A Novel Reverse Random Hyperplane Projection Scheme and Its Effect on Mining Sensor Streams

Antonios Skevis
School of ECE
Technical University of Crete
Chania, Greece
askevis@tuc.gr

George Klioumis
School of ECE
Technical University of Crete
Chania, Greece
gklioumis@tuc.gr

Nikos Giatrakos
School of ECE
Technical University of Crete
Chania, Greece
ngiatrakos@tuc.gr

***Abstract*—In this work we introduce a novel, reversible data summarization technique, namely the Reverse Random Hyperplane Projection (RRHP) scheme. RRHP is particularly useful in Wireless Sensor Network (WSN) settings because it enables individual sensors to compress their local data streams before transmitting them across the WSN. In that, RRHP saves communication and, thus, the residual energy of battery-powered sensors. Then, when the compressed sensor data streams reach a base station, the reversibility property of RRHP can be used to regain approximations of the original sensor streams to perform all kinds of data mining tasks. We provide formal theoretic guarantees on how RRHP directly trades the amount of compression for the approximation of original sensor streams' desired properties. We experimentally prove that RRHP is useful for performing various kinds of data mining tasks, over sensor data streams, by dramatically reducing the amount of communicated data, simultaneously achieving high accuracy.**

***Index Terms*—sensor networks, data streams, compression, data mining, locality sensitive hashing**

I. Introduction

Reversible data summaries [1]–[6] have the property of reducing the dimensionality of an initial vector u_i, from a ω-dimensional origin space O^ω, to a target vector v_i in an d-dimensional space T^d with $d \ll \omega$, simultaneously retaining the property of the reverse transformation. That is, given the transformed representation of $v_i \in T^d$, one can get back a vector $\hat{u_i} \in O^\omega$ with some quality guarantees on the relative distances, i.e., $distance(u_i, u_j) \approx distance(\hat{u_i}, \hat{u_j})$ for some distance metric.

The reversibility property is particularly useful in Wireless Sensor Networks (WSNs). This is because it enables individual sensors continuously compress their data streams, partitioned over windows, communicate them in a multi-hop fashion in the WSN and regain an approximation of the original windowed data stream, per sensor, at a base station. On the one hand, the importance of the compression property $O^\omega \rightarrow T^d$ in WSNs lies in the fact that communication is by far the biggest culprit in energy drain for battery powered sensors (motes) [7], [8]. Via compression, individual motes transmit less data and, therefore, the sensor network's lifetime is prolonged. On the other hand, the importance of the reverse transformation $T^d \rightarrow O^\omega$ is due to the fact that approximations of the original sensor measurements can be used to perform a variety of data mining tasks after reaching the base station.

The Random Hyperplane Plane (RHP) projection, Locality Sensitive Hashing (LSH) scheme [9], has been used in prior work [7], [8] for outlier detection [10], [11]. The TACO and Omnibus frameworks [7], [8] are the first in the literature that proposed a RHP-based outlier detection scheme with the unique advantage, among all other alternative techniques [10], [11], to straightforwardly trade bandwidth consumption for outlier detection accuracy.

However effective the bare RHP has been proven in outlier detection over WSNs, it is restricted to that and only that particular task. The reason for this limitation is that the bare RHP scheme converts original windows of sensor measurements from an ω-dimensional real space $\mathbb{R}^\omega$, to a bitmap representation in the $\{0,1\}^d$ Hamming space. The outlier detection process is performed judging sensor measurement similarity based on these bitmaps. But the bitmaps in $\{0,1\}^d$ cannot be used for anything else, because most meaningful mining tasks operate on non-binary features.

Other reversible data summaries do exist in the literature [12]–[16]. But given the distributed setting and the limited processing power and memory capacity of the motes, not all reversible data summaries are applicable in WSN settings due to high time and memory complexity or due to requiring first all data to be centrally available. Established and prominent reversible summaries such as the Discrete (and the Fast) Fourier Transform (DFT) [3], the Discrete Wavelet Transform (DWT) [4], the Discrete Cosine Transform (DCT) [1] or the Piecewise Aggregate Approximation (PAA) [2] have linear or quasi-linear time and space complexity, but also loose deterministic quality guarantees based on Parseval's theorem [17].

In this work, we fill this gap. We propose a new Reverse Random Hyperplane Projection (RRHP) scheme which enables sensors to communicate compressed bitmaps in the WSN to save communication and gives to the base station the ability to reverse these bitmaps as approximations of the original sensor measurements, i.e. from $\{0,1\}^d$ to $\mathbb{R}^\omega$, with predefined, probabilistic, tight quality guarantees on the relative distances between sensor data. Thus, any data mining task can be accurately performed on the reversed representations.

This work was supported by the EU project EVENFLOW under Horizon Europe agreement No. 101070430.

II. The New Reversible Data Summary Scheme

A. Background on the Random Hyperplane Projection Scheme

LSH [9] schemes have the property of compression via dimensionality reduction while preserving similarity between vectors, based on some distance metric. The Random Hyperplane Projection (RHP) LSH scheme is the basis of our reversible version. RHP is formally presented in Theorem 1.

Theorem 1. *[Random Hyperplane Projection [9]] Assume we are given a collection of vectors defined on an ω-dimensional space. We choose a family of hash functions as follows: We produce a spherically symmetric random vector r of unit length from this ω-dimensional space. We define a hash function h_r as:*

$$h_r(u_i) = \begin{cases} 1 & ,if\ r \cdot u_i \geq 0 \\ 0 & ,if\ r \cdot u_i < 0 \end{cases}$$

For any two vectors $u_i, u_j \in \mathbb{R}^\omega$ and their angle $\theta(u_i, u_j)$:

$$P = P[h_r(u_i) = h_r(u_j)] = 1 - \frac{\theta(u_i, u_j)}{\pi} \tag{1}$$

□

Equation 1 can be rewritten as:

$$\theta(u_i, u_j) = \pi \cdot (1 - P) \tag{2}$$

Note that Equation 2 expresses angle similarity as the product of the potential range of the angle between the two vectors (π), with the probability of equality in the result of the hash function application (P).

Thus, after repeating a stochastic procedure using d random vectors r and, thus d hash functions, we have:

$$u_i \cdot R = \xi_i \overset{h_{r_1}, h_{r_2}, \cdots h_{r_d}}{\longrightarrow} X_{u_i} \in \{0,1\}^d$$

where $R \in \mathbb{R}^{\omega \times d}$ a matrix composed of the d random vectors. The $u_i \in \mathbb{R}^\omega$ is multiplied by R producing a vector ξ_i and, then, the d hash functions are applied on ξ_i transforming it into a bitmap $X_{u_i} \in \{0,1\}^d$ depending on whether the respective coordinate is positive or not, based on Theorem 1. The final embodiment in the hamming cube for $u_i, u_j \in \mathbb{R}^\omega$ results in [7], [8]:

$$D_h(X_{u_i}, X_{u_j}) = d \cdot (1 - P) \tag{3}$$

where $D_h(X_{u_i}, X_{u_j}) = \sum_{\ell=1}^{d} |X_{u_i\ell} - X_{u_j\ell}|$ is the hamming distance of the respective bitmaps. Hence, from Equalities 2 and 3, we finally derive that, on expectation:

$$\frac{\theta(u_i, u_j)}{\pi} = \frac{D_h(X_{u_i}, X_{u_j})}{d} \tag{4}$$

Based on Equation 4, we can see that the angle and, thus, the cosine similarity distance is preserved between the original vectors in $\mathbb{R}^\omega$ and the produced bitmaps in $\{0,1\}^d$. Later on in Section II-D we are going to elaborate on how Equation 4 can accommodate other similarity metrics as well.

TABLE I: RRHP supported similarity metrics. Vectors $u_i, u_j^j \in \mathbb{R}^\omega$. $E(.)$, σ, cov refer to the mean, standard deviation and covariance. u_i', u_j' refer to normalized versions of u_i, u_j.

Similarity Metric	Calculation of Similarity
Cosine/Angle Similarity	$cos(\theta(u_i, u_j)) = \frac{u_i \cdot u_j}{\|\|u_i\|\| \cdot \|\|u_j\|\|}$ $\Leftrightarrow \theta(u_i, u_j) = arccos \frac{u_i \cdot u_j}{\|\|u_i\|\| \cdot \|\|u_j\|\|}$
Correlation Coefficient	$corr(u_i, u_j) = \frac{cov(u_i, u_j)}{\sigma_{u_i} \sigma_{u_j}} =$ $= \frac{E(u_i u_j) - E(u_i)E(u_j)}{\sqrt{E(u_i^2) - E^2(u_i)}\sqrt{E(u_j^2) - E^2(u_j)}}$
Euclidean Distance	$dist(u_i', u_j') = \|\|u_i' - u_j'\|\|$

B. The New Reverse Random Hyperplane Projection Scheme

Note that we consider a setting where $w < d$. This may seem counter intuitive given the discussion about compression and reversible data summaries in the intro, but it is not. The reason is that when we refer to compression in this work, we do not simply refer to the number of values between the original $\mathbb{R}^\omega$ and the target $\{0,1\}^d$ but, instead, we refer to the size of values. To understand why, consider the following simple example of a vector $u_i \in R^\omega$ consisting of 16 double precision numbers. The size of its values (for 8 bytes per number of double precision) is 1024 bits. Therefore, we can set, for instance, $d = 128 > 16 = \omega$ and compress original values to bitmaps with a compression ratio of 8.

Having clarified that, we progressively introduce the newly proposed Reverse Random Hyperplane Projection (RRHP) scheme in the following propositions.

Proposition 1. *For $R \in \mathbb{R}^{\omega \times d}$, $\omega < d$, composed of spherically symmetric random column vectors of unit length, the right inverse $R^\dagger \in \mathbb{R}^{d \times \omega}$ exists almost surely and $R \cdot R^\dagger = I_\omega$.*

Proof. Omitted due to space considerations. □

Proposition 2. *For sufficiently large ω, the spherically symmetric random vectors of unit length in $R \in \mathbb{R}^{\omega \times d}$, $\omega < d$, are approximately orthogonal and, thus, $R^\dagger \cdot R \approx I_d$.*

Proof. Omitted due to space considerations. □

Proposition 3. *For $X_{u_i}, X_{u_j} \in \{0,1\}^d$ produced from $u_i, u_j \in \mathbb{R}^\omega$ via the RHP scheme, and for vectors $\hat{u_i}, \hat{u_j} \in \mathbb{R}^\omega$ computed by $X_{u_i} \cdot R^\dagger = \hat{u_i}$ and $X_{u_j} \cdot R^\dagger = \hat{u_j}$, due to Proposition 2, $\hat{u_i} \cdot R = X_{u_i}$, $\hat{u_j} \cdot R = X_{u_j}$ is also an application of the RHP scheme. Therefore:*

$$\frac{\theta(\hat{u_i}, \hat{u_j})}{\pi} = \frac{D_h(X_{u_i}, X_{u_j})}{d} \tag{5}$$

Proof. Omitted due to space considerations. □

The multiplication leads directly to bitmaps, having started the computation from $X_{u_i} \cdot R^\dagger = \hat{u_i}$ and $X_{u_j} \cdot R^\dagger = \hat{u_j}$. These bitmaps are expectedly identical to those of u_i and u_j; notice that based on the computation X_{u_i}, X_{u_j} (without the hat (̂) notation on $_{u_i}$, $_{u_j}$) are on the right hand side of Equations 4 and 5. Finally, based on the above discussion, we formally present our Reverse Random Hyperplane Projection scheme.

Theorem 2. *[New Reverse Random Hyperplane Projection] Given $R \in \mathbb{R}^{\omega \times d}$ composed of spherically symmetric random column vectors of unit length and for $\omega < d$, with sufficiently large ω, from Equation 4 and Equation 5 we expectedly have:*

$$\frac{\theta(\hat{u}_i, \hat{u}_j)}{\pi} = \frac{\theta(u_i, u_j)}{\pi} \tag{6}$$

□

Our Reverse RHP scheme is composed of two phases:
Compression Phase: A vector $u_i \in \mathbb{R}^{\omega}$ is compressed by multiplying it with the $R \in \mathbb{R}^{\omega \times d}$ matrix:

$$u_i \cdot R = \xi_i \overset{h_{r_1}, h_{r_2}, \cdots h_{r_d}}{\longrightarrow} X_{u_i}$$

Reverse/Decompression Phase: The produced bitmap is multiplied by the right inverse of R deriving an estimation vector $\hat{u_i}$ of the original u_i:

$$\hat{u_i} = X_{u_i} \cdot R^{\dagger}$$

For any pair of vectors $u_i, u_j \in \mathbb{R}^{\omega}$ that have been compressed and decompressed in the above phases, their angle similarity is preserved as per Theorem 2. Obviously, RRHP has time and space complexity that is linear to ω and d.

C. On the Quality Guarantees of the New Scheme

RRHP provides probabilistic guarantees and directly trades the size of the constructed bitmaps d to the quality of the approximation of $\frac{\theta(u_i,u_j)}{\pi}$ in $\frac{\theta(\hat{u}_i,\hat{u}_j)}{\pi}$. The following theorem prescribes the way the size of the bitmaps d should be set.

Theorem 3. *To estimate $\frac{\theta(u_i,u_j)}{\pi}$ with precision ϵ and probability at least $1-\delta$ ($0 < \epsilon, \delta < 1$) using Equation 6, the bitmap size d should be set to $O(\ell og(4/\delta)/(2\epsilon^2))$ length.*

Proof. Omitted due to space considerations. □

The lower the allowed error ϵ, the greater the bitmap size d should be set, for fixed probability δ.

D. A Note on Distance Metrics and Multiple Dimensions

RRHP is not limited to the Angle Similarity. Table I presents some of relevant, RRHP- supported similarity metrics including the *Cosine Similarity*, the *Correlation Coefficient* and the *Euclidean Distance*. All these measures are supported commutatively via their relationship with Cosine and Angle similarity. Please see [18] for further details. Finally, one can use the new RRHP on matrices of multiple features instead of a single feature, preserving their similarity [8].

III. RRHP for Mining Sensor Data Streams

We outline the application of RRHP over a WSN setting composed of sensor nodes (also termed motes) organized in a clustered network architecture [19] (Figure 1). Nonetheless, RRHP can be applied in any hierarchical WSN such as tree-like networks [20]. The goal is to use RRHP to continuously perform one or more data mining tasks (clustering, classification, regression) over disjoint windows (tumbles) of ω measurements, as time passes and new such measurements are obtained. The way RRHP is applied on each window is not affected by the data mining task.

TABLE II: RRHP Performance vs ω for Compression Ratio 8

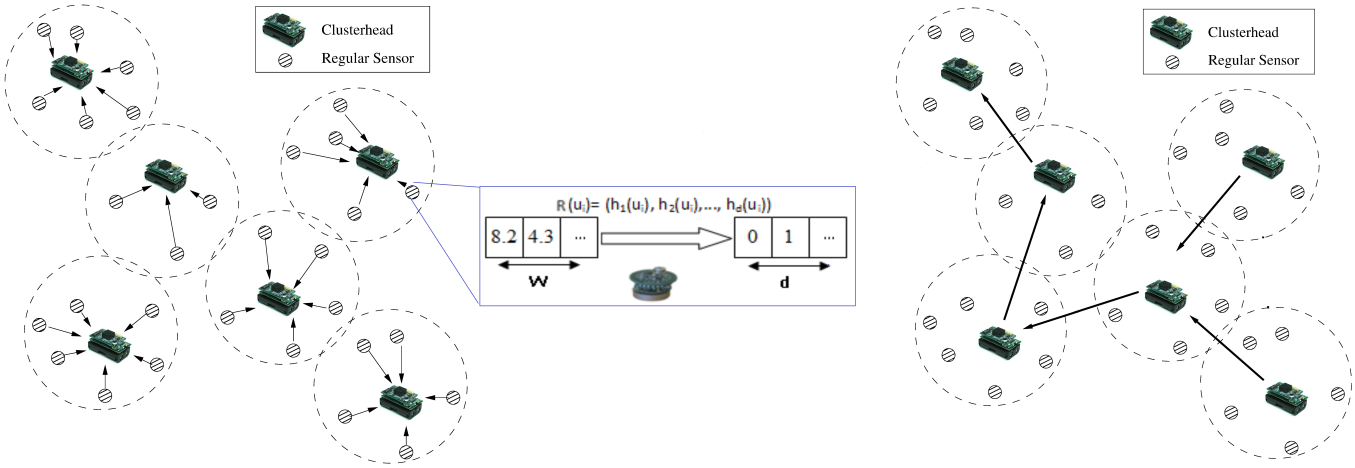
Window $\omega =$	16	32	64	128	Average per Metric
Clustering	0.95	0.96	0.975	1.0	ARI
Regression	0.12	0.12	0.12	0.07	RMSE
SVM/Neural Net	0.92	0.97	0.98	0.99	Accuracy
KNN	0.83	0.96	0.98	0.99	Accuracy

RRHP at the Sensor Level: Each one of the motes $S_1, S_2, \cdots, S_N$ in the WSN collects a window (tumble) of ω local measurements obtained from their operational realm, where ω is specified by the application. For a vector $u_i \in R^{\omega}$ formed by the latest ω measurements of mote S_i, S_i applies the Compression Phase as described in Section II-B. S_i compresses u_i producing a bitmap of $d \ll \omega \cdot B(u_{ik})$ bits, where $B(u_{ik})$ is the size of the binary representation of the k-th measurement. d is set by the application based on Theorem 3.

The matrix R should be common among all motes, but can change between windows. For $S_1, S_2, \cdots, S_N$ to be able to independently compute the common matrices R per window and apply the Compression Phase of RRHP, the application must broadcast a common $seed$ parameter at the beginning of the monitoring operation. Then, each mote S_i is able to produce the same d random vectors r in the R matrix used for each window, determine the d signs of $r \cdot u_i$ inner products as devised in the Compression Phase of Section II-B and come up with a bitmap $X_{u_i} \in [0,1]^d$ for the current u_i window.
Intracluster Communication: Compressed bitmaps of all motes are then transmitted to the clusterhead that each S_i belongs (Figure 1a). Clusterheads receive the measurements of multiple motes (dotted circles in Figure 1) of the same cluster. If motes transmit their original, uncompressed measurements, the transmission time is greater. Therefore, the probability of message collisions and the need for retransmissions increases. By applying the Compression Phase of RRHP we not only reduce the amount of the communicated data, but we also reduce the probability of retransmissions (see Section IV).
Inter-cluster Communication: Setting each clusterhead as the starting node and the base station as the destination, the intercluster communication is modeled as a shortest path problem. The compressed bitmaps are exchanged between clusterheads participating in the shortest path (Figure 1b) [7], [8], computed based on geographic aware routing [21].
RRHP at the Base Station: The base station uses the said common $seed$ to produce the R matrix. Based on R, the base station performs the Decompression Phase of the RRHP scheme (Section II-B –not shown in Figure 1) on each bitmap, obtaining approximations of the original vectors with their preserved distance and uses them as the feature vectors to perform the involved data mining task(s).

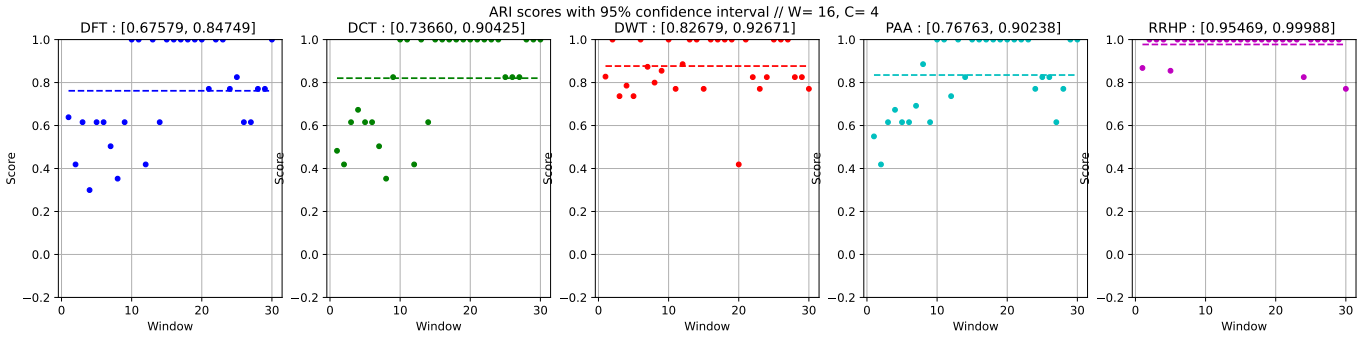
IV. Experimental Evaluation

Datasets. We utilize 2 real-world, publicly available datasets. The Intel Lab Data (https://db.csail.mit.edu/labdata/labdata.html) also used in [7], [8] with 48 sensors, and the Pump

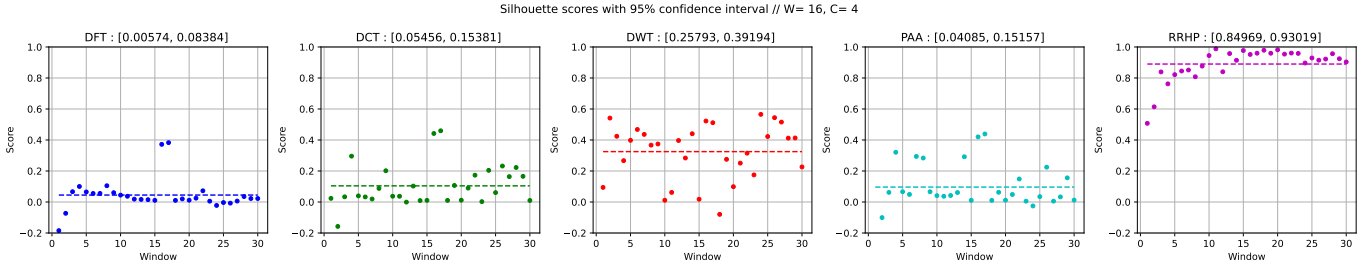


(a) RRHP Compression Phase at each mote and Intracluster Communication (b) Intercluster Communication

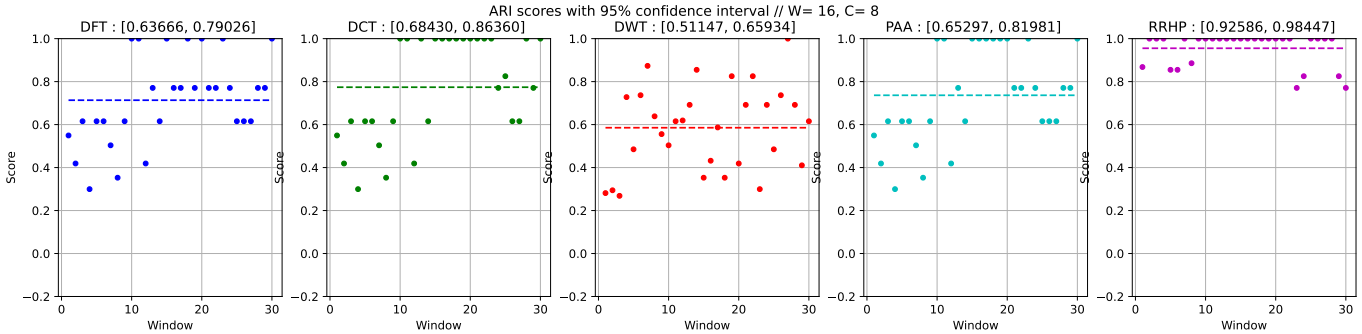
Fig. 1: Main Stages of the RRHP Application



(a) ARI Comparison for Compression Ratio C = 4



(b) Silhouette Score Comparison for Compression Ratio C=4



(c) ARI Comparison for Compression Ratio C=8

Fig. 2: DBSCAN Comparative Analysis ARI & Silhouette Scores (Vertical – The higher the better) vs Window Index (Horizontal) for $\omega = 16$ with compression ratio C=4 and C=8. At the top of each subplot 95% Confidence Intervals.

Sensor Data (https://www.kaggle.com/datasets/nphantawee/pump-sensor-data) with 50 motes measuring pump pressure. From the Intel Lab Data, we utilize the challenging (due to abrupt fluctuations) case of light measurements.

Data Mining Tasks. We build scenarios for a variety of data mining tasks carried out as per Section III. We use the unlabeled Intel data for DBSCAN clustering. The labeled Pump data (NORMAL, BROKEN, RECOVERING classes) are used for regression and classification tasks.

Candidates. We compare the new RRHP scheme against DFT [3], [22], DCT [1], DWT [4] and PAA [2]. For these comparisons, we develop Python code in Scikit Learn 1.5.1, Scipy 1.14.0, PyWavelets 1.6.0 and pyts 0.13.0. For the communication and energy savings of RRHP we develop simulations in TOSSIM [23] operating as described in Section III.

Preprocessing. To set the hyper-parameters for each mining algorithm, we analyzed the raw datasets, partitioned over the specified normalized windows each time, in an offline fashion without any compression, using grid search. Based on this, for DBSCAN $eps = 0.3$, for KNN $k = 4$. The default values for the rest of the hyper-parameters in the aforementioned Python libraries, were used. Our Neural Net includes 2 LSTM (42 neurons each) + 2 Dense Layers (32 – 3 neurons) trained for 50 epochs, with each window ω of each sensor being a batch.

Metrics. To judge the accuracy of each reversible data summary in each task versus performing the same task on the raw sensor data, we use the following metrics. For clustering, we use the Adjusted Rand Index (`ARI`) to quantify the similarity between the clusters formed upon using the raw data, versus the clusters formed upon using the reversible data summaries. We also use the `Silhouette` Score for measuring point distances within each and between clusters. We use the Root Mean Square Error (`RMSE`) for Linear Regression. In classification (KNN, SVM, Neural Net) we use the `Accuracy` score, i.e., percentage of instances each approach assigned the correct label/class per window.

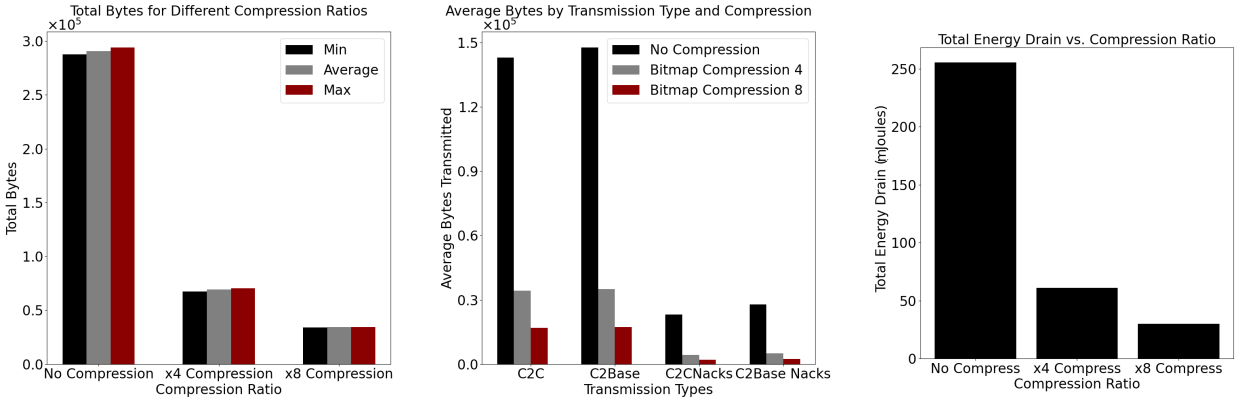
RRHP Performance across Mining Tasks. Due to Theorem 2, RRHP becomes more accurate for larger ω values (note that the theorem holds for sufficiently high ω). Additionally, due to Theorem 3, the higher the bitmap absolute size d, the greater the performance of RRHP. Instead of giving the d parameter in each experiment, we provide the compression ratio. That is the ratio between the binary representation of the raw window ω, in each experiment, over the bitmap size d. We utilize compression ratios of 4 and 8. The fact that high ω and d values favor the new RRHP scheme is experimentally validated. In Table II, we show that, for RRHP with a compression ratio of 8, all the performance metrics per data mining task receive their best values for $\omega = 128$, the highest of the cited windows. The RRHP performance is always high across mining tasks and respective metrics and progressively further improves upon increasing the window size from $\omega = 16$ to $\omega = 128$ (Table II). To stress test RRHP against other competitors, we next set a default $\omega = 16$.

Comparative Analysis. In Figure 2a and Figure 2c, we compare RRHP against DFT, DCT, DWT and PAA, indicatively, in a clustering task. We plot performance results of clustering, in terms of the ARI score for all candidates. RRHP clearly outperforms all other candidates with, on average, 20% improvement in ARI for compression ratios 4 and 8. Simultaneously, Figure 2b shows RRHP's 60% average improvement in Silhouette score versus the second best approach, indicatively for a compression ratio of 4. This performance comes with the tightest 95% confidence intervals (top of each subplot) among all candidates.

RRHP Simulation, Communication & Energy Savings. In Figure 3, we plot the communication and energy savings achieved by RRHP in our TOSSIM simulations. We utilized a clustered WSN architecture [19] composed of 4 sensor clusters with the same number of motes. Data are communicated in the network as described in Section III. We use the default packet size of TOSSIM equal to 29 bytes. In Figure 3(a) and Figure 3(c), we can see that for compression ratios of 4 and 8, both the (min, max, average) number of communicated bytes in the network and the consumed energy are reduced by 5 and 10 times, respectively, exceeding RRHP theoretic benefits (of 4 and 8). This is because RRHP reduces the size of messages and, consequently, the transmission time. Therefore, as mentioned in Section III, RRHP reduces the probability of message collisions in the intracluster (multiple motes send to their clusterhead – C2C communication) and intercluster phases (C2Base station communication). This fact is validated in Figure 3(b) which provides the transmitted bits categorization. C2C and C2Base communication keeps the ratio of 4 and 8. C2C Nacks (Negative acknowledgements) and C2Base Nacks are much higher when no compression is applied (black bars). Therefore, when these Nacks add up to the overall RRHP communication, they yield 5 instead of 4 times and 10 instead of 8 times reduced communication and lower energy for compression ratios of 4 and 8, correspondingly.

V. Related Work

Outlier detection is the most studied mining task [7], [8], [10], [11], [24] over WSNs. Approaches for performing broader data mining tasks over sensor data are surveyed in [25] with techniques tailored for specific sensor data mining tasks. RRHP differs by not being tailored to a specific mining task. Besides, it is orthogonal to such techniques which can use RRHP to compress transmitted data and decompress for mining sensor streams. The candidates we used in our experimental evaluation [1]–[5] are lightweight, reversible, time series data compression techniques, falling in the non data-adaptive category [26], as RRHP does. RRHP provides tight probabilistic error bounds, while the rest of the candidates rely on looser deterministic error bounds based on Parseval's theorem [17]. Plato [27] provides tight deterministic bounds only for specific queries over historical, centrally available sensor data, instead of mining tasks. Hung et al [16] compares another category, that of model based sensor data compression techniques. Some recently proposed time series compression techniques are not applicable on sensor networks due to either requiring first all data to be centralized [13]–[15], [28]–[31] or



(a) Total Comm. Cost (b) Comm. Cost Categories (c) Energy Consumption

Fig. 3: RRHP Communication Savings in TOSSIM Simulation. Sensors equally assgined to 4 WSN clusters. $\omega = 16$

due to high time/space complexity [12], [16]. RRHP is based on the RHP LSH [9] scheme. Wang et al [32] provides a comprehensive survey on relevant techniques.

VI. Conclusions

We introduced a novel Reversible Random Hyperplane Projection (RRHP) LSH scheme and presented its probabilistic quality guarantees on preserving relative vector distances, based on popular similarity metrics. Our experiments showed that RRHP can carry out data mining tasks with excellent accuracy, also exceeding its theoretically computed communication savings significantly. Our future work is on extending RRHP to matrices of multiple features, to different sliding window definitions and on representative bitmap election, to further reduce communication.

References

[1] N. Ahmed, T. R. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Trans. Computers*, vol. 23, no. 1, pp. 90–93, 1974.

[2] E. J. Keogh, K. Chakrabarti, M. J. Pazzani, and S. Mehrotra, "Dimensionality reduction for fast similarity search in large time series databases," *Knowl. Inf. Syst.*, vol. 3, no. 3, pp. 263–286, 2001.

[3] Y. Zhu and D. Shasha, "Statstream: statistical monitoring of thousands of data streams in real time," in *VLDB*, 2002.

[4] P. Chaovalit, A. Gangopadhyay, G. Karabatis, and Z. Chen, "Discrete wavelet transform-based time series analysis and mining," *ACM Comput. Surv.*, vol. 43, no. 2, pp. 6:1–6:37, 2011.

[5] A. Deligiannakis, Y. Kotidis, and N. Roussopoulos, "Compressing historical information in sensor networks," in *SIGMOD*, 2004.

[6] D. Shasha and Y. Zhu, *High Performance Discovery In Time Series: Techniques And Case Studies (Monographs in Computer Science)*. SpringerVerlag, 2004.

[7] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, and Y. Theodoridis, "TACO: tunable approximate computation of outliers in wireless sensor networks," in *SIGMOD*, 2010.

[8] N. Giatrakos, A. Deligiannakis, M. N. Garofalakis, and Y. Kotidis, "Omnibus outlier detection in sensor networks using windowed locality sensitive hashing," *Future Gener. Comput. Syst.*, vol. 110, 2020.

[9] M. Charikar, "Similarity estimation techniques from rounding algorithms," in *STOC*, J. H. Reif, Ed., 2002.

[10] Y. Zhang, N. Meratnia, and P. J. M. Havinga, "Outlier detection techniques for wireless sensor networks: A survey," *IEEE Commun. Surv. Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.

[11] T. Palpanas, "Real-time data analytics in sensor networks," in *Managing and Mining Sensor Data*, C. C. Aggarwal, Ed. Springer, 2013.

[12] G. Chiarot and C. Silvestri, "Time series compression survey," *ACM Comput. Surv.*, vol. 55, no. 10, feb 2023.

[13] A. Marascu, P. Pompey, E. Bouillet, M. Wurst, O. Verscheure, M. Grund, and P. Cudré-Mauroux, "TRISTAN: real-time analytics on massive time series using sparse dictionary compression," in *IEEE Big Data*, 2014.

[14] A. Khelifati, M. Khayati, and P. Cudré-Mauroux, "CORAD: correlation-aware compression of massive time series using sparse dictionary coding," in *IEEE Big Data*, 2019.

[15] S. Lin, W. Lin, K. Wu, S. Wang, M. Xu, and J. Z. Wang, "Cocv: A compression algorithm for time-series data with continuous constant values in iot-based monitoring systems," *Internet of Things*, vol. 25, p. 101049, 2024.

[16] N. Q. V. Hung, H. Jeung, and K. Aberer, "An evaluation of model-based approaches to sensor data compression," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 11, pp. 2434–2447, 2013.

[17] D. Rafiei and A. Mendelzon, "Similarity-based queries for time series data," *SIGMOD Rec.*, vol. 26, no. 2, p. 13–25, jun 1997.

[18] N. Giatrakos, Y. Kotidis, A. Deligiannakis, V. Vassalos, and Y. Theodoridis, "In-network approximate computation of outliers with quality guarantees," *Inf. Syst.*, vol. 38, no. 8, pp. 1285–1308, 2013.

[19] O. Younis and S. Fahmy, "Distributed clustering in ad-hoc sensor networks: A hybrid, energy-efficient approach," in *INFOCOM*, 2004.

[20] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad-hoc sensor networks," in *OSDI*, 2002.

[21] G. Tan, M. Bertier, and A. Kermarrec, "Visibility-graph-based shortest-path geographic routing in sensor networks," in *INFOCOM*, 2009.

[22] A. Kontaxakis, N. Giatrakos, and A. Deligiannakis, "A synopses data engine for interactive extreme-scale analytics," in *CIKM*, 2020.

[23] P. Levis, N. Lee, M. Welsh, and D. Culler, "Tossim: accurate and scalable simulation of entire tinyos applications," in *SenSys*, 2003.

[24] N. Giatrakos, Y. Kotidis, and A. Deligiannakis, "PAO: power-efficient attribution of outliers in wireless sensor networks," in *DMSN*, 2010.

[25] C. C. Aggarwal, *Mining Sensor Data Streams*. Springer, 2013.

[26] E. J. Keogh, "A decade of progress in indexing and mining large time series databases," in *VLDB*, 2006.

[27] C. Lin, E. Boursier, and Y. Papakonstantinou, "Approximate analytics system over compressed time series with tight deterministic error guarantees," *VLDB*, 2020.

[28] X. Kitsios, P. Liakos, K. Papakonstantinopoulou, and Y. Kotidis, "Sim-piece: Highly accurate piecewise linear approximation through similar segment merging," *Proc. VLDB Endow.*, vol. 16, no. 8, 2023.

[29] P. Liakos, K. Papakonstantinopoulou, and Y. Kotidis, "Chimp: Efficient lossless floating point compression for time series databases," *Proc. VLDB Endow.*, vol. 15, no. 11, pp. 3058–3070, 2022.

[30] C. M. Yeh, Y. Zheng, M. Pan, H. Chen, Z. Zhuang, J. Wang, L. Wang, W. Zhang, J. M. Phillips, and E. J. Keogh, "Sketching multidimensional time series for fast discord mining," in *IEEE Big Data*, 2023.

[31] S. J. Rotman, B. Cule, and L. Feremans, "Efficiently mining frequent representative motifs in large collections of time series," in *IEEE Big Data*, 2023.

[32] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014. [Online]. Available: https://arxiv.org/abs/1408.2927