Received: 26 October 2020 | Revised: 28 December 2020 | Accepted: 4 January 2021

IET Smart Grid

DOI: 10.1049/stg2.12010

ORIGINAL RESEARCH PAPER

Data reduction algorithm for correlated data in the smart grid

Zoya Pourmirza[1] | **Sara Walker**[1] | **John Brooke**[2]

[1]School of Engineering, Newcastle University, Newcastle upon Tyne, UK

[2]Freelance Consultant, Manchester, UK

Correspondence
Zoya Pourmirza, Newcastle University, Urban Sciences Building, 61 Wellington St, Newcastle upon Tyne, NE4 5TA, UK.
Email: zoya.pourmirza@newcastle.ac.uk

Funding information
Active Building Centre (ABC), Industrial Strategy Challenge Fund, Grant/Award Number: EP/S016627/1

Abstract

Smart grids are intelligent electrical networks that incorporate information and communication technology (ICT) to provide data services for the power grid. In this paper, the ICT requirements for monitoring and control of the neighbourhood area network level of the smart grid, with particular emphasis on making the ICT infrastructure energy efficient, are analysed. One approach to provide energy efficiency in the communication system is to develop a data reduction algorithm to reduce the volume of data prior to transmission. Thus, a data compression technique called DRACO (data reduction algorithm for correlated data) that shows a reasonable compression ratio while using network resources efficiently is designed and developed. DRACO can be applied to data with a high data sampling rate, and can transmit the essential information with compression ratios of 70%–99%. The results of applying DRACO on real data collected by devices located in the University of Manchester campus are discussed, followed by the evaluation and validation of DRACO by comparing it with other available techniques. Finally, it is concluded that DRACO is suitable for smart grid applications since it optimizes the network resource consumption and reduces the communication energy cost while maintaining the integrity and quality of data.

1 | INTRODUCTION

In planning for future electricity supplies, certain issues should be considered, such as increased electricity usage, climate change, and the conservation of natural resources. Some countries have investigated the transformation of their existing electricity grid into the smart grid. Smart grids have three main characteristics, which are to some degree antagonistic: provision of good power quality, cost reduction, and improvement in reliability. The need to ensure that these characteristics can be accomplished together brings a requirement to design and develop a rich information and communication technology (ICT) network alongside the electricity network [1]. Smart grids incorporate ICT to provide a data service for the generation, transmission, and distribution networks of the electricity grid. To reflect this structure, the ICT network is divided into three networks: the wide area network (WAN), neighbourhood area network (NAN), and home area network (HAN).

Deploying a large number of monitoring devices in the smart grid that transmits huge volumes of data can potentially saturate the devices' resources and consume energy at a rapid rate. Some of the key constraints of wireless sensor devices deployed in the smart grid are their limited resources, such as memory, battery, and processing power. These necessitate the development of techniques to utilise sensor resources more efficiently in order to achieve a better quality ICT network and a longer lifetime and time between maintenance sessions. As such, energy consumption considerations of ICT networks have emerged as a challenging concern. Energy awareness is important for both wired and wireless technologies, but for different reasons in each case. In wired networking, energy consideration is important because of the projected economic and environmental impacts, while in wireless networking, energy consideration is important because wireless sensor networks (WSNs) suffer from a lack of resources, such as a shortage in power supply. Difficulties arise when the deployed sensors in the smart grid are short of power; thus, a specific area of the grid is no longer being monitored. Given that real-time data is being used in the control layer, this problem may result in insufficiently accurate decision-making in the grid. In this research, we investigate a solution to ease this problem by developing a data reduction algorithm suitable for smart grid applications that can keep the integrity and quality of data.

According to the first order radio model [2], which quantifies the energy consumed for data transmission, there are two important factors affecting the energy consumption of data transmission. The first critical factor is d, the distance over which data will be transmitted. Considerations of grouping sensors to reduce the energy related to d are discussed in our previous paper [3]. The second important factor is k, the number of transmitted bits. Since there is a linear relation between the first order radio energy model and number of transmitted bits, by reducing the number of transmitted bits we could reduce the energy consumption of the wireless transmission process. Given that the energy consumed for transmitting one bit is equal to the energy for processing 1000 instructions [4], we can save energy by applying a data reduction technique before sending the data.

The novelty of our work is in developing a data reduction method within an ICT architecture at the NAN level of the smart grid. The lossless data reduction technique, which we will introduce here later, is called DRACO (data reduction algorithm for correlated data). Since DRACO is envisaged to be implemented on resource-constrained sensors, simplicity in the design of the algorithm is a key issue. DRACO should offer similar compression efficiency (for correlated data) as well-known compression algorithms such as Huffman [5], and Arithmetic [6]. It should be computationally more efficient, with fewer operations per logic gate, and therefore can require less time to compress a similar amount of data. This simplicity in terms of computation and time will ultimately imply energy efficiency over the whole process of data reduction. This energy awareness is important because we intend to reduce energy usage of wireless sensors per unit of data processed.

To summarise, we have designed and deployed an ICT architecture and have integrated DRACO within a working NAN. This is the medium voltage power network of the University of Manchester campus. Electricity grid monitoring is installed, and real data are available, at high frequency and accuracy. This is a rich test bed and has considerable variation in demand throughout the day and year (e.g. linked to the academic calendar). The core of our research is to design and use a lossless data reduction algorithm to extract data with a high data sampling rate and transmit the essential information, rather than sending all the data. This will ensure reduced energy consumption of data transmission by wireless sensors.

2 | RELATED WORK

We have classified data transmission techniques into three different categories. The first category is when sensors transmit data after receiving a request from the sink. The second category is when sensors send data whenever a threshold condition is violated. The third category is when sensors collect data and broadcast data continuously. The first and second categories are more energy efficient methods of data transmission because the data are being shipped with lower frequency. However, the NAN in our smart grid test bed necessitates the third category. This requirement stems from the fact that to understand the behaviour of the grid we need to sample data at a high rate at all times so that we can capture the fastest fluctuations. A review of literatures [7–9] reveals that energy efficient radio communication for continuous monitoring can be accomplished through different means such as duty cycling, optimising the routing algorithm, optimising the network topology, and in-network processing. In-network processing can be classified into two categories. The first is the data aggregation techniques being implemented in conjunction with WSN routing protocols. The second category of in-network processing methods is called data reduction, which is performed by implementing data reduction algorithms to reduce the communication cost by minimising the size of transmitted data. Applying data reduction will result in efficient bandwidth utilisation, and also in power saving, caused by minimised-size data transmission that will increase the network lifetime [10]. The technique used in this research to enhance the efficiency of the communication network belongs to the second category of in-network processing, that is, the data reduction method.

2.1 | Data reduction in smart grids

The current literature indicated that most of the studies in data reduction for smart grids concern smart meters [11], while data reduction for resource-constrained monitoring devices in smart grids (e.g. lightweight wireless sensors) has not been largely investigated. A study [12] has compared a number of data compression algorithms for smart meters by analysing their processing time and compression efficiency. The authors showed excellent compression efficiency can be achieved when investing a moderate amount of memory. However, to design DRACO, one of our requirements is to avoid using memory of the sensors. Authors in [13] have investigated the lossless compression of high-frequency smart meter data and have made recommendations on these algorithms. They have found that the compression ratio varies with data resolution and data type, which is in line with the finding in our research. However, these algorithms are implemented on smart meters, which are generally more powerful in terms of computation power and bandwidth compared with the wireless, resource-constrained sensors. Another study [14] proposed a new lossless compression algorithm to provide the best balance between the compression ratio and computational costs. The authors experimentally compared the data compression algorithms to improve energy efficiency in smart meters. These algorithms need memory and computation power to be implemented, as they either use dictionary or buffer, or a combination of multiple techniques. Therefore, they would not meet the resource-constrained limitation of sensors. Authors in [15] have proposed a compression algorithm for load profile data for smart meters using several data compression techniques combined. This is in contrast to the simplicity we require to design and implement our data reduction technique. In 2019, researchers [16] investigated an artificial neural network–based data mining technique to compress the meter readings on the customer side in an advance metering infrastructure (AMI) system, and decompressed the data at the data centre. Although this approach is suitable for the AMI, it cannot be implemented on resource-constrained sensors.

A study in 2018 [17] addressed the challenge of management of raw data in smart grids by comparing a number of lossless compression algorithms, to find the most suitable compression strategy for monitoring and analysis applications. Generally, these lossless algorithms are developed for aggregator or base stations, while our focus is on resource-constrained sensors. Klump et al. [18] proposed a two-stage lossless compression method for a synchrophasor measurements unit, which does not suffer from resource scarcity. Allalouf et al. [19] and Sari [20] have argued that the huge amount of data communication in smart grids will put the ICT network under substantial strain. They have examined the benefit that can be gained by applying lossy data reduction techniques by intermediate nodes to ease the flow of data. Khalifa et al. [21] have used a simulation to demonstrate that a centralised architecture, where hundreds of thousands of metering devices transmit their reading to the central data collection server, failed to adequately serve the smart grid infrastructure. Another study [17] compared 14 openly available lossless compression techniques to evaluate the compression efficiency and computation time for electricity grid data, and provided a set of recommendations. It proved that the compression efficiency strongly depends on the type of the datasets, and sampling rate, which is consistent with our results. A recent work by [22] studied how to map substation communication standard IEC 61850, with the constrained application protocol and the concise binary object representation. This work established more than 50% data reduction efficiency, compared with results based on http and web services. Other researchers [23,24] explored a data prediction method to reduce the amount of data in the communication channel of the smart grid. They used a lossy data reduction method, in which data is transmitted only if the predicted data and the actual data do not meet a satisfactory error threshold.

2.2 | Lossless data reduction

Since there is a limited amount of literature on lossless data reduction for resource-constrained sensors in the smart grid, we present the literature based on data reduction in other fields. Previously, researchers have investigated a range of data reduction techniques for different areas of science. However, some of these techniques are inappropriate for resource-limited networks, such as our test bed.

In this research we are interested in lossless data reduction techniques. Two of the most popular and most utilised lossless data reductions are Huffman coding and Arithmetic coding. Huffman coding [5] uses variable length coding and is the basis of much research in this area. The variable length coding converts the symbols into binary symbols on the basis of probability of occurrence of that symbol. Thus, most messages composed of repeated symbols can be compressed into a shorter bit stream. Arithmetic coding [6] is another method, which takes a stream of symbols and replaces them with a single number. Two main factors in the coding process in this method are the occurrence probability and the cumulative probabilities of a symbol sequence.

On the basis of the available literature, we have applied concepts for the WSN to a new domain. A key principle of our work is to develop a simple algorithm that can be performed on resource-constrained devices.

One of the techniques developed for the WSN, on which our data reduction algorithm is based, considers the differences between each sensor reading. The literature proposing a data reduction algorithm includes [4,25–29]. This technique is most appropriate where devices collect similar values, so values that are temporally adjacent are highly correlated.

Here we discuss two of these techniques that are based on the same principles as DRACO. The first one [30] has proposed a lossless data compression by exploiting the correlation between consecutive samples of data in the WSN and considering the principles of entropy compression. Considering these concepts, they have compressed the collected data with the help of a small dictionary. The algorithm functions as follows: It first finds the differences between each two successive values. Then, by utilising two's complement, it converts these differences into a set of least significant bits. Finally, it concatenates the compressed data with the Huffman variable length code. Authors have claimed that they have provided 66.99% and 67.33% compression efficiency for temperature data and humidity data, respectively. We believe the compression ratio depends on the characteristics of data and cannot be generalised. Later, we will compare DRACO's performance with that of this algorithm.

Another study [31,32] has modified the aforementioned approach [30] to enable data compression on a wider range of sensor types with higher standard deviations. This new version of the previously discussed algorithm is called Fixed Index. The authors state that the two approaches for applying data compression algorithms are either to implement a number of compression algorithms together on the data collected from a WSN or to develop a single compression algorithm that offers a satisfactory compression ratio. The new algorithm discussed in the research by Sornsiriaphilux et al. applies two modifications to the aforementioned algorithm [30]. In the original data compression algorithm [30], each set of compressed data is a combination of a group of high-order bits and low-order bits. High-order bits represent the number of bits needed to show the difference between each two consecutive values. Low-order bits represent the differences in the data. The first alteration is to use one's complement instead of using two's complement for showing the low-order bits. The second revision is to use Fixed Index instead of the Huffman variable length code in the high-order bits. The former modification will reduce the number of operations needed and the later modification will keep the length of bits constant, which is useful when the standard deviation of data increases. By this technique the length of the high-order bits is fixed to four bits. For example, when the difference between two values is eight, this eight should be represented in the low-order bits after one's complement has been applied to it. Therefore, eight will be represented as 0111. Next, the high-order bits represent the number of bits that low-order bits will occupy. These high - order bits will be identified through the four-bit Fixed Index table provided in the aforementioned paper. As such, since eight requires four bits (0100), therefore, the high-order bits will be 0100. Finally, the compressed data will be equal to 01000111. Sornsiriaphilux et al. have claimed that the Fixed Index algorithm performs better than the previous algorithm using the Huffman length code when the standard deviation increases. Here, we will later establish that DRACO offers a better compression ratio as the standard deviation of data increases, compared with these two algorithms.

3 | ICT ARCHITECTURE DESIGN FOR THE NAN

Electricity grids are commonly centrally monitored at the level of a national transmission grid, lacking monitoring at the NAN level. Here, we mainly focus on the NAN of the electricity distribution network. Having considered smart grid requirements at this level, both through the literature and a series of discussions with power engineer professionals, we developed a prototype ICT architecture, published in [33]. Here we briefly discuss the prototype architecture as a context to the data reduction.

Figure 1 depicts our proposed ICT architecture at the NAN level, which is based on hybrid communication technologies that integrate sensing and computation to enable monitoring, data gathering, and control and prediction of the future state of the network. The proposed ICT architecture has moved from a centralised architecture to a more decentralised system. A collection of single NANs (NAN 1, NAN 2 ...) communicate together to effectively construct a wider NAN.

The first layer of the architecture consists of smart meter monitoring systems that are the gateway from the HAN to the NAN and are used to monitor the building-level data. These monitoring devices are located in all the buildings in the campus test bed, recording electricity use data every 30 minutes. This data can be combined with data about real-time energy prices to offer an effective demand response control.

The second layer of the architecture is composed of hundreds of sensors situated in the streets. These are wireless sensors that are used to monitor environmental data such as temperature, light, humidity, and carpark monitoring. This information can be logged every second. We have utilised a cluster based WSN topology to reduce the data transmission range, which reduces energy demand in the system [34]. This layer helps in controlling the electricity network, since the sensors can provide information to help predict demand and to improve control actions. Examples of the controls offered by this layer include control of smart parking, control on the battery charging of electric cars, and control of street lighting.

The third layer incorporates substation monitoring and control. The reason we need a different layer of abstraction for this layer is that layer 2 is responsible for monitoring the environmental factors, whereas layer 3 is responsible for monitoring and control of the electricity network; these are two distinct functionalities. Regarding the test bed, this layer consists of eleven 6.6 kV substations that are equipped with

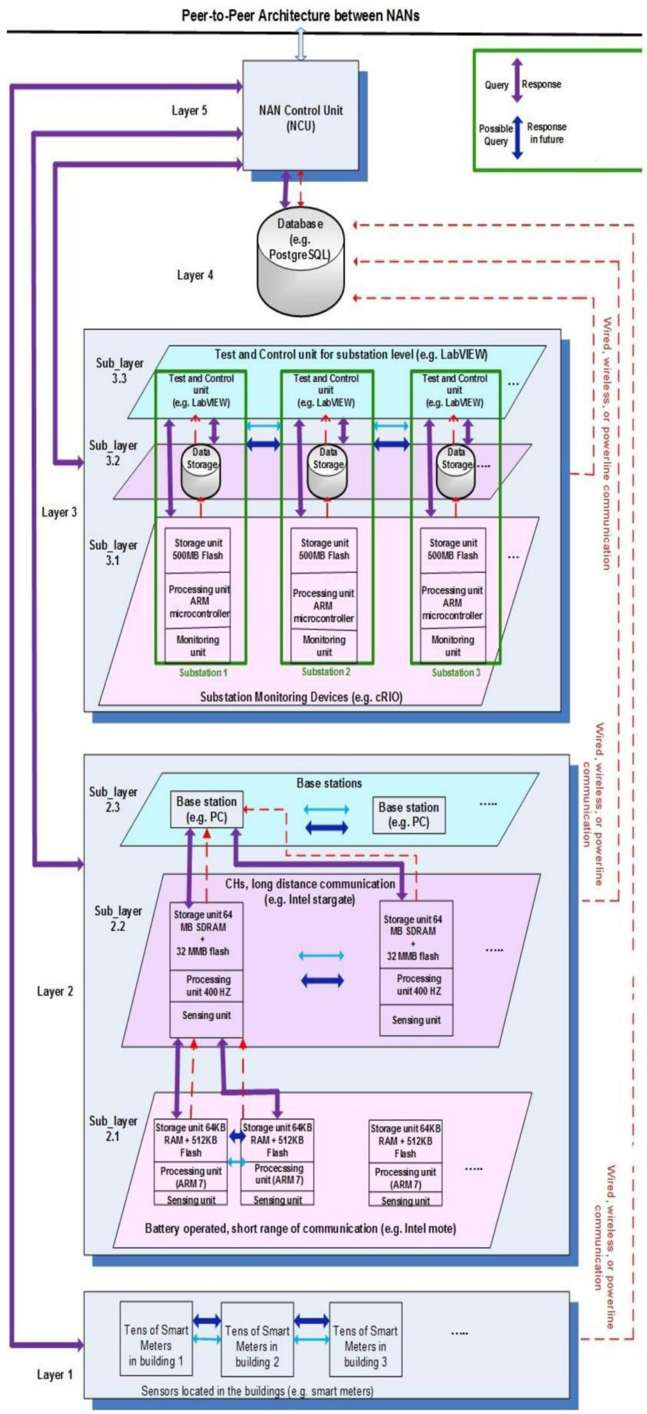

FIGURE 1 Proposed ICT architecture

sixteen monitoring systems, such as CompactRIOs (cRIOs). These cRIOs are running at 1–4 Hz (sensing one to four samples per second), measuring three-phase voltage, current, active power, power factor, voltage spectrum (eight channels for each phase), current spectra (eight channels), and frequency. For example, measurement of these parameters can enable identification of faults, for power quality analysis.

The fourth layer is the database (DB) layer that will store data received from the layers below and will feed them to the

NAN control unit (NCU) (layer 5). This layer has been implemented by a PostgreSQL database. The fifth layer (NCU) will apply control over the entire NAN area (a single unit of NAN) on the basis of information received from the DB layer. It should access sensing units directly in emergency situations, and indirectly through the DB in normal operating.

Figure 1 illustrates a single NAN only. We will consider here only one such NAN, namely, a university campus. Since each NAN can take optimal decisions for its own region, which are not necessarily the optimal decisions for the whole network, another layer of communication should be added to the top layer to enable each NAN to coordinate its decisions.

4 | DESIGN OF DRACO

In smart grid applications where the metering devices collect data with a high acquisition rate and transmit them to the NCU, a great degree of data correlation is common. We have therefore developed a simple data reduction algorithm that discards the redundant parts between each two consecutively measured values and transmits the changing parts only. These parts are a small portion of the original binary representation. This algorithm can improve the energy efficiency of the communication network by transmitting a smaller volume of data while keeping data integrity.

4.1 | Smart grid application requirements

Each of the smart grid applications has its own requirements in terms of sampling rate, data payload, latency, and reliability. The design of DRACO should not negatively impact these requirements. DRACO will meet the sampling rate requirement, by proving that when the sampling rate increases, the DRACO efficiency will increase as well. Also, DRACO does not impact the data payload, as DRACO is designed to reduce the transmitted data. It also does not reduce the reliability, as DRACO is a lossless algorithm and data can be fully recovered at the receiver side, and it does not affect the underlying transport mechanism that guarantees the delivery of messages. However, generally data compression algorithms can distress the communication latency.

Communication latency requirement 'T' is defined as [35]:

$$T = ta + tb + tc < Dth \tag{1}$$

where

- *ta* and *tc* are the delays for processing the message (e.g. data compression) at the source and destination;
- *tb* is the communication link delay;
- *Dth* is the required message delay that depends on the smart grid application.

While DRACO does not affect tb, the compression and decompression time effect ta and tc, respectively. DRACO execution time is in the order of seconds (on average 0.25 seconds for each compression and decompression) and can meet the low latency requirements for some of the smart grid applications for the NAN area that are in the order of seconds, including [36] meter reading (on demand from meters to utility) < 15s; electric service prepayment (from utility to customers) < 30s; distribution automation < 5s; customer information and messaging customers < 15s; distribution customer storage (charge/discharge command from distribution application controller to the storage) < 5s; electric transportation (utility sends price info to plug-in hybrid electric vehicles) < 15s.

4.2 | DRACO-1

The DRACO algorithm works as follows in order to reduce the size of the text files being transmitted by the monitoring devices located in our test bed. At the sender side of our proposed algorithm, the digit-based representation of signed decimal values is ready for transmission. DRACO first reads and keeps the sign. It converts the value to a positive digit-based decimal value. Then it converts the positive digit-based decimal value to a positive digit-based integer value by multiplying by a sufficiently large power of 10. After these modifications, at the beginning of each round of transmission, the sender will transmit the modified full value of the first measured data.

This full value indicates the start of each round of transmission. Frequent transmission of the updated full value of the measured data will reduce the risk of data loss in the communication network. If a number of measured data are missed in the transmission channel, then the receiver side will decode the received values incorrectly. In order to prevent such data loss, our strategy is to send the modified full measured value on a regular basis. A decision on how often the full value needs to be transmitted depends on the requirements of the user of the system. In the test scenario, we are dealing with file transmission. Each file contains data collected for the past one hour, which are logged every second. In this case, the first value of each file will be transmitted as the original full value, and the data reduction will be applied on the second value onwards, until the end of the file. The decision on the size of the original file depends on the application for which these data are collected.

To discard the redundancy between the two consecutive values, DRACO works as follows. After taking the digit-based representations of a decimal value, read and keep their signs, and change them to digit representations of absolute integer values; then convert these absolute integer values into digit-based binary representations. Subsequently, we initiate the data comparison on the binary representations by applying XOR on each two consecutive values, for example, beginning with the far left digit of the binary representation 111001 and the consecutive 111110. The first three-digit pairs of the first value and consecutive value are pairs of 1, so XOR returns three 0. The next digit pair are 0 and 1 and so XOR represents this as 1.

Similarly, the next digit pair are 0 and 1, so XOR returns a 1. The final digit pair are 1 and 0, so XOR represents this as 1. The difference between the binary representation of value one (+5.7) and consecutive value two (−6.2) is therefore 000111, or 111, as shown in Table 1. Next, the XORed values will be converted back to absolute digit-based representations of integers. Finally, they will be multiplied by their signs and will be transmitted to the receiver point. Thus, in this process, the signed digit-based representations of integer value that are the result of the XOR process will be transmitted.

On the receiver side, the reception device will receive a file comprising the first full value and subsequent changed parts only. It will first read and keep the signs of each value and change them to positive values. It then converts the base 10 representations to the binary representations. Subsequently, it can reconstruct the original value by applying the XOR to the value 'n' and to the reconstructed value 'n-1'. For example, 111 ('n') can be reconstructed by comparing with 111001 ('n-1'), to give 111110. Binary reconstructed values are then converted into digit-based representations of integer values and multiplied by their signs. Finally, these values will be converted into the original values with a decimal fraction part, on the basis of how many decimal points are needed.

This algorithm is called DRACO-1. Tables 1 and 2 demonstrate a simple example of the compression and decompression of DRACO-1 on the sender and receiver sides.

4.3 | DRACO-2

After several rounds of testing the data reduction method and analysing the results, we recognised that DRACO-1 could be improved to offer more compression efficiency for data with a higher correlation degree. The improved DRACO-1 is called DRACO-2, which, however, can only be applied in cases where correlation between the collected data is very high (e.g. this is the case for frequency and voltage readings).

The difference between DRACO-1 and DRACO-2 is that, on the transmitter side, after applying XOR and converting the binary values back into integer values, if any consecutive value appears as '0', DRACO-2 will only send one instance of previous value together with the number of repetitions. Tables 3 and 4 demonstrate a simple example of the compression and decompression of DRACO-2 on the sender and receiver sides. Although DRACO-2 is not as stable and general as DRACO-1, DRACO-2 is valuable for data in high volumes with strong correlations, and in these cases, it can perform better than DRACO-1.

DRACO offers similar execution time for compression and decompress, since similar computation power and similar functions are used on both sides of sender and receiver. This is beneficial for the performance of power systems and applications where latency is the main concern.

DRACO can also provide a low level of security for communication between devices, since we are transmitting a modified or cipher data, and not the original data. Similar to one-time pad cryptography, which uses information theory to create a cipher text, DRACO uses XOR gate to create a cipher text. DRACO avoids the need for the sender and receiver to carry a copy of the cipher key (which makes the one-time pad cryptography communication vulnerable to revealing the key). However, DRACO uses the preceding value as the key to cipher the consecutive value. This approach avoids the insecure-implementation vulnerability of a cipher key.

TABLE 1 The transmitter side (DRACO-1)

Measured value	Matrix of signs	Absolute value	Binary representation	XORed value	Absolute value of reduced part	Matrix of signs	Sent value (signed reduced value)
+5.7	+1	57	111001	111001	57	+1	+57
−6.2	−1	62	111110	111	7	−1	−7
+6.1	+1	61	111101	11	3	+1	+3
−5.7	−1	57	111001	100	4	−1	−4
+6.3	+1	63	111111	110	6	+1	+6

TABLE 2 The receiver side (DRACO-1)

Received value	Matrix of signs	Absolute value	Binary representation	Reconstructed XORed value	Absolute reconstructed value	Absolute reconstructed value with decimal points	Matrix of signs	Signed final reconstructed value (original value)
+57	+1	57	111001	111001	57	5.7	+1	+5.7
−7	−1	7	111	111110	62	6.2	−1	−6.2
+3	+1	3	11	111101	61	6.1	+1	+6.1
−4	−1	4	100	111001	57	5.7	−1	−5.7
+6	+1	6	110	111111	63	6.3	+1	+6.3

TABLE 3 The transmitter side (DRACO-2)

Measured value	Matrix of signs	Absolute value	Binary representation	XORed values	Absolute value of reduced part	Matrix of signs	Signed value of reduced part	Final sent value (signed value with number of repetition)
+49.5	+1	495	111101111	111101111	495	+1	+495	+495,0
+50.3	+1	503	111110111	11000	24	+1	+24	+24,4
+50.3	+1	503	111110111	0	0	+1	0	−53,0
+50.3	+1	503	111110111	0	0	+1	0	+49,0
+50.3	+1	503	111110111	0	0	+1	0	+194,2
+50.3	+1	503	111110111	0	0	+1	0	
−45	−1	450	111000010	110101	53	−1	−53	
+49.9	+1	499	111110011	110001	49	+1	+49	
+30.5	+1	305	100110001	11000010	194	+1	+194	
+30.5	+1	305	100110001	0	0	+1	0	
+30.5	+1	305	100110001	0	0	+1	0	

TABLE 4 The receiver side (DRACO-2)

Received value	Ordered value	Matrix of signs	Absolute value	Binary representation	Reconstructed XORed value	Absolute reconstructed values	Absolute reconstructed value	Matrix of signs	Final reconstructed value
+495,0	+495	+1	495	111101111	111101111	495	49.5	+1	+49.5
+24,4	24	+1	24	11000	111110111	503	50.3	+1	+50.3
−53,0	0	+1	0	0	111110111	503	50.3	+1	+50.3
+49,0	0	+1	0	0	111110111	503	50.3	+1	+50.3
+194,2	0	+1	0	0	111110111	503	50.3	+1	+50.3
	0	+1	0	0	111110111	503	50.3	+1	+50.3
	−53	−1	53	110101	111000010	450	45	−1	−45
	+49	+1	49	110001	111110011	499	49.9	+1	+49.9
	+194	+1	194	11000010	100110001	305	30.5	+1	+30.5
	0	+1	0	0	100110001	305	30.5	+1	+30.5
	0	+1	0	0	100110001	305	30.5	+1	+30.5

4.4 | Subtraction

Instead of applying XOR we could also use subtraction. When testing both subtraction and XOR for our collected data and comparing the size of the compressed data, we found that both these techniques result in a similar compression ratio. This is the result of the technique we use to represent data.

Table 5 compares the compression ratio for four parameters—voltage, current, frequency, and total power factor—using DRACO-1 and subtraction.

The logic diagram of XOR is simpler than that of subtraction; thus, it is easier to be executed on various types of devices, such as energy-constrained sensors. Therefore, XOR gives us almost the same compression as subtraction does but with simpler and fewer operations. In order to implement XOR, two inputs (A and B) and one output (Q) are required. The truth table in Table 6 shows the behaviour of XOR. XOR is true (the output is equal to 1) when only one of the inputs is true, expressed using Boolean algebra.

XOR can be implemented with only three gates as shown in Figure 2. In order to implement the subtraction, three inputs (A (minuend), B (subtrahend) and (borrowin)) and two outputs (D (difference bit) and (borrow out)) are required. Table 7 shows the behaviour of the subtraction process.

Accordingly, the following logical expression can be used to implement subtraction.

TABLE 5 Compression efficiency of DRACO and subtraction

	Original size (bit)	DRACO-1 size (bit)	Ratio (%)	Subtraction size (bit)	Ratio (%)
Voltage	230464	14822	93	14410	93
Current	230464	17700	92	15734	93
Frequency	371456	14408	96	14408	96
Total power factor	371288	14520	96	14406	96

TABLE 6 Truth table for XOR

A	B	Q	$A.\bar{B} + \bar{A}.B \equiv (A + B).(\bar{A} + \bar{B})$ (2)
0	0	0	Using de Morgan's Law we can convert this to
0	1	1	$(A + B).(\bar{A} + \bar{B}) \equiv (A + B).(\overline{A.B})$ (3)
1	0	1	
1	1	0	

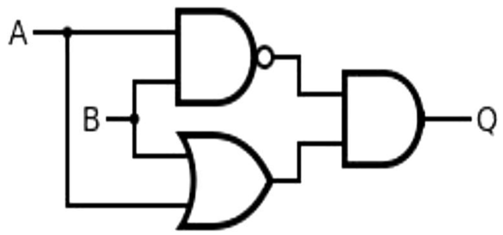

FIGURE 2 Logic diagram of XOR

$$D = (A \oplus B) \oplus Bor_{in} \tag{4}$$
$$Bor_{out} = \bar{A}.(B \oplus Bor_{in}) + B.Bor_{in} \tag{5}$$

Figure 3 shows the logic diagram of subtraction that requires seven gates, two of which are XOR gates, to be implemented. Thus, it has been shown that implementing a subtraction operation is more complex than implementing a XOR operation.

4.5 | Bit by bit comparison

The reason we chose XOR over bit by bit comparison is that we realise that a problem exists when using a normal comparison between bits.

In a normal comparison, to discard the redundancy between values, we start by comparing each of two consecutive digit-based binary representation values. The comparison of these values starts with comparing the most significant digit in the binary representation of that value. This process will continue until the first difference between two digits is found. Then the comparison will be halted, and the redundant digit will be discarded. The bits starting from the first dissimilar digit (which are called the changed digits) are then converted into base 10 representations and multiplied by their signs before transmission.

TABLE 7 Truth table for subtraction

A	B		D	
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

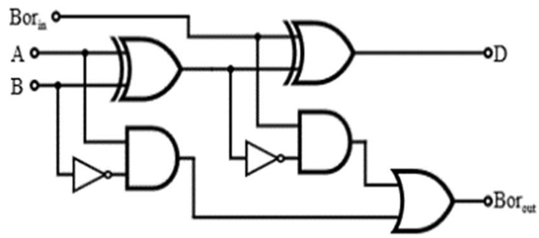

FIGURE 3 Logic diagram of subtraction

The receiver side will first read the signs and convert the unsigned numbers into positive digit-based integers. Then it will convert the base 10 representations into binary representations. This process is needed to implement the binary comparison. Subsequently, it can reconstruct the original value by knowing only a sample of the modified full data that was transmitted at the beginning of each round of transmission. Having one sample of modified full data discloses enough information to generate the original values out of the received compressed values. This information could be such as the number of bits each value occupies, and the pattern of bits. Then the signs will be added to the digit-based integer values, and finally, these values will be converted into the original values with a decimal fraction part.

Our experiments reveal that a problem with this system occurs when the first changed digit appears as '0' rather than '1'. In this case, when converting the binary representation of changed digits back into the digit-based integer, the '0's before the first '1' will be ignored. Thus, on the receiver side, when the reception unit receives the digit-based integer value, it is not able to reconstruct the correct original value because the number of '0' before the first '1' is unknown. Tables 8 and 9 demonstrate examples of the case when the bit by bit-based algorithm malfunctions.

For example, the binary representations of 61 and 57 in the third and fourth rows are '111101' and '111001'. After comparing these values with their previous values and discarding the redundant digits, the remaining digits are '01' and '001', respectively. When converting them back into the integer representation, the '0's before the '1' will be ignored. So, when the receiver side receives the changed parts of these two rows (integer 1), it cannot identify how many '0's appear before the first '1'. When rebuilding the values 61 and 57, both reconstructed values will be 63.

Therefore, to avoid the aforementioned problem, we proposed a solution, which is to use exclusive OR (XOR). The difference between utilising the XOR and the normal bit by bit comparison is that when the XOR is used, the first changed digits always appear as '1'. This is a result of the fact that the XOR between two similar digits will result in 0 ($1 \bigoplus 1 = 0$ and $0 \bigoplus 0 = 0$) and the XOR between two dissimilar digits will result in 1 ($1 \bigoplus 0 = 1$ and $0 \bigoplus 1 = 1$). Thus, when converting the changed digits (which always start with 1) into the integer format, we trust that we are not losing any digits.

5 | IMPLEMENTATION

As discussed earlier, we implemented DRACO in a university electricity network at the University of Manchester. Our technique is purely a compression method that keeps the quality and integrity of data. DRACO is lossless after the truncation of data. However, the decision to apply truncation or not depends on the user requirements. Although the proposed algorithm is applicable to diverse data sizes and characteristics, it is most suitable for data where consecutive values vary only in the least significant digits when represented as binary, that is, the rate of change of the data is slow with respect to the sensing rate. The efficiency of this compression process, which depends on the degree of data correlation, can be described by the following formula (if the data size is inflated, we indicate this in the results by a minus sign, as given by the formula).

$$\eta_c = \frac{S_o - S_c}{S_o}\% \tag{6}$$

where η_c is the compression ratio;

S_o is the original data size;

S_c is the compressed data size.

DRACO is economic in terms of the execution time, and the compression and decompression time are similar. The tests on different sources of data indicate that the execution time of DRACO is acceptable, which will be discussed later. It is efficient in terms of communication energy consumption, since this is dependent on the number of transmitted bits.

5.1 | Evaluation of DRACO on simulated data

In this section, first we evaluate DRACO with seven data sets that were simulated via LabVIEW and cRIO. In Figure 4, we present the results from the performance of algorithm for seven different data types, each comprising 17,982 data

TABLE 8 The transmitter side (bit by bit)

Counter	Measured value	Measured binary representations	Binary representations after comparison	Reduced value ready for transmission
1	57	111001	111001	57
2	62	111110	110	6
3	61	111101	01	1
4	57	111001	001	1
5	63	111111	111	7

TABLE 9 The receiver side (bit by bit)

Counter	Received value	Received binary representations	Binary reconstructed representations	Original (reconstructed decimal value)
1	57	1111001	111001	57
2	6	110	111110	62
3	1	1	111111	63
4	1	1	111111	63
5	7	111	111111	63

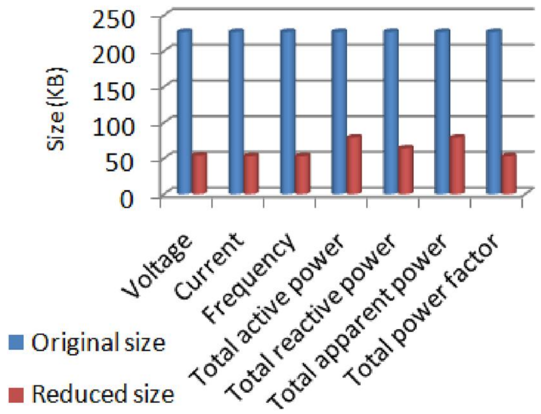

FIGURE 4 Effect of DRACO on simulated data [37]

values. These are the simulated data for one hour, being calculated approximately five times a second. Using formula (1.5) our simulation reveals that we are able to achieve over 70% efficiency, on average, in terms of reduced data volume. This would almost equate to a saving of 70% in energy consumption for data transmission if using DRACO.

5.2 | Evaluation of DRACO on real data measured at substation level

We now examine the results of applying the DRACO compression algorithm using data collected from the substation level of the university electricity network. The following are the results from some of our experiments on real data collected by the cRIO devices located in the 6.6 kV-substations in the University of Manchester campus. Comparing the efficiency of the algorithm on both simulated data and real data, we obtained similar results.

Both DRACOs were tested against a period of data to implement our experimentation. A sample one-day period is provided here. We demonstrate the DRACOs over 24 hours of real data (collected from 8:00 AM on 24 April 2013 to 8:00 AM on 25 April 2013) to assess the efficiency of the data reduction algorithm during different periods within a day, covering peak hours and non-peak hours. We used MATLAB to implement DRACO and ran codes on a standard Windows 10 x64 PC (Intel(R) Core(TM) i7-8650U CPU @ 1.90 GHz 2.11 GHz). In order to measure the average execution time of DRACO we used MATLAB functions tic and toc before and after the command line tool execution. The tic function records the current time, and the toc function uses the recorded value to calculate the elapsed time. We ran all the experiments in the same environment using the same laptop and MATLAB version to fairly compare DRACO and other existing compression algorithms.

In these experiments we have considered voltage, current, and frequency data because their characteristics are different. The variations in voltage are less, whereas variations in current are much more substantial, with a higher rate of changes. Additionally, the rate of change in the frequency data is very low, unless some problem occurs in the electricity grid. Therefore, our data reduction algorithm will perform differently on these three types of data.

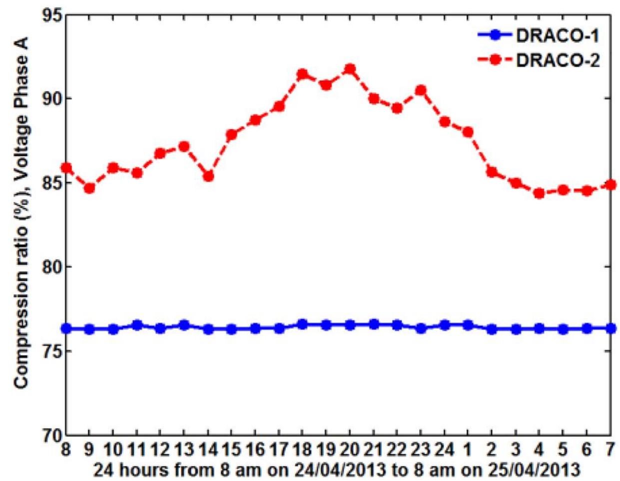

FIGURE 5 24 hours of compressed voltage data

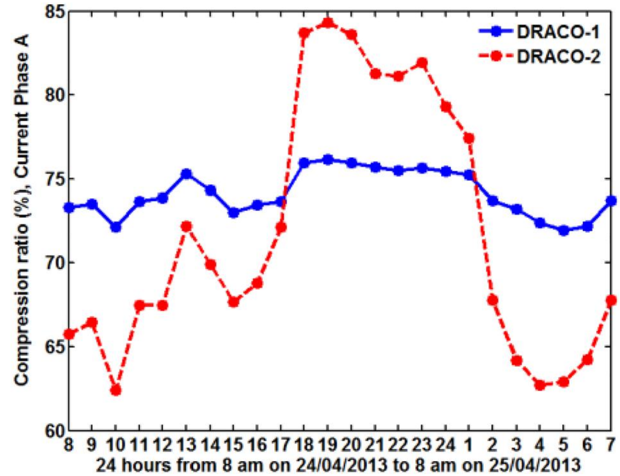

FIGURE 6 24 hours of compressed current data

Figure 5 compares DRACO-1 (blue line) and DRACO-2 (red dotted line) using voltage data. It shows that DRACO-2 has a better compression ratio, and is therefore a more efficient algorithm, for voltage compression. Moreover, between 16:00 and 1:00 we achieve a better compression (over 89%) ratio, which means the network voltage is steadier over this period, and as a result, the data correlation is higher during this period of time. It is recommended that DRACO-2 be used for compression of voltage data at all times.

Figure 6 compares DRACO-1 (blue line) with DRACO-2 (red dotted line) using the data for current amplitude. First, we achieve a lower compression ratio for both algorithms for current data compared with voltage data, due to its greater variation. Second, results show the best compression ratio is achieved roughly between 17:00 hours to 1:00, which (as for the voltage data) indicates a period of relatively steady network current. Comparing DRACO-1 and DRACO-2, we suggest a switching algorithm for the monitoring system for current

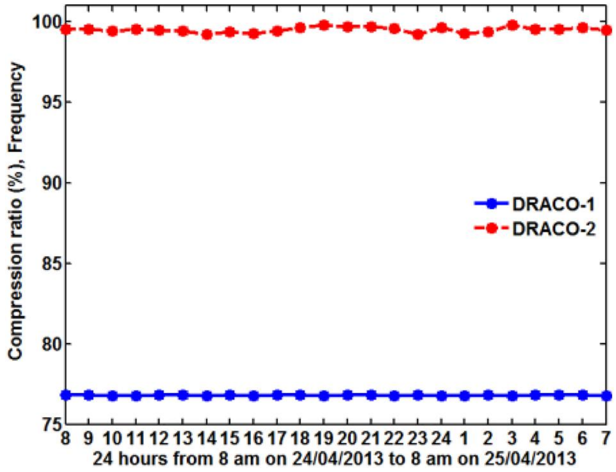

FIGURE 7 24 hours of compressed frequency data

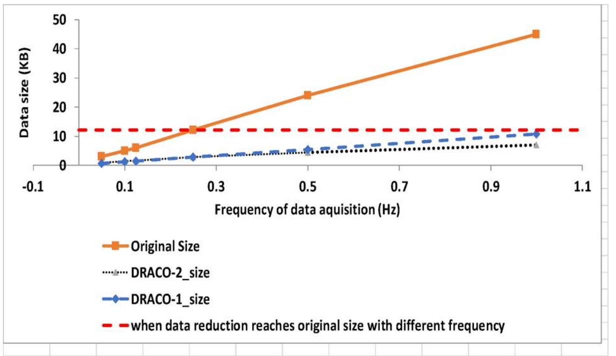

FIGURE 8 Data acquisition rate evaluation

data. Between 17:00 and 1:00 the compression ratio of DRACO-2 exceeds the compression ratio of DRACO-1; therefore, during this time period we could use DRACO-2 as the compression algorithm for the current data, and for the rest of the day DRACO-1 could be applied.

Figure 7 compares DRACO-1 (blue line) and DRACO-2 (red dotted line) using the frequency data. The frequency data has a lower rate of change compared with the voltage and current data. DRACO-1 gained over 76% compression efficiency, while DRACO-2 achieved over 99% compression efficiency. Thus, it is suggested that DRACO-2 be used for compressing the frequency data at all times. This is an interesting result for frequency, because without using DRACO-2, it would be very wasteful in terms of data transmission and hence energy to send data at such a high sampling rate. It is very important to monitor frequency data at a high sampling rate because frequency could go out of range very quickly, which is potentially catastrophic for the system. As a result, this is an excellent example of the value of DRACO compression, with an over 99% compression ratio.

5.3 | Evaluation of the effect of different data sampling frequencies

An experiment was designed to assess the effect of various sampling rates on the efficiency of DRACO. We examined the data being logged with different frequencies, such as once every second (1 Hz), once every two seconds (0.5 Hz), once every four seconds (0.25 Hz), once every eight seconds (0.125 Hz), once every 10 seconds (0.1 Hz), and finally, once every 20 seconds (0.05 Hz). Figure 8 shows that, as the frequency of the data acquisition rate increases, the original size of the data will increase. However, as we start to sample more frequently, the correlation between each two consecutive values is higher and DRACO performs best on data with stronger correlations. So the difference between the original data size and the DRACO reduced data size also grows. Thus, with a higher sampling rate we could transmit more data about the network, and with the use of the DRACOs we could send this data more efficiently in terms of data volume.

This means by using DRACOs we can achieve a better compression ratio for a higher acquisition rate. This result confirms the fact that, when data are being sampled at a faster rate, the correlation between each two consecutive values is higher and DRACO performs best on data with stronger correlations. In a simplified scenario, the result can also be beneficial in terms of bandwidth utilisation. In case we want to transmit data with a higher frequency, which results in a higher volume of data, we can transmit more information using the same amount of bandwidth by applying DRACO.

5.4 | Evaluating the effect of different bit rates

In this section we examine the effect of DRACO on the bit rate. This experiment was carried out to determine the link between significant events in the actual data profile and the maximum/minimum bit rate. Since our analysis shows that the total active power data has one of the highest variations of all the data types, here we have selected this as the test data. Consequently, the changes in the bit rate can be visibly seen. We have estimated the bit rate by dividing the size of the reduced data by time, which is 30 seconds in this experiment (byte/second). Figure 9 shows the actual profile of the total active power and the bit rate [38] after applying DRACO. Analysing both the behaviour of the original data and the data transfer rate, we realise that, when there are fewer spikes and minimum changes between consecutive data, the data transfer rate is low. Conversely, when there is a big change in the data values and when the data variation is high, we observe a higher transfer rate. This result can be seen in the two ringed areas in Figure 9 corresponding to the areas where minimum and maximum rate of change happen in the top figure, and their corresponding transfer rate in the bottom figure. The solid black ring (880s–910s) in the bit-rate graph shows the maximum data transfer rate, where data values vary significantly and has the biggest change among the time periods in

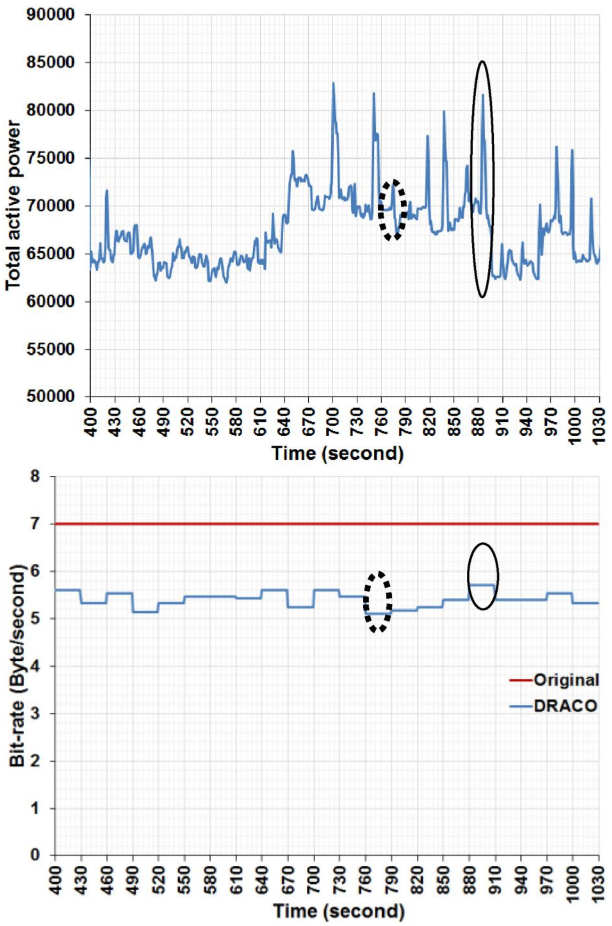

FIGURE 9 Total active power (kW) (top figure) and the corresponding bit rate (bottom figure)

this profile. The dashed black ring (760s–790s) demonstrates the lowest bit rate, for a period where data are very similar. Although there is a small jump in the dashed black ring time period, the rest of the data have the most similarities (very steady), which leads to a bit-rate minimum. In summary, the correlation between the two graphs indicates the dependency of the data transfer rate on the rate of change of the quantity being measured (e.g. total active power).

6 | COMPARISON WITH OTHER WORKS

To assess the efficiency of the DRACOs we have compared their performance with that of other data reduction algorithms. Thus, we have followed two approaches: In the first approach, we have produced data as described in other published works, then applied DRACOs to this generated data, and compared the compression performance of DRACOs and other data reduction algorithms. In the second approach, we have applied DRACOs and other developed algorithms to our test data and compared the compression performance of DRACOs and other data reduction algorithms.

6.1 | The first approach: comparison of DRACO with other algorithms on generated data

Tests were carried out to compare the compression efficiency of DRACOs with that of two other existing data reduction algorithms specifically developed for sensor data on the basis of the Huffman and Fixed Index methods (both methods have been discussed earlier in section 2). These two algorithms were chosen because they have been designed particularly for sensor networks with resource limitation problems, and use low-complexity data reduction techniques suitable for such environments. Moreover, their designs are also based on the correlations between each of two consecutive data values, which will reduce the number of transmitted bits.

In this scenario we tested the DRACOs on data that were generated using the method we describe next. Thus, we are evaluating the DRACOs with more general data that are not tailored just for electricity networks. Since we can collect data across different sensing environments with a wide range of standard deviations, we can compare compression algorithms for data with a range of standard deviations.

In this experiment the data reduction algorithm, which is based on Huffman, is called Differential_Huffman, and the one that uses Fixed Index is simply called Fix_index.

In order to initiate a comparison, we first started generating test data. The data was generated according to the procedure described in the research by [32]. We produced 14,400 samples of random data using normal or Gaussian distribution. It was assumed that these data were generated by a sensor at a rate of once a minute during a 10-day period ($60\times24\times10=14{,}400$ data points). This is a realistic assumption for data measurements derived from environmental phenomena. The mean of the simulation was assigned to 0, while the standard deviations ranged from 0 to 250.

A high standard deviation and less correlated data might result in lower compression using DRACO. The DRACOs are efficient for compression when the data are scattered fairly narrowly around the mean, rather than being widespread. However, the standard deviation does not indicate how data are linked in time (e.g. with a same standard deviation, values could be differently linked in time). These two issues (high standard deviation and lower data correlation) will decrease the effectiveness of DRACO. Therefore, a range of standard deviation data samples were used.

Then we applied DRACO-1, DRACO-2, Differential_Huffman, and Fix_index on these generated data and determined the compression ratio. Therefore, we have compared these four algorithms in similar situations.

In the investigations by Sornsiriaphilux et al. [31], the decimal precision for test data is not given when executing their algorithms. However, the decimal precision is important for DRACOs. Hence, we have considered different precisions in two scenarios, called the 'best-case scenario' and the 'worst-case scenario'. The best-case scenario is when we considered only one decimal point and the worst-case scenario is when we considered full decimal points. Note that DRACO is designed for electrical network monitoring, where small variations in the decimal points are not important from the control point of view. For monitoring of the University of Manchester Campus electricity network, the precision requirement is therefore a best case. However, future investigations may require the full precision of the sensor readings. This would exploit the lossless nature of our technique, which can still produce compression in this limit of precision.

The results of these comparisons are shown in Figure 10. It shows that, in the best-case scenario, both DRACOs performed better than Differential_Huffman and Fix_index. Moreover, the compression ratio of our methods is more stable over the ranges of standard deviations, whereas the compression ratio of the other two algorithms reduces as standard deviation increases. Thus, as the standard deviation of the data increases, we do not lose significant compression efficiency for DRACOs.

In the worst-case scenarios, when the standard deviation value is smaller, the Differential_Huffman and Fix_index perform better than the DRACOs. However, as the standard deviations increase, DRACO-1 performs better than these two algorithms. Also, at the largest standard deviation of 180 and above, DRACO-2 performs better than the Differential_Huffman. DRACO-2 is a steadier compression algorithm (has a consistent compression ratio performance) compared with the Differential_Huffman and Fix_Index (for which the compression ratio decreases rapidly as the standard deviation increases, for both algorithms).

6.2 | The second approach: Comparison of DRACO with other algorithms on electricity network data

In the second round of tests we compared DRACO-1 with two different compression algorithms. These algorithms are the Huffman and Arithmetic, which are among the best-known data reduction techniques. We considered the binary transmission technique in this analysis as discussed earlier. Comparisons between these algorithms were made by applying our data as an input to the Huffman, Arithmetic, and DRACO in similar situations. These data were collected from monitoring devices during a period of one hour, sampled once every second, located in the electricity network substation. The input data were rounded off to the same precision for all algorithms.

Earlier we showed that DRACO-1 is a more stable algorithm compared with DRACO-2. DRACO-2 is suitable for downloaded data with higher correlation, such as voltage and frequency, while DRACO-1 is suitable for live data stream and can be generalised and be applied across all the measured values. Therefore, we kept DRACO-2 out of the comparisons for the size and compression ratio in this subsection. The results, as shown in Table 10, indicate that Huffman compression ratio is only 1%–2% better than DRACO-1 for data streams with smaller variations, such as voltage, frequency, and total power factor. However, for the data streams with higher

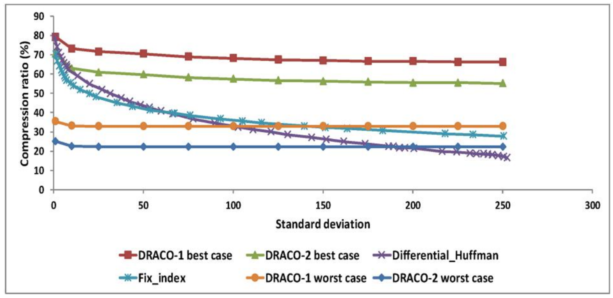

FIGURE 10 Comparison of data reduction algorithms

variation, such as current, DRACO-1 performs 4% better than Huffman. This is a result of the fact that Huffman obtains its compression efficiency on the basis of frequency of occurrence of each value. Since the rate of reappearance of each value in current was low in our data, this makes Huffman perform poorly compared with DRACO-1. Also, our experiments revealed that the average of Huffman's execution time is higher than the DRACO's (see Table 11). Comparing the results of the Arithmetic algorithm with those of DRACO-1 reveals that, for data streams with low variations and high frequency of occurrence, such as total power factor and frequency, Arithmetic compression ratio is 2% higher than that of DRACO-1. However, for data streams with higher variations, such as voltage, the compression ratio of DRACO-1 is 2% higher than Arithmetic's. For data streams with even greater variations, such as current, the compression ratio of DRACO-1 is 7% higher than Arithmetic's. This is the result of the fact that compression efficiency of Arithmetic coding depends on the frequency of occurrence of data. Also, our experiments revealed that the average execution time of the Arithmetic is higher than the DRACO's (see Table 11).

7 | CONCLUSION

We have developed an energy aware architecture to enable sensor networks to transmit data at a reduced energy consumption. In order to incorporate energy awareness in our architecture, we have developed data reduction algorithms that can be used as an appropriate technique.

Our survey of data compression algorithms shows that there is no one method that is superior for all forms of data streams. Therefore, we have devised a practical data reduction algorithm, DRACO, on the basis of readings from monitoring devices that are typical of electricity network data patterns. The efficiency of the proposed technique depends on the degree of correlation between data points.

We have been able to validate the new algorithm, DRACO, on data from real electrical networks, which were produced at a very high sampling rate, and transmit the essential data with compression ratios of 70%–99%. High sampling rates are typical for this application, in terms of identifying important changes in the dynamic behaviour of electrical systems. Since, according to the first order radio model, the energy

TABLE 10 Size and compression ratio for three algorithms applied to electricity network data

	Original size (b)	Huffman size (b)	Huffman ratio (%)	Arithmetic size (b)	Arithmetic ratio (%)	DRACO-1 size (b)	DRACO-1 ratio (%)
Voltage	230464	13664	94	19600	91	14822	93
Current	230464	26841	88	33392	85	17700	92
Frequency	371456	4979	98	4176	98	14408	96
Total Power Factor	371288	7209	98	6648	98	14520	96

TABLE 11 Execution time for compression of different data reduction algorithms

	Huffman (s)	Arithmetic (s)	DRACO-1 (s)	DRACO-2 (s)
Voltage	0.36	3.61	0.25	0.24
Current	1.06	3.52	0.25	0.27
Frequency	0.31	0.53	0.24	0.19

consumption of data transmission has a linear relation to the number of transmitted bits, one significant contribution of this research is that the DRACO can reduce energy consumption associated with data transmission, and can improve the overall energy efficiency of the communication network in the proposed architecture.

We have shown that DRACO performs well in comparison to other data reduction algorithms, by comparing compression ratio for algorithms applied to a general data set as well as the specific case of the electricity network data.

In addition to energy consumption reduction, DRACO is able to provide an efficient flow of information by reducing data traffic, which needs further investigation. In complex sensor networks, bottlenecks are caused by the fact that thousands of sensors are sending their data to the central point. In some cases, by applying DRACO, we may reduce the risk of bottlenecks. This is an issue for further research work. Moreover, the growth in the number of monitoring devices in the smart grid in the near future will lead to an explosion in data volume. This will cause storage and network congestion problems. At this stage we are not typically prepared to manage such a volume of data. We need to develop new methods and techniques to ease these forthcoming issues of network congestion and data storage. DRACO could be an initial point for addressing these problems.

Here we have focused on the design and implementation of DRACO, but in the future, storage limitation and network congestion problems can be considered as possible further applications of DRACO.

ACKNOWLEDGEMENT

Our research was supported by the Active Building Centre, Industrial Strategy Challenge Fund, under Programme Grant EP/S016627/1.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are commercially sensitive and cannot be released to the public domain.

ORCID

Zoya Pourmirza https://orcid.org/0000-0001-8985-128X

REFERENCES

1. Miller, J., Pullins, S., Bossart, S.: The Modern Grid US Department of Energy. Office of Electricity Delivery and Energy Reliability (2008)
2. Heinzelman, W. R., Chandrakasan, A., Balakrishnan, H.: Energy-efficient communication protocol for wireless microsensor networks. In: Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, p. 10. IEEE (2000)
3. Pourmirza, Z., Brooke, J. M.: The wireless sensor network and local computational unit in the neighbourhood area network of the smart grid. In: SENSORNETS, pp. 84–88 (2013)
4. Sacaleanu, D.I., et al.: Compression scheme for increasing the lifetime of wireless intelligent sensor networks. In: 2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO), pp. 709–713. IEEE (2012)
5. Huffman, D.A.: A method for the construction of minimum-redundancy codes. Proc. IRE. 40(9), 1098–1101 (1952)
6. Campos, A.S.E. (1999). Arithmetic Coding. Barcelona, Spain. http://www.arturocampos.com/ac_arithmetic.html.
7. Kimura, N., Latifi, S.: A survey on data compression in wireless sensor networks In: International Conference on Information Technology: Coding and Computing (ITCC'05), vol. 2, pp. 8–13. IEEE (2005)
8. Talele, A.K., Patil, S.G., Chopade, N.B.: A survey on data routing and aggregation techniques for wireless sensor networks. In: 2015 International Conference on Pervasive Computing (ICPC), pp. 1–5. IEEE (2015)
9. Saraswat, J., Bhattacharya, P.P.: Effect of duty cycle on energy consumption in wireless sensor networks. Intl. J. Comput. Netw. Commun. 5(1), 125 (2013)
10. Săcăleanu, D.I., et al.: Data compression in wireless sensor nodes with LoRa. In: 2018 10th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), pp. 1–4. IEEE (2018)
11. Wen, L., et al.: Compression of smart meter big data: A survey. Renew. Sust. Energy Rev. 91, 59–69 (2018)
12. Ringwelski, M., et al.: The hitchhiker's guide to choosing the compression algorithm for your smart meter data. In: 2012 IEEE International Energy Conference and Exhibition (ENERGYCON), pp. 935–940. IEEE (2012)
13. Unterweger, A., Engel, D.: Lossless compression of high-frequency voltage and current data in smart grids. In: 2016 IEEE International Conference on Big Data (Big Data), pp. 3131–3139. IEEE (2016)
14. Spiegel, J., Wira, P., Hermann, G.: A comparative experimental study of lossless compression algorithms for enhancing energy efficiency in smart

meters. In: 2018 IEEE 16th international Conference On Industrial Informatics (INDIN), pp. 447–452. IEEE (2018)
15. Unterweger, A., Engel, D.: Resumable load data compression in smart grids. IEEE Trans. Smart Grid. 6(2), 919–929 (2014)
16. Huang, X., et al.: Electric load data compression and classification based on deep stacked auto-encoders. Energies. 12(4), 653 (2019)
17. Jumar, R., Maaß, H., Hagenmeyer, V.: Comparison of lossless compression schemes for high rate electrical grid time series for smart grid monitoring and analysis. Comput. Electr. Eng. 71, 465–476 (2018)
18. Klump, R., et al.: Lossless compression of synchronized phasor measurements. In: IEEE PES general meeting, pp. 1–7. IEEE (2010) July.
19. Allalouf, M., et al.: Data-quality-aware volume reduction in smart grid networks. In: 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), pp. 120–125. IEEE (2011)
20. Sari, E.M.: Data Compression for Smart Grid Infrastructure, in School of Science and Engineering, p. 58. Şık Üniversitesi (2018)
21. Khalifa, T., et al.: Transport protocol for smart grid infrastructure. In: 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN), pp. 320–325. IEEE (2010)
22. Iglesias-Urkia, M., et al.: Integrating electrical substations within the IoT Using IEC 61850, CoAP, and CBOR. IEEE Internet Thing. J. 6(5), 7437–7449 (2019)
23. Mohamed, M.F., et al.: An adaptive framework for real-time data reduction in AMI. J. King Saud Univ-Comput. Inform. Sci. 31(3), 392–402 (2019)
24. Nassar, J., et al.: Heterogeneous data reduction in WSN: application to smart grids. In: Proceedings of the 4th ACM MobiHoc Workshop on Experiences with the Design and Implementation of Smart Objects, pp. 1–6 (2018)
25. Vidhyapriya, R., Vanathi, P.T.: Energy efficient data compression in wireless sensor networks. Int. Arab J. Inf. Technol. 6(3), 297–303 (2009)
26. Ju, H., Cui, L.: Easipc: A packet compression mechanism for embedded WSN. In: 11th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'05), pp. 394–399. IEEE (2005)
27. Aquino, J.F., et al.: A differential coding algorithm for wireless sensor networks. In: 2008 IEEE 19th International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 1–5. IEEE (2008) September
28. Arici, T., et al.: PINCO: A pipelined in-network compression scheme for data collection in wireless sensor networks. In: Proceedings of 12th International Conference on Computer Communications and Networks, IEEE Cat. No. 03EX712IEEE, pp. 539–544 (2003)
29. Săcăleanu, D.I., et al.: In node data processing for increasing the lifetime of a wireless sensor network. In: 5th WSEAS International Conference on Sensors and Signals (SENSIG'12), Sliema, Malta, pp. 77–81 (2012)
30. Marcelloni, F., Vecchio, M.: A simple algorithm for data compression in wireless sensor networks. IEEE Commun. Lett. 12(6), 411–413 (2008)
31. Sornsiriaphilux, P., et al.: On applying simple data compression to wireless sensor networks. 14th International Annual Symposium on Computational Science and Engineering (ANSCSE14), Chiang Rai, Thailand (2010)
32. Sornsiriaphiluxi, P., et al.: Performance comparison of data compression algorithms based on characteristics of sensory data in wireless sensor networks. International Conference on Information and Communication Technology for Embedded Systems (IC-ICTS), Thailand (2010)
33. Pourmirza, Z., Brooke, J.M.: A Realistic ICT Network Design and Implementation in the Neighbourhood Area of the Smart Grid (2013)
34. Pourmirza, Z., Brooke, J.M.: Energy aware communication in the smart grid. In: International Conference on Smart Grids and Green IT Systems. SCITEPRESS – Science and Technology Publications (2014)
35. Patrik, R., et al.: Traffic Modelling, Communication Requirements and Candidate Network Solutions for Real-Time Smart Grid Control (2015)
36. Kuzlu, M., Pipattanasomporn, M., Rahman, S.: Communication network requirements for major smart grid applications in HAN, NAN and WAN. Comput. Netw. 67, 74–88 (2014)
37. Pourmirza, Z., Brooke, J.M.: Data reduction for an ICT network in the smart grid. In: 2013 9th International Conference on intelligent Environments, pp. 224–227. IEEE (2013)
38. Forouzan, A.B.: Data Communications & Networking (SIE). Tata McGraw-Hill Education (2007)

How to cite this article: Pourmirza Z, Walker S, Brooke J. Data reduction algorithm for correlated data in the smart grid. *IET Smart Grid*. 2021;1–15. https://doi.org/10.1049/stg2.12010