Abstract—Network anomaly detection in large scale sensor networks is a fundamental task in many Internet of Things (IoT) applications. Given an incomplete set of corrupted observations of the network matrices, the problem of network matrix recovery and anomaly detection can be formulated as a low-rank matrix completion problem with a fraction of the observed entries being corrupted by outliers. Although many centralized algorithms have been proposed to solve the low-rank matrix completion problem, they generally require the observations to be centrally available, which can incur many problems in practical applications, e.g., power budget constraints, single point failure and privacy concern. Recently, a decentralized nuclear-norm minimization-based algorithm was developed to solve the low-rank matrix completion problem in a mesh network. In this paper, we consider a two-tier network and propose a new decentralized approach based on the Riemannian optimization. Our proposed clustering and consensus sharing method achieves a balance between the performance guarantee and the computational cost: the proposed method distributes the computational burden over the agent nodes, while exhibits a recovery performance close to its centralized counterpart. In addition, our Riemannian optimization-based approach scales well with the size of the problem, hence it is more favoured for handling large data sets in IoT than the nuclear-norm minimization-based algorithm. Numerical simulations are performed to demonstrate the effectiveness of the proposed approach, which is able to solve problems of size $2000 \times 2000$ of rank 5 with 50 agent nodes in 10 seconds.

Index Terms—Internet of things, network anomaly detection, Riemannian optimization, decentralized approach

I. INTRODUCTION

With hundreds or even thousands of nodes coordinating simultaneously in Internet of Things (IoT) applications, the faults due to communication unreliability or malicious attacks will significantly degrade the system performance [1]-[3]. Network anomaly detection is crucial to guarantee resilient IoT solutions with fault-tolerance and recovery mechanisms. Real-world network can experience missing data and anomalies due to failure in measurement system and abrupt unusual changes, such as network security threats, denial of service attacks and flash crowds [4], [5]. To interpolate the missing data and detect anomalies in network matrices, many studies have proposed exploiting the low-rank structure in the data, e.g., traffic matrices [6], Internet delay matrices [7] and received signal strength (RSS) matrices [8]. The task of network matrix completion and anomaly detection has been formulated as a low-rank matrix completion problem with outliers among the observed entries [6], [8], [9]. Mathematically, the problem can be described as below. Suppose $X \in \mathbb{R}^{m \times n}$ is a low-rank network matrix with rank$(X) = r \ll m \leq n$, and $A \in \mathbb{R}^{m \times n}$ is a sparse matrix with $\|A\|_0 \leq s$. Consider a set $\Omega$ of the complete set of index pairs $\{(i, j) : i \in [m], j \in [n]\}$, the operator $P_\Omega(\cdot)$ represents the effect of missing measurements, i.e., $P_\Omega(X_{ij}) = X_{ij}$ if $(i, j) \in \Omega$ and $P_\Omega(X_{ij}) = 0$ otherwise. Given the corrupted measurements

$$P_\Omega(Y) = P_\Omega(X + A + N),$$

(1)

where $N \in \mathbb{R}^{m \times n}$ denotes the dense noise, the aim is to estimate both the network matrix $X$ and the anomalies $A$. One typical approach is to solve the following convex optimization $[8]–[11]$

$$\min_{X,A} \frac{1}{2} \|P_\Omega(Y - X - A)\|_F^2 + \lambda_1\|X\|_* + \lambda_2\|A\|_1,$$

(2)

where $\| \cdot \|_*$ and $\| \cdot \|_1$ represent the nuclear norm and the $l_1$ norm respectively, and $\lambda_1$, $\lambda_2$ are the regularization parameters.

It is noted that all the above works rely on a centralized method which requires the measurements $P_\Omega(Y)$ to be available at a fusion center. However, sending all raw measurements for centralized processing can be challenging due to the limited network capacity and the low energy constraint of the sensor nodes, hence impractical for IoT applications [12], [13]. In some cases, the agent nodes might prefer sharing only the result over the whole measurements due to privacy concern [14]. Furthermore, a centralized approach is not robust to single point (the fusion center) failure [15]. Recently, a decentralized approach for a mesh network was proposed in [16]. It separates the matrix demixing algorithm by leveraging an alternative characterization of the nuclear norm. Each node locally solves a low-rank and sparsity regularization algorithm iteratively based on its own raw measurements. In each iteration, every node shares a low dimensional matrix $((r \times n))$ with its neighbours, so that the final decentralized solution would be close to that obtained by the centralized approach.

In this paper, we focus on design of a new decentralized approach for a two-tier network to reduce the communication overhead and achieve much faster convergence. This will help mitigate the network congestion, and energy consumption of the sensor nodes, hence enhance the efficiency of the overall network. Our contribution is twofold. First, we propose a clustering method to reduce the communication overhead at
the sensor nodes. In the existing method [16], each sensor needs to share a low-dimensional matrix with its neighbors and locally compute an updated consensus at each iteration. We propose deploying cluster head nodes to form a two-tier network such that each node only shares a low-dimensional matrix with its cluster head at each iteration. Cluster heads will communicate directly with each other, and derive a consensus matrix before sharing it to the fellow nodes. In this way, each node only needs to receive an updated consensus matrix from the cluster head, instead of receiving and processing all the matrices from the neighbor nodes and computing a consensus by itself.

Our second contribution is a decentralized Riemannian algorithm for robust low-rank matrix completion. Recently, the centralized Riemannian optimization-based approach has been proposed to complete low rank matrix from its corrupted measurements, and shown to be very fast and efficient when the size of the problem is large [17], [18]. Here, we adopt the Riemannian optimization and propose the decentralized algorithm. More importantly, our consensus derivation is incorporated with the proposed clustering approach for fast convergence. By distributing the computational task over the agent nodes and sharing the consensus efficiently, the proposed decentralized approach runs much faster than its centralized counterpart without causing much performance degradation.

The rest of the paper is organized as follows. In Section II, we review the essential concepts of optimization on manifolds and the existing Riemannian optimization-based works on low-rank matrix completion. We introduce our decentralized approach in Section III, and present numerical experiments in Section IV. Conclusions are drawn in Section V.

Notation: We denote matrices and vectors by uppercase and lowercase boldface letters respectively. We represent a sequence of matrices by \( \mathbf{A}_1, \ldots, \mathbf{A}_L \). For a matrix \( \mathbf{A} \), \( \mathbf{A}_{pq} \) denotes the element on its \( p \)-th row and \( q \)-th column. \( \mathbf{A}^T \) represents the transpose of \( \mathbf{A} \). The Frobenius norm and nuclear norm of \( \mathbf{A} \) is denoted by \( \| \mathbf{A} \|_F = \sqrt{\text{tr}(\mathbf{A}^T \mathbf{A})} \) and \( \| \mathbf{A} \|_* = \sum_i \sigma_i(\mathbf{A}) \) respectively, where \( \sigma_i(\mathbf{A}) \) denotes the \( i \)-th singular value of \( \mathbf{A} \). The Frobenius norm \( \| \mathbf{A} \|_F \) is given by \( \| \mathbf{A} \|_1 = \sum_{p,q} |A_{pq}| \). The diagonal matrix diag(\( a \)) has the entries of \( a \) on its diagonal.

II. PRELIMINARIES

In this section, we briefly review the Riemannian optimization and its applications on robust low-rank matrix completion. Our approach will be based on the same tools.

Besides the nuclear norm relaxation, another way to solve the low-rank matrix completion problem is through Riemannian optimization. Suppose the rank \( r \) of the matrix of interest \( \mathbf{X} \) is known in advance (which makes sense in many real world applications, e.g., the real Internet traffic matrices have \( \text{rank}(\mathbf{X}) \in [5, 8] \) [4]), the problem of recovering \( \mathbf{X} \) from the measurements \( \mathbf{P}_\Omega(\mathbf{Y}) = \mathbf{P}_\Omega(\mathbf{X} + \mathbf{N}) \) can be written as [17]

\[
\min_{\mathbf{X} \in \mathcal{M}_r} \frac{1}{2} \| \mathbf{P}_\Omega(\mathbf{X} - \mathbf{Y}) \|_F^2, \tag{3}
\]

where \( \mathcal{M}_r \) is a smooth Riemannian manifold embedded in \( \mathbb{R}^{m \times n} \) of dimension \( r(m + n - r) \), i.e.,

\[
\mathcal{M}_r := \{ \mathbf{X} \in \mathbb{R}^{m \times n} : \text{rank}(\mathbf{X}) = r \}. \tag{4}
\]

Since the objective function is also smooth, (3) is a smooth optimization problem which can be solved by methods from Riemannian optimization. Vandereycken [17] proposed using a generalization of classical non-linear conjugate gradient (CG) to perform optimization on manifolds. The low-rank matrix manifold \( \mathcal{M}_r \) can be equivalently characterized as

\[
\mathcal{M}_r = \{ \mathbf{U} \Sigma \mathbf{V}^T : \mathbf{U} \in \mathbb{S}^r_n, \mathbf{V} \in \mathbb{S}^n_r, \Sigma = \text{diag}(\sigma_i), \sigma_1 \geq \cdots \geq \sigma_r > 0 \}, \tag{5}
\]

where \( \mathbb{S}^r_n \) is the Stiefel manifold of \( m \times r \) real, orthonormal matrices, and \( \sigma_i \) denotes the \( i \)-th singular value. With the closed form expressions on the Riemannian gradient and the mapping for tangent vectors from one tangent space to another, the Riemannian CG algorithm can efficiently solve the low-rank matrix completion problem involving very large data set [17].

In [18], the Riemannian optimization is extended to low-rank matrix completion where a fraction of the observations are corrupted by outliers, i.e., \( \mathcal{P}_\Omega(Y) = \mathcal{P}_\Omega(X + A) \). The formulation is given by

\[
(P0) \min_{\mathbf{X} \in \mathcal{M}_r} \| \mathcal{P}_\Omega(\mathbf{X} - \mathbf{Y}) \|_1.
\]

To handle the nonsmooth objective function of (P0), a smoothing technique with increasing accuracy is proposed. In particular, the objective function is replaced by

\[
\min_{\mathbf{X} \in \mathcal{M}_r} f_\delta(\mathbf{X}) := \sum_{(i,j) \in \Omega} \sqrt{\delta^2 + \| \mathbf{X}_{ij} - \mathbf{Y}_{ij} \|^2}.
\]

The idea is to solve \( f_\delta(\mathbf{X}) \) for decreasing values of \( \delta \), and for each instance of \( \delta \) the Riemannian CG algorithm is applied. In this paper, we aim to develop a decentralized algorithm for robust low-rank matrix completion based on the Riemannian optimization. The algorithm is expected to possess the following features: 1) affordable communication overhead at agent nodes; 2) each agent node should obtain an estimate of its local network matrix and anomalies, which are comparable with the solution of the centralized approach.

III. DECENTRALIZED RIEMANNIAN CLUSTER ALGORITHM

We consider \( L \) networked agents being able to perform local computations, and represent the set of agents as a set of nodes \( \mathcal{I} := \{1, \ldots, L\} \). Each agent \( i \in \mathcal{I} \) collects a few incomplete and corrupted rows of matrix \( \mathbf{Y} \in \mathbb{R}^{m \times n} \).

For convenience, we assume that the local data available to agent \( i \) is matrix \( \mathcal{P}_{\Omega_i}(\mathbf{Y}_i) \), where \( \mathbf{Y}_i \in \mathbb{R}^{d \times n} \), \( dL = m \), \( \mathbf{Y} = [\mathbf{Y}_1^T, \ldots, \mathbf{Y}_L^T]^T = \mathbf{X} + \mathbf{A} \), and \( \{\Omega_i\} \subseteq \Omega \) represents the set of index pairs observed by agent \( i \). In the network, a set \( \mathcal{K} \) of cluster head nodes (\( |\mathcal{K}| = K \)) are deployed such that each agent node is grouped into a specific cluster. We assume that each agent has a direct connection to its cluster head, and each
cluster head can communicate with the other cluster heads in a single-hop.

To derive the decentralized algorithm for (P0), we start from a slightly different formulation

\[
(P1) \min_{\mathbf{X} \in \mathcal{N}_1} \| \mathcal{P}_\Omega (\mathbf{X} - \mathbf{Y}) \|_1,
\]

where

\[
\mathcal{N}_1 := \{ (\mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T, \ldots, \mathbf{U}_L \Sigma_L \mathbf{V}_L^T) : \mathbf{U}_i \in \text{St}_d^d, \Sigma_i = \text{diag}(\sigma_{ij}), \sigma_1 \geq \cdots \geq \sigma_r > 0, \mathbf{V} \in \text{St}_r^n \}.
\]

By construction, every element in \( \mathcal{N}_1 \) is a rank-\( r \) matrix, which reveals \( \mathcal{N}_1 \subset \mathcal{M}_r \), and that the optimal solution of (P1) is a feasible solution of (P0). Since \( l_1 \)-norm is separable, (P1) can be equivalently expressed as

\[
\sum_{i=1}^L \min_{\mathbf{X}_i \in \mathcal{M}_r} \| \mathcal{P}_\Omega (\mathbf{X}_i - \mathbf{Y}_i) \|_1,
\]

s.t. \( \mathbf{V}_i = \mathbf{V}_j \) \( \forall i, j \in 1, \ldots, L \),

where

\[
\mathcal{M}_r := \{ \mathbf{U} \Sigma \mathbf{V}^T : \mathbf{U} \in \text{St}_d^d, \Sigma = \text{diag}(\sigma_{ij}), \sigma_1 \geq \cdots \geq \sigma_r > 0, \mathbf{V} \in \text{St}_r^n \}.
\]

The consensus constraint, \( \mathbf{V}_i = \mathbf{V}_j \) \( \forall i, j \in 1, \ldots, L \), is used to guarantee the equivalence. The problem can be further reduced if the consensus condition is omitted.

\[
(P2) \min_{\mathbf{X} \in \mathcal{N}_2} \| \mathcal{P}_\Omega (\mathbf{X} - \mathbf{Y}) \|_1,
\]

where

\[
\mathcal{N}_2 := \{ (\mathbf{U}_1 \Sigma_1 \mathbf{V}_1^T, \ldots, \mathbf{U}_L \Sigma_L \mathbf{V}_L^T) : \mathbf{U}_i \in \text{St}_d^d, \Sigma_i = \text{diag}(\sigma_{ij}), \sigma_1 \geq \cdots \geq \sigma_r > 0, \mathbf{V} \in \text{St}_r^n \}.
\]

Similarly, (P2) can be decomposed as

\[
\sum_{i=1}^L \min_{\mathbf{X}_i \in \mathcal{M}_r} \| \mathcal{P}_\Omega (\mathbf{X}_i - \mathbf{Y}_i) \|_1,
\]

where each agent independently solves the problem

\[
\min_{\mathbf{X}_i \in \mathcal{M}_r} \| \mathcal{P}_\Omega (\mathbf{X}_i - \mathbf{Y}_i) \|_1.
\]

If \( \text{rank}(\mathbf{X}_i) = \cdots = \text{rank}(\mathbf{X}_L) = \text{rank}(\mathbf{X}) = r \) (which is generally true for high-dimensional low-rank matrices with \( r \ll d \)), the optimal solution of (P0) will be a feasible solution of (P2). Based on the relationship of (P1) and (P2) with (P0), it is expected that a regularized combination of (P1) and (P2) can provide a good decentralized approximation of the centralized problem (P0). Thus, we propose the following decentralized problem formulation.

\[
(P3) \min_{\mathbf{X}_1, \ldots, \mathbf{X}_L \in \mathcal{M}_r} \sum_{i=1}^L \min_{\mathbf{X}_i \in \mathcal{M}_r} \| \mathcal{P}_\Omega (\mathbf{X}_i - \mathbf{Y}_i) \|_1 + \frac{\rho}{2} \sum_{i,j} d(V_i, V_j)^2,
\]

where \( \mathbf{X}_i = \mathbf{U}_i \Sigma_i \mathbf{V}_i^T \) and \( d(V_i, V_j) \) represents the Riemannian geodesic distance between \( V_i \) and \( V_j \). With \( \rho = 0 \), (P3) is equivalent to (P2). When \( \rho > 0 \), the penalty term on the Riemannian geodesic distance is used to guarantee that the consensus matrix \( \mathbf{V}_i \) of each agent is close to other consensus matrices. In this way, the consensus constraint of (P1) can be naturally incorporated in (P3).

**Decentralized Riemannian Cluster (DRC) Algorithm.** To implement (P3) on the network such that each agent can make both local computation and efficient consensus sharing, we need further adjustments by appealing to the property of the cluster heads. The full algorithm is stated in Algorithm 1. In the proposed approach, we need two nested loops. In the inner loop, each agent solves \( f_\delta(X_i) \) for a fixed \( \delta \) by using the Riemannian CG algorithm. The parameter \( \delta \) is then decreased gradually in the outer loop in order to increase the accuracy. This follows the smoothing technique as proposed in [18], while the difference is that it is now applied on each agent individually instead of the fusion center. However, the

---

**Algorithm 1 DRC algorithm per agent \( i \in \mathcal{I} \) and cluster head \( k \in \mathcal{K} \)**

**Input:** \( \mathbf{Y}_i, \Omega_i, \theta, \rho \)

**Initialize:** \( \mathbf{X}_i^{(0)}, \delta^{(0)}, \eta^{(0)} \)

for \( t = 1, 2, \ldots \) do

[S1] **Computation at agent \( i \) in cluster \( k \)**

if \( t > 1 \) then

Update \( \mathbf{X}_i^{(t-1)} \) based on \( \mathbf{V}_i^{(t-1)} \)

Consider the function \( \frac{\rho}{2} d_i^2 \)

Compute the gradient via the logarithmic mapping

\[
\nabla \| \mathbf{y} \|_2^2 = -\log(\mathbf{y})^T \frac{\mathbf{y}}{\| \mathbf{y} \|_2^2}
\]

Update \( \mathbf{V}_i^{(t-1)} \) by taking a gradient descent step with the projective retraction

\[
\mathbf{V}_i^{(t-1)} = R_{\mathbf{y}}(\mathbf{V}_i^{(t-1)} - \eta(t) \rho \nabla \| \mathbf{y} \|_2^2)
\]

where \( \eta(t) \) represents the stepsize

end if

Solve \( \mathbf{X}_i^{(t)} = \arg\min_{\mathbf{X}_i \in \mathcal{M}_r} f_\delta^{(t-1)}(\mathbf{X}_i) \) with

\[
f_\delta^{(t-1)}(\mathbf{X}_i) = \sum_{(p,q) \in \Omega_i} \sqrt{(\delta^{(t-1)})^2 + (\mathbf{X}_i^{(t)} - \mathbf{Y}_i^{(t)})}
\]

using Riemannian CG algorithm starting from \( \mathbf{X}_i^{(t-1)} \)

Pass the consensus \( \mathbf{V}_i^{(t)} \) to the cluster head \( k \)

[S2] **Computation at cluster head \( k \)**

Compute the Karcher mean of \( \mathcal{I}_k \) subspaces

\[
\mathbf{V}_k^{(t)} = \text{Karcher mean}(\mathbf{V}_i^{(t)}, i \in \mathcal{I}_k)
\]

Broadcast \( \mathbf{V}_k^{(t)} \) to all agents \( i \in \mathcal{I}_k \)

[S3] **Computation at cluster head \( k \)**

Compute the global Karcher mean

\[
\mathbf{V}_k^{(t)} = \text{Karcher mean}(\mathbf{V}_k^{(t)}, g \in \mathcal{K})
\]

[S4] **Update auxiliary variable**

\[\delta(t) = \delta^{(t-1)} \cdot \theta\]

end for

return \( \mathbf{X}_i \)
smoothing technique is not good enough since it can only handle the $l_1$ norm term of (P3). To satisfy the Riemannian geodesic distance constraint of (P3), each agent requires the $V_i$ information from all the other agents, which would be very communication and storage consuming and impractical for the network topology. Instead, we propose a suboptimal solution via efficient consensus sharing and computation through the network topology. Instead, we propose a suboptimal solution

\[
\text{Karcher mean}(V_1, V_2, \ldots) = \arg\min_{V_k \in S^k} \sum_i d(V_i, V_k)^2,
\]

based on the algorithm proposed in [20], [21]. Before the start of a new inner loop, the cluster heads pass the mean to their agents, where the starting point for the new inner loop is updated based on the mean. By moving the starting points towards the mean at each iteration in the outer loop, the solution can be gradually tuned to approach the geodesic distance constraint of (P3) (hence, the consensus constraint of (P1)). We note that the regularization parameter $\rho$ of (P3) is incorporated in the retraction step; higher (lower) penalty on the geodesic distance results in larger (smaller) scalar of the direction in the retraction step. Here, both the logarithmic mapping and the projective retraction have closed-form expressions [17], [22].

At each agent, the starting point of the algorithm, $X_i^{(0)}$, is chosen simply using the rank-$r$ SVD of $P_{[1]}(Y_i)$. We set $\delta^{(0)} = 1$, $\theta = 0.05$, $\rho = 1$ and $\eta^{(0)} = \eta^{(0)} = 0.5$ for good results in our numerical experiments. In real applications, these parameters need to be tuned by cross-validation. For both the inner and outer loop, the stopping criterion is based on the convergence condition or the maximum number of iterations. As in the centralized Riemannian optimization-based low-rank matrix completion algorithms [17], [18], the SVD-like representation is used throughout the algorithm for computational efficiency, i.e., $X_i^{(t)} = U_i^{(t)} \Sigma_i^{(t)} (V_i^{(t)})^T$.

**Remark 1** (Communication overhead). The transmission of a consensus matrix $V_i$ does not cost much overhead when the matrix is low-rank (i.e., $r$ is small). In addition, the consensus sharing only needs to be conducted in the outer loop, which makes the overall communication overhead independent of the number of iterations of the inner loop.

**Remark 2** (Comparison with related literature). In the proposed algorithm, the inner loop is solved by following the smoothing technique in the centralized Riemannian approach [18]. As shown in the simulations, simply executing the centralized approach on each agent individually fails to provide good recovery performance. In our work, an additional penalization on the geodesic distance is introduced in the formulation to guarantee a better performance. Efficient consensus sharing and computation method is also proposed to implement this additional constraint in a two-tier network. In [23], a Riemannian gossip algorithm is proposed for solving the matrix completion problem in a decentralized setting. Our study departs from it in the following aspects: [23] considers the matrix completion under the corruption of additive Gaussian noise and relies on the Frobenius norm in the objective function, while we focus on the situation where the observations are corrupted by anomalies and choose $l_1$ norm in the objective function. Furthermore, our network topology is different from that in [23].

We note that the proposed algorithm can be extended to the case where the cluster heads do not have direct communication with each other.

**IV. SIMULATIONS**

In this section, we demonstrate the performance of the proposed approach on synthetic numerical experiments. The synthetic network matrix $X$ is generated as proposed in [17], [18]: $X$ is the product of two matrices $U \in \mathbb{R}^{m \times r}$ and $V \in \mathbb{R}^{r \times n}$, i.e., $X = UV$, where $U$ and $V$ are with independent and identically distributed (i.i.d.) Gaussian entries. We then sample $0.8mn$ entries uniformly at random, which is equivalent to the situation of 20% missing data [16]. 5% of the observed data is selected uniformly at random to add the outliers, where each outlier is a realization of the random variable $A = S \cdot N$ with $S$ being a Bernoulli random variable (+1, −1) and $N$ being a Gaussian random variable with mean $\mu$ and variance $\sigma^2$. Suppose the estimated network matrix is $\hat{X}$, we employ the root mean squares error (RMSE) to measure the estimation performance, i.e.,

\[
\text{RMSE}(X, \hat{X}) = \sqrt{\frac{\|X - \hat{X}\|_F^2}{mn}}.
\]

We compare the proposed ‘Decentralized Riemannian cluster’ algorithm (Algorithm 1) with the following algorithms: ‘Centralized Riemannian’ algorithm, which is the centralized benchmark obtained by solving (P0) using the algorithm in [18]; ‘Decentralized no consensus’ algorithm, which is to solve (P2) distributively at each agent without the consensus sharing; and ‘Centralized robust PCA’ algorithm, which is another centralized benchmark by solving a sparsity-regularized nuclear-norm minimization problem via the alternating-direction method of multipliers (ADMM) [11]. For the ‘Decentralized Riemannian cluster’ algorithm, we set the maximum of iterations to 40 for the inner loop, and 15 for the outer loop. The convergence tolerance is set to $10^{-8}$ for all three Riemannian optimization-based algorithms. Regarding the ‘Centralized robust PCA’ algorithm, we made a modification to the starting condition. Since the algorithm in [11] was not implemented for missing data, we assume...
Manopt toolbox [24] is used to perform the CG method on matrix, where the corrupted partial observations \( \mathcal{P} \) using MATLAB R2016b on a full observation on \( L \) dual-core Intel i7 processor, and the number of agent nodes is set to \( K = 5 \). All experiments were run on a 1000×1000 matrix, with still 80% observed data and 5% outliers in the observed entries when \( L=25 \), \( K=5 \), \( \mu = 1 \) and \( \sigma = 1 \).

The number of cluster head nodes is set to \( K = 5 \), \( \mu = 1 \) and \( \sigma = 1 \) are used to create the outliers. The RMSE results with respect to the running time is depicted in Fig. 1a. It takes the ‘Centralized Riemannian’ algorithm approximately 7 seconds to reach an RMSE of \( 10^{-7} \), while the ‘Decentralized no consensus’ algorithm reaches convergence faster (2 seconds) but with a worse RMSE of \( 10^{-2} \). On the other hand, the proposed approach demonstrates a good balance between convergence rate and accuracy: an RMSE of about \( 3 \times 10^{-5} \) is achieved in 3 seconds. We present the result of the ‘Centralized robust PCA’ algorithm separately (Fig. 1b) since its convergence is too slow. In Fig. 2, the average RMSE results with respect to the iteration index taking over 10 random instances is presented. Here, the iteration index represent the overall iteration, which includes both inner and outer loop for Riemannian optimization-based algorithms. Although the ‘Centralized Riemannian’ algorithm is able to reach convergence with less number of iterations, it takes longer time on completing each iteration than the decentralized algorithms (as in Fig. 1).

The ‘Centralized robust PCA’ algorithm can estimate both the network matrix and the outliers. As for the three Riemannian optimization-based algorithms, we add a simple thresholding step to extract the estimated outliers. In particular, from an estimation \( \hat{X} \), we implement threshold \( \mathcal{P}_\Omega(\hat{X}) - \mathcal{P}_\Omega(\mathbf{Y}) \), which identifies the indexes of the estimated outliers if the corresponding entries of \( \mathcal{P}_\Omega(\hat{X}) - \mathcal{P}_\Omega(\mathbf{Y}) \) have a magnitude above the tolerance level tol. In this experiment, we set \( \text{tol} = 10^{-4} \). The detection of an outlier is claimed to be successful if its index can be correctly detected. In Table I, the average anomaly detection results for each algorithm are listed. All algorithms can accurately detect the anomalies at low false positive rates.

We then conduct the same experiment on a larger 2000×2000 matrix, with still 80% observed data and 5% outliers. The number of agent nodes is set to \( L = 50 \), and the number of cluster head is remained as \( K = 5 \). Fig. 3 shows the RMSE results versus the running time. The ‘Centralized robust PCA’ algorithm is omitted due to its slow convergence rate. The proposed decentralized Riemannian cluster approach scales very well with the size of the problem; it is able to solve the problem of size 2000×2000 in 10 seconds with an RMSE of \( 3 \times 10^{-5} \).

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>True positive</th>
<th>False positive</th>
</tr>
</thead>
<tbody>
<tr>
<td>Centralized Riemannian</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Centralized robust PCA</td>
<td>0.9935</td>
<td>0.0065</td>
</tr>
<tr>
<td>Decentralized no consensus</td>
<td>0.9999</td>
<td>2.98×10^{-4}</td>
</tr>
<tr>
<td>Decentralized Riemannian cluster</td>
<td>0.9999</td>
<td>1.37×10^{-4}</td>
</tr>
</tbody>
</table>

V. CONCLUSION

We have proposed a decentralized network anomaly detection method based on Riemannian optimization. The computation task is distributed among a number of agents and...
cluster heads. Consensus sharing and computation can be processed efficiently through the proposed approach. Numerical experiments demonstrate the superior performance of the proposed approach on different benchmarks. The decentralized method is robust and fast in detecting network anomalies, e.g., abnormal traffic or abnormal RSSI. In future work, we will consider improving the proposed decentralized Riemannian cluster approach with adaptive rank estimation and better dense noise tolerance, and testing the algorithm on real-world datasets.

Fig. 2. Average RMSE versus iteration index: low-rank matrix completion of rank-5 $800 \times 800$ matrix from 80% data and with 5% outliers in the observed entries when $L=50$, $K=5$, $\mu = 1$ and $\sigma = 1$.

Fig. 3. RMSE versus running time: low-rank matrix completion of rank-5 $2000 \times 2000$ matrix from 80% data and with 5% outliers in the observed entries when $L=50$, $K=5$, $\mu = 1$ and $\sigma = 1$.

REFERENCES


