Author(s): Väänänen, Olli; Zolotukhin, Mikhail; Hämäläinen, Timo

Title: Linear approximation based compression algorithms efficiency to compress environmental data sets

Year: 2020

Version: Accepted version

Please cite the original version:

Väänänen, O., Zolotukhin, M. & Hämäläinen, T. (2020). Linear approximation based compression algorithms efficiency to compress environmental data sets. In L. Barolli, F. Amato, F. Moscato, T. Enokido & M. Takizawa (Eds.), Web, Artificial Intelligence and Network Applications. Proceedings of the Workshops of the 34th International Conference on Advanced Information Networking and Applications (WAINA-2020), 110-121. Advances in Intelligent Systems and Computing, vol 1150. Springer, Cham. DOI: 10.1007/978-3-030-44038-1_11

URL: https://doi.org/10.1007/978-3-030-44038-1_11

Linear Approximation Based Compression Algorithms Efficiency to Compress Environmental Data Sets

Olli Väänänen[1], Mikhail Zolotukhin[2] and Timo Hämäläinen[2]
[1] Industrial Engineering, School of Technology, JAMK University of Applied Sciences, Jyväskylä, Finland
olli.vaananen@jamk.fi
[2] Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
mikhail.m.zolotukhin@jyu.fi, timo.t.hamalainen@jyu.fi

Abstract. Measuring some environmental magnitudes is a very typical application in the field of Internet of Things. Wireless sensor nodes measuring these environmental magnitudes are often battery powered devices. Thus, the energy efficiency is an important topic in these measuring devices. The most efficient method to reduce energy consumption in wireless devices is to reduce the amount of data needed to transmit via wireless connection. A simple method to reduce the amount of the data is to compress sensor data. Environmental data behaves quasi linearly in short time window and many compression algorithms utilize this data behavior. In this paper the different environmental data sets characteristics and their effect on compression algorithms' compression ratio are evaluated. The results can be used to evaluate and choose the suitable compression algorithm for the application and to predict the lifetime of the battery powered device.

1 Introduction

In the field of Internet of Things (IoT), sensors measuring some environmental magnitudes are very typical applications. The IoT applications measuring and utilizing some environmental data can be found and used in many sectors in the society. The need to measure some environmental magnitudes is especially typical in agricultural applications [1]. In agricultural applications, the devices are often spread across the field and thus the resources available are often limited, e.g. reliable power supply and good quality wireless connections, which often also means limited computational power.

In agriculture, the Internet of Things applications can be used for e.g. crop management, crop protection, soil monitoring and water management. [2, 3] Many IoT applications and solutions in the field of agriculture are still in their infancy; however, the field is changing very fast [4].

Energy efficiency and energy saving are very important aspects in battery powered wireless sensor nodes [5, 6]. One very efficient way to reduce energy consumption in wireless sensor nodes is to compress the sensor data. By compressing the sensor data, it is possible to reduce the amount of data needed to transmit via a wireless connection. Wireless connection is known to be the most energy consuming operation in the wireless sensor node. [7]

Compressing the amount of actual sensor data needed to transmit wirelessly is only one way to reduce energy consumption. [6] However, due to the simplicity of many compression methods presented, it is a very easy and powerful method for maximizing the lifetime of a battery powered device.

In this paper, several data linearity-based compression methods have been evaluated and compared to each other. The efficiency of the compression methods to compress certain environmental data sets are evaluated and the effect of the data sets' characteristics in the compression ratio achieved have been evaluated. The correlation of certain data sets' characteristics to the compression ratio has been evaluated. With the correlation found, it is possible to choose the suitable compression method for a certain application.

2 Compression Methods Based on Time Series Data Linearity

Various sensor data compression methods have already been introduced several decades ago. After the proliferation of the wireless sensor networks (WSN) and the Internet of Things (IoT), the topic of sensor data compression has received a great deal of new attention in the field of research. [8] Typical sensor data sets are for example some environmental variable data sets such as temperature, humidity, air pressure and wind speed. Additionally, different Wireless Body Sensors measuring some parameters or behavior of the human body have gained much attention [9]. They are used in different wearable and wellness devices and applications.

There are various types of compression methods presented in research papers. There are time domain and transform domain methods. Well-known transform domain methods are for example Wavelets, Chebyshev Transform, Discrete Fourier Transform (DFT) and Discrete Cosine Transform (DCT). Many times the domain methods are based on data linearity. Linearity based methods are for example Piecewise Linear Approximation (PLA), Lightweight Temporal Compression (LTC), Piecewise Aggregate Approximation (PAA) and Piecewise Constant Approximation (PCA). These methods are lossy compression methods. [6, 10-12]

Transform domain methods are not well suited for constrained wireless sensor nodes due to their computational complexity and limited memory. [10] Many compression methods such as LTC also suffer from latency and are not well suited for real-time or near real-time applications [13]. DFT and DCT also suffer latency dependent on the window size N used.

Even though linear approximation and data linearity-based methods are well known and simple methods, there is recent and ongoing research on the topic. Several different variations of these methods have been introduced during recent years. [7, 11, 14, 15]

The LTC is very efficient compression method for environmental data with a linear nature at least in short time window. Its compression ratio can be quite high. For temperature data with 10 minutes measuring rate and error bound $\varepsilon = 0.5$ °C, the LTC algorithm can achieve a compression ratio 10 to 1. The compression ratio is very dependent on the data set characteristics and the error bound used. [11] At the same time, it is rather a simple compression algorithm and thus it can be used in constrained IoT devices with limited memory and processing power.

3 Effect of Environmental Data Set on Compression Ratio

Many environmental magnitudes behave near linearly if the observation window is short enough. [16] For example, if the environment microclimate temperature is rising, it can be predicted that it will continue rising at least in the near future. This linear behavior can be used to compress the amount of data needed to transmit from the sensor node. Simple sensor data compression methods utilizing this behavior are based on data linearity. Perhaps the simplest compression algorithm for this kind of data is to use linear regression of n ($n \geq 3$) measured values and with allowing certain error bound $\pm\varepsilon$ to the calculated regression line. The calculated regression line with error bound can be used to predict the following values. There are many different versions of linearity-based compression methods presented in the literature. [11, 16, 17] This kind of methods are lossy methods.

Environmental data has a linear behavior if the observation window is short. The more constant the measured magnitude remains, the more efficiently these data linearity-based compression algorithms compress the data. [12] However, the environmental magnitudes do not remain constant; instead, the values are mostly changing. There is natural variation in the values of environmental magnitudes in function of time but with allowing some random variation in values, the main trend is often rather stable for some time period. Many compression methods utilize this behavior.

3.1 Data Set Characteristics Evaluated

In this paper, the efficiency of different linearity-based time series compression algorithms to compress environmental data is tested for different environmental data sets. Data sets' characteristics are evaluated, and the parameters affecting the compression ratio have been evaluated.

The tested and evaluated data set parameters are:

- *AC,* the average absolute change between consecutive measurements in the whole data set
- *SD*, the standard deviation of the change between consecutive measurements in the whole data set

For the measured values x_i: $i \in [1, n]$, the average change (AC) between consecutive measurements is calculated with the equation (1):

$$Average\ change = AC = \frac{\sum_{i=1}^{n-1} |x_{i+1} - x_i|}{n-1}. \quad (1)$$

Standard deviation (*SD*) is calculated from the consecutive measurement change values with the equation (2):

$$Standard\ deviation\ =\ SD\ =\ \sqrt{\frac{\sum_{i=1}^{n-1}\left((x_{i+1}-x_i)-(\overline{x_{i+1}-x_i})\right)^2}{n-2}}. \tag{2}$$

where,

$$(\overline{x_{i+1}-x_i})\ =\ \frac{1}{n-1}\sum_{i=1}^{n-1}(x_{i+1}-x_i)\,. \tag{3}$$

3.2 Data Sets

The used data sets were gathered from the Finnish Meteorological Institute's (FMI) open data service [18]. The data sets gathered from FMI service were Naruska measurement station data from whole year 2018. The Naruska measurement station is located in Eastern Lapland in Finland. It is one of the official measurement stations in Finland. Temperature, air pressure and wind speed data with a 10-minute measurement interval were used. The data sets were divided into monthly data sets, and the whole year data set was also used. 20-minute, 30-minute, 40-minute, 50-minute and 1-hour measurement interval data sets were derived from the original 10-minute measurement interval data set by cancelling the values from the original data set.

Thus, there were in total 78 data sets for each environmental variable (temperature, air pressure and wind speed).

The whole year 2018 data set with a 10-minute measurement interval was the largest data set with 51 961 measured values for each variable. The smallest data set used was February 2018 with 1-hour measurement interval with 672 measured values for each variable.

The average change *AC* values and standard deviation *SD* values were compared to the compression ratios achieved with different time series compression algorithms. The compression ratios were calculated with the equation (4):

$$CR\ =\ \frac{original\ data}{compressed\ data}. \tag{4}$$

where the *original data* is the amount of values in original data set and the *compressed data* is the amount of values in a compressed data set.

4 Compression Algorithms' Compression Ratio Compared to the Characteristics of Selected Data Set

Compression algorithms tested and evaluated were Lightweight Temporal Compression (LTC) [16] and Linear Regression based Temporal Compression (LRbTC) [11]. The LTC algorithm is originally presented in reference [16]. LTC uses the piecewise linear function to estimate data points. LTC calculates the upper and lower bound from every new data point by using the selected error bound. The LTC algorithm is explained in detail in references [7, 11, 16]. LRbTC algorithm uses the n measured values to calculate the regression line which can be used to predict following values with allowing a certain error bound $\pm\varepsilon$ from the line. When the measured value falls out from the allowed area, the new regression line is calculated which predicts the future values. [11] LRbTC algorithms were tested with 3, 4 and 5 values used to calculate the linear regression line. The error bound used was 0.5 °C for temperature data sets, 0.5 hPa for air pressure data sets and 0.5 m/s for wind speed data sets. The compression ratios achieved were compared to the data sets' characteristics *SD* and *AC,* which have been previously explained in this paper. The compression algorithms were programmed on Matlab as in reference [11]. The LRbTC algorithm used was the slightly modified version M-LRbTC [11], and the LTC algorithm used was the original version originally presented in the reference [16].

4.1 Temperature Data Sets

For temperature data sets (78 data sets in total) the results can be seen in Fig. 1. Fig. 1 presents the compression ratio for each temperature data set with LTC and LRbTC algorithms. Discrete Cosine Transform (DCT) algorithm with window size of 5 values was used just for comparison. The results are presented in the function of the standard deviation (*SD*) of the data set's consecutive measurements change (on the left) and in the function of average change (*AC*), as previously explained in this paper. The trend line (solid line) is visually the best fit polynomial regression line of the data points presented.

Fig. 1 clearly indicates that the LTC is the most effective compression algorithm compared to the others. The similar results have been achieved in reference [11]. The highest compression ratio (19.24) has been achieved with LTC algorithm from December 2018 data set with a 10-minute measurement interval. The highest compression ratio with LRbTC is 8.21 from the same data set. LRbTC with 3 values used to calculate regression line is slightly worse than the versions with 4 and 5 values used to calculate the regression line. The difference between 4 and 5 values used to calculate regression line is almost negligible.

The correlation between the compression ratio and the standard deviation of the consecutive measurements change is clear; however, the correlation is not linear. A small standard deviation means that the value changes are small from measurement to measurement, which means more constant and linearly behavior data. When the SD value decreases from the value 1 to 0.2, the compression ratio raises strongly.

In Fig. 1 on the right side, the same data sets' compression ratios were compared to the average change (*AC*) in the absolute value of the consecutive measurements' change as explained previously in this paper. Similar results can be seen with the *AC* as with the *SD*. The correlation is similar as in *SD* comparison except the dispersion is smaller in *AC* comparison. Thus, it seems that the *AC* predicts the compression ratio achieved better than the *SD*.

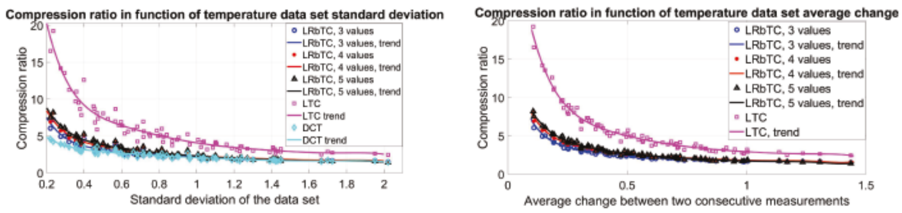

Fig. 1. Compression ratio in function of *SD* (on the left) and *AC* (on the right) from the temperature data sets.

The trend lines (fitting lines) in Fig. 1 are 8^{th} degree polynomials (y = p1*x^8 + p2*x^7 + p3*x^6 + p4*x^5 + p5*x^4 + p6*x^3 + p7*x^2 + p8*x + p9). The 8^{th} degree polynomials were chosen here because they give visually the best fit for the data. Additionally, the norm of residuals value, which is the measure of the goodness of the fit, was best or almost the best of the basic fitting functions. The smaller the norm of residuals value is, the better the fit. The polynomial coefficients and norm of residuals values for each compression algorithm in function of *SD* and *AC* can be seen in Table 1 and Table 2.

Table 1. Correlation between the compression ratio and standard deviation for temperature data sets, polynomial coefficients and the norm of residuals values.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	6.7423	6.2854	6.8386	6.548
p2	-72.267	-67.776	-70.398	-80.376
p3	325.72	307.75	306.91	406.94
p4	-805.25	-767.62	-739.53	-1118.2
p5	1192.3	1148.7	1076.8	1831.5
p6	-1081.4	-1055.3	-971.24	-1837.2
p7	587.81	583.09	533.42	1109.9
p8	-178.12	-180.98	-167.79	-378.84
p9	26.555	28.115	27.281	63.733
Norm of residuals	2.4946	2.5065	2.5082	7.0922

Table 2. Correlation between the compression ratio and average change for temperature data sets, polynomial coefficients and the norm of residuals values.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	177.09	181.2	141.07	35.377
p2	-1166.5	-1174.1	-896.79	-331.22
p3	3232.7	3207.3	2407.7	1253.5
p4	-4906.3	-4813.2	-3566.6	-2552
p5	4447.9	4335.5	3197.7	3089
p6	-2465.2	-2405	-1791.4	-2295
p7	821.88	811.44	625.16	1036
p8	-156.24	-158.91	-131.74	-270.25
p9	16.452	17.68	16.636	37.549
Norm of residuals	1.4334	1.3387	1.4053	3.9424

The norm of residuals values demonstrate that the correlation between compression ratio and *AC* is better than with *SD*.

If the data values are varied a great deal from measurement to measurement, it means faster changes in data in function of time. This indicates that this kind of data have higher frequencies. The frequency spectrum of the data set can be calculated with Discrete Fourier Transform (DFT). The data set DFT was calculated in Matlab with FFT (Fast Fourier Transform) function. The comparison of the frequency spectrum of two very different temperature data sets can be seen in Fig. 2. The red line is FFT from data set: Naruska July 2018 with a 1-hour interval. It has the standard deviation SD = 2.021 and average change *AC* = 1.437. The compression ratio with LTC is 2.439. The blue line is the FFT from data set: Naruska December 2018 with a 10-minute interval. It has the *SD* = 0.235 and *AC* = 0.106. The compression ratio with LTC algorithm is 19.241. Data sets have been normalized as both have been sampled with the same sampling rate. The frequency spectrums indicate that December 2018 with the 10-minutes measurement interval behaves more linearly because it has lower energy in high frequencies. The higher levels in high frequencies in July 2018 data indicate quick changes from measurement to measurement. Both data sets can be seen in function of time in Fig. 3.

In summertime, the temperature is changing on a daily basis approximately 15-20 degrees according to July 2018 measurement as can be seen in Fig. 3 on the left. With 1-hour measurement interval that means a significant change in value between two consecutive measurements. In wintertime, the change is not that big daily and there are long periods when the temperature remains quite constant as can be seen from December 2018 data (Fig. 3 on the right side) and specially when the measurement interval is short like 10 minutes in this example, then the data behaves quite linearly and remains during many consecutive measurements quite constant. Standard deviation and average change values indicate this.

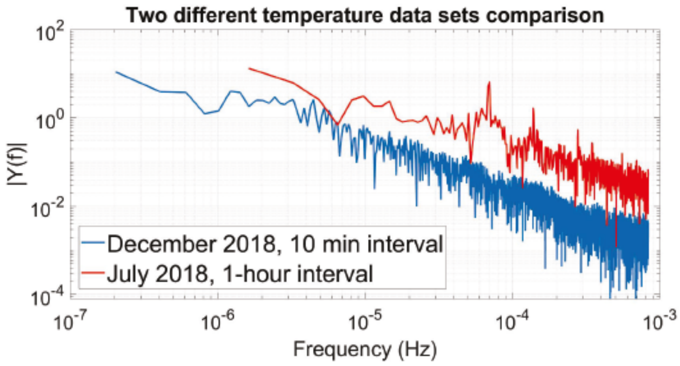

Fig. 2. The frequency spectrum of two temperature data sets.

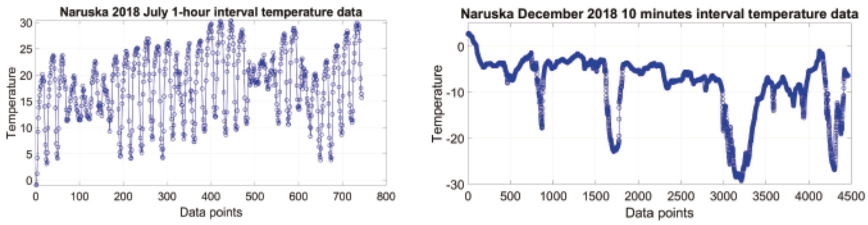

Fig. 3. The data sets with the highest *AC* and lowest *AC*.

The results presented here can be used to evaluate the suitability of these compression algorithms to compress certain data sets if the data characteristics are known. The choice between three different versions of the LRbTC can be made by evaluating the compression efficiency and computational complexity. In real measurement applications the future data set's characteristics are not known but the history data can be used to predict the probable data behavior and thus to choose the suitable algorithm.

4.2 Air Pressure Data Sets

The similar observations as for temperature data sets were done for air pressure data. The results can be seen in Fig. 4 where the left side illustrates the compression ratios of different compression algorithms in function of the *SD*, as described in this paper.

It can be seen in Fig. 4 on the left side that some data sets are dispersed slightly far from the other points. Those data sets are October 2018 data sets with all measurement intervals used. There is a clear error in the data because during October 2018 in air pressure data there is two times over 10 hPa air pressure change in 10 minutes. The air pressure change of more than 5 hPa/hour is rare and occurs only if there is an incoming thunderstorm [18]. Those two big sudden and atypical changes in air pressure rise the *SD* value relatively much; yet, it is not seen in *AC* value. For example, in October 2018 the air pressure data with 10-minute measurement interval have 4 064 measured values. Thus, those two big changes in measured values do not affect the *AC* value much. Thus, this behavior cannot be seen in Fig. 4 on the right side which is the compression ratio in function of *AC*.

The trend lines in Fig. 4 are fitting lines which are in this case 4^{th} degree polynomials (y = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5). The coefficients of fitting lines and the norm of residuals can be seen in Table 3 and Table 4.

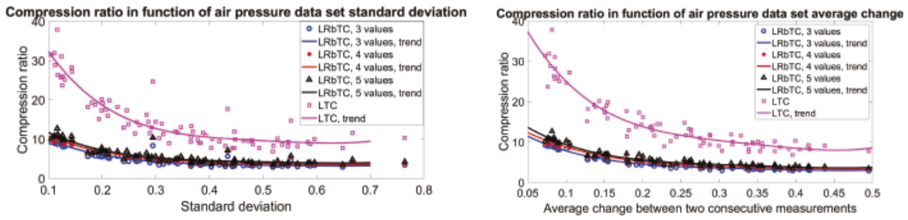

Fig. 4. Compression ratio in function of the *SD* (on the left side) and the *AC* (on the right side) for the air pressure data.

Table 3. Correlation between the compression ratio and standard deviation for air pressure data sets, polynomial coefficients and norm of residuals.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	168.35	148.3	215.54	620.34
p2	-351.23	-316.4	-439.59	-1272.5
p3	277.08	257.58	338.11	990.82
p4	-99.784	-96.847	-119	-353.54
p5	17.371	18.119	20.632	58.903
Norm of residuals	6.0527	7.6021	7.9608	23.117

Table 4. Correlation between the compression ratio and average change for air pressure data sets, polynomial coefficients and norm of residuals.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	743.39	555.94	961.93	2788
p2	-1051.1	-837.53	-1332.6	-3821.2
p3	561.64	482.03	694.26	1982.9
p4	-139.09	-129.9	-167.6	-481.17
p5	17.099	17.713	20.423	56.847
Norm of residuals	3.4408	4.2411	5.0525	18.369

The correlation in general is similar as with temperature data sets. The correlation is not linear but the general behavior can be seen in Fig. 4. The compression ratios are much higher for air pressure data than for temperature data, which indicates that air pressure data is rather linear in behavior and it is changing slowly. The compression algorithms based on data linearity are rather effective for this kind of data. The error bound used was 0.5 hPa which can be rather high in some applications. If the error bound is smaller, then the compression ratio is smaller.

4.3 Wind Speed Data Sets

Wind speed data from the Finnish Meteorological Institute's open data service is measured in a 10-minute average value [18]. In general, the wind speed has a slightly different behavior compared to the other environmental data. Wind speed can remain in 0 m/s for a while, and wind speed can also change quickly and there can be gusts. A 10-minute average measurement evens out the quick variation; however, the wind speed value remains in 0 m/s sometimes for long periods.

The results can be seen in Fig. 5. The correlation is again quite clear, and it is not as non-linear as with temperature and air pressure data sets.

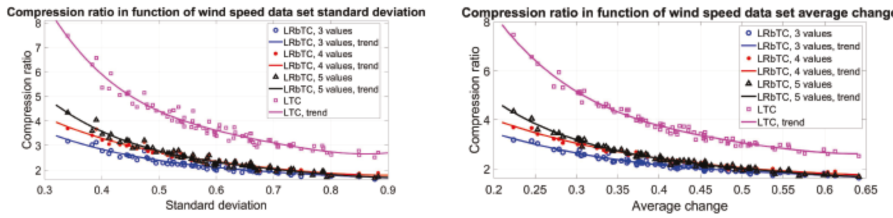

Fig. 5. Compression ratio in function of the *SD* (on the left side) and the *AC* (on the right side) for wind speed data.

The trend lines in Fig. 5 are 4[th] degree polynomials (y = p1*x^4 + p2*x^3 + p3*x^2 + p4*x + p5). The polynomial coefficients and the trend line norm of residuals can be seen in Table 5 and Table 6.

Table 5. Correlation between the compression ratio and standard deviation for wind speed data sets, polynomial coefficients and norm of residuals.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	11.793	22.697	43.109	127.38
p2	-38.01	-61.805	-126.05	-352.64
p3	47.695	67.138	141.3	372.1
p4	-28.642	-36.706	-74.163	-181.75
p5	8.804	10.612	17.597	38.372
Norm of residuals	0.82744	0.83827	0.92181	1.7419

Table 6. Correlation between the compression ratio and average change for wind speed data sets, polynomial coefficients and norm of residuals.

Coefficients	LRbTC, 3 values	LRbTC, 4 values	LRbTC, 5 values	LTC
p1	-13.996	23.544	113.51	274.47
p2	2.7715	-54.641	-240.49	-561.96
p3	21.602	54.168	196.51	443.05
p4	-18.535	-27.962	-76.098	-164.2
p5	6.2976	7.8401	13.9	27.494
Norm of residuals	0.55859	0.5111	0.63325	1.1643

5 Conclusions

According to the research presented in this paper, it is possible to predict the compression ratio for selected compression methods according to average change (*AC*) and standard deviation (*SD*) values of the data. The correlation is better between the compression ratio and *AC* than compression ratio and *SD* in every type of environmental data tested. This can be seen by comparing the norm of residuals value between *AC* and *SD* results. The *AC* value is also very easy to calculate from the history data. The history data can be used to predict the compression ratio with the selected compression method. The results can be used to choose a suitable compression method and with the estimated compression ratio it is possible to predict the battery powered wireless sensor node lifetime.

The best correlation is with wind speed data sets and the worst with air pressure data set. At the same time, the compression methods selected and tested are most efficient for the air pressure data, and the least efficient for the wind speed data, whereas the temperature data is between these.

References

1. Poornima., Ayyanagowadar, M.S.: Internet of things in agriculture: A review. In: Agricultural Reviews, Volume 39, issue 4. (2018) 338-340. doi: 10.18805/ag.R-1836
2. Salam, A., Shah, S.: Internet of Things in Smart Agriculture: Enabling Technologies. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland, (2019) 692-695. doi: 10.1109/WF-IoT.2019.8767306
3. Reddy, S.S., Azharuddin, M.R., Khan, K.: Importance of Internet of Things in Agriculture. In: International Journal of Recent Trends in Engineering Research. Vol.4(4), (2018) 372-373
4. Tzounis, A., Katsoulas, N., Bartzanas, T., Kittas, C.: Internet of Things in agriculture, recent advances and future challenges. In: Biosystems engineering 164, Elsevier. (2017) 31-48. doi: 10.1016/j.biosystemseng.2017.09.007

5. Abbasi, M., Yaghmaee, M.H., Rahnama, F.: Internet of Things in agriculture: A survey. In: 2019 3rd International Conference on Internet of Things and Applications (IoT), Isfahan, Iran. (2019) 1-12. doi: 10.1109/IICITA.2019.8808839
6. Väänänen, O., Hämäläinen, T.: Requirements for Energy Efficient Edge Computing: A Survey. In: The 18th International Conference on Next Generation Wired/Wireless Advanced Networks and Systems NEW2AN 2018, St.Petersburg, Russia. (2018) doi: 10.1007/978-3-030-01168-0_1
7. Sarbishei, O.: Refined Lightweight Temporal Compression for Energy-Efficient Sensor Data Streaming. In: 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), Limerick, Ireland. (2019) 550-553. doi: 10.1109/WF-IoT.2019.8767351
8. Luo, G., *et al*.: Piecewise linear approximation of streaming time series data with max-error guarantees, 2015 IEEE 31st International Conference on Data Engineering, Seoul. (2015) 173-184. doi: 10.1109/ICDE.2015.7113282
9. Grützmacher, F., Beichler, B., Hein, A., Kirste, T. and Haubelt, C.: Time and Memory Efficient Online Piecewise Linear Approximation of Sensor Signals. Sensors, 18(6), (2018) 1672. doi: 10.3390/s18061672
10. Li, J., Li, G. and Gao, H.: Novel ε-Approximation to Data Streams in Sensor Networks. In IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 6, pp. 1654-1667, 1 June 2015. doi: 10.1109/TPDS.2014.2323056
11. Väänänen, O., Hämäläinen, T.: Compression methods for microclimate data based on linear approximation of sensor data. In: NEW2AN 2019: Internet of Things, Smart Spaces, and Next Generation Networks and Systems: Proceedings of the 19th International Conference on Next Generation Wired/Wireless Networking, and 12th Conference on Internet of Things and Smart Spaces, Lecture Notes in Computer Science, 11660. Cham: Springer, (2019) 28-40. doi: 10.1007/978-3-030-30859-9_3
12. Hung, N. Q. V., Jeung, H. and Aberer, K.: An Evaluation of Model-Based Approaches to Sensor Data Compression. In*:* IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 11, pp. 2434-2447, Nov. 2013. doi: 10.1109/TKDE.2012.237
13. Giorgi, G.: A Combined Approach for Real-Time Data Compression in Wireless Body Sensor Networks. In: IEEE Sensors Journal, vol. 17, no. 18, pp. 6129-6135, 15 Sept.15, 2017.
14. Wee, C. K., and Nayak, R.: Alternate approach to Time Series reduction. In*:* 2018 International Conference on Soft-computing and Network Security (ICSNS), Coimbatore. (2018) 1-4. doi: 10.1109/ICSNS.2018.8573685
15. Belov, A. A. and Proskuryakov, A. Y. Time Series Compression in Telecommunication Systems for Environmental Monitoring of Polluting Emissions. In*:* 2018 XIV International Scientific-Technical Conference on Actual Problems of Electronics Instrument Engineering (APEIE), Novosibirsk. (2018) 391-395. doi: 10.1109/APEIE.2018.8545336
16. Schoellhammer, T., Osterwein, E., Greenstein, B., et al.: Lightweight temporal compression of microclimate datasets. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks IEEE Computer Society, pp. 516-524. (2004)
17. Aggarwal, Charu C.: Managing and Mining Sensor Data. Springer. 2013. doi: 10.1007/978-1-4614-6309-2
18. Finnish Meteorological Institute's open data–service. https://en.ilmatieteenlaitos.fi/opendata