SpecKriging: GNN-based Secure Cooperative Spectrum Sensing

Yan Zhang, Ang Li, Jiawei Li, Dianqi Han, *Student Member, IEEE,* Tao Li, *Member, IEEE,* Rui Zhang, *Member, IEEE,* and Yanchao Zhang, *Fellow, IEEE*

***Abstract*—Cooperative spectrum sensing (CSS) adopted by spectrum-sensing providers (SSPs) plays a key role for dynamic spectrum access and is essential for avoiding interference with licensed primary users (PUs). A typical SSP system consists of geographically distributed spectrum sensors which can be compromised to submit fake spectrum-sensing reports. In this paper, we propose SpecKriging, a new spatial-interpolation technique based on Inductive Graph Neural Network Kriging (IGNNK) for secure CSS. In SpecKriging, we first pretrain a graphical neural network (GNN) model with the historical sensing records of a few trusted anchor sensors. During system runtime, we use the trained model to evaluate the trustworthiness of non-anchor sensors' data and also use them along with anchor sensors' new data to retrain the model. SpecKriging outputs trustworthy sensor reports for spectrum-occupancy detection. To the best of our knowledge, SpecKriging is the first work that explores GNNs for trustworthy CSS and also incorporates the hardware heterogeneity of spectrum sensors. Extensive experiments confirm the high efficacy and efficiency of SpecKriging for trustworthy spectrum-occupancy detection even when malicious spectrum sensors constitute the majority.**

***Index Terms*—Wireless security, GNN, cooperative spectrum sensing.**

I. Introduction

With ever-growing wireless/mobile devices and the resulting explosive data traffic, RF spectrum has emerged as an imperative resource worldwide with limited supply. Cooperative spectrum sensing (CSS) adopted by spectrum-sensing providers (SSPs) plays a key role for dynamic spectrum access and is essential for avoiding interference with licensed primary users (PUs). A typical SSP system consists of geographically distributed spectrum sensors. Most spectrum sensors are deployed by the SSP itself, and some can be recruited mobile users accepting crowdsourced spectrum-sensing tasks [1–3]. The SSP distributes spectrum-sensing tasks to spectrum sensors and then explores the collected spectrum data to provide an integrated spectrum-analytics platform to various users. There has been a lot of work on CSS-based applications such as spectrum patrolling [4], spectrum occupancy query [5], and transmitter localization [6]. Although promising, CSS is vulnerable to false spectrum reports from compromised or dysfunctional spectrum sensors. An extreme example is when the majority of spectrum sensors are compromised to submit fake spectrum measurements and/or location information, the spectrum-analysis results can be totally wrong.

The prior work to defend against compromised spectrum sensors can be classified into several categories. The first category [7, 8] uses anomaly detection and fails if malicious sensors are the majority. The second category maintains reputations for spectrum sensors based on their historical logs or statistical results [9–11]. These methods could not handle the sudden change in sensor behavior and thus are vulnerable to the instantaneous attack. The third category relies on a few trusted sensors to exclude the malicious nodes whose data significantly deviate from trusted sensors' [1, 12, 13]. Although attractive, they require either real signal-propagation data from PUs that are often difficult to obtain [12] or signal-propagation models that are unpredictable in real scenarios [1]. The latest category utilizes various spatial-interpolation techniques such as Inverse Distance Weighting (IDW) [14] and Ordinary Kriging (OK) [5, 13]. With growing demand for fine-granular and large-scale spectrum data, these spatial-interpolation techniques show great potential for RF signal processing [5, 13]. The underlying intuition is that the spectrum measurement in a location can be inferred from those at neighboring locations with spatial interpolation. By comparing the predicted and measured spectrum data of individual spectrum sensors, the SSP could evaluate their trustworthiness.

Although promising, spatial-interpolation techniques suffer from a few drawbacks. First, it can be difficult to implement these techniques when the number n of spectrum sensors is large. For example, the state-of-the-art techniques [5, 13] based on OK [15] uses large-matrix multiplication with computational complexity of $\mathcal{O}(n^4)$. Second, these methods cannot handle the multi-dimensional RF datasets (e.g., spectrum signals from multiple frequency bins). Last and most importantly, these methods cannot apply when spectrum sensors are heterogeneous. In particular, when spectrum sensors have heterogeneous device features such as a different number of antennas, various sampling rates, or measurement mechanisms, their RF signal measurements might relate to not only spatial information but also these device features. In this case, spatial-interpolation methods become less valid.

This paper explores graph neural networks (GNNs) for secure CSS for the first time in literature. We are motivated to adopt GNNs because they are powerful in mapping spatiotemporal spectrum data from different sensor location at a certain time slot into a single graph. Then we can adopt GNN-based interpolation techniques such as [16] to detect

Y. Zhang, A. Li, J. Li, D. Han, and Y. Zhang are with the School of Electrical, Computer and Energy Engineering, Arizona State University, Tempe, AZ 85287, USA. (E-mail: {yanzhangyz, anglee, jwli, dqhan, yczhang}@asu.edu).

T. Li is with the Computer and Information Technology Department, Indiana University-Purdue University Indianapolis, Indianapolis, IN 46202 (E-mail: tli6@iupui.edu).

R. Zhang is with Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716, USA. (E-mail: ruizhang@udel.edu)

malicious spectrum sensors. The biggest advantage of GNN-based interpolation is that the GNN-based model need not be retrained once trained well. In contrast, the OK-based model needs to recalculate the variance matrix of all sensors every time when new data arrive, resulting in huge computational overhead for a large number of spectrum sensors. Our main contributions can be summarized as follows.

- First, we propose **SpecKriging**, a new spatial-interpolation technique based on Inductive Graph Neural Network Kriging (IGNNK) [16] for secure CSS. In SpecKriging, we first pretrain the GNN model with the historical sensing records of a few trusted sensors called *anchor sensors* during system initialization. During system runtime, we use the model to evaluate the trustworthiness of non-anchor sensor's data and also use them along with anchor sensors' new data to retrain the model. The model retraining stops when the model quality is sufficiently high. Compared with the OK-based methods with computational complexity of $\mathcal{O}(n^4)$ [15], SpecKriging' computational complexity is only $\mathcal{O}(n^2)$ for each time slot and is thus more suitable for large-scale CSS systems.
- Second, we incorporate spectrum-sensor heterogeneity into SpecKriging. As far as we know, SpecKriging is the first work that takes spectrum-sensor heterogeneity into consideration for interpolation-based trustworthiness evaluation of spectrum sensors. In particular, we assign a linear transformation matrix for each sensor to represent its hardware configuration before feeding its sensing reports into the GNN model. These matrices are updated with other model parameters during the training phase.
- Third, we evaluate the efficacy of SpecKriging for trustworthy spectrum-occupancy detection with and without malicious spectrum sensors. Our evaluations use SVM and Random Forest as examples. The experiments show that SpecKriging can distinguish benign and malicious sensors with the classification accuracy up to 98.10% for homogeneous sensors and 97.69% for heterogeneous sensors. When the sensors are homogeneous, SpecKriging can perform correct spectrum-occupancy detection with accuracy as high as 98.45%, which is comparable to the performance of the OK-based techniques [5, 13]. When the sensors are heterogeneous, SpecKriging can still achieve the detection accuracy up to 98.18%, while the OK-based techniques [5, 13] no longer work.

The rest of this paper is organized as follows. Section II gives the problem formulation and threat model. Section III illustrates the SpecKriging design. Section IV presents the experimental evaluation of SpecKriging. Section V reviews the related work. Section VI concludes this paper.

II. Problem Formulation and GNN Primer

A. System Model

We consider an SSP who provides spectrum analytics over a large geographic area divided into many sensing zones. There are one or more static PUs in each sensing zone. Since SpecKriging applies to one or more PUs without any operational difference, we assume only one PU in each sensing zone hereafter for ease of presentation. Fig. 1 shows the CSS architecture where the snapshot of one sensing zone is shown. As in [1, 12, 13], we assume that the SSP deploys several anchor sensors at strategic locations. Anchor sensors are highly safeguarded, tamper-resilient, and remotely monitored by the SSP; they can also undergo periodic remote attestation by the SSP and are excluded if tested as compromised. Although anchor sensors provide trustworthy sensing reports, they are too expensive to deploy in large numbers and can thus only serve as "ground-truth" reports at their locations. the SSP resorts to a lot of cheaper non-anchor sensors along with a few anchor sensors to provide high-quality spectrum analytics. Most non-anchor sensors are expected to be deployed by the SSP itself, and some can be recruited mobile users accepting crowdsourced spectrum-sensing tasks [1–3]. Anchor and non-anchor sensors can be mobile or static and report their location-tagged spectrum-sensing reports to the SSP either periodically or on demand. For simplicity, we assume that each report contains the sensor location and detected RSS (radio signal strength) values, and the extension of our work to other types of spectrum-sensing reports is left as future work. Most sensing reports come from non-anchor sensors are initially untrustworthy. In the following, we use Λ_{anchor} to denote anchor sensors and Λ to denote non-anchor sensors, where $|\Lambda| \gg |\Lambda_{\text{anchor}}|$.

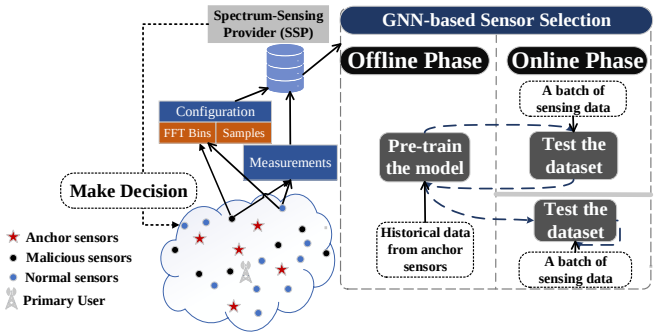


Fig. 1: System architecture.

B. Threat Model

We assume that anchors sensors cannot be compromised, but the adversary can compromise and fully control arbitrary non-anchor sensors. Each malicious sensor hereafter refers to a compromised non-anchor sensor. The adversary is fully aware of our SpecKriging scheme and can use malicious sensors to launch the following attack. First, a malicious sensor submits high RSS values when the PU is absent, aiming to block the spectrum access of legitimate spectrum users by increasing the probability of false alarms. Second, a malicious sensor may report low RSS values when the PU is present, aim to cause interference to the PU by increasing the probability of missed detection. Third, the malicious sensor submits a fake location to attempt inducing a wrong CSS result. Given this threat model, we assume that anchor sensors' reports are trustworthy, but the trustworthiness of all non-anchor sensor's

reports needs to be carefully evaluated before they are explored for spectrum-occupancy detection.

C. Problem Formulation

The goal of our SpecKriging technique is to develop a secure sensor-selection scheme for trustworthy spectrum-occupancy detection when malicious non-anchor sensors are present and even constitute the majority. SpecKriging uses a GNN framework to achieve this goal by mapping the RF signals of all sensors to an undirected graph. It works by using GNN-based spatiotemporal interpolation to predict the sensing data of each non-anchor sensor based on trustworthy anchor-sensor reports and then evaluating the trustworthiness of each non-anchor sensor, which is commensurate with the similarity between their respective interpolation results and reports. In particular, the SSP first pretrains a GNN model with the historical datasets from anchor sensors in the off-line mode during system initialization, as shown in Fig. 1. Note that a limited number of anchors are always not sufficient to train a GNN model that is good enough for testing. Therefore, retraining by using new sensors is necessary for the first several rounds of the online mode. In particular, each time the SSP receives new spectrum-sensing reports from non-anchor and anchor sensors during system runtime, i.e., in the online mode, it tests non-anchor sensor reports in batches with the pre-trained model and new anchor sensor reports, distinguish benign and malicious non-anchor sensors, and then retrain the GNN model. The model retraining can stop once the retrained GNN model satisfies the detection-accuracy requirement. In this scenario, only dataset testing is needed as illustrated in Fig. 1.

D. IGNNK Method for Spatiotemporal Interpolation

We use GNNs to characterize the spatiotemporal relationships of all sensing reports generated by different spectrum sensors at different time. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ denote an undirected graph, where $\mathcal{V}$ and $\mathcal{E}$ denote the node and edge sets, respectively. We also define a Euclidean distance-based adjacency matrix W as follows,

$$W_{i,j} = \exp\left(-\left(\frac{\mathsf{dist}(v_i, v_j)}{\sigma}\right)^2\right), \quad (1)$$

where $\mathsf{dist}(v_i, v_j)$ denotes the Euclidean distance or edge weight between arbitrary sensors i and j. We also use X_i to represent the data of node $i \in \mathcal{V}$.

We adopt the IGNNK model in [16] as the basic model for spatiotemporal representation and interpolation of sensing signals. One superior feature of IGNNK is that it is an inductive GNN which has parameters independent of the scale of graph nodes and can be used for previous unseen nodes or even an unseen graph. This feature makes the graph available even when the input data are from non-anchor sensors (e.g., mobile crowdsourcing users) that have never appeared.

IGNNK works as follows. When a period of datasets that contain some masked measurements are fed into the IGNNK model, this model learns the corresponding spatiotemporal graph and then reconstruct the data of all nodes including both observed and masked nodes. Here the measurements of masked nodes—called masked or unknown measurements—are interpolated based on the spatiotemporal relationships with their neighbors in the graph. This model has three layers where the first layer is the initial representation of signals. IGNNK adopts Diffusion Graph Convolutional Networks (DGCNs) [17] as the basic building block show below,

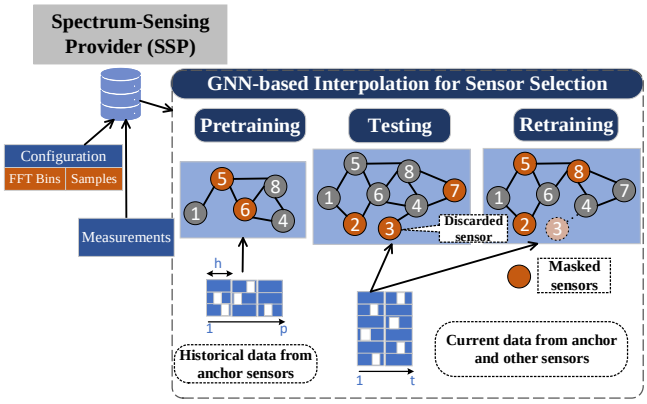


Fig. 2: Overview of learning and prediction in SpecKriging.

$$H_1 = \sum_{k=1}^{K} T_k(\overline{W}_{\text{sample}}) H_0 \Theta_{b,0}^k + T_k(\overline{W}_{\text{sample}}) H_0 \Theta_{f,0}^k \,. \quad (2)$$

Here $H_0 = X_{\text{sample}}^M$ denote observed and masked measurements fed into the first layer, and $\overline{W} = W_{\text{sample}}/\mathsf{rowsum}(W_{\text{sample}})$ represents the forward/backward transition matrix. T_k, $\Theta_{b,0}^k$, and $\Theta_{f,0}^k$ are the model's inner function and parameters representing the Chebyshev polynomial and learning parameters.

Note that the masked nodes only pass 0 to their neighbors in this layer and their features are also masked hindering them to produce desired representation. Therefore, the second layer is added to produce more generalized representations of masked nodes, which is represented by

$$H_2 = \sigma\Big(\sum_{k=1}^{K} T_k(\overline{W}_{\text{sample}}) H_1 \Theta_{b,1}^k + T_k(\overline{W}_{\text{sample}}) H_1 \Theta_{f,1}^k\Big) + H_1 \,. \quad (3)$$

The third layer is to reconstruct the input data as

$$\hat{X}_{\text{sample}} = \sum_{k=1}^{K} T_k(\overline{W}_{\text{sample}}) H_2 \Theta_{b,2}^k + T_k(\overline{W}_{\text{sample}}) H_2 \Theta_{f,2}^k \,. \quad (4)$$

Here the loss function is defined as the total reconstruction error on both unmasked and masked samples as follows

$$l = \sum_{\text{sample}} ||\hat{X}_{\text{sample}} - X_{\text{sample}}||^2 \,. \quad (5)$$

With the IGNNK method, the measurements from new locations can be predicted using the trained model. In particular, these samples are masked as unobserved data and fed into the model along with the observed ones, and the output of the model is the interpolation results for the masked locations.

III. SpecKriging Design

A. SpecKriging GNN for Homogeneous Sensors

We use the graph model $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ in Section II-D to characterize the spatiotemporal sensing data reports. Each anchor or non-anchor sensor corresponds to a unique node in $\mathcal{V}$. We use sensor or node i and V_i interchangeably hereafter. The edge set $\mathcal{E}$ contains an edge between any two spectrum sensors (anchor or non-anchor). The edge weights form the Euclidean distance-based adjacency matrix $\mathcal{W}$ defined in Eq. (1). We also use $\mathcal{X}$ to denote all the time series of sensing data (i.e., RSS values), and $X_i \in \mathcal{X}$ to represent the data of each node $i \in \mathcal{V}$. Fig. 3 illustrates the graph model where Type-n means the configuration information of each sensor investigated in Section III-B.

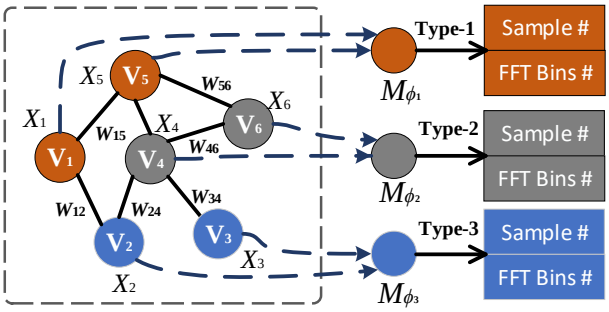


Fig. 3: Graph model illustration.

Algorithm 1 illustrates the basic SpecGriging GNN architecture when all spectrum sensors are homogeneous. In this algorithm, the model goes through pretraining-testing-retraining steps, and new data reports are finally classified as either trustworthy or malicious. Trustworthy reports are used for both spectrum-occupancy detection and model retaining if needed, while malicious reports are simply dropped. It is worth noting that malicious reports are similar to wrong reports submitted by dysfunctional but benign spectrum sensors. So it is up to the SSP to decide whether the corresponding spectrum sensors are excluded permanently or temporarily suspended from the system. Note that in the testing phase, non-anchor sensors are not added in one time. Instead, each time we add a batch of nodes that have the nearest distance to anchor sensors (see Step 2). Fig. 6a shows that adding all non-anchor sensors in one time results in worse performance than our scheme because too many negative samples can reduce the sensor-classification performance.

Fig. 2 illustrates the workflow of SpecKriging. In the training phase, we mask the anchor nodes whose reports are trustworthy and improve the model performance by minimizing the prediction error of the masked nodes. In the testing phase, we mask the newly added nodes for trustworthiness classification. If the difference between the predicted and measured value collected in one time slot exceeds a predefined empirical threshold, the corresponding node is classified as malicious and discarded in this time slot.

Algorithm 1 SpecKriging algorithm for homogeneous sensors

Require: Historical RSS data X from anchor sensors over period $[1, p]$ (size $n \times p$); adjacent matrix W of these anchor sensors; New data X^{new} from both anchor and non-anchor sensors over a short period $[1, t]$ ($t = h *$ batch) (size $(n + k) \times t$). Parameters: parameters of the IGNNK architecture P_0.

1: Train IGNNK algorithm using X and W; ▷ ***Initial training step***
2: Sort all non-anchor sensors according to the distance from anchor sensors, and then partition them into sets $S_1, S_2, ..., S_M$ with $|S_1| < |S_2|$, $S_2 < S_3$,...,$|S_{M-1}| < |S_M|$;
3: Let the current node set $S^{\text{cur}} = S^{\text{anc}}$, and the added node set $S^{\text{add}} = \emptyset$;
4: **for** m=1 : M **do**
5: Update $S^{\text{add}} = S_m$. Concatenate the sensing data, X_m^{add} and X_m^{cur}, from S^{add} and S^{cur} during $[1, t]$, and obtain a new data matrix X_m^{new} and adjacent matrix W_m^{new};
6: Test X_m^{new} with masked X_m^{add} using the trained model and W_m^{new}; obtain a predicted signal matrix $\hat{X}$; ▷ ***Testing step***
7: **for** $x_m[i, j]$ in X_m^{new} **do**
8: **if** $\left|\hat{x}_m[i, j] - x_m[i, j]\right| > \tau_1$ **then**
9: $x_m[i, j] = 0$;
10: **end if**
11: **end for**
12: Make the zero elements in X_m^{new} as missing values which will not participant in the following retraining process of MODIFIED_IGNNK algorithm;
13: Update $S^{\text{cur}} = \{S^{\text{anc}}, S_1, S_2, ..., S_m\}$;
14: Retrain IGNNK algorithm; ▷ ***Retraining step***
15: **end for**

B. SpecKriging GNN for Heterogeneous Sensors

Now we illustrate how to modify the basic SpecKriging GNN architecture to accommodate heterogeneous spectrum sensors. A heterogeneous graph is associated with a sensor-configuration mapping function $\phi : \mathcal{V} \to \mathcal{M}$ where $\mathcal{M}$ denotes the set of sensor configuration [18, 19]. For each configuration type of sensors (e.g., those of type ϕ_i), we use the type-specific linear transformation matrix M_{ϕ_l} to project the features of different sensor configurations into the same feature space as

$$x_i' = M_{\phi_l} \cdot x_i \ . \tag{6}$$

Here x_i and x_i' denote the original and projected feature of sensor i, respectively.

Thus we incorporate these configuration matrices into the model's parameter set and update them during the training phase. Algorithm 2 shows the brief structure of the heterogeneous SpecKriging GNN, where MODIFIED_IGNNK represents a new model for reconstructing sensing data based on the IGNNK model. The key difference is MODIFIED_IGNNK requires the data matrix to be multiplied by the linear transformation matrix—referred to as a configuration matrix

Algorithm 2 SpecKriging algorithm for heterogeneous sensors

Require: Historical data X from anchor sensors S^{anc} over period $[1, p]$ (size $n \times p$); adjacent matrix W of these sensors; current data X^{new} from both anchor and non-anchor sensors over a short period $[1, t]$ (size $(n+k) \times t$); a configuration feature list F^r containing $(n+k)$ elements with each of size q (the number of configuration feature). Parameters: parameter set of the IGNNK architecture, P_0; time slot length h.

1: Create two lists, $F_{\text{ded}}^{\text{anc}}$ and $T_{\text{ded}}^{\text{anc}}$ (of size $h \times h$), for anchor sensors; Create two empty lists, $F_{\text{ded}}^{\text{non}}$ and $T_{\text{ded}}^{\text{non}}$, for current empty non-anchor sensor set; add these four lists to MODIFIED_IGNNK algorithm as learnable parameters.
2: Train MODIFIED_IGNNK model in Training-Mode using X and W; update $F_{\text{ded}}^{\text{non}}$ and $T_{\text{ded}}^{\text{non}}$; ▷ ***Initial training step***
3: Sort all non-anchor sensors according to the distance from anchor sensors; partition them into sets $S_1, S_2, ..., S_M$ with $|S_1| < |S_2|$, $S_2 < S_3$,...,$|S_{M-1}| < |S_M|$;
4: Let the non-anchor sensor set $S^{\text{non}} = \emptyset$, and the added node set $S^{\text{add}} = \emptyset$;
5: **for** m=1 : M **do**
6: Let current node set $S^{\text{cur}} = \{S^{\text{anc}}, S^{\text{non}}\}$ consisting of anchor and current non-anchor sensor set;
7: Update $S^{\text{add}} = S_m$. Concatenate the data from S^{add} and S^{cur} during $[1, t]$, X_m^{add} and X_m^{cur}, and obtain a new data matrix X_m^{new} and adjacent matrix W_m^{new} from the concatenated nodes; derive a redundant configuration feature list $F_{\text{red}}^{\text{add}}$ only from S_m
8: Test X_m^{new} with masked X_m^{add} using the model, W_m^{new}, and $F_{\text{red}}^{S_m}$ in Testing-Mode, and obtain a predicted data matrix $\hat{X}$; ▷ ***Testing step***
9: **for** $x_m[i, j]$ in X_m^{new} **do**
10: **if** $\left|\hat{x}_m[i, j] - x_m[i, j]\right| > \tau_1$ **then**
11: $x_m[i, j] = 0$;
12: **end if**
13: **end for**
14: Make the zero elements in X_m^{new} as missing values;
15: Update $S^{\text{non}} = \{S_1, S_2, ..., S_m\}$; augment $F_{\text{ded}}^{\text{non}}$.append($F_{\text{ded}}^{S_m}$) and $T_{\text{ded}}^{\text{non}}$.append($T_{\text{ded}}^{S_m}$);
16: Retrain MODIFIED_IGNNK algorithm in Training-Mode; update $F_{\text{ded}}^{\text{anc}}$, $T_{\text{ded}}^{\text{anc}}$, augmented $F_{\text{ded}}^{\text{non}}$, and augmented $T_{\text{ded}}^{\text{non}}$; ▷ ***Retraining step***
17: **end for**

hereafter—before feeding the data to the IGNNK model. Note that instead of adding the redundant configuration matrices into the model's parameters, we add the de-duplicated configuration matrices. In the following, we use three new metrics defined in Table I—F_{ded}, T_{ded}, and T_{red}—to characterize the configuration knowledge.

In Algorithm 2, we first randomly generate a de-duplicated list of configuration matrices for all anchor sensors corresponding to their known configuration features. Then we train MODIFIED_IGNNK model using the signal matrix, adjacency matrix, and configuration matrix of all anchor sensors. In particular, the model updates the parameters containing not only the parameter set of the IGNNK algorithm but also the configuration parameters of all anchor sensors, $T_{\text{ded}}^{\text{anc}}$, as shown in Algorithm 3. In the testing step, because the model does not know the configuration matrices of the new nodes in advance, we propose to use a small neural network (NN) to predict these matrices using the matrices from the previously known nodes, as detailed in Algorithm 4. With the predicted configuration matrices, MODIFIED_IGNNK could determine if the measurements from new nodes are trustworthy or not. Then it adds trustworthy measurements into the dataset and discard the untrustworthy ones. Then we retrain MODIFIED_IGNNK using the augmented dataset, the augmented adjacency matrix, and the augmented configuration matrix list which has been initialized in the testing step. The testing and retraining steps are executed in batches. In summary, the training mode is responsible for updating model parameters, while the testing mode is responsible for predicting the configuration matrices of new nodes and then decide if the nodes are trustworthy or malicious.

TABLE I: Definitions of some variables used in Algorithm 2.

Variables	Description
F_{ded}	a de-duplicated list of the collected configuration features
T_{ded}	a de-duplicated list of linear transformation matrices, initialized by random generation for anchor sensors while by NN for non-anchor sensors, and updated with other model parameters
T_{red}	a list of the redundant linear transformation matrices used to multiply the original data X

Algorithm 3 MODIFIED_IGNNK in training mode

Require: Data matrix X and the corresponding adjacent matrix W. Parameters: two de-duplicated lists, $F_{\text{ded}}^{\text{anc}}$ and $T_{\text{ded}}^{\text{anc}}$, from anchor sensors; two de-duplicated lists, $F_{\text{ded}}^{\text{non}}$ and $T_{\text{ded}}^{\text{non}}$, from current non-anchor sensors (if exist); P_0;

1: Obtain $T_{\text{red}}^{\text{anc}}$ from anchor sensors with $T_{\text{ded}}^{\text{anc}}$ and $F_{\text{ded}}^{\text{anc}}$;
2: **if** the non-anchor sensor set is NOT empty **then**
3: Obtain $T_{\text{red}}^{\text{non}}$ from the non-anchor sensors with $T_{\text{ded}}^{\text{non}}$ and $F_{\text{ded}}^{\text{non}}$;
4: **end if**
5: Concatenate $T_{\text{red}}^{\text{anc}}$ and $T_{\text{red}}^{\text{non}}$, and obtain a new matrix list $T_{\text{red}}^{\text{con}}$;
6: Multiply X with $T_{\text{red}}^{\text{con}}$, and get X';
7: obtain reconstructed data $\hat{X}'$ = IGNNK(X', W, P_0);
8: **return** $\hat{X}'$

IV. Performance Evaluation

In this section, we experimentally evaluate the efficacy and efficiency of SpecKriging for both homogeneous and heterogeneous sensors with comparison to OK-based interpolation techniques [5, 13].

A. Experimental Setup

We design experiments to verify that we could use SpecKriging to select trustworthy RSS measurements for coop-

Algorithm 4 MODIFIED_IGNNK in testing mode

Require: Data matrix X and the corresponding adjacent matrix W. Parameters: the same parameters as that on the Training-Mode; a redundant configuration feature list $F_{\text{red}}^{\text{add}}$ from the new non-anchor sensors;

1: Concatenate $T_{\text{ded}}^{\text{anc}}$ and $T_{\text{ded}}^{\text{non}}$ from anchor and current non-anchor sensors, and get a new list $T_{\text{ded}}^{\text{con}}$;
2: Concatenate $F_{\text{ded}}^{\text{anc}}$ and $F_{\text{ded}}^{\text{non}}$, and obtain a new list $F_{\text{ded}}^{\text{con}}$;
3: Use a small NN network to learn the mapping from $F_{\text{ded}}^{\text{con}}$ to $T_{\text{ded}}^{\text{con}}$; use the network to predict $T_{\text{ded}}^{\text{add}}$ from the new non-anchor sensors with $F_{\text{red}}^{\text{add}}$;
4: Obtain the overall redundant configuration matrix list $T_{\text{red}}^{\text{o}}$ from anchor, current non-anchor and added sensors using $T_{\text{ded}}^{\text{con}}$, $F_{\text{ded}}^{\text{con}}$, $T_{\text{ded}}^{\text{add}}$, and $F_{\text{ded}}^{\text{add}}$;
5: Multiply X with $T_{\text{red}}^{\text{o}}$, and obtain X';
6: obtain reconstructed data $\hat{X}'$ = IGNNK(X', W, P_0);
7: **return** $\hat{X}$

erative spectrum-occupancy detection. We use the following two datasets corresponding to homogeneous-node and heterogeneous-node scenarios, respectively.

- *Real data from homogeneous sensors*. We use the datasets from [13, 20]. The sensing data are collected on the campus of the University of Colorado Boulder and cover an area of 1.5km×1.1km. We test the sensing data from 100 randomly chosen sensors, and there are 4 randomly placed transmitters emitting signals in different time slots in the corresponding coverage area. Three measurements are collected for each sensor in each time slot, so we let $h = 3$. Here we just show the interpolation performance because the datasets contains only the data for transmitter presence.
- *Synthetic data from homogeneous sensors*. We use the RF signal-analysis tool SPLAT! [21] and the terrain database of US Geological Survey [22] to generate the path-loss model for spectrum sensors. We consider a geographic area of 1 km by 1 km where we randomly deploy 100 spectrum sensors with the PU at the center. Actually, if the PU moves during a sensing period, we can collect the time-varying sensing reports and train a new spatiotemporal GNN model, which could be our future work. We also simulate spatially correlated noise samples of the same number as the signal samples for sensor selection and spectrum-occupancy detection.
- *Synthetic data from heterogeneous sensors*. We adopt the simulation data in [4], which defines 36 different sensor-configurations with each corresponding to a tuple (N, NFFT). Here N denotes the number of I/Q samples, and NFFT denotes the resolution of the FFT algorithm. Because there is only one energy value in every location, we can only generate a 1-dimensional feature vector which is not sufficient for spectrum-sensing experiments. Therefore, we augment the dataset by adding some Gaussian noise of power -80dBm.

In our experiments, we claim that a malicious sensor launches an attack with *attack strength T* dB if it reports x_i+T instead where x_i denotes the true RSS value [1, 23]. Table II lists the default experiment parameters.

TABLE II: Default experiment parameters.

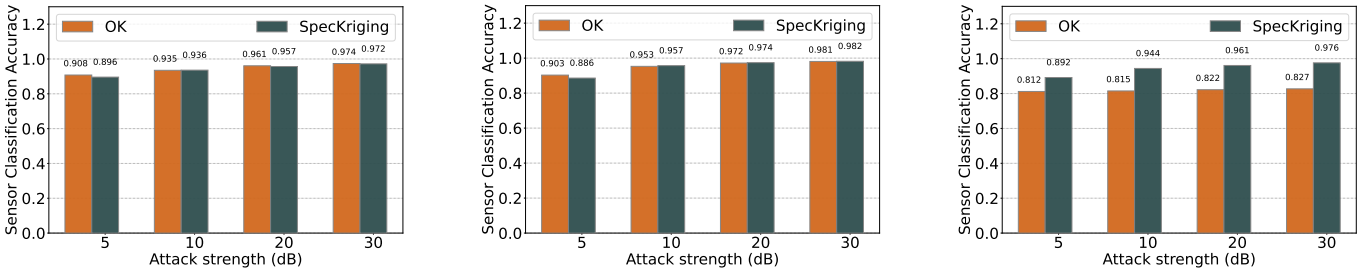
Para.	Value	Description
k	70	Number of sensors
k_a	20	Number of malicious sensors
n	10	Number of anchor sensors
P	27-33 dBm [6]	Transmitter power
N_0	-80dBm	Noise power
t	200	Number of time slots of anchor sensors' historical data
t^{new}	50	Number of time slots of new data
T	10 dB	Attack strength
h	2	Number of measurements in each time slot
q	2	Configuration feature length
P_0	[16]	Parameters of IGNNK algorithm

B. Classification and Interpolation Performance for Evaluating Sensor Trustworthiness

Fig. 4 compares SpecKriging with OK-based interpolation techniques [5, 13] for distinguishing between benign and malicious non-anchor sensors. The performance metric here is the classical classification accuracy defined as the number of correct predictions divided by the total number of predictions. Fig. 4a and Fig. 4b show the results for homogeneous spectrum sensors. For both methods, the classification accuracy increases with the attack strength because benign and malicious sensors are obviously much easier to tell apart for larger attack strength. In addition, SpecKriging and the OK-based methods have comparable classification performance. Fig. 4c shows the results for heterogeneous sensors. It is clear that SpecKriging outperforms the OK-based techniques regardless of the attack strength. The reason is that the OK algorithm calculates the weights between two nodes based on the prediction variance, but the sensor-configuration information is unpredictable and not related to the variance matrix constructed by the OK algorithm. This issue makes the OK-based methods less applicable when spectrum sensors have highly heterogeneous hardware configurations.

Fig. 5 compares the interpolation performance of SpecKriging and the OK-based methods when all the non-anchor sensors are benign. The performance metric here is Mean Absolute Error (MAE), which indicates the difference between the interpolated (predicted) and measured data values. We test the real data with 30 sensors during the training phase. As shown in Fig. 5a, the MAE values of SpecKriging algorithm decrease sharply in the first several epochs and finally reach a lower extreme (smaller than 6) than that of the OK-based algorithm. This result verifies the effectiveness of SpecKriging in real scenarios.

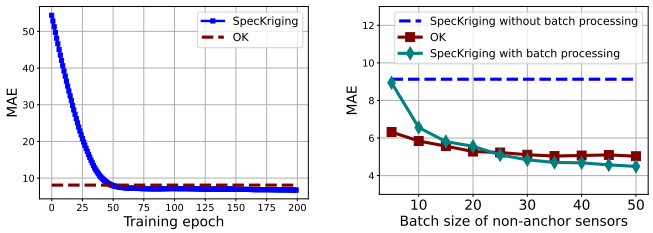
Recall that SpecKriging adds non-anchor sensors in batches during the testing and model-retraining phases. Next, we test the MAE changes because of the batch processing. We assume that non-anchor sensors are partitioned into random batches for the model training. The dataset contains 50 new non-anchor sensors with each reporting 3 RSS values (i.e., covering $3/h = 1$ time slots). These sensors are used for validation



(a) homogeneous sensors with real data (b) homogeneous sensors with synthetic data (c) heterogeneous sensors with synthetic data

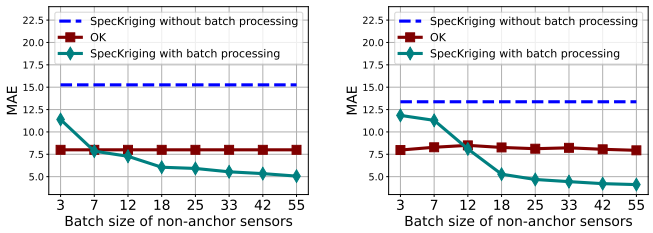
Fig. 4: Sensor-classification accuracy for homogeneous and heterogeneous sensors.

and thus not used in the model-training phase. We use the trained model to predict the data values of these sensors and then compute the average MAE. Each data point in Fig. 5b represents the average of 100 runs. Other experiment parameters follow Table II.



(a) homogeneous sensors with real data (b) homogeneous sensors with real data

Fig. 5: MAEs of homogeneous sensors corresponding to real data.



(a) homogeneous sensors with synthetic data (b) heterogeneous sensors with synthetic data

Fig. 6: MAEs of homogeneous and heterogeneous sensors corresponding to synthetic data.

We also test the performance of synthetic data of homogeneous and heterogeneous sensors. The dataset contains 55 new non-anchor sensors with each reporting 100 RSS values (i.e., covering $100/h$ time slots). Fig. 6a plots the MAE for homogeneous sensors. we can clearly see SpecKriging has much better performance when batch processing is used. In addition, the larger the batch size, the more non-anchor sensors contributing to the model training, the better the interpolation accuracy, the lower the MAE, and vice versa. In addition, SpecKriging starts to outperform the OK-based methods when the batch size exceeds 7. Similar results can be observed in Fig. 6b for heterogeneous sensors. Note that the MAE of SpecKriging decreases significantly with the batch size, while the MAE of the OK-based methods has little change. The main reason is that the OK-based methods do not take sensor configuration into consideration, resulting in prediction errors for new sensors.

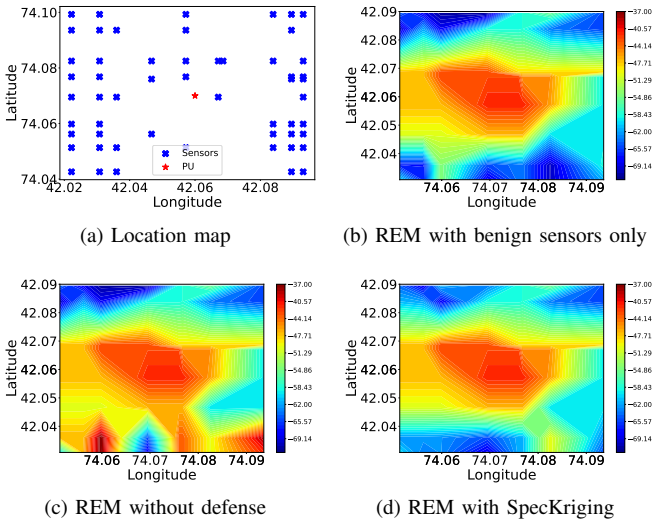
C. Performance for Spectrum-Occupancy Detection

In this section, we compare the performance of SpecKriging and the OK-based methods for spectrum-occupancy detection in the presence of malicious sensors. For this set of experiments, we use classification accuracy and also the reconstructed Radio Environmental Map (REM) as the main performance metrics. In this context, a classification error is said to happen if a present (or absent) PU is classified as absent (or present). We use two popular machine learning methods—SVM and Random Forest [24, 25]—for detecting spectrum occupancy based on trustworthy data reports output by SpecKriging or the OK-based methods. Each data sample is a time sequence of RSS measurements, and there can be missing measurements after SpecKriging or the OK-based methods are applied to detect and then discard malicious measurements. Since the data samples with missing measurements cannot be directly used, we use the interpolated measurements to substitute the discarded measurements.

1) Homogeneous sensors: This section reports the results with homogeneous spectrum sensors. In this group of experiments, we have 100 samples (each of size $k+n$) with the PU present and 100 noise samples of the same size. The simulation is done in Python, and anchor/benign/malicious sensors are picked randomly. Each point in Fig. 8 is the average of 100 runs. We also show the results without any countermeasure as a reference.

Fig. 7 visualizes the REMs when all nodes are benign (Fig. 7b), there are malicious sensors without any defense (Fig. 7c), and SpecKriging is adopted to counteract malicious sensors (Fig. 7d). Because the PU is at the center of the sensing region, its signal strength decreases gradually from the center to the corners. Malicious sensors report high RSS values to change the signal distribution in REM significantly. We can clearly see that SpecKriging could exclude the malicious sensors and reconstruct accurate measurements in their locations.

Fig. 8a demonstrates the classification accuracy with different numbers of anchor sensors. Both SpecKriging and the OK-based methods achieve comparably very high classification accuracy when malicious sensors are present. In addition, the classification accuracy increases with the number of anchor



(a) Location map (b) REM with benign sensors only

(c) REM without defense (d) REM with SpecKriging

Fig. 7: REMs for homogeneous sensors.

sensors because more trustworthy measurements could train a better model for evaluating sensor trustworthiness. In addition, even though the sensor-classification accuracy is relatively low for low attack strength as shown in Fig. 4b, the spectrum-occupancy classification accuracy can still be very high. The reason is that with the lower attack strength, malicious measurements escaping the detection are also more similar to benign measurements and thus have a weaker negative impact on spectrum-occupancy classification.

Fig. 8b shows the classification accuracy with different numbers of malicious sensors. As we can see, the classification accuracy decreases with the number of malicious sensors. This result coincides with our intuition but is still much better than without any defense. The good news is that even when there are more malicious sensors than benign ones (40 vs. 30), SpecKriging can still achieve classification accuracy over 90% better than that of the OK-based methods.

Fig. 8c illustrates the classification accuracy as the attack strength varies. When the attack strength becomes 0, the classification accuracy is clearly the highest. In addition, no matter whether the attacker reports higher or RSS values, the classification accuracy always drops. The reason is that higher RSS values increase the false positive rate (FPR) when the PU is absent, while lower RSS values increase the false negative rate (FNR) when the PU is present. In all simulated scenarios, SpecKriging can still achieve classification accuracy over 90%.

2) Heterogeneous sensors: Now we report the results with heterogeneous spectrum sensors. In this group of experiments, we feed the energy data with sensor-configuration information into the SpecKriging model.

We first show the REMs with benign sensors only (Fig. 9b), with malicious sensors but without any defense (Fig. 9c), and with SkecKriging adopted to detect malicious sensors (Fig. 9d). We can clearly see the high performance of SpecKriging in excluding malicious measurements and reconstruct accurate substitutes.

Fig. 10a shows the classification accuracy with different numbers of anchor sensors. SpecKriging achieves very high classification accuracy and significantly outperforms the OK-based methods which cannot accommodate sensor-configuration heterogeneity. Also as expected, the more anchor sensors, the higher the classification accuracy.

Fig. 10b illustrates the impact of different numbers of malicious sensors. SpecKriging always outperforms the OK-based methods and maintains high performance even when malicious sensors are more than benign ones (40 vs. 30).

Fig. 10c demonstrates the classification accuracy as the attack strength varies. SpecKriging always outperforms the OK-based methods and maintains high performance even when the attack strength is very high. Since high attack strength can significantly increase the attacker's cost, SpecKriging can significantly raise the bar of launching effective spectrum-sensing attacks.

Fig. 11 shows the effect of sensor density on the classification accuracy. As we can see, a large number of sensors can boost the classification performance, which also verifies that trustworthy anchor sensors alone are not enough for high-quality spectrum analytics.

D. Computational Time

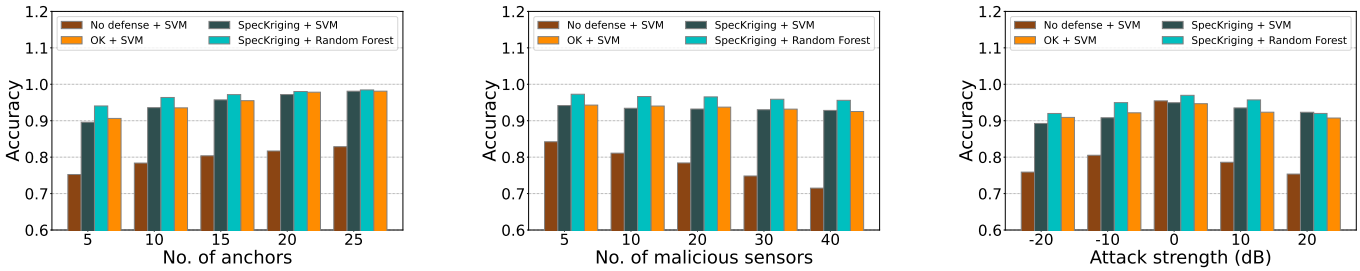
We also evaluate the computational time (mainly the dominating model-training latency) of SpecKriging and compare it with the OK-based methods. We train the model on a Dell desktop with 3.19 GHz CPU, 16 GB RAM, and Windows 10 64-bit Professional. As shown in Fig. 12a, SpecKriging has similar model-training latency to the OK-based methods when the data size is small and is significantly better when the data size ((especially the number of sensors) becomes large. This result is anticipated due to their different computational complexity. For the OK-based methods, the computational complexity is $\mathcal{O}(n^4)$ with n denoting the number of sensors [15]. Since IGNNK constructs three layers of DGCNs [17] with computational complexity $\mathcal{O}(n^2)$ in our scenario, SpecKriging also has computational complexity $\mathcal{O}(n^2)$ for each time slot. Therefore, SpecKriging outperforms the OK-based schemes in large-scale CSS systems.

We also evaluate the effect of the number of measurements, h, in each time slot for SpecKriging with heterogeneous sensors. This parameter h also denotes the size of the linear transformation matrix assigned to the input data. It determines the granularity of the data after being multiplied by h. Fig. 12b shows the classification accuracy with different h. As we can see, when h increases, the classification accuracy decreases due to the coarser data segmentation during linear transformation. However, a small h value means high computational complexity. Therefore, there is a trade-off between classification accuracy and computational complexity with regard to h.

V. Related Work

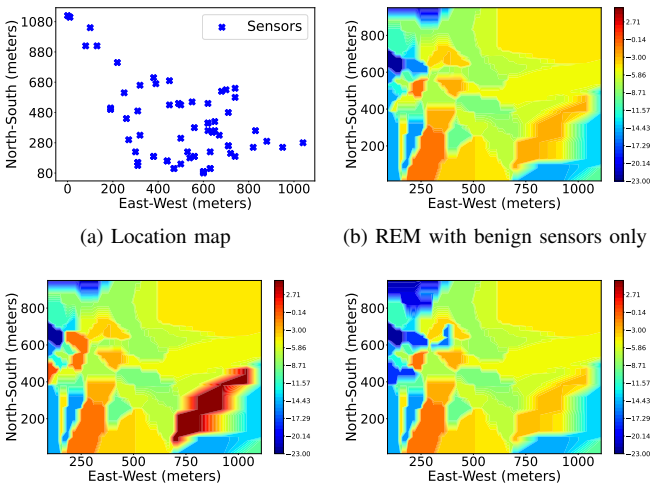
In this section, we briefly outline the prior work most germane to SpecKriging.

There are various countermeasures to identify the false sensing reports. The first category is anomaly detection [7, 8, 26, 27]. Some methods identify false sensing reports by differentiating them from other measurements of the normal



(a) Accuracy vs. # of anchor sensors (b) Accuracy vs. # of malicious sensors (c) Accuracy vs. attack strength

Fig. 8: Spectrum-occupancy classification accuracy with homogeneous spectrum sensors.



(a) Location map (b) REM with benign sensors only

(c) REM without defense (d) REM with SpecKriging

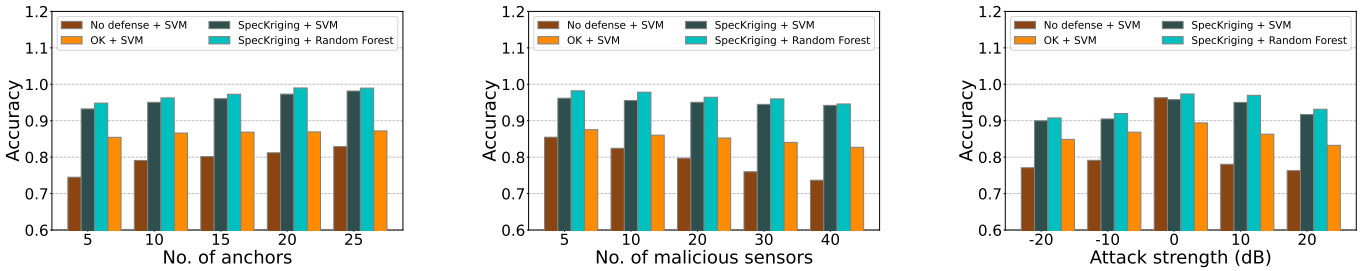
Fig. 9: REMs of heterogeneous nodes.

nodes in their neighborhood. For example, in [8], the authors use a moving box window to detect the abnormal locations where the measurement is very different from that submitted by its neighbors. Some other techniques use machine learning-based methods to predict attacking behaviors based on the legitimate sensors' previous patterns. For example, in [26], the authors use one-class SVM to learn the pattern of normal sensors' previous behavior and predict future abnormal reports from attackers. These methods fail when the malicious nodes become the majority in the sensing region. The second category detects adversaries based on the reputation scores from historical logs or statistical results [9–11]. In [9], the authors propose to use sensors' reputation levels as their fusion weights during spectrum detection. However, reputation-based methods cannot handle the sudden change in mobile sensors' behaviors and are thus vulnerable to instantaneous attacks.

Researchers also use the trusted nodes/anchors to detect the false measurements from the crowdsourcing users [1, 12, 13]. In [12], the authors observe that given the real signal propagation information from PU, neighboring cells can help to correct the sensing decision. In [1], the relationship between anchors and normal nodes are established to exclude abnormal nodes. However, the performance degrades if the propagation model is not accurate enough.

Recently, some papers propose interpolation techniques, such as Inverse Distance Weighting (IDW) and Ordinary Kriging (OK), to detect malicious nodes [5, 13, 14]. The intuition is that measurements from good nodes can be used to "interpolate" the measurements from unknown sensors. If the reports from some sensors are significantly different from the interpolated results, these sensors are probably malicious. For example, in [5], the authors use OK to select the proper sensors for the spectrum occupancy query. In [13], the authors use OK to select the normal sensors to construct the radio environment map. These methods are promising for sensor selection but they have some limitations as mentioned in I. First of all, these methods become less effective when the sensors have heterogeneous features (e.g., various sampling rates or different number of antennas) that affect the sensing reports. In contrast, SpecKriging explores a GNN model for the interpolation task and incorporates sensors' heterogeneous information into the model. By combining the spatial-temporal and configuration features, we can obtain good interpolated results from the unknown sensors. In addition, the existing interpolation methods are applied to spatial or temporal interpolation solely. Therefore, each time when a new dataset arrives, all sensors are required to update the interpolation matrix. SpecKriging, by contrast, does not need extra training efforts when new nodes are added once the model is trained.

Researchers have also explored the sensor heterogeneity in cooperative spectrum sensing [4, 28–32]. In [28, 29, 32], the authors consider the heterogeneity of SUs in terms of the SNR values, reporting errors, and data transmission rates. Besides, the authors in [30] investigate the heterogeneous sensing abilities including detection and false alarm rates. In addition, authors in [31] discuss the CSS performance of SUs with different antenna numbers and sampling rates. The most recent work [4] considers the scenario when spectrum sensors have heterogeneous sensing capabilities including sampling rates (i.e., the number of FFT bins) and sample numbers in one time slot. Similar to the work in [4], we also incorporate spectrum heterogeneity into SpecKriging, with regard to sampling rates and sample numbers. As far as we know, SpecKriging is the first work that takes spectrum-sensor heterogeneity into consideration for interpolation-based trustworthiness evaluation of spectrum sensors. In our setting, because the sample number and FFT bin number are both discrete values, we could classify each sample number (i.e., an FFT bin number pair) as one sensor type and assign it a type-specific linear transformation matrix. As for other heterogeneous configuration information such as SNR, discretization is needed before such information



(a) Accuracy vs. # of anchor sensors (b) Accuracy vs. # of malicious sensors (c) Accuracy vs. attack strength

Fig. 10: Spectrum-occupancy classification accuracy with heterogeneous spectrum sensors.

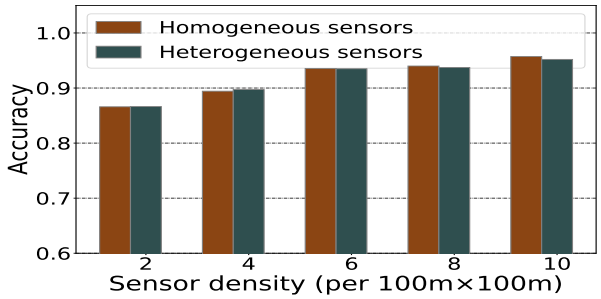
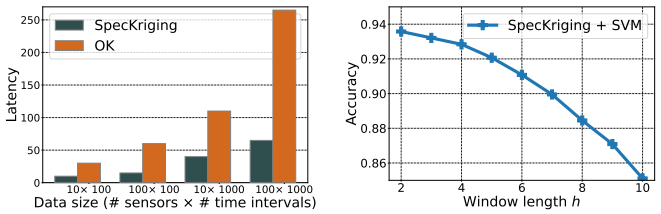


Fig. 11: Classification accuracy vs. sensor density.



(a) training latency (b) impact of window length h.

Fig. 12: Efficiency analysis of SpecKriging.

is assigned a transformation matrix and fed into the GNN model.

VI. Conclusion and Future Work

In this paper, we present the design and evaluation of SpecKriging, the first work that explores GNNs and considers sensor heterogeneity for secure cooperative spectrum sensing (CSS). Extensive experiments show that SpecKriging maintains very high spectrum-occupancy detection accuracy even when the majority of spectrum sensors are malicious. We also show that besides its unique support for sensor heterogeneity, SpecKriging has comparably high performance for trustworthy spectrum-occupancy detection and much lower computational complexity than the state-of-the-art prior work in large-scale CSS systems with homogeneous spectrum sensors.

References

[1] R. Zhang, J. Zhang, Y. Zhang, and C. Zhang, "Secure crowdsourcing-based cooperative spectrum sensing," in *IEEE INFOCOM*, Turin, Italy, April 2013.

[2] X. Jin, R. Zhang, Y. Chen, T. Li, and Y. Zhang, "Dpsense: Differentially private crowdsourced spectrum sensing," in *ACM CCS*, Vienna, Austria, October 2016.

[3] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *IEEE INFOCOM*, San Francisco, CA, April 2016.

[4] A. Chakraborty, A. Bhattacharya, S. Kamal, S. Das, H. Gupta, and P. Djuric, "Spectrum patrolling with crowdsourced spectrum sensors," in *INFOCOM*, Honolulu, HI, April 2018.

[5] A. Chakraborty, M. Rahman, H. Gupta, and S. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *INFOCOM*, Atlanta, GA, May 2017.

[6] A. Bhattacharya, C. Zhan, H. Gupta, S. Das, and P. Djuric, "Selection of sensors for efficient transmitter localization," in *INFOCOM*, Virtual Conference, July 2020.

[7] H. Li and Z. Han, "Catch me if you can: An abnormality detection approach for collaborative spectrum sensing in cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 9, no. 11, pp. 3554–3565, November 2010.

[8] O. Fatemieh, A. Farhadi, R. Chandra, and C. Gunter, "Using classification to protect the integrity of spectrum measurements in white space networks," in *NDSS*, San Diego, CA, February 2011.

[9] R. Chen, J. Park, and K. Bian, "Robust distributed spectrum sensing in cognitive radio networks," in *INFOCOM*, April 2008, pp. 1876–1884.

[10] K. Zeng, P. Paweczak, and D. Cabric, "Reputation-based cooperative spectrum sensing with trusted nodes assistance," *IEEE Communications Letters*, vol. 14, no. 3, pp. 226 –228, March 2010.

[11] Y. Sagduyu, "Securing cognitive radio networks with dynamic trust against spectrum sensing data falsification," in *IEEE MILCOM*, Washington, DC, October 2014.

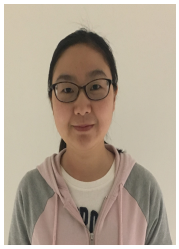
[12] O. Fatemieh, M. LeMay, and C. Gunter, "Reliable telemetry in white spaces using remote attestation," in *ACSAC*, Orlando, FL, 2011, pp. 323–332.

[13] Y. Hu and R. Zhang, "A spatiotemporal approach for secure crowdsourced radio environment map construction," *ACM TON*, vol. 28, no. 4, pp. 1790–1803, 2020.

[14] A. Achtzehn, J. Riihijärvi, and P. Mähönen, "Improving accuracy for tvws geolocation databases: Results from measurement-driven estimation approaches," in *IEEE DySPAN*, Mclean, VA, April 2014.

[15] B. Srinivasan, R. Duraiswami, and R. Murtugudde, "Efficient kriging for real-time spatio-temporal interpolation,"

in *AMS Program*, Atlanta, GA, January 2010.
[16] Y. Wu, D. Zhuang, A. Labbe, and L. Sun, "Inductive graph neural networks for spatiotemporal kriging," in *AAAI*, Virtual Conference, February 2021.
[17] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *ICLR*, BC, Canada, April 2018.
[18] W. Hamilton, P. Bajaj, M. Zitnik, D. Jurafsky, and J. Leskovec, "Embedding logical queries on knowledge graphs," in *NeurIPS*, Montreal, Canada, December 2018.
[19] X. Wang, H. Ji, C. Shi, B. Wang, Y. Ye, P. Cui, and P. Yu, "Heterogeneous graph attention network," in *WWW*, San Francisco, CA, May 2019.
[20] C. Phillips, M. Ton, D. Sicker, and D. Grunwald, "Practical radio environment mapping with geostatistics," in *IEEE DySPAN*, Bellevue, WA, October 2012.
[21] (2008) Splat! a terrestrial rf path analysis application for linux/unix. [Online]. Available: https://www.qsl.net/kd2bd/splat.html
[22] (2020) Us geological survey. [Online]. Available: https://tinyurl.com/zwkpk9h
[23] A. Min, K. Shin, and X. Hu, "Attack-tolerant distributed sensing for dynamic spectrum access networks," in *ICNP*, Princeton, NJ, October 2009, pp. 294–303.
[24] K. Thilina, K. Choi, N. Saquib, and E. Hossain, "Machine learning techniques for cooperative spectrum sensing in cognitive radio networks," *IEEE Journal on selected areas in communications*, vol. 31, no. 11, pp. 2209–2221, 2013.
[25] Y. Lu, P. Zhu, D. Wang, and M. Fattouche, "Machine learning techniques with probability vector for cooperative spectrum sensing in cognitive radio networks," in *IEEE WCNC*, Doha, Qatar, April 2016.
[26] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Pattern based anomalous user detection in cognitive radio networks," in *IEEE ICASSP*, Brisbane, Australia, April 2015.
[27] H. Wang and Y.-D. Yao, "Primary user boundary detection in cognitive radio networks estimated secondary user locations and impact of malicious secondary users," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4577–4588, 2018.
[28] B. Wang, K. Liu, and T. Clancy, "Evolutionary cooperative spectrum sensing game: how to collaborate?" *IEEE Transactions on Communications*, vol. 58, no. 3, pp. 890–900, March 2010.
[29] W. Zhang, Y. Yang, and K. Yeo, "Cluster-based cooperative spectrum sensing assignment strategy for heterogeneous cognitive radio network," *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2637–2647, 2014.
[30] L. Khalid and A. Anpalagan, "Adaptive assignment of heterogeneous users for group-based cooperative spectrum sensing," *IEEE Transactions on Wireless Communications*, vol. 15, no. 1, pp. 232–246, 2015.
[31] G. Yang, J. Wang, J. Luo, Y. Wen, H. Li, Q. Li, and S. Li, "Cooperative spectrum sensing in heterogeneous cognitive radio networks based on normalized energy detection," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 3, pp. 1452–1463, 2015.
[32] A. Celik, A. Alsharoa, and E. Kamal, "Hybrid energy harvesting-based cooperative spectrum sensing and access in heterogeneous cognitive radio networks," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, no. 1, pp. 37–48, 2017.

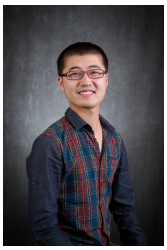


Yan Zhang received the B.S. in Information and Computing Science from Xi'an Jiaotong University, China, in 2014, the M.S. in Communication and Information System from Beijing Normal University, in 2017. Currently, she is a Ph.D. student in Computer Engineering from Arizona State University. Her research interest is about cyber security and privacy issues in mobile systems.

Dianqi Han received the B.S. in Information Security from University of Science and Technology of China, China, in 2010, the M.S. in Electrical and Computer Engineering from University of California, Davis, in 2015. Currently, he is a Ph.D. student in Computer Engineering from Arizona State University. His research interest is about indoor navigation, security and privacy issues in computer and networked systems.

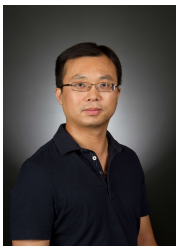
Ang Li received the B.E. in Network Engineering from Guangxi University, China, in 2010, the M.S. in Computer Science from Beihang University, China, in 2014. Currently, he is a Ph.D. student in Computer Engineering from Arizona State University. His research interest is about security and privacy in social networks, machine learning, wireless networks, and mobile computing.

Jiawei Li is a Ph.D. student in Computer Engineering at Arizona State University. He received the B.E. in Telecommunication Engineering from Nanjing University of Posts and Telecommunications at 2013. His research interest is on security and privacy issues in wireless network and wireless sensing.



Tao Li received a Ph.D. in Computer Engineering from Arizona State University in 2020, a M.S. in Computer Science & Technology from Xi'an Jiaotong University in 2015, and a B.E. in Software Engineering from Hangzhou Dianzi University in 2012. His primary research is on security and privacy issues in networked/mobile/distributed systems, smart sensing, and wireless networks. He is an Assistant Professor in the Department of Computer and Information Technology at Indiana University-Purdue University Indianapolis (IUPUI).

Rui Zhang received the B.E. degree in communication engineering and M.E. degree in communication and information system from the Huazhong University of Science and Technology in 2001 and 2005, respectively, and the Ph.D. degree in electrical engineering from Arizona State University in 2013. He was an Assistant Professor with the Department of Electrical Engineering at the University of Hawaii from 2013 to 2016. He has been an Assistant Professor with the Department of Computer and Information Sciences at the University of Delaware since 2016. His current research interests include network and distributed system security, wireless networking, and mobile computing. He received the NSF CAREER Award in 2017 and is a member of the IEEE.



Yanchao Zhang received the B.E. in Computer Science and Technology from Nanjing University of Posts and Telecommunications in 1999, the M.E. in Computer Science and Technology from Beijing University of Posts and Telecommunications in 2002, and the Ph.D. in Electrical and Computer Engineering from the University of Florida in 2006. He is an Professor in School of Electrical, Computer and Energy Engineering at Arizona State University. His primary research interests are network and distributed system security, wireless networking, and mobile computing. He is/was on the editorial boards of IEEE/ACM Transactions on Networking, IEEE Transactions on Mobile Computing, IEEE Wireless Communications, IEEE Transactions on Control of Network Systems, and IEEE Transactions on Vehicular Technology. He received the US NSF CAREER Award in 2009 and is an IEEE Fellow for contributions to wireless and mobile security. He also chaired the 2020 ARO Workshop on Autonomous and Proactive Defenses in Wireless Networks, the 2017 IEEE Conference on Communications and Network Security (CNS), the 2016 ARO Workshop on Trustworthy Human-Centric Social Networking, the 2015 NSF Workshop on Wireless Security, and the 2010 IEEE GLOBECOM Communication and Information System Security Symposium.