclusions and future works in Section V.

II. RELATED WORKS

Various localization algorithms for WSNs/IoT have been proposed in recent years. Concerning our work, the consideration of security aspects in localization is the most relevant factor. Therefore, we present overviews of related works classified into four groups: range-based, range-free, Artificial Intelligence (AI)-based and security-aware approaches.

A. RANGE-BASED APPROACHES

As introduced in section I, many range-based localization approaches have been proposed. Luo *et al.* proposed an

RSS-based Localization using Uncertain Data Mapping (LUDM) for WSN [12]. Simulation results show that the proposed approach outperforms other solutions in terms of the absolute mean localization error. However, the four anchor nodes are statically fixed at the corners of the experimental area. Moreover, the localization accuracy may decrease greatly when resorting to the RSS attenuation model to improve the generality in unknown localization environments.

In 2019 Wang *et al.* proposed a Time of Flight (ToF)-based localization algorithm for asynchronous WSN [18]. The simulation results show that the proposed approach outperforms conventional algorithms in terms of localization accuracy. However, again the anchor nodes are statically deployed in the network and localization depends on a centralized server, which must provide sufficient computational capacity for estimating clock skews and performing the localization procedure. This can cause the entire localization process to break down due to a single point of failure if server issues occur.

In 2020, the same authors proposed another time-based joint synchronization and localization algorithm for asynchronous WSN which utilizes TDoA [19]. It also relies on a centralized server for launching the localization procedures. Simulation results show that this new approach is superior in scenarios where anchor positions are imperfectly known. However, its centralized structure means that there is a single point of failure.

In 2021, Ding *et al.* proposed an indoor localization algorithem based on Error variance and measurement Noise Weighted Least Squares, named ENWLS [4], which is a weighted algorithm for localization in 3D WSN based on RSS/AOA measurements. Simulation results show that ENWLS outperforms other existing hybrid RSS/AOA localization algorithms when there are more than three anchor nodes in the scenario. This implies that ENWLS is not reliable in case of malfunctioning of anchor nodes. Moreover, multipath effects and NLoS are not considered.

Range-based approaches aim at enhancing localization accuracy by measuring ranges between unknown nodes and anchor nodes. However, to perform such measurements, they often require strict clock synchronization or special hardware. Additionally, multipathing or other environmental conditions may interfere with such systems. This makes them harder and more expensive to deploy.

B. RANGE-FREE APPROACHES

As presented in Section I, MCL [8] is a representative range-free localization approach that requires no additional hardware. More importantly, MCL allows all nodes including anchors in the network to move arbitrarily over time. Node mobility is leveraged to further increase localization accuracy through the use of a particle filter. Strong mobility support is one of the outstanding features of MCL-based approaches.

Inspired by MCL, many range-free approaches have been proposed, improving localization accuracy and sampling efficiency [15], [25]. However, these solutions cannot solve the problem of localization failure due to connectivity loss caused by dynamic changes in the network topology and low anchor node density. In order to solve this problem, Hartung *et al.* proposed the Sensor-Assisted Monte Carlo Localization (SA-MCL) method in [7]. The algorithm can specifically handle the temporary loss of all connectivity to anchor nodes in the network, which greatly improves the shortcomings of the original MCL algorithm. Low-cost 9-axis Inertial Measurement Unit (IMU) sensors are used to perform dead reckoning to bridge periods without connectivity to anchor nodes.

In order to improve localization accuracy under low anchor node density, Qin and Zhu [15] adapted MCL through the use of the Differential Evolution optimization algorithm (MCL-DE). Instead of using the regular sample filtering and resampling algorithms, MCL-DE selects the sample weight as the objective function for optimization and implements the differential evolution algorithm to obtain valid samples for location prediction. The authors found that MCL-DE has enhanced localization accuracy. However, the computational and communication costs of the proposed scheme are not studied and no security implications are considered.

Range-free localization approaches not based on MCL also exist. One common group of range-free approaches are those based on DV-Hop. In 2020, Gui *et al.* [6] introduced a decentralized, range-free approach based on the DV-Hop family of localization approaches. Centralized Connectivity based DV-Hop (CCDV-Hop) and Distributed Connectivity based DV-Hop (DCDV-Hop) have relatively low computational complexity and achieve low localization error. However, malfunctioning or malicious anchor nodes are not considered and simulation results are provided only for small and medium-sized networks of up to 30 nodes. Additionally, it was only evaluated with relatively high numbers of anchor nodes (50 % of unknown nodes in the small network of 6 unknown nodes, 33.3 % in the medium network of 9 unknown nodes) with high communication ranges relative to the small simulation areas (20 m in a $40 \times 40\,\mathrm{m}^2$ or $60 \times 60\,\mathrm{m}^2$ areas), allowing each anchor node to cover a high percentage of the experimental area. The authors do not provide an anchor node density measure.

Overall, range-free approaches trade off some localization accuracy for ease of deployment, cost-effectiveness and more generalized applicability. Since the only requirement to deploy common range-based localization approaches is the ability to decide whether a node is within radio range or not, this type of approach can be employed even with very basic hardware.

C. AI-BASED APPROACHES

Along with range-based and range-free localization algorithms, researchers have also applied artificial intelligence (AI) techniques to develop new indoor and outdoor localization solutions. Chen *et al.* proposed ConFi, the first Convolutional Neural Network (CNN)-based indoor Wi-Fi localization method, in 2017 [3]. It uses Channel State Information (CSI) to build a time-frequency matrix that is utilized

as the feature for localization. The authors conducted extensive experiments to select the parameters for the CNN and also show that ConFi outperforms the existing solutions. However, the amount of samples required to train the CNN is high. To apply ConFi in a new environment that is not comparable to the current one, the model must be trained with a new dataset from that environment.

Gharghan *et al.* [5] proposed an adaptive neural fuzzy inference system to estimate the distance between a moving bicycle (i.e., player) and a static coordinator node (i.e., coach) for the indoor and outdoor velodromes. Simulation results show that the proposed approach outperforms other state-of-the-art systems in terms of mean absolute error. However, the offline phase for Adaptive Neural Fuzzy Inference System (ANFIS) training is time-consuming and depends heavily on the complexity of the fuzzy inference system. Furthermore, the current approach requires the training of two ANFIS systems for the localization coordinates seperately which increases the overall training cost.

Besides using CSI and RSS to train the neutral model, researchers have also proposed approaches that apply AI techniques based on Wi-Fi fingerprints, such as HybLoc [1], which is a hybrid indoor localization system for both room-level and latitude-longitude predictions. Simulation results show that HybLoc has better performance in terms of accuracy and precision. However, the hardware requirements for the sensor nodes to achieve a short response time in the prediction phase, which is very important to evaluate whether HybLoc is applicable in real scenarios, are not presented. In addition, without re-training the model, HybLoc is not applicable for localization in a new environment, such as a building that is not in the dataset.

Munadhil *et al.* [14] proposed a neural network-based localization system for WSNs in an indoor environment that is specially designed to determine the position of an Alzheimer's patient. It utilizes the RSS with respect to the anchor nodes. Simulation results show that the proposed approach outperforms other previous techniques in terms of mean localization error. However, in the experiment, the mobile node carried by the patient must be connected to a laptop to record the RSS samples, configure the wireless connection, and supply power. This is an impractical hardware requirement for most real-world scenarios. Moreover, due to the static anchor nodes and the offline training phase, the approach cannot simply be transferred to other environments or mobile scenarios.

Overall, these AI-based approaches are computationally intensive and usually only apply to specific environments they were trained for. Furthermore, all anchor nodes in these approaches are positioned in static locations. Additionally, they do not consider security implications of malfunctioning or malicious anchor nodes, which may pose issues in real-world applications. In the end, the system complexity and the high offline training cost of these AI models should not be underestimated. However, more general and lightweight approaches also exist, which are more suitable for environments like WSNs.

D. SECURITY-AWARE APPROACHES

Besides improving localization accuracy, sampling efficiency, and coping with transient connectivity loss, addressing security aspects of localization is also an active research topic. Many security-aware localization algorithms have been proposed. In [21], Xie *et al.* proposed a lightweight secure ToA-based localization algorithm in WSNs, exploiting the noise features caused by external distance attacks. This approach aims mainly to defend against impersonation attacks launched by external attackers.

Liu *et al.* proposed a Malicious Node Detection algorithm based on Clustering and consistency evaluation (MNDC) along with an enhanced secure localization version called EMDC, both of which are range-based localization schemes [11]. They use density-based spatial clustering to detect the abnormal clusters of nodes. A sequential probability ratio test is then used to identify malicious nodes that compromise the networks. The conducted simulations show that the proposed algorithms outperforms other state-of-art schemes in terms of detection accuracy and effectiveness. However, the computational overhead caused by the clustering algorithm and the sequential probability ratio test is not explained. In addition, there is no mechanism for anchor nodes that are identified as malicious to recover their reputations. As a range-based scheme, it also has additional requirements for deployment compared to range-free schemes.

Yuan *et al.* [24] proposed a secure APIT-based range-free scheme in 2018, which attempts to detect Sybil nodes inside the network. Sybil-free APIT (SF-APIT) is evaluated with a relatively low number of anchor nodes compared to unknown nodes (10 %), but relatively high communication ranges (60 m in a $300 \times 300\,\mathrm{m}^2$ area). Under these conditions, it achieves low localization errors. However, SF-APIT is only applicable to static networks and no consideration is given to anchor nodes that behave in malicious ways without performing Sybil attacks. No anchor density measure as defined by Hu and Evans [8] is provided.

Existing security-aware approaches often focus on network structure-based attacks (e.g. wormhole or Sybil attacks), have high computational requirements, assume static networks or make strong assumptions (e.g. trustworthiness of anchor nodes) that may not be true in real-world scenarios. In many cases, where anchor or unknown nodes are assumed to behave maliciously, no specific attack model for the behavior of malicious nodes is given or such malicious nodes are only detected, but no further handling of the detected nodes is specified.

We propose RESA-MCL, which combines the advantages of range-free approaches, such as low cost and ease of deployment, with the enhanced robustness against attacks of secure localization schemes while still being very lightweight with

TABLE 1. Symbols with references to sections and listings.

Symbol	Value	First use	Meaning
V_{Min}	10 m/s	S.III-B	Minimum speed of nodes
V_{Max}	20 m/s	L.1	Maximum speed of nodes
t		L.1	Time t
L_t		L.1	Set of particles at time t
L_S, L_R		L.1	Strict/relaxed set of particles
N	50	L.1	Target number of particles
l^k_{t-1}		L.1	Particle k from L_{t-1}
p_k		L.1	Particle k
r	50 m	L.2	Radio range in meters
δ	5 m	S.III-B	Increased radio range for relaxed acceptance
r_D	2.5	L.5	Anchor node plausibility check range factor (direct)
r_I	4.5	L.5	Anchor node plausibility check range factor (indirect)
r_f		L.5	A radio range factor
Δpos		L.4	Sensor measured movement distance of node since last MCL call
c		L.1	MEETCONDITION result
A_D, A_I		L.2	Set of anchor nodes in range (direct, indirect)
a^i		L.5	Anchor node i
a^i_t		L.5	Anchor node i and position at time t
$a^i_{t_{\text{last}}}$		L.5	Anchor node i's last known position before time t
P_i	0	L.5	Anchor node i's distrust points
$d(a, b)$		L.2	Distance between two given particles or points a and b
s_ϕ		S.III-D2	Modulo value used for subsetting
s_λ		S.III-D2	Threshold value used for subsetting

respect to computation. Our scheme is fully decentralized, meaning that there are no infrastructure requirements beyond the nodes themselves and that no single point of failure exists. Malfunctioning or malicious anchor nodes can be detected and their effect on localization accuracy is mitigated. We define three different attack models for anchor nodes and evaluate our proposed scheme, as well as previous schemes, under these conditions. As the approach is based on MCL, RESA-MCL fully supports mobility for both unknown nodes and anchor nodes. There is no offline training phase or dependence on any location-specific data, meaning that RESA-MCL can be used in arbitrary environments, rather than being dependent on a fixed location. Our approach performs well with low numbers of anchor nodes (less than 5 % of unknown nodes) and low communication ranges relative to the experimental area (50 m in a 500 × 500 m^2 area).

Overall, RESA-MCL can be deployed with minimal hardware requirements. It leverages sensor data from low cost and low powered 9-axis Inertial Measurement Unit (IMU) sensors, similar to SA-MCL, to both enhance localization accuracy and allow the detection of malfunctioning or malicious anchor nodes. As RESA-MCL mainly consists of computationally inexpensive modifications to SA-MCL, the real-world power measurements presented by Hartung *et al.* [7] apply to RESA-MCL as well, demonstrating that its deployment on low powered IRIS sensor nodes [13] based on an 8 bit microcontroller and XBee (2.4 GHz 802.15.4) radio is feasible. In addition, since

```
1:  procedure MCL(attempts, enableSubset=false)
2:      L_t ← {}
3:      count ← 0
4:      while size(L_t) < N and count < attempts do
5:          count ← count+1
6:          L_S ← {}
7:          L_R ← {}
8:          for all k ∈ [1, 2, ..., N] do
9:              p_k ← RESAMPLEINRADIUS(l^k_{t-1}, V_Max)
10:             c ← MEETCONDITION(p_k, enableSubset)
11:             if c = strict then
12:                 L_S ← L_S ∪ {p_k}
13:             else if c = relaxed then
14:                 L_R ← L_R ∪ {p_k}
15:         N_left ← N − SIZE(L_t)
16:         if SIZE(L_S) ≥ N_left then
17:             L_t ← L_t ∪ CHOOSE(L_S, N_left)
18:         else
19:             L_t ← L_t ∪ CHOOSE(L_S ∪ L_R, N_left)
```

LISTING 1. MCL algorithm.

RESA-MCL does not make any assumptions about the area it is deployed in, it is suitable for both mobile and stationary networks as well as indoor and outdoor environments.

III. LOCALIZATION SCHEME

RESA-MCL uses the idea of the SA-MCL scheme introduced by Hartung *et al.* [7] as a basis and adds a number of improvements, which both increase its overall accuracy and make it more robust in networks with faulty or malicious nodes. SA-MCL itself is based on the MCL [8] approach. In the following, we shortly reiterate these approaches and finally detail the improvements made in RESA-MCL.

A. NOTATION

This section provides an overview of all symbols used in the sections detailing our scheme. Table 1 provides short explanations, references to where each symbol is first used and, where applicable, values of constants. The "First use" column specifies which section (S.) or listing (L.) the symbol is used in first.

B. MCL

The MCL algorithm [8] is made up of two phases. The first phase is network communications and the second phase is a particle filter used for location estimation.

The necessary information to update location information is gathered during a phase of network communications. Here, anchor nodes broadcast their locations. Unknown nodes that receive this information directly from an anchor node broadcast it again, marking it as a rebroadcast. Nodes that receive either type of broadcast store the received location data and node IDs. In regular intervals, the nodes use this collected

```
1: function MEETCONDITION(p_k, enableSubset=false)
2:     result ← strict
3:     for all a_t^i ∈ A_D do
4:         if not enableSubset ∨ ((k + t + i) mod s_φ < s_λ)
   then
5:             if d(p_k, a_t^i) ≥ r + δ then
6:                 return filtered
7:             else if d(p_k, a_t^i) ≥ r then
8:                 result ← relaxed
9:     for all a_t^i ∈ A_I do
10:        if not enableSubset ∨ ((k + t + i) mod s_φ < s_λ)
   then
11:            if d(p_k, a_t^i) < r − δ ∨ d(p_k, a_t^i) ≥ 2r + δ then
12:                return filtered
13:            else if d(p_k, a_t^i) < r ∨ d(p_k, a_t^i) ≥ 2r then
14:                result ← relaxed
15:    return result
```

LISTING 2. Particle filter condition function.

```
1: procedure SA-MCL(attempts)
2:     if A_D = ∅ ∧ A_I = ∅ then
3:         DEADRECKONING
4:     else
5:         MCL(attempts, enableSubset=false)
```

LISTING 3. SA-MCL algorithm.

```
1: procedure DEADRECKONING
2:     L_t ← {}
3:     Δx ← GETMOVEMENTXFROMSENSORS
4:     Δy ← GETMOVEMENTYFROMSENSORS
5:     Δpos ← (Δx, Δy)
6:     for all l_{t−1} ∈ L_{t−1} do
7:         l_{t−1}.x ← l_{t−1}.x + Δx
8:         l_{t−1}.y ← l_{t−1}.y + Δy
```

LISTING 4. Particle dead reckoning.

information to update their location estimates by using a particle filter.

The particle filter itself is split into two steps: prediction and filtering. Initially, each unknown node initializes N particles. These particles are points in 2D space and represent possible locations of the unknown node. They are distributed randomly over the area of possible locations. These particles are an approximation of the probability distribution of possible node locations. During the prediction step, each particle is reassigned to a new location within the radius corresponding to the maximum speed of the node. This mechanism ensures that, if no observations are made to constrain the distribution, its uncertainty grows over time in accordance with the node's movement. In the filter step, the resampled particle is checked against received broadcasts from anchor nodes and forwarded 2-hop retransmissions of such broadcasts. If the particle is not within the radio range of all anchor nodes, from which transmissions are received, or within the ring between one and two radio ranges in the case of 2-hop retransmissions, it is discarded and a new particle is sampled instead, restarting the process.

Since the original description of the algorithm leaves some room for interpretation, Listing 1 gives a more detailed description, following the actual code given in the simulator released by the authors, which contains additional detail, such as a relaxed acceptance criterium for particles during the filter step.

The ''attempts'' parameter determines how many iterations and thereby time should be spent on trying to find a set of particles fulfilling the range MCL criteria while resampling. When running the algorithm for the first time, the ''attempts'' parameter is given as 10000, while subsequent runs use an ''attempts'' value of 200. Additionally, during the more thorough initialization run, the while loop is run through first fully discarding relaxed samples and then once more, if necessary, keeping them. For the sake of simplicity, this additional behavior for the initialization is not described in Listing 1.

The original MCL algorithm uses a meetCondition function (shown in Listing 2) without the additional particle subsetting (see Section III-D2) introduced for RESA-MCL. To avoid having two slightly different copies of the pseudocode, we introduce the ''enableSubset'' parameter to MCL and meetCondition. For the original MCL approach the ''enableSubset'' parameter is always ''false'' in both MCL and meetCondition. The meetCondition function determines whether a given sample passes the filter criterium fully, only in a relaxed manner or not at all. If enough samples are found to fully fulfill the strict range criterium, the relaxed condition is not used. Otherwise, the range is extended by δ meters and particles fulfilling this relaxed condition are accepted as well.

For a given point p and radius r, the function resampleInRadius(p, r) returns a random, new point around that point p within a radius of r, but still within the bounds of the experimental area. Given a set S and $N \in \mathbb{N}$, the choose(S, N) function randomly selects at most N elements from a set S, but never returns more than $|S|$ elements.

C. SA-MCL

The main addition to MCL contributed by SA-MCL is that dead reckoning is used to update node locations when no anchor nodes are within communication range. This allows it to bridge the time between encountering anchor nodes with enhanced localization accuracy compared to the original approach. However, since this process is only run when no anchor nodes are within range, in denser networks its performance is not improved over that of MCL.

As shown in Listing 3, in SA-MCL, particle positions are updated according to the direction of the node as measured by the IMU sensor installed on it. The deadReckoning procedure described in Listing 4 performs these updates to the particle positions. This procedure is run instead of the regular particle resampling and filtering if both A_D and A_I are empty sets.

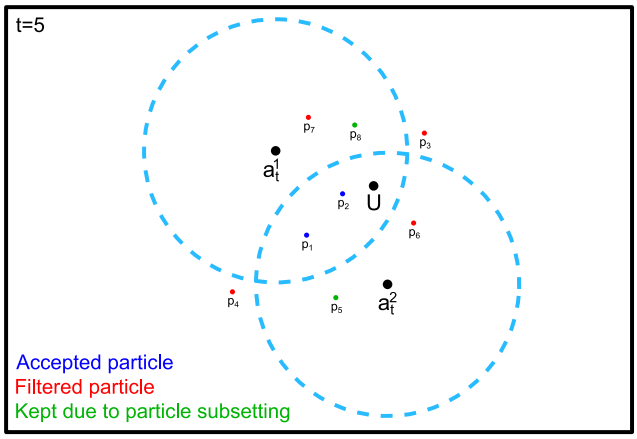

FIGURE 1. Example illustrating the particle subsetting process (S.III-D2) with two anchors, one unknown node U and $k \in \{1, 2, \ldots, 8\}$, $t = 5$, $j \in \{1, 2\}$.

In a real-world test bed implementation, SA-MCL uses low-cost MPU-9150 9-axis IMU sensors with low power consumption [7]. The authors show that the use of this type of sensor is feasible in WSN localization approaches, both from a cost (monetary and power) as well as from an accuracy perspective.

D. RESA-MCL

RESA-MCL introduces three modifications to the SA-MCL algorithm, making it more accurate and improving its resilience to adversarial network conditions. Each of the improvements by itself is both effective and can be implemented efficiently even on low powered hardware such as 8 bit microcontroller-based IRIS sensor nodes.

1) MOTION-BASED PARTICLE UPDATES

As detailed in Section III-C, SA-MCL leverages dead reckoning to update particle positions when no anchor nodes are within communication range. RESA-MCL goes one step further and also applies these motion-based particle updates even when anchor nodes are within range. In RESA-MCL, the deadReckoning procedure is executed before the loop of prediction and filtering operations, leading to two differences. Firstly, RESA-MCL always updates its position estimate according to the sensed motion data. Secondly, if no anchor nodes are within range, both 1-hop or 2-hop, particle resampling is still performed after the motion-based position updates.

2) PARTICLE SUBSETTING

To prevent a single malicious node from completely throwing off the position estimate of a node by providing misleading positional information, in RESA-MCL anchor node information is only applied to a subset of particles.

The particle subsetting is implemented in the form of an additional condition that allows skipping the MCL range criterium check with respect to a certain anchor node. In the original MCL, a particle that is not within the range of an anchor node that was heard by the unknown node is filtered. With RESA-MCL's particle subsetting, the particle may instead be kept, even if it is not within the radio range of that specific anchor node. More specifically, a particle p_k is subject to filtering at time t with respect to anchor a_t^j, only if $k + t + j \mod s_\phi < s_\lambda$. Here $k + t + j \mod s_\phi$ functions in a hash-like manner to pseudo-randomly select different particles at each time step and for each anchor node. If this condition is not true, the given particle p_k is not be affected by the anchor a_t^j's positional information at the given time step.

Figure 1 shows an example at time $t = 5$ with eight particles $p_k, k \in \{1, 2, \ldots, 8\}$, two anchor nodes $a_t^j, j \in \{1, 2\}$ and an unknown node U. The particles represent the position estimate of the unknown node. For each particle, the MCL range condition is checked with respect to both anchor nodes. Particles are color-coded according to whether they are kept (blue), filtered (red) due to not fulfilling the MCL range criterium. Green means that particle subsetting prevented them from being filtered despite the fact that they would fail an MCL range condition check. Particles p_1 and p_2 fall into the intersection of a_t^1's and a_t^2's radio ranges and are therefore kept. The filtered particles p_3, p_4, p_6, p_7 are removed, because they fail the range MCL criterium; p_3, p_4 are outside of both radio ranges, p_6 is not inside the radio range of a_t^1 and p_7 is not inside the radio range of a_t^2.

Finally, p_5 and p_8 are kept only because the particle subsetting. Specifically, p_5 fulfills the range condition for a_t^2, but would be filtered due to failing it with respect to a_t^1. This is because $5 + 5 + 1 \; mod \; 4 = 3$, which is not less than $s_\lambda = 3$. Since the result of the modulo operation is not less than the threshold value s_λ, the range criterium is not checked for p_5 with respect to a_t^1 and the particle is not filtered out. In the case of p_8, it fulfills the range criterium for a_t^1, but it would be filtered due to failing it with respect to a_t^2. However, for this particle holds that $8+5+2 \; mod \; 4 = 3$, which is again not less than the threshold value s_λ. Therefore the particle p_8 is kept because the range criterium is ignored.

The particle subsetting functionality is added in RESA-MCL's meetCondition function, which is shown in Listing 2. Its general operation matches that of MCL's meetCondition, other than the addition of the particle subsetting mechanism through the additional conditions on lines 4 and 10. The additional "enableSubset" parameter allows disabling this new functionality and makes the given function equivalent to the original MCL meetCondition function.

Particle subsetting can prevent potentional malicious nodes in the network from strongly affecting location estimates, but it also leads to a lower rate of information use in the case that all provided information is legitimate. However, the use of dead reckoning already improves the accuracy of the particle filter-based location estimation significantly, making up for the lower rate of positional correction through anchor node information.

1: **function** CHECKANCHOR(a^i, r_f)
2: **if** no $a^i_{t_{\text{last}}}$: **return** true
3: **return** $d((a^i_t - a^i_{t_{\text{last}}}), \Delta\text{pos}) < r \cdot r_f$

LISTING 5. Anchor position plausibility check function.

1: **procedure** UPDATEPOINTS($A_{\text{bad}}, A_{\text{good}}$)
2: **for all** $a^i \in A_{\text{bad}}$ **do**
3: $P_i \leftarrow \begin{cases} 20 \text{ if } P_i < 20 \\ P_i + 5 \end{cases}$
4: **for all** $a^i \in A_{\text{good}}$ **do**
5: $P_i \leftarrow \begin{cases} 0 \text{ if } P_i < 1 \\ P_i - 1 \end{cases}$

LISTING 6. Anchor distrust point update function.

3) ANCHOR POSITION PLAUSIBILITY CHECK

RESA-MCL nodes also apply a plausibility check to received positional information, using the checkAnchor function (Listing 5). When receiving position information from an anchor node, from which previously position information has already been received, a movement vector is calculated from the difference in positions ($a^i_t - a^i_{t_{\text{last}}}$) and compared with the movement of the node (Δpos) trying to localize itself according to its IMU sensed movement data. The difference between both movement vectors is then calculated and compared to the radio range multiplied by a factor r_f, which relaxes the condition to allow for inaccuracies in radio range and motion detection. The idea is that, if the anchor node and unknown node can hear each other before and after moving, and have a radio range of r, then r_f is chosen, such that they cannot have moved by a distance greater than $r \cdot r_f$. Specifically, the factor r_f is chosen as $r_f = 2 \cdot hops + 0.5$, where the hop count *hops* determines the maximum distance between nodes and both the factor of 2 and the additive term of 0.5 are a safety margins to reduce false positives with respect to the plausibility check. Therefore, in the 1-hop case, $r_f = 2.5$, which corresponds to the diameter of the circles representing the radio ranges around nodes, plus the safety margin of 0.5. As the total distance between nodes may be doubled in the 2-hop case, r_f is chosen as 4.5 following the same formula and reasoning. In the 2-hop case, the maximum distance between anchor and unknown node can be twice that of the 1-hop case. Due to that, r_f is also chosen to be higher to make up for the increased variance in movement and range introduced by the retransmission of anchor information.

If implausible information is detected, the anchor node is removed from the sets of anchor nodes used in meetCondition and added to a list of untrustworthy anchor nodes. The updatePoints (Listing 6) procedure assigns distrust points to such anchor nodes with implausible positions and reduces distrust points for anchor nodes with plausible positions. This allows trustworthy anchors, which were mistakenly classified as non-trustworthy, to recover over time and allows its location information to be used for localization, once its distrust points fall back to 0. At the same time, nodes that consistently misbehave are not be used for localization.

1: **procedure** RESA-MCL(attempts)
2: DEADRECKONING
3: $A'_D \leftarrow \{a^i | a^i \in A_D \text{ if CHECKANCHOR}((a^i, r_D))\}$
4: $A'_I \leftarrow \{a^i | a^i \in A_I \text{ if CHECKANCHOR}((a^i, r_I))\}$
5: UPDATEPOINTS($((A_D \setminus A'_D) \cup (A_I \setminus A'_I), A'_D \cup A'_I)$)
6: $A_D \leftarrow \{a_i | a_i \in A'_D \wedge P_i = 0\}$
7: $A_I \leftarrow \{a_i | a_i \in A'_I \wedge P_i = 0\}$
8: MCL(attempts, enableSubset=true)

LISTING 7. RESA-MCL algorithm.

This functionality synergizes well with particle subsetting. If an anchor node broadcasts positional information which would greatly impact localization accuracy, this is likely to be detected the second time location information is received. At the same time, particle subsetting limits the impact of the inaccurate information received in the first time step.

4) PLAUSIBILITY CHECKING OF 2-HOP FORWARDING

Applying the same plausibility check described for anchor nodes in the 2-hop case to unknown nodes is also considered, but not used in RESA-MCL. The idea of this approach is to disable all 2-hop data reception when implausible forwarded data is received following a similar points system as described for anchor plausibility checking. However, during the ablation experiments in Section IV-E, we find limited benefits of this mitigation strategy in our considered attack models and therefore do not employ it as a part of RESA-MCL. We plan to reevaluate this method in the future under different attack models.

5) PUTTING EVERYTHING TOGETHER

The full algorithm can be described as given in Listing 7. First particle positions are updated according to dead reckoning, then heard anchor lists are filtered according to the positional plausibility check and their distrust points are updated. Finally, the basic MCL algorithm is run with particle subsetting enabled and anchor sets A_D, A_I containing only anchor nodes with zero distrust points.

The estimated position is the average position of the particles in L_t after running the MCL function.

IV. EVALUATION

In this section, we introduce our methodology for evaluating RESA-MCL. First, we introduce our experimental setup and simulation parameters. Following that, we show and discuss our experimental results with respect to baseline performance, motivate parameter choices and demonstrate RESA-MCL's robust behavior under three different types of attacks.

A. EXPERIMENTAL SETUP

All experiments are done using an extended and improved version of the Java-based simulator originally developed and

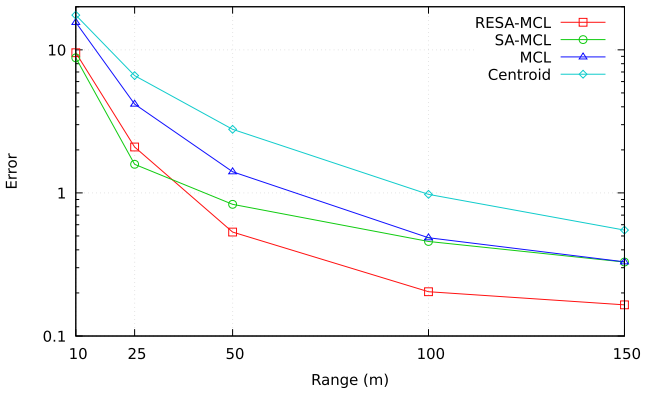

FIGURE 2. Comparison at different communication ranges without attack.

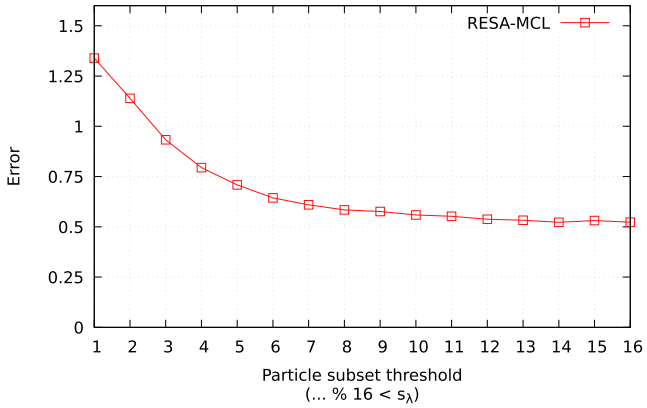

FIGURE 3. Evaluation of particle subset sizes without attacks and $s_\phi = 16$.

used for the evaluation MCL and also used in the evaluation of SA-MCL. The source code of our version of the simulator as well as simulation results are publicly available on Code Ocean.[1] This simulator is used to ensure the comparability of results to other MCL-based algorithms (e.g. SA-MCL), which commonly also use the same code base.

All experiments are run with 300 nodes. Initial positions are chosen randomly at the start of each run. Of these, unless otherwise specified, 10 are used as anchor nodes. The experimental area is $500 \times 500\,\mathrm{m}^2$ with a radio communication range of of 50 m. Simulations are run for 1000 steps, with each step representing a time interval of 1 s and one iteration of the localization algorithm. Each such run is repeated 10 times with a different random seed. Nodes follow a modified random waypoint movement model, where a minimum movement speed of V_{Min} and a maximum path segment duration constraint are applied to prevent movement from degrading to low average speeds as described by Yoon *et al.* [23]. Specifically, movement speeds are randomly selected from the range of $10\,\mathrm{m\,s}^{-1}$ to $20\,\mathrm{m\,s}^{-1}$ and a limit of at maximum five time steps per path segment is imposed. An error of 20 % is applied to both the sensed speed and direction to include the effect of noisy sensors in the evaluation of our scheme and enable a fair comparison to SA-MCL. RSS or link quality between nodes is not considered beyond basic connectivity, which is all that is required in range-free localization schemes.

Hu and Evans [8] define the anchor density as the average number of anchor nodes within a 1-hop distance to unknown nodes. Due to node mobility and random initialization, it is not possible to give a fixed anchor density with this definition. With the given experimental parameters, we measure a mean anchor density of 0.327 with a standard deviation of 0.054 over ten runs of 1000 steps each. Localization error in all figures is given as the error in meters divided by the radio range, which is the common measure for range-free approaches.

[1] https://codeocean.com/capsule/5799c9e7-22c4-450a-8783-a3e3eb2b5829/tree/v1

Due to the results shown in Section IV-C, the particle subsetting parameters are chosen as $s_\lambda = 3, s_\phi = 4$ (equivalent to $s_\lambda = 12, s_\phi = 16$) for all experiments other than those specifically varying those parameters. This mostly avoids impacting the localization performance in attack-free scenarios while still mitigating the impact of malicious nodes.

The behavior of four range-free localization algorithms (Centroid, MCL, SA-MCL, RESA-MCL) under three different attack models is evaluated. In the following, we analyze the results of our experiments.

B. BASELINE

Figure 2 shows a comparison of RESA-MCL with previous approaches in a scenario with no attacks over different radio ranges. Since only 10 anchor nodes are used for an area of $500 \times 500\,\mathrm{m}^2$, in scenarios with very low communication ranges, nodes rarely encounter an anchor node. In this case, RESA-MCL and SA-MCL behave quite similarly, as both use sensor data for dead-reckoning. Since SA-MCL has no built-in detection of malfunctioning or malicious nodes, it can make full use of the very limited information it receives, letting it perform slightly better than RESA-MCL in scenarios with extremely low communication ranges and thus anchor density.

As communication range increases, the performance of SA-MCL approaches that of MCL, because nodes almost always have at least one anchor node within range, making it disregard its motion sensor data. RESA-MCL meanwhile keeps employing dead-reckoning to improve its localization accuracy even further in comparison to other approaches.

C. OPTIMAL PARTICLE SUBSETTING

In this section we determine the optimal parameters for the particle subsetting process given in Section III-D2. Figure 3 shows the performance of RESA-MCL in a scenario without attacks, different values for s_λ (threshold) with $s_\phi = 16$ (modulus). The case of $s_\lambda = s_\phi = 16$ is equivalent to no particle subsetting, while $s_\lambda = 1$ means that only 1 in 16 particles is affected by an anchor node. It can be seen that for $s_\lambda < 8$, the

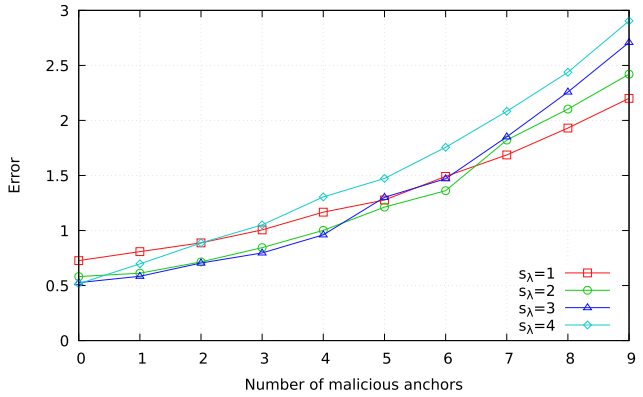

FIGURE 4. Evaluation of particle subset sizes under fixed position attack with $s_\phi = 4$.

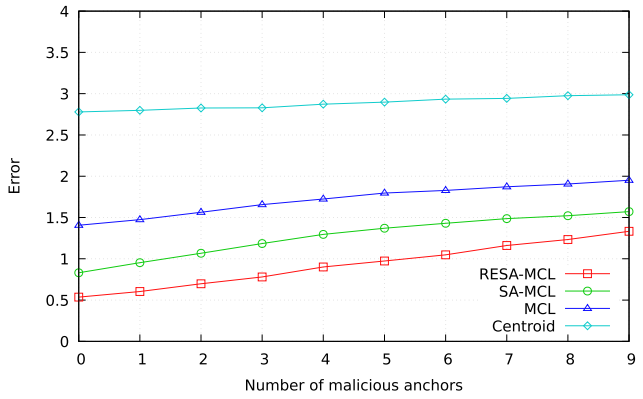

FIGURE 5. Biased position attack.

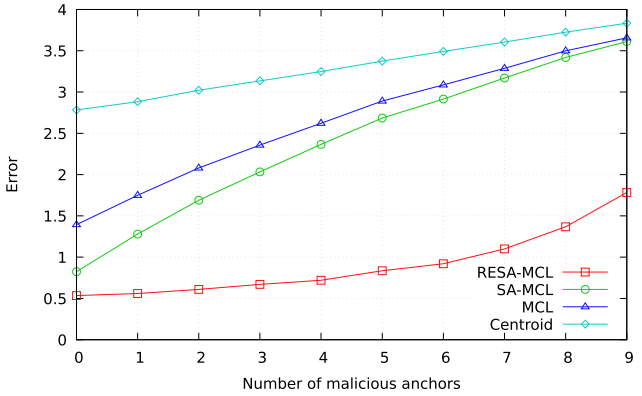

FIGURE 6. Random position attack.

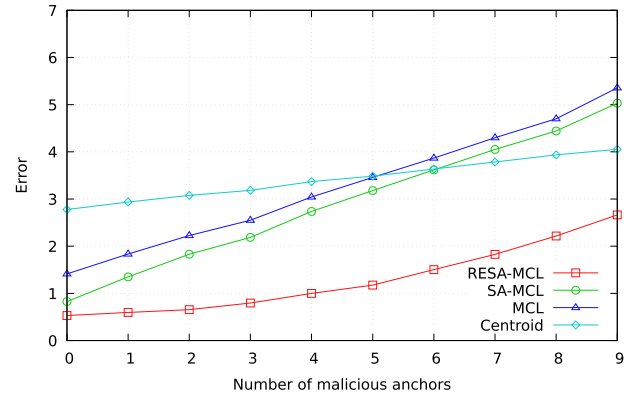

FIGURE 7. Fixed position attack.

localization error sharply increases with declining s_λ, while above $s_\lambda = 12$ the localization performance is barely affected by particle subsetting.

Figure 4 shows that more restrictive particle subsetting makes RESA-MCL more resilient against attacks. The fixed position attack, which is shown to be the most effective attack in Section IV-D3, is chosen for this evaluation. Due to the results from Figure 3 and to increase the legibility of the figure, we divide s_λ and s_ϕ by 4 and show results for $s_\lambda \in \{1, 2, 3, 4\}$, representing the most relevant tradeoff points. As per Section IV-B, the least subsetting performs the best in the scenario with no attacks, but $s_\lambda = 3$ performs almost equally as well. In scenarios with 10 % to 40 % malicious anchor nodes, $s_\lambda = 3$ performs the best with more aggressive subsetting being more effective in scenarios where half or more anchor nodes are malicious. As a network where the majority of nodes is malicious seems less likely, we decide on $s_\lambda = 3$ as a reasonable trade-off between robustness and localization accuracy in networks without attacks. Therefore, $s_\lambda = 3$ is used for all further experiments.

D. PERFORMANCE UNDER ATTACKS

In the following, we present results showing the performance of RESA-MCL under three different types of attack models.

1) BIASED POSITION ATTACK

In the biased position attack, malicious anchor nodes send their true locations, with an offset of (50, 50) (in meters) added to it. This type of attack is harder to detect for RESA-MCL, because the movement vectors of anchor nodes match their true movement, thereby usually satisfying the plausibility check. At the same time, since only an offset is added on top of the real position, the effect of the attack is also lower than that of other attacks, as the malicious data still contains real information.

Figure 5 shows the performance of MCL, SA-MCL and RESA-MCL, as well as Centroid, which serves as a baseline. It can be seen that all approaches are affected by the attack, but as expected, the effect of the attack is relatively low. RESA-MCL performs the best in all cases, due to its overall superior localization accuracy capabilities granted through its use of dead reckoning.

2) RANDOM POSITION ATTACK

Malicious anchor nodes send out completely random, freshly chosen positions in the random position attack. The effect of this attack is stronger than that of the biased position attack as no real data is part of the malicious broadcasts.

The results in Figure 6 show that in this attack, the same relative ordering between approaches is maintained as with the

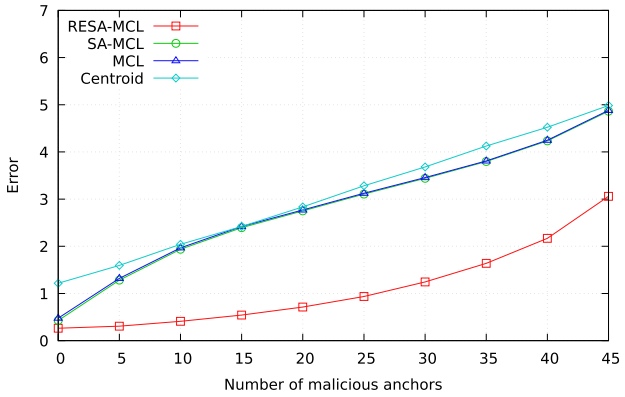

FIGURE 8. Fixed position attack with 50 anchor nodes.

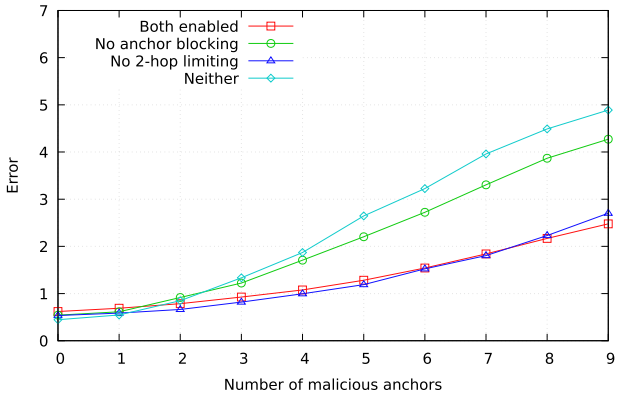

FIGURE 9. Ablation experiments.

biased position attack. However, RESA-MCL performs much better, with only a slight increase in its location estimation error up until about 50 % malicious anchor nodes. The other approaches are affected much more strongly. This shows that anchor position plausibility checks and particle subsetting are effective at mitigating this attack.

3) FIXED POSITION ATTACK

The fixed position attack is found to have the strongest effect of all three attacks. Malicious anchor nodes broadcast their position as a fixed point of (70, 70) (in meters) on the map. This pulls location estimates towards one corner of the map. Its effect is stronger than that of the random position attack because in the random position attack positions closer to the actual position of a node may be sent at random.

One obvious difference to the previous attacks, which is clearly visible in Figure 7, is that after 50 % of anchor nodes are malicious, MCL and SA-MCL start being affected more strongly by the attack than Centroid. This is likely the case due to the low anchor density (0.327) in this scenario. When Centroid localization is not within range of any anchor nodes, which according to the anchor density measurement is the case for 67 % of location estimates, it will assume a location in the center of the map, which is an averagely bad estimate for all points of the map. The other approaches are stateful and their current estimate is influenced by past information received from anchor nodes. With a majority of anchor nodes being malicious, these estimates will be negatively affected at nearly all times, rather than just in 32.7 % of the time as is the case with Centroid.

RESA-MCL's robustness features can once again be seen to be effective in this scenario, with its error rate remaining way below Centroid even at 90 % of malicious anchor nodes. Up until 30 % of anchor nodes being malicious, the increase in localization error is very small.

Figure 8 shows the same type of results for a scenario with 300 nodes in total, of which 50 nodes are chosen as anchor nodes, leading to a much higher anchor node density. As most unknown nodes are in the range of at least one anchor node almost all the time, SA-MCL and MCL fall onto one line here, because SA-MCL will not perform its dead reckoning while within range of an anchor node. RESA-MCL outperforms all other approaches despite being optimized for low anchor node densities.

E. ABLATION EXPERIMENTS

Taking inspiration from the field of machine learning, we perform ablation experiments to evaluate the performance of RESA-MCL with certain components disabled. This experiment includes the 2-hop plausibility check described in Section III-D4 as one component, and the anchor position plausibility check from Section III-D3 as the second component. Particle subsetting is not included as results from it being disabled can also be seen in Section IV-C.

Figure 9 shows the results of disabling either or both of the components under a fixed position attack in comparison with full RESA-MCL. It can be seen that 2-hop plausibility checks only provide a benefit with 70 % or more malicious anchor nodes and otherwise worsen performance. For this reason, we do not include this component in RESA-MCL, although it may be useful under specific circumstances. Blocking malicious anchor nodes however, is shown to be highly beneficial under attack.

F. COMPARISON UNDER ANCHOR DENSITY

In the earlier parts of this section, we have thoroughly compared the localization accuracy of Centroid, MCL, SA-MCL and RESA-MCL. In the following, we provide a comparison of RESA-MCL and MCL-DE [15], another recent MCL-based approach.

For RESA-MCL, we measure an anchor density of 0.327 with a standard deviation of 0.054 over ten runs of

TABLE 2. Error comparison between RESA-MCL and MCL-DE without attacks.

Approach	Anchor density	Error
RESA-MCL	0.33	0.54
	1.63	0.26
MCL-DE	0.5	1.0
	1.5	0.5

1000 steps each in scenarios with 300 nodes, of which 10 nodes are anchor nodes. In the scenario is presented in Figure 8, with 300 nodes in total, of which 50 nodes are anchor nodes, we measure an average anchor density of 1.63 with a standard deviation of 0.09. As can be seen in Table 2, MCL-DE achieves a localization error of approximately 1.0 at an anchor density of 0.5 and an error of approximately 0.5 at an anchor density of 1.5.

Our measured anchor densities in RESA-MCL do not perfectly match up with those of MCL-DE. However, with an error of 0.54 at anchor density 0.33, RESA-MCL achieves an error comparable to that of MCL-DE at anchor density 1.5 and 45 % lower than the error of 1.0 achieved by MCL-DE at the closer anchor density value of 0.5. Similarly, with an error of 0.26 at an anchor density of 1.63, RESA-MCL's error is 48 % lower than the error of 0.5 at MCL-DE's comparable anchor density of 1.5. In both cases, RESA-MCL achieves significantly lower localization errors (up to 48 %) at comparable anchor densities or, conversely, requires significantly less anchor nodes to achieve a given localization accuracy.

V. CONCLUSION AND FUTURE WORKS

In this work, we introduce RESA-MCL, a novel MCL-based, range-free, security-aware localization algorithm for WSNs and the IoT that strongly outperforms comparable approaches both in safe situations and under attack by malicious anchor nodes. Without attacks, it outperforms a recent comparable approach with 48 % lower localization error at similar anchor densities. RESA-MCL employs three techniques to both enhance general localization accuracy and robustness against malicious anchor nodes. Localization accuracy is enhanced both in very sparse networks with low numbers of anchor nodes and in very dense networks with high numbers of anchor nodes.

Additional contributions include the introduction of three different attack models against localization approaches in WSNs and the IoT, based on the assumption of malicious or malfunctioning anchor nodes. We evaluate the previous approaches Centroid, MCL and SA-MCL under these attack models and find them to be strongly affected by the attacks. In contrast, we demonstrate RESA-MCL's resilience against attacks under all three attack models and show that its localization error only increases slightly even with 30 % malicious anchor nodes under the most effective "fixed position" attack model.

Furthermore, we introduce the idea of 2-hop plausibility checking, which may increase resilience against malicious unknown nodes and provide a highly detailed reformalization of the original MCL approach through pseudo-code. This reformalization includes implementation details previously found only in the original author's simulation. To facilitate easier future investigations of MCL-based approaches, we also publish an optimized version of the simulator with performance optimizations and bug fixes.

To ease future works comparing these approaches, we optimize and improve the original simulation software provided by Hu and Evans [8] and Hartung *et al.* [7] and make it available on CodeOcean (see Section IV).

In the future, we plan to investigate further attack models, such as mixed attack strategies, malicious unknown nodes, as well as the effectiveness of refined versions of the 2-hop plausibility check functionality. We also plan to investigate ways of increasing the effectiveness of implausible location data checking while reducing false positives. Furthermore, we also plan to evaluate our approach in a real-world testbed, to validate simulation results. Finally, making the approach topology-aware, allowing it to exclude impassable terrain from particle locations is another avenue of enhancing localization accuracy that we plan to explore.

REFERENCES

[1] B. A. Akram, A. H. Akbar, and O. Shafiq, "HybLoc: Hybrid indoor Wi-Fi localization using soft clustering-based random decision forest ensembles," *IEEE Access*, vol. 6, pp. 38251–38272, 2018.

[2] N. Castell, F. R. Dauge, P. Schneider, M. Vogt, U. Lerner, B. Fishbain, D. Broday, and A. Bartonova, "Can commercial low-cost sensor platforms contribute to air quality monitoring and exposure estimates?" *Environ. Int.*, vol. 99, pp. 293–302, Feb. 2017.

[3] H. Chen, Y. Zhang, W. Li, X. Tao, and P. Zhang, "ConFi: Convolutional neural networks based indoor Wi-Fi localization using channel state information," *IEEE Access*, vol. 5, pp. 18066–18074, 2017.

[4] W. Ding, S. Chang, and J. Li, "A novel weighted localization method in wireless sensor networks based on hybrid RSS/AoA measurements," *IEEE Access*, vol. 9, pp. 150677–150685, 2021.

[5] S. K. Gharghan, R. Nordin, A. M. Jawad, H. M. Jawad, and M. Ismail, "Adaptive neural fuzzy inference system for accurate localization of wireless sensor network in outdoor and indoor cycling applications," *IEEE Access*, vol. 6, pp. 38475–38489, 2018.

[6] L. Gui, F. Xiao, Y. Zhou, F. Shu, and T. Val, "Connectivity based DV-Hop localization for Internet of Things," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8949–8958, Aug. 2020.

[7] S. Hartung, A. Bochem, A. Zdziarstek, and D. Hogrefe, "Applied sensor-assisted Monte Carlo localization for mobile wireless sensor networks," in *Proc. Int. Conf. Embedded Wireless Syst. Netw. (EWSN)*, Graz, Austria, Feb. 2016, pp. 181–192.

[8] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proc. 10th Annu. Int. Conf. Mobile Comput. Netw.*, Phildadelphia, PA, USA, 2004, pp. 45–57.

[9] A. Kanev, A. Nasteka, C. Bessonova, D. Nevmerzhitsky, A. Silaev, A. Efremov, and K. Nikiforova, "Anomaly detection in wireless sensor network of the 'smart home' system," in *Proc. 20th Conf. Open Innov. Assoc. (FRUCT)*, Apr. 2017, pp. 118–124.

[10] M. Liu, K. Yang, N. Zhao, Y. Chen, H. Song, and F. Gong, "Intelligent signal classification in industrial distributed wireless sensor networks based industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 7, pp. 4946–4956, Jul. 2021.

[11] X. Liu, S. Su, F. Han, Y. Liu, and Z. Pan, "A range-based secure localization algorithm for wireless sensor networks," *IEEE Sensors J.*, vol. 19, no. 2, pp. 785–796, Jan. 2019.

[12] Q. Luo, Y. Peng, J. Li, and X. Peng, "RSSI-based localization through uncertain data mapping for wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 9, pp. 3155–3162, May 2016.

[13] MEMSIC. (2010). *IRIS OEM Datasheet*. Accessed: Jan. 22, 2022. [Online]. Available: http://static6.arrow.com/aropdfconversion/8f126c9f83e22fabed80e2c01b513ad4960db1ea/6020-0123-02_a_iris_oem_edition-t.pdf

[14] Z. Munadhil, S. K. Gharghan, A. H. Mutlag, A. Al-Naji, and J. Chahl, "Neural network-based Alzheimer's patient localization for wireless sensor network in an indoor environment," *IEEE Access*, vol. 8, pp. 150527–150538, 2020.

[15] M. Qin and R. Zhu, "A Monte Carlo localization method based on differential evolution optimization applied into economic forecasting in mobile wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2018, no. 1, pp. 1–9, 2018.

[16] P. Sommer, J. Liu, K. Zhao, B. Kusy, R. Jurdak, A. McKeown, and D. Westcott, "Information bang for the energy buck: Towards energy- and mobility-aware tracking," in *Proc. Int. Conf. Embedded Wireless Syst. Netw.* Junction, KS, USA: Junction Publishing, 2016, pp. 193–204.
[17] L. Tõnisson, J. Voigtländer, M. Weger, D. Assmann, R. Käthner, B. Heinold, and A. Macke, "Knowledge transfer with citizen science: Luft-Leipzig case study," *Sustainability*, vol. 13, no. 14, p. 7855, Jan. 2021.
[18] T. Wang, H. Ding, H. Xiong, and L. Zheng, "A compensated multi-anchors TOF-based localization algorithm for asynchronous wireless sensor networks," *IEEE Access*, vol. 7, pp. 64162–64176, 2019.
[19] T. Wang, H. Xiong, H. Ding, and L. Zheng, "TDOA-based joint synchronization and localization algorithm for asynchronous wireless sensor networks," *IEEE Trans. Commun.*, vol. 68, no. 5, pp. 3107–3124, 2020.
[20] Y. I. Wu, H. Wang, and X. Zheng, "WSN localization using RSS in three-dimensional space—A geometric method with closed-form solution," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4397–4404, Mar. 2016.
[21] N. Xie, Y. Chen, Z. Li, and D. O. Wu, "Lightweight secure localization approach in wireless sensor networks," *IEEE Trans. Commun.*, vol. 69, no. 10, pp. 6879–6893, Jul. 2021.
[22] J. Yin, Q. Wan, S. Yang, and K. C. Ho, "A simple and accurate TDOA-AOA localization method using two stations," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 144–148, Jan. 2016.
[23] J. Yoon, M. Liu, and B. Noble, "Random waypoint considered harmful," in *Proc. 22nd Annu. Joint Conf. IEEE Comput. Commun. Societies*, vol. 2, Mar./Apr. 2003, pp. 1312–1321.
[24] Y. Yuan, L. Huo, Z. Wang, and D. Hogrefe, "Secure APIT localization scheme against sybil attacks in distributed wireless sensor networks," *IEEE Access*, vol. 6, pp. 27629–27636, 2018.
[25] Y. Zhang, L. Cui, and S. Chai, "Energy-efficient localization for mobile sensor networks based on RSS and historical information," in *Proc. 27th Chin. Control Decis. Conf. (CCDC)*, May 2015, pp. 5246–5251.

ARNE BOCHEM was born in Bad Mergentheim, Germany. He received the B.Sc. and M.Sc. degrees in applied computer science from the University of Goettingen, where he is currently pursuing the Ph.D. degree.

His research interests include secure localization for wireless sensor networks, the Internet of Things, and blockchain technology.

HANG ZHANG received the B.Sc. degree in information and computer science from the Changsha University of Science and Technology, in 2005, and the M.Sc. and Ph.D. degrees in computer science from the University of Goettingen, in 2012 and 2018, respectively.

Her research interests include designing secure communication protocols in wireless mobile *ad-hoc* networks, wireless sensor networks, and *ad-hoc* vehicular networks using bio-inspired optimization algorithms and machine learning/deep learning algorithms. Her current research interest includes design of security-aware protocols for vehicle-to-everything (V2X) communications.