Evaluating Fidelity of Lossy Compression on Spatiotemporal Data from an IoT Enabled Smart Farm

Aekyeung Moon[a], Jaeyoung Kim[a], Jialing Zhang[b], Seung Woo Son[b]

[a]*Electronics and Telecommunication Research Institute, Daegu, South Korea*
[b]*University of Massachusetts Lowell, Lowell, MA, USA*

Abstract

As the volume of data collected by various IoT sensors used in smart farm applications increases, the storing and processing of big data for agricultural applications become a huge challenge. The insight of this paper is that lossy compression can unleash the power of compression to IoT because, as compared with its counterpart (a lossless one), it can significantly reduce the data volume when the spatiotemporal characteristics of IoT sensor data are properly exploited. However, lossy compression faces the challenge of compressing too much data thus losing data fidelity, which might affect the quality of the data and potential analytics outcomes. To understand the impact of lossy compression on IoT data management and analytics, we evaluated four classification algorithms with reconstructed agricultural sensor data based on various energy concentration. Specifically, we applied three transformation-based lossy compression mechanisms to five real-world weather datasets collected at different sampling granularities from IoT weather stations. Our experimental results indicate that there is a strong positive correlation between the concentrated energy of the transformed coefficients and the compression ratio as well as the data quality. While we observed a general trend where much higher compression ratios can be achieved at the cost of a decrease in quality, we also observed that the impact on the classification accuracy varies among the data sets and algorithms we evaluated. Lastly, we show that the sampling granularity also influences the data fidelity in terms of the prediction performance and compression ratio.

Keywords: Smart farm, lossy compression, IoT, signal processing, data fidelity.

1. Introduction

The advent of IoT revolutionizes the knowledge discovery paradigm for various domains (Ludena and Ahrary (2013); Al-Fuqaha et al. (2015)). Suggestive actionable knowledge can be extracted from a continuous stream of raw data collected from IoT devices. This paper is particularly interested in IoT enabled smart farming. In short, smart farming with data analytic capabilities can provide more precise forecasts and thus could potentially improve crop yields as well as reduce production costs by removing the use of non-essential pesticides or fertilizers.

Recent years have witnessed a plethora of IoT solutions beneficial to agricultural domains. In the agriculture industry, advanced decision support systems through IoT technologies are increasingly gaining attention because they enable precision farming. After processing the collected data, they provide forecast services to farmers and growers so that they can make smarter decisions. The three major features that can affect weather-based predictions are as follows:

- A smart greenhouse is a facility that helps in the steady production of high-quality plants all year round by artificially controlling the cultivation environment. Different kinds of plants require different conditions (e.g., temperature, humidity, etc.) for their growth. If proper growth environments were provided, it would enable one to control plant growth rate (e.g., either promote or prevent flowering), thereby bringing huge economic benefits to framers and growers. An IoT enabled greenhouse control system collects information for managing plant growth and controls the facilities promoting optimal growth environments.

- Frost and freeze damage to flowers and buds at or near the bloom stage could result in significant crop failures (Jaradat et al. (2008); Matzneller et al. (2016)). For example, Chung et al. (2004) forecasted frost using global climate and weather data. If there were an accurate frost forecast, it would prevent damages from frost proactively, e.g., by moving a frost fan around the crop.

- Plant pathogens and pests including insects, mites, weeds and fungi can negatively affect crop productivity and profitability. Tripathy et al. (2013) reported on the interrelationship of weather, crops, and pests. Crop pest are also sensitive to particular weather condition such as humidity.

One of the key challenges to enable IoT smart farming is how to manage big data collected from various sensors efficiently (Ukil et al. (2015)). One solution for handling a large volume of data is to apply data compression techniques such that the storage and communication overheads are reduced (Bose et al. (2016); Huiibbe et al. (2013)). Various compression algorithms have been applied to satisfy different application needs, and many of them considered lossy algorithms with examples being ZFP, SZ, ISABEL, and wavelet-based (Li et al. (2017a,b)).

Lossy compression (Chou and Piegl (1992)) can help reduce the data size significantly, but error rates and thus loss of data quality are not easy to bound. Several recent approaches have proposed techniques to bound the error introduced by applying lossy compression methods (Sustika and Sugiarto (2016); Abo-Zahhad et al. (2015); Tao et al. (2017)). In Tao et al. (2017), it focuses on the scientific applications where often exhibits fairly sharp or spiky data changes in small data regions. Sustika and Sugiarto (2016); Abo-Zahhad et al. (2015) exploit sparse data pattern.

Nevertheless, as reported in several prior studies such as analyses of turbulent flow data (Li et al. (2015)) and climate data(Baker et al. (2014)), data reconstructed from lossy compression still allows meaningful analysis to be carried out. In (Baker et al. (2014)), the reconstructed data were able to reveal the same mean climate as the original data because climate data with compression rates of up to 5:1 can be reconstructed to be statistically indistinguishable from the original. However, lossy compression techniques are subjective to data fidelity issues (Li et al. (2017b)). Often data fidelity is dependent on specific application domain because acceptable information loss varies among variables of interest (Baker et al. (2014)).

In our previous paper (Moon et al. (2017b,a)), we showed that transformation based lossy compressions are useful for minimizing data reconstruction errors as well as important for maintaining errors within a tolerable range. Furthermore, we have studied the effect of lossy data compression on data fidelity before and after applying IoT analytics. However, from the viewpoint of data fidelity, there is a lack of verification for the relationship with the data collection or sampling frequency.

To manage IoT data efficiently and reliably, we collect, compress, and store climate data, and then reconstruct them for later analysis. We evaluate the fidelity of the reconstructed weather sensor data using lossy compression algorithms based on three transformations, namely, the Discrete Cosine Transform (DCT) (Razzaque et al. (2013)), Fast Walsh-Hadamard Transform (FWHT) (Fino and Algazi (1976)), and Discrete Wavelet Transform (DWT) (Abo-Zahhad et al. (2015)). Our objective was to evaluate the impact of the lossy compression and restoration on data reliability. Our experimental results using five sensor datasets show that lossy data compression can achieve 30x-100x compression ratios with marginal information loss. We collected weather sensors data using two sampling granularities (every minute and every hour) to evaluate how the sampling rate affects the amount of data reduction and quality of the data analysis.

Our compression mechanism is also simple in that it does not require complex quantization methods. In our comparison of the four classification algorithms for predicting frost, we observed that the prediction accuracy using compressed data containing only 90% of the total energy from the transformed coefficients did not drop much compared with that using the original data. In most cases, the frost prediction performance based on the reconstructed data is comparable with the performance based on the original data. Interestingly, in some cases, the prediction performance improves when the reconstructed data are used. These results clearly demonstrate that lossy compression leads to efficient management of big IoT data by reducing the data storage and transmission time while still maintaining the data quality.

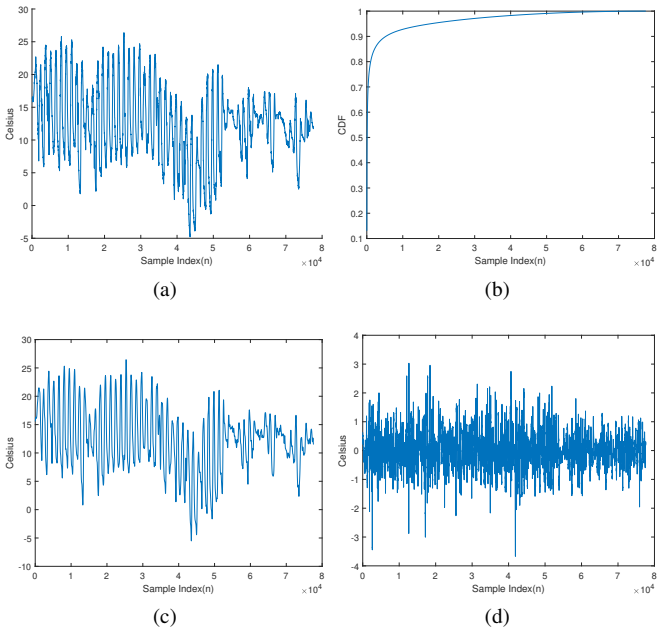
2. Materials and Methods

2.1. Design of the Transform-Based Lossy Compression

Many of the lossy compression techniques exploit the fact that, while individual data values in the dataset might show some randomness, their overall patterns are spatiotemporally smooth. Because of this, compression techniques in conjunction with data transformation can be more effective because the transformed data usually reveal the correlation of the data explicitly. For example, let us consider the temperature data (shown in Figure 1a) which is one of the datasets we evaluated in this paper. The details about the dataset we collected and evaluated will be described in Section 3.1. As shown in the figure, the data measured every minute exhibited diurnal temperature variations over the period of the measurement.

Figure 1b shows the CDF (cumulative density function) for the energy of the DCT coefficients after applying the DCT to the original temperature data for which each value is transformed to a DCT coefficient. As we can see, most of the energy is concentrated on a small number of low-frequency DCT coefficients and the remaining high-frequency coefficients are close to zero. Once the data is represented in the frequency domain, we can easily find the relationship between the percentage of informative DCT coefficients (i.e., low-frequency ones) and the amount of energy carried by them. To demonstrate the effect of this relationship on data compression, we chose DCT coefficients containing 99.9% of the energy attained by the original data, which accounts for only 3.16% (245 out of 77,590 total data points) of the entire data points. We then apply the inverse DCT to these selected coefficients to evaluate the difference between the reconstructed and the original data. As shown in Figure 1d, the difference is small, thus confirming that one can reconstruct IoT data like temperature data by maintaining a very small fractions of the original data. It should be noted that this error can be reduced even further if a proper quantization method is applied.

In signal processing theory, data in one domain (or basis) are transformed to another domain so that a signal is represented in a more concise format. The outcome of such a transformation can help reduce storage and data transmission overheads. If a sufficiently small number of non-zero coefficients from the data transformation can represent the characteristics of the original data, high compression ratios can be achieved (Abo-Zahhad et al. (2015)). Many transformation methods for representing signals in a compact format and minimizing errors during the reconstruction phase have been proposed by Chaturvedi and Yadav (2013). For example, in Sustika and Sugiarto (2016), DCT, DWT, and WHT (Walsh-Hadamard Transform) were evaluated using weather data. Their simulation results show that using DCT-transformed data as a basis has a better performance on weather data recovery compared with other transformation methods such as the WHT and DWT. However, they showed



(a) (b) (c) (d)

Figure 1: The variation of "temperature" values. (a) Original data. (b) Cumulative density function (CDF) of the energy concentrated in DCT coefficient sequences. (c) Reconstructed data from the inverse DCT of coefficients containing 99.9% energy of the original data. All other coefficients are set to zero, thus eliminating quantization on high-frequency data for higher data fidelity. (d) Difference between the reconstructed and the original data.

that the compressed signal is reconstructed with some error (Chaturvedi and Yadav (2013)). Bicer et al. (2013) proposed an online compression algorithm for climate data by exploiting spatial and temporal characteristics exhibited in climate data, thereby improving data the retrieval performance.

This section describes three lossy algorithms based on spatial data characteristics, which we evaluated as data transformation methods. To describe each transformation in detail, let us consider a one-dimensional discrete-time data x of length N, which is denoted as $N \times 1$ column vector with the elements $x[n]$, where $n = 1, 2, \ldots, N$.

Discrete Cosine Transform (DCT)

We first considered the discrete cosine transformation (DCT), which transforms data from the spatial domain into the frequency domain. A signal in the DCT is represented as a sum of varying magnitudes and frequencies, and DCT has been used in the lossy compression of audio and images (Razzaque et al. (2013)). DCT is defined as follows:

$$y(k) = w(k) \sum_{n=1}^{N} x(n) cos(\frac{\pi(2n-1)(k-1)}{2N}), k = 1, 2, \ldots, N,$$
$$w(k) = \begin{cases} \frac{1}{\sqrt{N}}, k = 1 \\ \sqrt{\frac{2}{N}}, 2 \le k \le N, \end{cases}$$

where $x(n)$, $y(k)$, N denote the original data, the transformed data, and the length of x, respectively.

Discrete Wavelet Transform (DWT)

We next considered the DWT. Wavelet transforms have been widely adopted in various compression algorithms because of its high-energy compaction properties (Abo-Zahhad et al. (2015)). In one-level DWT, two sets of coefficients are computed: approximated coefficients (C_1) and detailed coefficients (D_1). These two vectors are obtained by convolving x with the low-pass filter h_0 for C_1 and with the high-pass filter h_1 for D_1. The $\downarrow 2$ represents the down sampling operator by a factor of 2. The length of each filter is equal to $2L$. For a signal of length N, the signal F and G are of length $N + 2L - 1$, and then the coefficients C_1 and D_1 are of length $\frac{N-1}{2} + L$. A single-level DWT is denoted below, and this process can be applied recursively.

$$C_1 = (x * h_0) \downarrow 2,$$
$$D_1 = (x * h_1) \downarrow 2.$$

Fast Walsh-Hadamard Transform (FWHT)

FWHT is a faster implementation of the Walsh-Hadamard Transform (WHT). The FWHT requires only $N \log N$ additions or subtractions whereas a naive implementation of the WHT would have a time complexity of $O(N^2)$ (Fino and Algazi (1976)). The FWHT for a signal $x(n)$ of length N is calculated as follows:

$$y_n = \frac{1}{N} \sum_{i=1}^{N} x_i WAL(n, i),$$

where $i = 1, 2, \ldots, N$ and WAL(n, i) is Walsh functions.

2.2. Requirements for IoT Enabled Smart Farming

Weather stations with the purpose of performing predictive analytics collect climate and weather data from various sensors. In those predictive stations, gathering time-based climate data is particularly essential for making analytics and predictions more accurate because such data indicate changes in certain weather conditions at particular locations over time, thereby enabling one to evaluate climate and weather patterns and to perform short- and long-term forecasts. However, because the sensor nodes typically have a limited storage space, the sensors need to periodically send data to the cloud, which is expensive in terms of time and energy.

AgWeatherNet[1], for example, uses 177 stations installed since 1988 to collect climate data for providing agricultural services. AgWeatherNet collects various sensor data such as air temperature, relative humidity, dew point temperature, soil temperature, rainfall, wind speed, wind direction, solar radiation, leaf wetness, etc. Some stations also measure atmospheric pressure. A data logger collects these data every 5 seconds and summarizes them every 15 minutes. The number of stations is constantly increasing, thus increasing the amount of data dramatically. The IoT smart farm application itself is constrained in terms of bandwidth, energy and storage. As such, these data needs to be managed efficiently such that the storage and transmission costs can be reduced (Bose et al. (2016)).

To efficiently manage and transmit data, we need to use compression techniques on the data, especially a lossy one to

[1] `http://weather.wsu.edu/`

achieve a comparably higher compression ratio. However, the reconstructed data in the cloud needs to be analyzed later. In other words, without careful considerations, lossy compression could filter potentially important information critical to analytic workloads running on the cloud. To explain the need for determining acceptable quality requirements for the climate data and analytics, let us consider the following two IoT smart farm use-cases. First, plum pocket is a fungal disease that causes plum fruit to become hollow and irregularly shaped or to drop early, thereby resulting in a crop loss. Cool and wet spring weather is known to significantly promote the development of such fruit diseases. Weather conditions are important not only in early season diseases but also during the harvest where excess moisture can cause several rot diseases (Johnson (1975)). Second, frost can cause heavy losses in agricultural production; thus, it is important to minimize potential frost damage. Frost refers to the formation of ice crystals on surfaces or a meteorological event when crops and other plants experience freezing injury.

For both plant diseases and frost, when weather conditions are in favor of their occurrence, an instant warning is required to successfully control them. Therefore, it would be necessary to send, store and process all surrounding climate data for several regions to determine the possibility of their occurrence. For example, if we collect 5 climate data variables and assume each data variable is a real type every minute at N weather stations, the total amount of data that need to be transmitted per day is $1{,}440 \times 5 \times 4 \times N$ bytes. To predict more accurately, one might want to include more diverse sensors or increase the number of weather stations or sensor nodes. All these will increase the amount of data that need to be transmitted and analyzed. We already know that data compression works for highly efficient data management. However, what lacks is a comparative evaluation that establishes the criteria for selecting the degree of compromise in data quality yet still makes reasonably accurate forecast services.

2.3. Design of the Predictive Weather Platform

Our predictive weather platform, depicted in Figure 2, periodically collects climate information similar to other weather platforms, but performs data analysis for a frost forecast service using four machine learning (ML) algorithms: decision tree, boosted tree, random forest, and regression. Details about how we applied the individual MLs are as follows. Given a set of features x_i and a label $y_i \in \{0, 1\}$, logistic regression interprets the probability that the label is in one class as a logistic function of a linear combination of the features, which is represented as follows:

$$f_i(\theta) = p(y_i = 1|x) = \frac{1}{1 + \exp(-\theta^\gamma x)}. \quad (1)$$

The decision tree and boosted tree can also be used as a classifier for our purposes. In contrast to linear models like logistic regression, these algorithms can model nonlinear interactions between the features and the target values. Boosted tree is based on a collection of base learners, i.e., decision tree classifiers, and combines them using gradient boosting. It should be noted that our focus in this paper is to evaluate four widely used ML algorithms on their classification performance for predicting the presence/absence of frost.

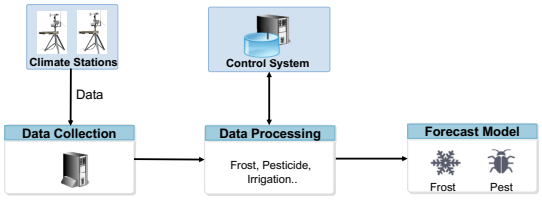


Figure 2: The architecture of IoT based predictive weather platform.

The proposed platform's overall goal is to provide web and mobile services so that farmers and plant growers can subscribe to agricultural services, frost prediction in this study, to help them make farming decisions. Our platform makes the frost forecast every night at 11 PM based on our frost prediction models built on top of the climate collected throughout the day. We make this forecast data available on the web and to mobile services so that the subscribers (farmers) can proactively implement preventive actions. For farmers who subscribe to push services, they are automatically notified with updated, more accurate frost forecast information at 1 AM. Moreover, it provides an interface for farmers to easily provide the system feedback for more accurate data collection. The location of observation stations are displayed on the map, and frost prediction/occurrence information, micro-weather information, etc. are displayed in real time on our project website[2]

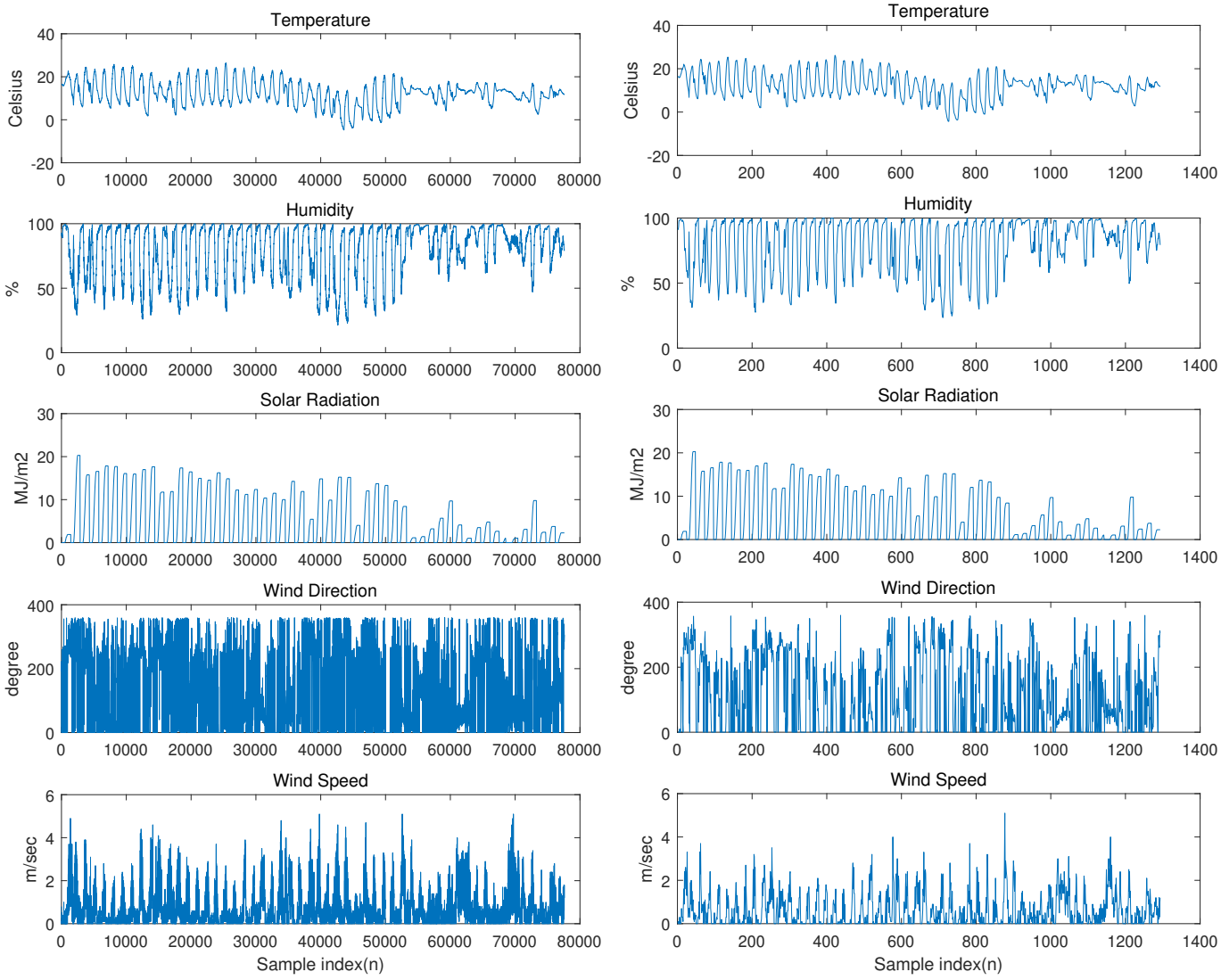
3. Results and Discussion

3.1. Datasets

We evaluated the effectiveness of the compression algorithms discussed in Section 2.1 on climate data. In our evaluation, we used a real-world dataset from the wireless climate stations located in a small orchard in Youngcheon, South Korea. We chose the following five most important variables, namely temperature, humidity, solar radiation, wind direction and wind speed, from the climate data collected during October 2015 at the deployed weather station. The data were continuously monitored and collected using two measuring (or sampling) periods: every minute and every hour. The reason that we had two sampling periods is to evaluate how different sampling rates affect the amount of data reduction and quantify its impact on the data analysis. In our evaluation, there were 77,590 data points for the collection period consisting of every minute and 1,294 data points for the collection period consisting of every hour, respectively. The entire duration of the sampling period was about 54 days. The original data samples for both measuring periods are shown in Figure 3.

Table 1 presents the statistical properties of the original data in terms of the standard deviation (STD), normalized standard deviation (NSTD), skewness, and kurtosis. The NSTD is calculated as $\frac{STD(x)}{Mean(x)}$. Skewness is a measure of data asymmetry

[2] `http://183.106.117.219/`. (Korean website)



(a) every minute (b) every hour

Figure 3: Data value variations exhibited in the original dataset (temperature, humidity, solar radiation, wind direction, and wind speed) during the entire sampling period (54 days or 7.7 weeks). The x-axis in each graph represents sampling points.

around the mean value. Normal distribution, which is symmetric around its mean, gives zero skewness. Negative skewness values mean that more data are scattered to the left of the mean whereas positive skewness values mean more data are scattered to the right. The measures of kurtosis in Table 1 indicate how outlier-prone a distribution is. As the kurtosis of any normal distribution is 3, distributions with a kurtosis higher than 3 are more outlier-prone. The distributions of a kurtosis lower than 3, on the other hand, are less outlier-prone. Solar radiation and wind speed show a higher STD than that of the other datasets. Table 1 shows the characteristics of the data collected using different sampling periods (every minute and every hour). In both sampling periods, we observe similar data characteristics for all metrics (deviation, skewness, and kurtosis). In the case of the skewness, solar radiation, wind speed, and wind direction have a positive value. Solar radiation, wind direction and wind speed also have a higher STD compared to the other datasets. For those three datasets, the kurtosis is also more deviated from 3 when compared with the other two datasets. Because the kurtosis for wind speed is higher than 3, distribution of the wind speed data has a heavier tail and a sharper peak than that of the normal distribution. However, the kurtosis for wind direction and solar radiation is lower than 3, which indicates that the distribution for wind speed has lighter tails and flatter peak than the normal distribution.

We used the following three metrics to evaluate the performance of each compression algorithm: compression ratio, the Normalized version of Root Mean Square Error (NRMSE) and the Peak Signal-to-Noise Ratio (PSNR).

- Compression Ratio: The compression ratio achievable by

Table 1: The evaluated datasets and their characteristics.

	Sampling Period	Standard Deviation	Normalized Standard Deviation	Skewness	Kurtosis
Temperature	1 min	5.6807	0.4659	-0.0907	3.0002
	1 hour	5.6843	0.4663	-0.0848	2.9999
Humidity	1 min	20.3439	0.2548	-0.9132	2.5916
	1 hour	20.3855	0.2553	-0.9180	2.5917
Solar Radiation	1 min	5.9199	1.1748	0.8546	2.2679
	1 hour	5.9772	1.1393	0.7969	2.1666
Wind Direction	1 min	112.6286	1.0666	0.5782	1.8434
	1 hour	113.7976	1.0364	0.5224	1.7903
Wind Speed	1 min	0.7574	1.2284	1.3920	4.5762
	1 hour	0.7805	1.2233	1.4652	5.0640

each compression method, R_M, is given by:

$$R_M = \frac{|D| - |D'|}{|D|} \times 100\%,$$

where $|D|$ is the size of D, $|D'|$ is the reduced size, and M is the individual compression method. Consequently, in the case of the DCT, DWT, and FWHT transformations, we compute k, which is reduced to $(n-k)/n$.

- NRMSE: Let $x = x_1, x_2, x_3, ..x_n$ be the original data and $\hat{x} = \hat{x}_1, \hat{x}_2, \hat{x}_3, ...\hat{x}_n$ be the reconstructed data. Then, the NRMSE for each compression method M can be defined as:

$$NRMSE_M = \frac{RMSE_M}{Mean(x)} = \frac{1}{\bar{x}} \sqrt{\frac{\sum_{n=1}^{N}(x(n) - \hat{x}(n))^2}{N}},$$

where N is the number of data points and $\bar{x}$ is mean of the original data x, and $\hat{x}$ is the reconstructed value of x.

- PSNR: For evaluating the effectiveness of our lossy compression method, we measured the peak signal-to-noise ratio (PSNR), a commonly used average error metric, especially in visualization (Tao et al. (2017)). It is calculated as follows:

$$PSNR_M = 20log_{10}(\frac{Max(x) - Min(x)}{RMSE_M})$$

3.2. Evaluation

We first describe how we compress the evaluated dataset using the DCT, DWT and FWHT transformations. For the DWT wavelet transform, we used the Daubechnies d4 wavelet (or db4 wavelet). The specific steps for calculating the compression ratio for each algorithm are as follows (Moon et al. (2017b)):

1. Decompose the original data into the DCT, DWT, and FTWH basis vectors, i.e., coefficient vectors.
2. Sort the coefficient vector $|S|$ in descending order of coefficient values. The sorted coefficient vector is denoted as: $SS = SS_1, SS_2, ...SS_n$.
3. Find k, which determines how many coefficients are required to represent the δ amount of the energy in the signal, where $0.0 \leq \delta \leq 1.0$ or $0\% \leq \delta \leq 100\%$. The norm is computed by the Euclidean norm (p-norm, p=2) of vectors SS.

$$\frac{norm(SS(1:k))}{norm(SS)} < \delta. \quad (2)$$

We note that the sum of the energy stored in the entire DCT coefficient is 1.0 (or 100%).

4. Coefficients smaller than the threshold value δ are set to zero. In other words, those nonsignificant values are discarded.

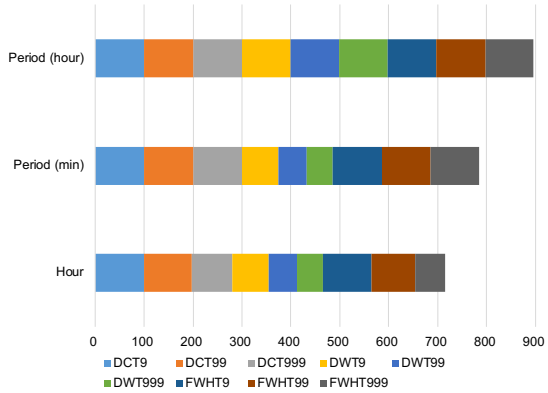


Figure 4: Compression ratio for the temperature dataset.

In our evaluation, we use $\delta = 0.9$ (90%), 0.99 (99%) and 0.999 (99.9%) in Equation 2 and compress the five data variables with 77,590 data points each. Assuming that each data has a real value of 4 bytes, so the total size of data nearly become 1.52 MB. Note that, the data size is per sensor node, and we expect the data volume will increase as we deploy more sensors and nodes. For DCT, DWT, and FWHT, each compression time takes about 0.014, 0.031, and 0.02 seconds, respectively. Each decompression time of DCT, DWT and FWHT are 0.006, 0.014 and 0.011 seconds each. Overall, we were able to achieve the compression speed of 108, 49 and 76 MB/s and the decompression speed of 253, 108 and 138 MB/s, respectively.

Figure 4 shows the compression ratio for the temperature dataset with the different data collection periods, i.e., every minute and hourly. Each clustered row combines all compression ratios using different transforms and different amounts of energy concentrations. In the case of the hourly data, the number of data points is reduced to 1,294 from a total of 77,590 data points. Therefore, in the case of 'Period (hour)', the number of data points is basically decreased from 77,590, and the final compression ratio is close to 99.9%. The last bar, denoted as 'Hour', is the compression ratio calculated using the 1,294 data points. In this case, the compression ratio is lower than the data collected every minute, denoted as 'Period (min)'. In other words, the compression ratio of 'Period (min)' is increased by 10% compared to 'Hour'.

Table 2 shows the compression ratio for the data collected every minute when $\delta = 0.9$ (90%), 0.99 (99%) and 0.999 (99.9%) in Equation 2, respectively. As shown in Table 2, DCT and FWHT show higher compression ratios than DWT. DWT

Table 2: Comparison of the compression ratios (min).

Algorithm	Threshold (δ)	Temperature	Humidity	Solar Radiation	Wind Direction	Wind Speed
DCT	0.9	99.9987	99.9987	99.9897	99.3479	99.8170
	0.99	99.9459	99.9794	99.8015	62.1098	70.5877
	0.999	99.6842	99.7306	97.8142	30.2513	35.1708
DWT	0.9	74.2106	68.8065	88.5707	84.8808	89.0282
	0.99	58.4483	54.6797	80.1057	70.1315	72.9952
	0.999	53.4760	50.8158	73.8459	56.7818	55.0419
FWHT	0.9	99.9954	99.9977	99.9390	99.0959	99.6643
	0.99	99.7902	99.8970	99.2340	60.8818	68.5310
	0.999	99.0524	99.2844	95.2011	29.537	46.8398

Table 3: Comparison of the error rates in Period (min).

Algorithm	Threshold (δ)	Temperature	Humidity	Solar Radiation	Wind Direction	Wind Speed
DCT	0.9	0.4659	0.2548	0.6672	0.6372	0.6900
	0.99	0.1555	0.1445	0.2171	0.2062	0.2234
	0.999	0.0493	0.0460	0.0690	0.0654	0.0708
DWT	0.9	0.4809	0.4498	0.6724	0.6373	0.6904
	0.99	0.1556	0.1456	0.2176	0.2062	0.2234
	0.999	0.0493	0.0461	0.0690	0.0654	0.0.0708
FWHT	0.9	0.4521	0.3951	0.6489	0.6267	0.6750
	0.99	0.1460	0.1335	0.2089	0.1735	0.1950
	0.999	0.0459	0.0431	0.0676	0.0511	0.0561

Table 4: Comparison of error rates in Period (hour)

Algorithm	Threshold (δ)	Temperature	Humidity	Solar Radiation	Wind Direction	Wind Speed
DCT	0.9	0.4661	0.2552	0.6598	0.6274	0.6874
	0.99	0.1552	0.1456	0.2135	0.2027	0.2224
	0.999	0.0491	0.0461	0.0677	0.0643	0.0703
DWT	0.9	0.4814	0.4509	0.6567	0.6279	0.6853
	0.99	0.1555	0.1447	0.2117	0.2034	0.222
	0.999	0.0493	0.0462	0.0671	0.0644	0.0701
FWHT	0.9	0.4693	0.3768	0.6313	0.5854	0.6423
	0.99	0.1442	0.1337	0.1986	0.1669	0.183
	0.999	0.0431	0.0401	0.0611	0.0526	0.056

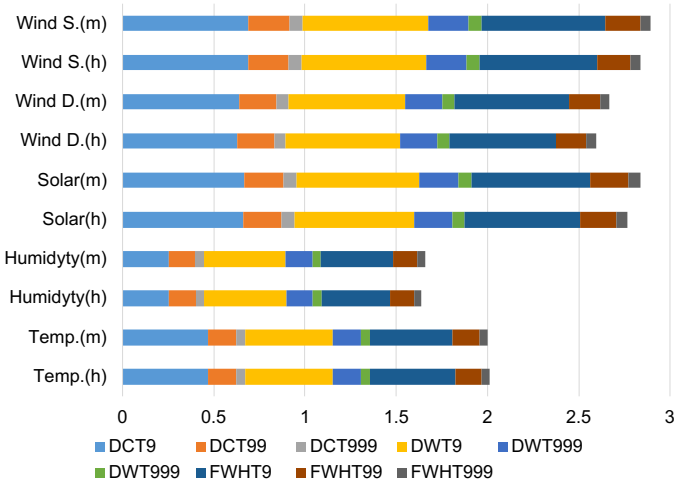


Figure 5: Clustered stacked chart of the error rates for datasets collected in different sampling periods.

shows the lowest compression ratios because we used a 1-level wavelet transformation, and therefore, k in Equation 2 is increased. In the case of the DCT and FWHT, compression ratios vary depending on the characteristics of data. The compression rates of the wind direction using the DCT are about 1.4 times and 2.8 times decreased according to threshold (δ). As previously shown in Table 1, in the the case of the kurtosis, the degree of deviation from 3 in those datasets (wind direction, wind speed, and solar radiation) is high. Additionally, the NSTD showed relatively high values for the wind speed, solar radiation, and wind direction. Combining those observed data characteristics with the compression ratios, we can see that there is a clear correlation between those two. The compression ratios of the wind direction data in the DCT are about 1.5 times and 3.2 times is decreased according to threshold (δ). Additionally, the compression ratios of the wind speed data in the DCT are about 1.4 times and 2.8 times decreased according to the threshold (δ). In the case of the hourly collected data, because the collection period itself already reduces the amount of data significantly compared with the data gathered per minute, the compression rate is close to 99%. This is because the compression ratio is calculated from all the data points (77,590).

To measure how much the reconstructed data deviate from the original data, we evaluated the normalized version of RMSE (NRMSE), a frequently used distortion estimate. The reconstructed data from our lossy compression method are shown in Tables 3 and 4. Overall, an the error threshold of 90% shows a slightly higher error rate than that of 99% (or 0.99), but the effect of the error rate varies depending on the datasets. Specifically, the error rates for wind speed, wind direction and solar radiation increase in the case of the DCT, DWT, and FWHT compared to that for the temperature. From these tables, we can see that the reconstructed data almost coincide with the original data in the case representing 99.9% (or 0.999). In the humidity data, the DWT shows higher variances than that of the other algorithms although it is still similar to the other reconstructed data. These results clearly demonstrate that the data can be recovered by using a small number of measurements or a small sampling rate. It shows the importance of the compression coefficients being within a range tolerable by the application.

Tables 3 and 4 show the error rates between the original data and the reconstructed data from our lossy compression method for 'Period (hour)' and 'Period (min)', respectively. To better understand the error rates, Figure 5 shows the cumulative errors for each compression algorithm and each dataset (Wind Speed, Wind Direction, Solar Radiation, Humidity, and Temperature). Period (hour) is denoted by h and Period (min) is denoted by m. As shown in this figure, Wind Speed, Wind Direction, and Solar Radiation show higher error rates than the other datasets. Specifically, in the case of the wind speed, the error rate of 'Period (m)' is 0.1x greater than the error rate of 'Period (h)'. The error rate is also affected by the characteristics of the data. For those three datasets with high error rates, the kurtosis also exhibited a relatively high deviation from 3 compared with the other two datasets. The skewness of those datasets has a positive value only.

Because PSNR measures the size of the RMSE relative to the peak size of the signal, a higher value of PSNR represents less error whereas lower values of RMSE/NRMSE indicate better quality. Figure 6 shows the PSNR for each compression algorithm. Overall, an error threshold of 90% shows a slightly lower PSNR than that of 99% (or 0.99), but the effect of the error rate varies depending on the datasets. Specifically, the error rates

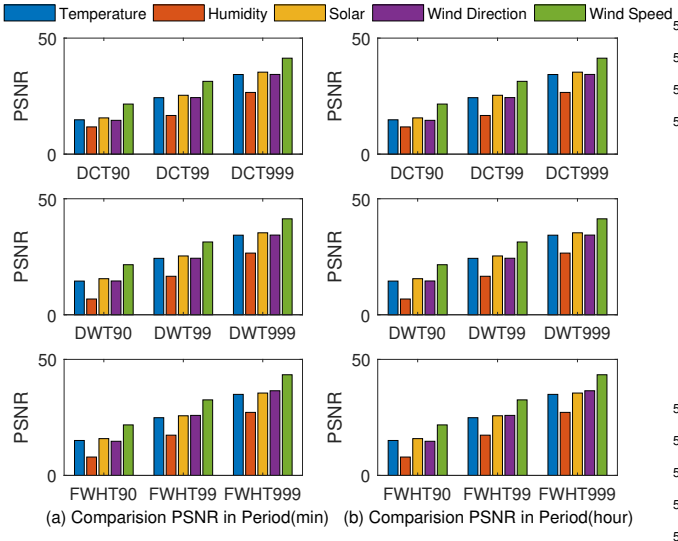


Figure 6: PSNR in different sampling periods.

for wind speed, wind direction and solar radiation increase in the case of the DCT, DWT, and FWHT compared to that for the temperature.

3.3. Data Fidelity

For the evaluation of data fidelity, we also collected frost data from four regions of Yeoungcheon, South Korea from October 1 to November 23, 2015. The number of actual frost occurrences is 19 out of entire 216 observed data points (54 days per each station). We predict the possibility of frost the next morning using the microclimate data. We used machine learning toolkits available on the GraphLab website[3] to train and evaluate five machine learning algorithms. We use 80% of the data for training and the remaining 20% for the testing.

We used the following two metrics to evaluate the impact of fidelity on the prediction accuracy: P (Precision) and R (Recall). In the case of the frost forecast, it is important to predict which days are likely to have frost occur, so we chose performance indicators that can calculate the fraction of the relevant instances. Precision is referred to as the positive predictive value while Recall in this context is referred to as the true positive rate. Precision is the number of correct results divided by the number of all returned results. Recall, on the other hand, is the number of correct results divided by the number of results that should have been returned. In other words, the recall value means the probability of predicting the actual event, a frost day in our study. Similarly, the precision value means the possibility of an actual frost day among the predicted frost days. For example, if the recall value is 0.9, it means that 90% of the actual frost was predicted. In this case, 1 out of 10 means that a frost day cannot be predicted. A precision value of 0.9 means that when it predicts 10 frost days, 1 out of 10 is not correct. In other words, suppose that we forecast 10 frost days; there must be frost for 9 days; however, 1 day may not have frost. Therefore, in the case of the frost forecast, the recall value is more important, and we chose to evaluate algorithms that generate higher recall values for this reason.

$$Precision = \frac{TP(TruePositive)}{TP(TruePositive) + FP(FalsePositive)} \quad (3)$$

$$Recall = \frac{TP(TruePositive)}{TP(TruePositive) + FN(FalseNegative)} \quad (4)$$

Tables 5-7 show the results of the four classification algorithms, namely the Decision Tree (DT), Logistic Regression (LR), Random Forest (RF), and Boosted Tree (BT) used on our evaluated datasets. To evaluate the performance of the prediction for our evaluated datasets, we did experiments for three cases: data collected every minute, data collected every hour, and averaged data per hour collected data every minute. For the predictive performance shown in Figure 8, the first case, i.e., running classification algorithms on the data collected every minute, is better than all the other algorithms used. For example, in the case of the FWHT999 (99.9%) in conjunction with the DT, the data collected per minute is about 10% and 50% better than the hourly collected data and time-averaged data, respectively.

Both the DCT and FWHT based lossy compressions can achieve significantly higher compression ratios with a tolerable loss of data quality. Especially, it is interesting to observe that in the case of the FWHT, the prediction performance using the reconstructed data is better than that using the original data. The reason for this is that, for the days where frost is highly likely to occur, the prediction algorithm predicts better if the data value is represented more accurately. For example, as investigated in several prior studies (Kwon et al. (2008); Han et al. (2009)), the possibilities of frost are higher when the wind speed is below a certain level. Figure 7 shows the temperature, humidity, wind speed, and amount of solar radiation using the FWHT. We excluded the wind direction because it does not have much influence on frost. Among the four variables, a drop in the temperature is an important factor for frost conditions. For example, when the average temperature difference is over 12 degree Celsius, there is a higher possibility of frost. The average of the lowest temperature, e.g., below -0.4 degrees Celsius, also indicates a higher possibility of frost. Lastly, higher solar radiation, for example, 12 MJ/m^2,is another key indicator of higher chances of frost.

Our evaluation using the original and reconstructed data from the compression shows that the classification models can effectively predict the possibility of frost in advance such that farmers can proactively take preventive actions to protect their crops from frost damage. More specifically, Figure 8 the comparison of the classifiers with different data sampling granularity. In each chart, M means 'Period (minute)'; H means 'Period (hour)', and HA means time-average per hour. P and R denote precision and recall values, respectively, as shown in Equation 3 and 4. Again, in the case of frost forecast, Recall is the more

[3] https://turi.com/products/create/docs/graphlab.toolkits.classifier.html

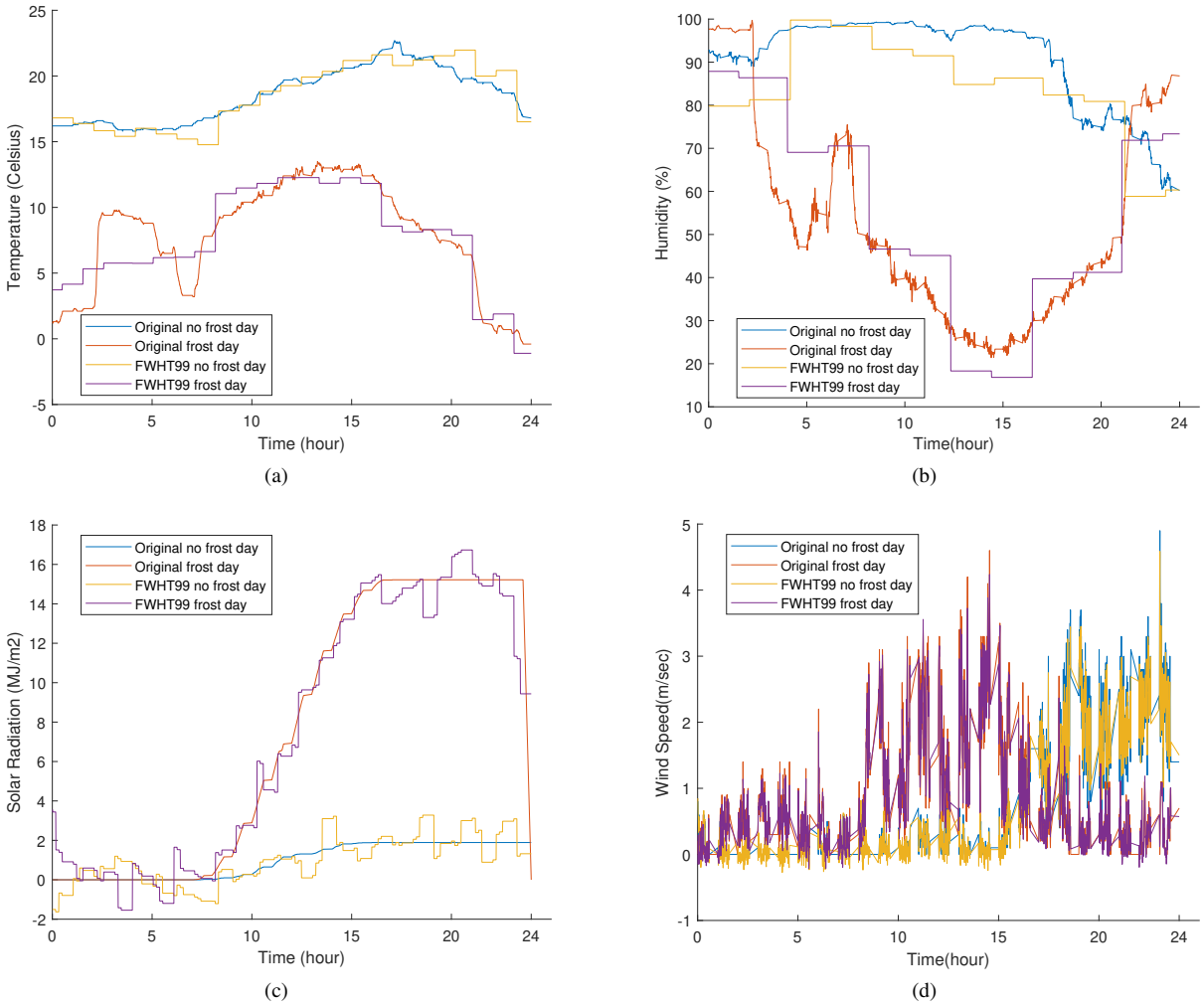


Figure 7: Comparison of frost and no-frost day for four weather variables: (a) temperature, (b) humidity, (c) solar radiation, and (d) wind speed. The reconstructed data is based on FWHT with 99% of energy.

Table 5: Comparison of original and reconstructed data for Period (min)

	DT(P)	DT(R)	LR(P)	LR(R)	RF(P)	RF(R)	BT(P)	BT(R)
Original	0.862	0.619	0.774	0.577	0.983	0.555	0.927	0.751
DCT90	0.678	0.084	NaN	0	1	0.061	1	0.095
DCT99	0.876	0.704	0.716	0.699	0.932	0.685	0.926	0.862
DCT999	0.936	0.581	0.744	0.546	0.98	0.492	0.963	0.761
DWT90	0.884	0.225	0.525	0.23	1	0.198	0.839	0.311
DWT99	0.885	0.61	0.771	0.569	0.95	0.553	0.939	0.772
DWT999	0.861	0.614	0.78	0.564	0.912	0.615	0.952	0.786
FWHT90	0.971	0.996	0.878	1.0	0.971	0.996	1	1
FWHT99	0.861	0.797	0.707	0.55	0.91	0.796	1	0.997
FWHT999	0.918	0.665	0.767	0.555	0.925	0.614	0.989	0.906

Table 6: Comparison of original and reconstructed data for Period (hour)

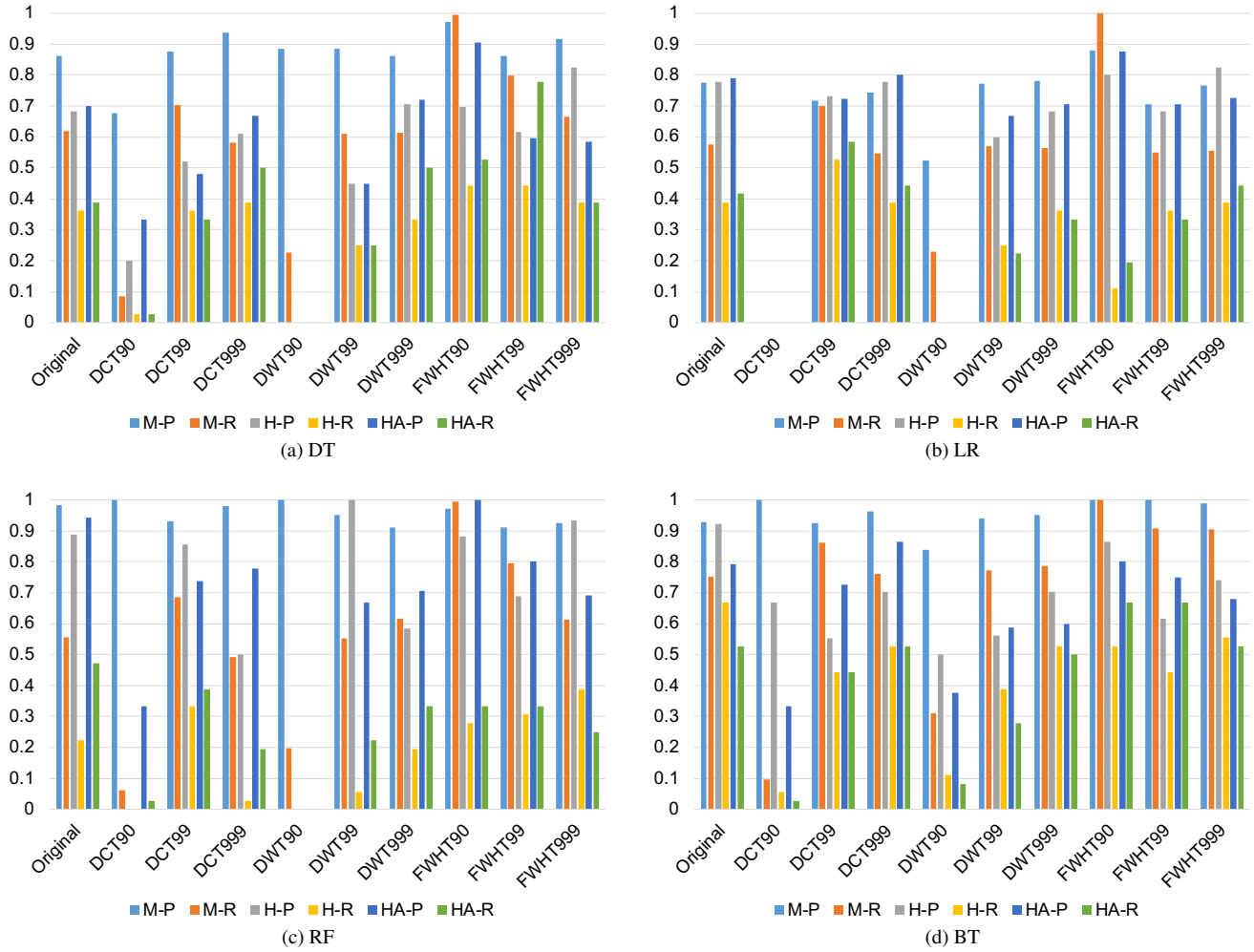
	DT(P)	DT(R)	LR(P)	LR(R)	RF(P)	RF(R)	BT(P)	BT(R)
Original	0.684	0.361	0.778	0.389	0.889	0.222	0.923	0.667
DCT90	0.2	0.028	NaN	0	NaN	0	0.667	0.056
DCT99	0.52	0.361	0.731	0.528	0.857	0.333	0.552	0.444
DCT999	0.609	0.389	0.778	0.389	0.5	0.028	0.704	0.528
DWT90	NaN	0	NaN	0	NaN	0	0.5	0.111
DWT99	0.45	0.25	0.6	0.25	1	0.056	0.56	0.389
DWT999	0.706	0.333	0.684	0.361	0.583	0.194	0.704	0.528
FWHT90	0.696	0.444	0.8	0.111	0.883	0.278	0.864	0.528
FWHT99	0.615	0.444	0.684	0.361	0.688	0.306	0.615	0.444
FWHT999	0.824	0.389	0.824	0.389	0.933	0.389	0.741	0.556

important value. As shown in Figure 8, M-R is greater than H-R and HA-R. In the case of the DT and FWHT90, M-R is about 2.2x better than H-R and 1.8x better than HA-R. Therefore, in terms of the prediction performance, it is better to use data with a finer sampling granularity for both compression and analysis of the reconstructed data.

It should be noted that, for a more precise forecast, a predictive weather platform needs to carefully monitor the numerical stability and consistency of the results because it evaluates compression artifacts. Most IoT big data will likely require lossless compression methods. However, compression schemes might be useful if they are applied to the more compressible variables ensuring data reliability.

4. Conclusion

Emerging IoT-based smart farming produces a large volume of diverse data, which needs to be stored efficiently and re-



(a) DT (b) LR (c) RF (d) BT

Figure 8: Comparison of the classifiers. (a) DT (Decision Tree) (b) LR (Logistic Regression) (c) RF (Random Forest) (d) BT (Boosted Tree). Note that some bars, e.g., all DCT90 bars in the LR classifier, do not appear because the result (Precision and Recall) is NaN.

Table 7: Comparison of original and reconstructed data on average.

	DT(P)	DT(R)	LR(P)	LR(R)	RF(P)	RF(R)	BT(P)	BT(R)
Original	0.7	0.389	0.789	0.417	0.944	0.472	0.792	0.528
DCT90	0.333	0.028	NaN	0	0.333	0.028	0.333	0.028
DCT99	0.48	0.333	0.724	0.583	0.737	0.389	0.727	0.444
DCT999	0.667	0.5	0.8	0.444	0.778	0.194	0.864	0.528
DWT90	NaN	0	NaN	0	NaN	0	0.375	0.083
DWT99	0.45	0.25	0.667	0.222	0.667	0.222	0.588	0.278
DWT999	0.72	0.5	0.706	0.333	0.706	0.333	0.6	0.5
FWHT90	0.905	0.528	0.875	0.194	1	0.333	0.8	0.667
FWHT99	0.596	0.778	0.706	0.333	0.8	0.333	0.75	0.667
FWHT999	0.583	0.389	0.727	0.444	0.692	0.25	0.679	0.528

liably. In this paper, we evaluated the effectiveness of data compression on five of the most important variables in real climate/weather data as an exemplar of IoT applications. Specifically, we compared the performance of the predictive analytics on the reconstructed data using the DCT, FWHT, and DWT to evaluate the feasibility of applying lossy compression to IoT big data. Our experimental results show that lossy compressions based on the DCT and FWHT can achieve significantly higher compression ratios with a marginal loss of data quality. Interestingly, the prediction performance using the reconstructed data based on the FWHT achieved better results than that of the original data. We also observed that the reconstructed data from the DCT and FWHT almost coincide with the original data for all five datasets. Overall, our results are promising in that the compression schemes might be useful if they are applied to more compressible variables ensuring data reliability. Additionally, in terms of the prediction performance, it is better to use data with a finer sampling granularity for both compression and analysis of the reconstructed data. Therefore, it is important to select compression coefficients and data sampling granularity within a range tolerable by the application with reliable precision.

Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant No. 1751143. This work was supported by the National Research Council of Science & Technology (NST) grant by the Korea government (MSIP) (No. CRC-15-01-KIST).

References

Abo-Zahhad, M. M., Hussein, A. I., Mohamed, A. M., 2015. Compressive Sensing Algorithms for Signal Processing Applications: A Survey. International Journal of Communications, Network and System Sciences 8 (6), 197–216.

Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., Ayyash, M., 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. IEEE Communications Surveys Tutorials 17 (4), 2347–2376.

Baker, A. H., Xu, H., Dennis, J. M., Levy, M. N., Nychka, D., Mickelson, S. A., 2014. A Methodology for Evaluating the Impact of Data Compression on Climate Simulation Data. In: The 23rd International Symposium on High Performance Parallel and Distributed Computing. pp. 203–214.

Bicer, T., Yin, J., Chiu, D., Agrawal, G., Schuchardt, K., 2013. Integrating Online Compression to Accelerate Large-Scale Data Analytics Applications. In: 27th IEEE International Symposium on Parallel and Distributed Processing (IPDPS). pp. 1205–1216.

Bose, T., Bandyopadhyay, S., Kumar, S., Bhattacharyya, A., Pal, A., June 2016. Signal Characteristics on Sensor Data Compression in IoT – An Investigation. In: 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). pp. 1–6.

Chaturvedi, R., Yadav, Y., 2013. A Survey on Compression Techniques for ECG Signals. International Journal of Advanced Research in Computer and Communication Engineering 2 (9).

Chou, J. J., Piegl, L. A., May 1992. Data reduction using cubic rational B-splines. IEEE Computer Graphics and Applications 12 (3), 60–68.

Chung, U., Seo, H. C., Yun, J. I., 2004. Site-Specific Frost Warning Based on Topoclimatic Estimation of Daily Minimum Temperature. Korean Journal of Agricultural and Forest Meteorology 6 (3), 164–169.

Fino, B. J., Algazi, V. R., Nov. 1976. Unified Matrix Treatment of the Fast Walsh-Hadamard Transform. IEEE Trans. Comput. 25 (11), 1142–1146.

Han, J. H., Choi, J. J., Chung, U. R., Cho, K. S., 2009. Frostfall Forecasting in the Naju Pear Production Area Based on Discriminant Analysis of Climatic. Korean Journal of Agricultural and Forest Meteorology 11 (4), 135–142.

Huiibbe, N., Wegener, A., Ling, Y., Ludwig, T., June 2013. Evaluating Lossy Compression on Climate Data. In: International Supercomputing Conference (ISC). pp. 343–356.

Jaradat, M. A. K., Al-Nimr, M. A., Alhamad, M. N., Dec. 2008. Smoke Modified Environment for Crop Frost Protection: A Fuzzy Logic Approach. Comput. Electron. Agric. 64 (2), 104–110.

Johnson, J. D., 1975. Diseases of Peaches and Plums. `http://hdl.handle.net/1969.1/160892`.

Kwon, Y., Lee, H., Kwon, W., Boo, K., 2008. The Weather Characteristics of Frost Occurrence Days for Protecting Crops against Frost Damage. The Korean Geographic Society, 824–842.

Li, S., Gruchalla, K., Potter, K., Clyne, J., Childs, H., 2015. Evaluating the Efficacy of Wavelet Configurations on Turbulent-Flow Data. In: IEEE Symposium on Large Data Analysis and Visualization (LDAV). pp. 81–89.

Li, S., Marsaglia, N., Chen, V., Sewell, C., Clyne, J., Childs, H., 2017a. Achieving Portable Performance For Wavelet Compression Using Data Parallel Primitives. In: Eurographics Symposium on Parallel Graphics and Visualization.

Li, S., Marsaglia, N., Chen, V., Sewell, C., Clyne, J., Childs, H., 2017b. Performance Impacts of In Situ Wavelet Compression on Scientific Simulations. In: Proceedings of the In Situ Infrastructures on Enabling Extreme-Scale Analysis and Visualization. pp. 37–41.

Ludena, R. D. A., Ahrary, A., Oct 2013. A Big Data Approach for a New ICT Agriculture Application Development. In: 2013 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. pp. 140–143.

Matzneller, P., Götz, K.-P., Chmielewski, F.-M., 2016. Spring frost vulnerability of sweet cherries under controlled conditions. International Journal of Biometeorology 60 (1), 123–130.

Moon, A., Kim, J., Zhang, J., Liu, H., Son, S. W., 2017a. Understanding the Impact of Lossy Compression on IoT Smart Farm Analytics. In: 2017 IEEE Big Data. pp. 4602–4611.

Moon, A., Kim, J., Zhang, J., Son, S. W., 2017b. Lossy Compression on IoT Big Data by Exploiting Spatiotemporal Correlation. In: 2017 IEEE HPEC. pp. 1–7.

Razzaque, M. A., Bleakley, C. J., Dobson, S., 2013. Compression in wireless sensor networks: A survey and comparative evaluation. ACM Transactions on Sensor Networks 10 (1), 5.

Sustika, R., Sugiarto, B., 2016. Compressive Sensing Algorithm for Data Compression on Weather Monitoring System. TELKOMNIKA Telecommunication, Computing, Electronics and Control, 974–980.

Tao, D., Di, S., Chen, Z., Cappello, F., 2017. Significantly Improving Lossy Compression for Scientific DataSets Based on Multidimensional Prediction and Error-Controlled Quantization. In: 2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS). pp. 1129–1139.

Tripathy, A. K., Adinarayana, J., Sudharsan, D., June 2013. Data Mining and Wireless Sensor Network for Groundnut Pest / Disease Interaction and Predictions - A Preliminary Study. In: International Journal of Computer Information Systems and Industrial Management Applications. pp. 427–436.

Ukil, A., Bandyopadhyay, S., Pal, A., April 2015. IoT Data Compression: Sensor-Agnostic Approach. In: 2015 Data Compression Conference. pp. 303–312.