A Two Vector Data-Prediction Model for energy-efficient Data Aggregation in Wireless Sensor Network

Khushboo Jain¹ and Akansha Singh²

¹ DIT University, Dehradun, India, ² NIET, Greater Noida, India
{khushboojain2806@gmail.com, singhakansha1@banashhali.in
https://orcid.org/0000-0002-4166-2591

Abstract. Most ecological management applications use Wireless Sensor Networks (WSNs) to collect data regularly, with great temporal redundancy. As a result, a significant amount of energy is used transmitting redundant data, making it tremendously problematic to attain a satisfactory network lifetime, which is a bottleneck in enduring such environmental monitoring applications. A two-vector prediction model based on Normalized Quantile Regression (NQR) for Data Aggregation is proposed to proficiently accomplish energy reduction in synchronized data collecting cycles. The introduced NQR algorithm provides high-accuracy prediction. With accurate estimates, energy usage is reduced. Furthermore, it extends the network’s lifetime. In intracluster transmissions, NQR uses a two-vector data-prediction algorithm to coordinate the anticipated sensor's reading and, as a result, minimize cumulative inefficiencies from uninterrupted predictions. NQR algorithm can be integrated with both homogeneous and heterogeneous WSNs. When compared to existing methods, the suggested NQR methodology is shown to have high energy efficiency. The results show greater prediction accuracy, more positive predictions with high data quality, which help the network last longer.

Keywords: Data Quality; Energy Efficiency; Heterogeneous WSN; Normalized Quantile Regression; No Transmission Cycle;

1. Introduction

WSNs are considered a well-known technology that can transform people's life. Sensor nodes (SNs) are typically deployed with the target of collecting a large amount of data through environmental monitoring [1]. As a result, the main aim of WSN is to maximize the network lifetime by reducing the consumption of energy and guaranteeing the accuracy of gathered data to conduct extensive analysis. WSNs with a large number of SNs typically present unique design limitations and issues. Energy resources and computational capability are two significant challenges they face. By performing data aggregation, it is critical to establish resource management methods in WSN to ensure their long-term viability. [2].
Data aggregation technique eliminates duplicates in highly changeable time series and reduces redundant transmission of data, leading to significant energy savings and improved proficiency [3]. An additional significant problem for WSN is the dynamic elements of temporal and spatial diversity observed in the process of sensing data as a result of constantly changing environmental parameters. Furthermore, dynamic queries are created as a result of changing application needs and end-user expectations, necessitating the ability to retrieve information [4][5].

The deployment of SNs and connectivity and the constant decline in the network's residual energy levels are all aspects that make the WSN more dynamic. SNs typically gather data over a period of time and deliver fused data to the cluster-head (CH) or directly to the base station (BS) on a regular basis[6]. Many data gathering techniques allow SNs to communicate all data aggregates frequently in order to accomplish exact data measurements. However, due to the substantial energy drain caused by excessive transmissions, such data collection strategies result in early sensor death [7]. Data prediction, which uses projected values instead of real ones to acquire sensor data, is an efficient technique to minimize the data transmission in the continuously observed sensor networks. However, ensuring prediction accuracy within the threshold is a challenge [8].

A two-vector data-prediction model based on Normalized Quantile Regression (NQR) for data aggregation is proposed in this paper to minimize the data transmission and enhance the network lifespan by estimating the future data based on prior sensed readings. In this work, a two-vector data-prediction model for energy-efficient data aggregation is implemented, which is designed to organize the predicted data sets evaluated by sensor nodes and cluster heads (CHs) to avoid cumulative error from continuous predictions. NQR Algorithm predicts the sensor's reading in the next time slot at both SN and CH based on data stored on the vectors. When the next data is sensed, each SN in the cluster compares its predicted data with the real sensed data. When the prediction error is lower than a predetermined application-specific threshold, the SN will not relay the real sensor reading to the CH. When the CH does not receive a value from the SN, it uses the same data prediction algorithm to forecast the next data in the series. As a result, duplicate and useless communications are prevented, and the SN's energy is preserved. This cycle will be no data transmission cycle. When the prediction error exceeds a preset application-specific threshold, the SN must send the real sensed data to the CH.

The rest of the work is organized as follows. Section 2 presents the literature review of the related data prediction models in WSN. Section 3 describes the Framework for a Two Vector Data-Prediction Model and the network architecture and energy model. Section 4 defines the Normalized Quantile Regression (NQR) Algorithm for Data Prediction in detail. In Section 5, we present the experimentation along with the comparative result analysis. Finally, in Section 6, we state the conclusions and future aspects.
2. Literature Survey

This section describes existing techniques for data aggregation based on data prediction to enhance WSNs lifetime.

In 2011, Wei et al. [9] proposed a novel prediction-based data collection protocol in which they designed a double queue mechanism so that cumulative error of continuous prediction is minimized. The proposed approach has reduced communication redundancy and manages the energy consumption to enhance network lifetime. In 2015, Sinha & Lobiyal [10] proposed data gathering rounds based on temporal prediction models for data aggregation (TDPA) to minimize the energy consumption. They also provided a cluster head rotation algorithm for load balancing within the cluster that uses energy prediction. Results show improvement in energy consumption.

Avinash et al. [11] used the Kalman filter in WSN to predict the lost data in 2015 and evaluated the network’s performance. In 2015, Raza et al. [12] presented a technique for data prediction named derivative-based prediction (DBP). They evaluated this technique on a real WSN setup in a road tunnel. The proposed approach shows significant improvement with respect to the mainstream periodic reporting.

In 2017, Yue et al. [13] used kernel function for fault prediction in ribbon wireless sensor networks. Results yield that the proposed approach has higher accuracy in predicting data. In 2018, Wang & Wang [14] proposed a technique for data prediction named derivative-based prediction (DBP). They evaluated this technique on a real WSN setup in a road tunnel. The proposed approach shows significant improvement with respect to the mainstream periodic reporting.

Correlation analysis of multi-model sensor data is given by Rajesh & Chaturvedi [15] in 2019 for environmental sensor networks. In this work, performance analysis is done, and results are compared based on different types of correlation coefficients. A novel data prediction model is proposed by Cheng et al. [16] in 2019 for multistep sensory data in WSN. According to them, the proposed model can predict data in the short term and midterm sensory data.

In 2020, a prediction model named Extended Cosine Regression (ECR) was proposed by Jain and Kumar [17] for Data Aggregation in WSN. This approach is based on two vector models in intracluster transmission to synchronize the predicted data and is suitable for homogeneous WSN. Various NS-2.34 simulation results in high accuracy in data prediction and improves energy efficiency. Jaya Singh [18] worked on augmented data prediction efficiency aiding artificial technology and machine learning algorithm.

For homogeneous network setup in WSN, Nels and Singh [21] proposed a data aggregation scheme in 2020. Two-way model generation was introduced for reducing correlated spatial, temporal data. El-Sayed et al. [20] proposed a Distributed Data Predictive Model For predicting dissemination faults. This approach has decreased energy consumption and improves network lifetime.

For data aggregation in WSN, Nels and Singh [21] proposed a Hierarchical Fractional quantized kernel least mean square filter in the year 2021. The proposed data aggregation approach prolonged the network lifetime due to less energy consumption. Recently a data prediction model (ELR) based on correlation for data reduction was
proposed by Jain et al. [22]. This approach exempts the sensor node from sending a huge volume of data to conserve energy and thus improve the network lifetime.

The TDPA, ELR, and other successive algorithms also claimed that most of the algorithms developed for homogeneous networks are not well suited for heterogeneous WSNs.

Despite having numerous literature on data aggregation and prediction as discussed above, the proposed prediction model, Normalized Quantile Regression (NQR) for data aggregation, will be more promising for resource-constrained WSN.

3. Framework for a Two Vector Data-Prediction Model

This work considers a two-vector data-prediction model for energy-efficient data aggregation in WSN. The network model consists of a layered architecture. The first layer formulates the SNs, which uninterruptedly forward the sensed data to the respective CH a via single-hop and generate the sensed characteristics of data at different time $t$. The CH is at the framework’s second layer, which manages and coordinates the intra-cluster communication. Finally, the BS is the third layer that receives the aggregated data from the CHs.

![Fig 1. Three-layer network model of WSN](image-url)

The following are the major constraints in developing a two-vector data-prediction model:

1. What are some options for keeping the two vectors synchronized as maintained by the SN and the CH manages the other? Since both CH and BS use the same data sequence and same data forecast method in our proposed model. Furthermore, SN has real sensed readings, whereas the CH does not have because some real sensed data is not forwarded to the CHs. After all, associated forecasts are positively accomplished beforehand.
2. In case CH does not obtain real sensed reading from the SN, how will it differentiate between the positive data prediction and packet loss in the network? When the SN outbound the threshold, it must transmit the real sensed data to the CH. But still, there may be a case in which the CH may also fail to receive the real sensed reading because of faulty transmission. From this viewpoint of CH, this case is extremely similar to the positive data-prediction cycle.

3. How can cumulative error in positive data prediction cycles be avoided? Because the data collected for implementing the forecasts may comprise predicted results, the cumulative error may arise inadvertently.

The proposed two-vector data-prediction model for energy-efficient data aggregation is provided as follows to address these concerns.

3.1. Prerequisites:
- The time period of each SN is partitioned into a fixed time period, and only one data is sensed during each time period.
- The same instance of the data-prediction model will be employed both at the SNs and CHs.
- The CH must provide an acknowledgment back to the source SN in order for trustworthy data and end-to-end communication to occur.
- Both homogeneous and heterogeneous WSNs can benefit from the approach.

3.2. System Preliminaries and Setup:
- The BS will communicate to all CHs and SNs its acceptable prediction threshold denoted as $\delta$ and cumulative threshold denoted as $\varepsilon$. They are pre-deployed based on the application for which it is used. Any change in the values of $\delta$ and $\varepsilon$ is communicated to all CHs and SNs.
- Each SNs within a cluster will construct two vectors of the same period (length): one holds the sequence of real sensed reading and is known as a real data vector ($RDV_{SN}$). While the other stores the values which will be used to forecast the sensor readings in CHs and SNs and are termed as forecast data-vector ($FDV_{SN}$). Each CH will also build an equivalent vector for its cluster member nodes (CMNs), termed $FDV_{CH}$. Also, for $i^{th}$ SN the $FDV_{CH}^i = FDV_{SN}^i$.
- Each SN stores the first $n$ sensed reading into its $RDV_{SN}$ and $FDV_{SN}$ and communicates the to the respective CH to construct $FDV_{CH}^i$. During initialization, $RDV_{SN}^i = FDV_{CH}^i = FDV_{SN}^i = (r_1, r_2, \ldots, r_n)$, where $r_i$ represents the sensed data in the vector of an SN.
Also, if the WSN is homogeneous, the role of the CHs can be interchanged based on the energy threshold. Control packets are reduced using this strategy. And if the WSN is heterogeneous, the role of the CH is fixed as the SNs have lower initial energy as compared to CH in heterogeneous WSNs. To test the heterogeneity, the SNs are divided into two classes. The normal SNs have the initial energy of $E_0$. The $m$ number of advanced SNs has $a$ times more energy as compared to the normal SNs. Thus, there will be $mN$ advanced SNs equipped with $E_0(1 + a)$—initial energy.

### 3.3. Data Prediction:

Let us assume that $r_{n+1}$ be the real sensed data, $p_{n+1}$ be the predicted value using $RDV^i_{SN}$, and $\hat{p}_{n+1}$ be the predicted value using $FDV^i_{SN}$, respectively. The CH will receive the $p_{n+1}$ value from the $FDV^i_{CH}$ vector. If $|p_{n+1} - r_{n+1}| \leq \delta$, then the prediction threshold is within the error limit, else outside the error limit. Also, if the $|\hat{p}_{n+1} - p_{n+1}| \leq \delta$, then the cumulative threshold is also within the error limit, outside the error limit. Both prediction threshold $\delta$ and cumulative threshold $\epsilon$ should be within the error limit in order to consider it a positive prediction.

In positive prediction, the SNs don't have to transmit $r_{n+1}$ to their respective CH. It will take the forecasted data $\hat{p}_{n+1}$ as the $r_{n+1}$ in this time interval. No data will be transmitted in positive prediction, also known as no transmission cycle. After the positive prediction cycle, all the vectors will be updated as:

\[
RDV^i_{SN} = (r_1, r_2, \ldots, r_n, r_{n+1})
\]
\[
FDV^i_{SN} = (r_1, r_2, \ldots, r_n, \hat{p}_{n+1})
\]
\[
FDV^i_{CH} = (r_1, r_2, \ldots, r_n, \hat{p}_{n+1})
\]

Though for negative prediction, the real sensed reading would be transmitted to the CH, and all the vectors will be updated as:

\[
RDV^i_{SN} = (r_1, r_2, \ldots, r_n, r_{n+1})
\]
\[
FDV^i_{SN} = (r_1, r_2, \ldots, r_n, r_{n+1})
\]
\[
FDV^i_{CH} = (r_1, r_2, \ldots, r_n, r_{n+1})
\]

### 3.4. Energy Model:

A first-order radio communication model [23] [24] is used for the energy estimation of the proposed work. The energy dissipation with an SN depends on the sensing unit, the processing, and the radio communication subsystem. Commonly, the consumed energy relates to four main tasks, namely, data sensing, data transmission, data aggregation, and data processing, as represented in eq. (1)

\[ E_{Total} = E_{SEN} + E_{DA} + E_{DC} + E_{DP} \]  

$E_{SEN}$ is the energy required for sensing one data reading, which is the energy needed to convert the physical quantities into digital one. The following eq. (2) computes it.

(13) where $b$ represents the number of bits in the sensed data packet after conversion
to the digital signal, \( V \) represents the voltage supply, \( I_{SEN} \) is the total current required for the sensing task, and \( T_{SEN} \) is the total time-duration of sense.

\[
E_{SEN} = b V I_{SEN} T_{SEN}
\]

\( E_{DA} \) is the energy required to perform data aggregation and \( E_{AGG} \) is the energy dissipated per bit. To aggregate \( b \) data bits, the energy dissipation is calculated in Eq. (3) as:

\[
E_{AGG}(b) = bE_{DA}
\]

Based on the transmitter-receiver distance, the first order radio communication model uses both types of the channel, such as “free-space (\( f_s \))” and “multipath (\( m_p \))” fading channels. A threshold is used to control the transmission over the communication channel. Therefore, to transmit \( b \)-bits data over a distance, the radio expends:

\[
E_{TX}(b, d) = E_{TX-\text{elec}}(b) + E_{TX-\text{amp}}(b, d)
\]

\[
E_{TX}(b, d) = \begin{cases} 
E_{\text{elec}}b + b\epsilon_{fs}d^2 & \text{if } d < d_0 \\
E_{\text{elec}}b + b\epsilon_{mp}d^4 & \text{if } d \geq d_0 
\end{cases}
\]

Where threshold \( d_0 \) is given as

\[
d_0 = \sqrt{\frac{\epsilon_{rs}}{\epsilon_{mp}}}
\]

Energy spends by the radio to receive a data packet is given by

\[
E_{RX}(k) = E_{RX-\text{elec}}(k) = kE_{\text{elec}}
\]

The proposed data prediction model considers two communication phases: intra-cluster communication during data prediction and inter-cluster transmission with the BS. The number of nodes as \( N \) SNs in the sensor network from which the number of CHs is \( C \), and each SNs gathers \( R \) data values. Let us assume that \( E_p \) be energy consumption for a single prediction and \( E_{SN} \) be energy to transmit a data reading with the likelihood \( l \). The CHs prediction is initiated at prospect \( 1 - l \). Thus, the predictable energy cost \( E_{pSN} \) is defined by equation (7).

\[
E_p = C (1 - l) + l E_{SN}
\]

Therefore, the energy consumption of an intra-cluster can be defined as follow:

\[
E_{\text{INTRA}} = R \left\{ E_p (N - C) (2 - l) + E_{SN} (N - C) l \right\}
\]

Similarly, in the inter-cluster SN transmission to CH, the \( E_{\text{INTER}} \) values for \( C_{CH} \) are given as \( E_{CH} R_C \). Henceforth, the energy consumption of communication within-cluster is defined as follows.
\[ E_{\text{INTER}} = R \sum_{k=1}^{c} E_{CH} R_{C} \]  
(9)

By combining eq (8) and eq. (9), the total prediction energy cost can be computed as follow:
\[ E_{DP} = E_{\text{INTRA}} + E_{\text{INTER}} \]  
(10)

4. Normalized Quantile Regression (NQR) for Data Prediction

We proposed a two-vector data-prediction model based on Normalized Quantile Regression (NQR) for data aggregation in WSN. The goal is to reduce the data transmission and to enhance the network lifespan by estimating future data based on prior sensed readings. The proposed NQR algorithm can be integrated with both homogeneous and heterogeneous WSNs and is implemented as a discrepancy of the cosine distance on the Quantile Regression model, which is an extension of linear regression. It is used when outliers, heteroskedasticity, and high skewness exist in the data. Here, we try to estimate the quantile of the dependent variable \( P \) given the values of \( R' \). The suggested NQR algorithm is implemented in three stages: the first stage is an initialization, the second stage is NQR algorithmic model building, and the last third stage is working in which prediction is computed.

4.1. NQR Algorithm: Initialization Stage

The BS will broadcast its predefined application-based prediction threshold \( \delta \) to all SNs and CHs during the setup stage. All SNs within each cluster will use single-hop communication to send their real sensed data value to their corresponding CHs. All of these data readings will be collected by the respective CH and cached as an ordered pair with the matching SN's ID. For a certain time, for instance, \( T \), the SN is authorized to sense and transmit their data value. This period of time has been divided into equal time slots. \( T = (t_1, t_2, ..., t_n) \).

Some previous sensed values are stored in the SN to initialize the data vectors for the NQR model designated as \( R \) before predicting the new data value and expressed in eq. (11).

\[ R = r_1[t_1], r_2[t_2], ..., r_n[t_n] \]  
(11)

For \( i^{th} \) SN of each cluster, it will hold real sensed readings into \( RDV_{SN}, FDV_{CH} \), and \( FDV_{SN} \). During model setup stage, \( RDV_{SN} = FDV_{CH} = FDV_{SN} = R \). After the initialization stage, each CH will contain a sensed data till \( t_n \) rounds of communication which will go for the model building phase of NQR based data prediction model.

| Algorithm 1- NQR Model: Initialization stage |
|---------------------------------------------|
| **Parameters:**                             |
| \( T \) \( \leftarrow \) SN time interval of equal length: \( T = (t_1, t_2, ..., t_n) \) |
| \( R \) \( \leftarrow \) Real sensed data: \( R = r_1[t_1], r_2[t_2], ..., r_n[t_n] \) |
| \( P \) \( \leftarrow \) Predicted data: \( P = r_{n+1}[t_{n+1}] \) |
| \( \delta \) \( \leftarrow \) Acceptable prediction threshold |
| **Algorithm BEGIN**                         |


1. for SNs in each cluster
2. BS broadcasts threshold $\delta$ to all CHs and SNs
3. for each $t$ in $T[n]$
4. If each real sensed reading $r_i[t_i]$
5. Cache actual data values into $RDV_{SN}^i = FDV_{CH}^i = FDV_{SN}^i$
6. $RDV_{SN}^i = FDV_{CH}^i = FDV_{SN}^i = R = r_1[t_1], r_2[t_2], \ldots, r_n[t_n], p_{n+1}[t_{n+1}]$
7. end if
8. end for
9. end for

4.2. NQR Algorithm: Model-Building Stage

As a prediction model generally need some previous knowledge about the data to be predicted, we have presumed that the values are dispersed normally. In this stage, the NQR model is developed by enhancing the cosine distance over the Quantile Regression model. The fundamental idea behind NQR is to combine the two autonomous prediction strategies in order to avoid any insufficiencies that a single prediction technique could cause. To increase the prediction accuracy in WSN, the Quantile Regression model and the cosine distance are combined. Since the Quantile Regression model is used when data contains outliers, heteroskedasticity, and high skewness, and the cosine distance is useful for improving prediction accuracy on data sequences that frequently fluctuate, the combination of the Quantile Regression model and the cosine distance reduces uncertainty and improves prediction accurateness.

$$P_i = R_i \beta + \delta_i$$  \hspace{1cm} (12)

Here $P_i$ is the dependent variable, $R_i$ is the independent variable and $\delta_i$ is the prediction threshold. The objective function to minimize is follow:

$$q \sum |\delta_i| + (1 - q) \sum |\delta_i|$$  \hspace{1cm} (13)

If $q$ is the $q^{th}$ quantile, then Quantile Regression Model expression for $q^{th}$ quantile will be

$$Q_q P_i = R_i \beta(q) + \delta_i(q)$$  \hspace{1cm} (14)

The best Quantile regression line is originated by reducing the median absolute deviation ($MAD$).

$$MAD = \frac{1}{n} \sum \rho_q(P_i - \beta(q))$$  \hspace{1cm} (15)

Now, $\rho_q$ is the check function that gives asymmetric values to the threshold depending on the quantile and the overall sign of the threshold. Fig. 2 shows the func-
tion graph of the Quantile Regression Model. Mathematically, \( \rho_q \) is represented as follow:

\[
\rho_q(\mu) = q \max(\mu, 0) + (1 - q) \max(-\mu, 0)
\]  
(16)

Here, \( \mu \) is the threshold of a single sensed data, and the maximum function returns the largest value in the parentheses. This means that if the threshold is positive, then the check function multiplies the error by \( q \), and if the error is negative, then the check function multiplies the error by \( (1 - q) \).

A cosine distance is the measure of the variation in the threshold. An angle of 0° means that \( \cos \theta = 1 \); i.e., the sensed data sets are exactly similar to each other. An angle of 90° means that \( \cos \theta = 0 \) and the compliant values are perpendicular to each other.

At the end of a data sensing time \( T \), each CHs implements data aggregation on the data vectors created on the basis of \( \text{MAD} \). The data aggregation will be accomplished by aggregations functions as \( \frac{1}{n} \sum \rho_q(P_i - \beta(q)) \). Consequently, similarity in data values is expressed as follow:

\[
\text{Sim}(\text{MAD}) = \cos \theta
\]  
(17)

\[
\cos \theta = \frac{1}{n} \sum \rho_q(P_i - \beta(q))
\]  
(18)

As the angle between them reduces, is \( \cos \theta \) approaches to 1, that means the two vectors are getting closer so the likeness of data which is signified by the vectors rises. Therefore, we have extended the Quantile Regression Model to the NQR model as follow:

\[
P_i = \cos \theta R_i + \delta_i
\]  
(19)

\[
P_i = R_i \beta + \delta_i
\]  
(20)

Here \( \cos \theta = \beta \) will be the normalized cosine function and \( \delta_i \) is the fitting constant. The values of \( \beta \) and \( \delta_i \) is calculated in each CH as follow:
\[ \beta = \frac{1}{n} \sum_{i} \rho_i (P_i - \beta(q)) \]  
(21)

\[ \delta_i = \frac{\sum_{i} (r_i - \bar{r})(p_i - \bar{p})}{\sum_i (r_i - \bar{r})^2} \]  
(22)

\( P_i \) represents the predicted data of \( i^{th} \) SN.

\[ P_i = P_{n+1}[t_{n+1}] \]  
(23)

\[ P_{n+1}[t_{n+1}] = \beta R_i + \delta_i \]  
(24)

Now, \( R_{n+1}[t_{n+1}] \) is the read sensed value at time \( t_{n+1} \) for \( i^{th} \) SN of each cluster, and \( P_{n+1}[t_{n+1}] \) is the predicted value of the SN using RDV\(_{SN}^j\) as we haven’t considered the cumulative threshold, so \( FDV_{CH}^j = FDV_{SN}^j \). The CH will also obtain \( P_{n+1}[t_{n+1}] \) from the \( FDV_{CH}^j \) vector.

**Algorithm 2- NQR Model: Model Building Stage**

| Parameters: |
|-------------|
| \( \beta \) \( \rightarrow \) normalized cosine function |
| \( \delta_i \) \( \rightarrow \) fitting constant |
| \( T \) \( \rightarrow \) SN time interval of equal length: \( T = (t_1, t_2, ..., t_n) \) |
| \( R \) \( \rightarrow \) Real sensed data: \( R = r_1[t_1], r_2[t_2], ..., r_n[t_n] \) |
| \( P \) \( \rightarrow \) Predicted data: \( P = P_{n+1}[t_{n+1}] \) |
| \( \delta \) \( \rightarrow \) Acceptable prediction threshold |

**Algorithm**

1. for each cluster, the CH and SN will calculate the following:
2. \[ \beta = \frac{1}{n} \sum_{i} \rho_i (P_i - \beta(q)) \]
3. \[ \delta_i = \frac{\sum_{i} (r_i - \bar{r})(p_i - \bar{p})}{\sum_i (r_i - \bar{r})^2} \]
4. \[ P_{n+1}[t_{n+1}] = \beta R_i + \delta_i \]
5. Store predicted data values into \( FDV_{SN}^j \)
6. \[ FDV_{CH}^j = FDV_{SN}^j \]
7. end for

4.3. NQR Algorithm: Working Stage

Suppose the prediction error be \( \Delta[t_{n+1}] = |p_{n+1}[t_{n+1}] - r_{n+1}[t_{n+1}]| \) and \( \delta \) is the prediction threshold. After estimating them, the SN compares \( \delta \) with the acceptable threshold. If the value of \( \Delta[t_{n+1}] < \delta \), the SN will not send the real sensed reading \( r_{n+1}[t_{n+1}] \) to CH. The CH will also execute the same NQR algorithm on \( FDV_{CH}^j \). Consequently, it will also predict the identical value as forecast by the SN. Though, the CH does not compute the prediction error as it does not have a real sensed read-
ing $r_{n+1}[t_{n+1}]$. If no data is obtained by CH in a fixed time $T_0$, the CH will set $\Delta[t_{n+1}] < \delta$ and deliberates $p_{n+1}[t_{n+1}]$ as $r_{n+1}[t_{n+1}]$ in the present sensing interval. To synchronize the prediction vectors, $FDV_{CH}^{i} = FDV_{SN}^{i}$.

When, $\Delta[t_{n+1}] \geq \delta$ the SN uses predicted data $r_{n+1}[t_{n+1}]$ as the value of $[t_{n+1}]^{th}$ interval is in the next prediction vector because the CH does have $r_{n+1}[t_{n+1}]$. Later, the positive prediction, the two vectors used in data prediction, will be updated in eq. 25 and eq. 26 as follow:

$$RDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], r_{n+1}[t_{n+1}]$$

(25)

$$FDV_{CH}^{i} = FDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], p_{n+1}[t_{n+1}]$$

(26)

But, in the case of negative prediction, real sensed reading $r_{n+1}$ of $[t_{n+1}]^{th}$ instance is transmitted to the CH. The vectors are updated in this case as expressed in eq. 27 as follow:

$$RDV_{SN}^{i} = FDV_{CH}^{i} = FDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], r_{n+1}[t_{n+1}]$$

(27)

Algorithm 3- NQR Model: Working stage

Parameters:
- $T$ ← SN time interval of equal length: $T = (t_1, t_2, ..., t_n)$
- $R$ ← Real sensed data: $R = r_1[t_1], r_2[t_2] ,..., r_n[t_n]$
- $P$ ← Predicted data: $P = r_{n+1}[t_{n+1}]$
- $\delta$ ← Acceptable prediction threshold
- $\Delta[t_{n+1}]$ ← Calculated prediction threshold

Algorithm BEGIN

1. for each cluster, the CH and SN will calculate the following:
2. \[ \Delta[t_{n+1}] = |p_{n+1}[t_{n+1}] - r_{n+1}[t_{n+1}]| \]
3. if $\Delta[t_{n+1}] \leq \delta$, then // No transmission cycle
4. \[ \text{Corresponding SN will not Transmit } r_{n+1}[t_{n+1}] \text{ to respective CH} \]
5. \[ \text{CH will use the predicted values} \]
6. \[ \text{Vectors are updated as follow:} \]
7. \[ RDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], r_{n+1}[t_{n+1}] \]
8. \[ FDV_{CH}^{i} = FDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], p_{n+1}[t_{n+1}] \]
9. else
10. \[ \text{SN will send } r_{n+1}[t_{n+1}] \text{ to respective CH} \]
11. \[ \text{Vectors are updated as follow:} \]
12. \[ RDV_{SN}^{i} = FDV_{CH}^{i} = FDV_{SN}^{i} = r_1[t_1], r_2[t_2] ,..., r_n[t_n], r_{n+1}[t_{n+1}] \]
13. end if
14. end for
15. CH will send the $FDV_{CH}^{i}$ data to the BS

END Algorithm
5 Experimentation and Result Analysis

The proposed two-vector data-prediction model based on Normalized Quantile Regression (NQR) for data aggregation is implemented to reduce the data transmission and to enhance the network lifespan by estimating the future data based on prior sensed readings. In this section, we will test the validity of the proposed NQR model and also perform comparisons with No prediction, ELR [22], and TDPA [10]. We have performed experiments on NS [25] which is an event-driven simulator that has proved useful for studying the dynamic nature of communication protocols. NS2 is written in the C++ programming language. The Object-Oriented Tool Command Language (OTCL) provides communication between various objects based on the user's corresponding behavior. We have implemented a simulation environment same as the EESCT [19], which is based on a probabilistic threshold to select the CHs, and the probability is set to 0.1. We also estimate the cumulative energy consumption after every communication cycle. The simulation settings are listed in Table 1.

| Parameter                        | Symbols | Value     |
|----------------------------------|---------|-----------|
| Initial energy                   | $E_0$   | 1 J       |
| Prediction Threshold             | $\delta$ | 0.5, 1.0  |
| Sensor Nodes                     | $N$     | 500       |
| Prediction Algorithms            |         | No Prediction, ELR, TDPA, and NQR |
| Network Area                     | $L \times B$ | 1000m $\times$ 1000m |
| Data packet                      | $D$     | 500 bytes |
| Simulation time interval (T)     | $T$     | 150 s     |
| Transmission energy              | $E_{TX}$ | 150 nJ/s for 1 bit, 10 m |
| Reception energy                 | $E_{RX}$ | 50 nJ/s for 1 bit |
| Aggregation energy               | $E_{DA}$ | 5 (nJ/bit)/s |
| Free space amplifier energy      | $\epsilon_{fs}$ | 10 (pJ/bit)/m² |
| Multipath fading amplifier energy | $\epsilon_{mp}$ | 0.0013 (pJ/bit)/m⁴ |

In the proposed NQR algorithm, $\delta$ is the acceptable prediction threshold, and $\epsilon$ is the cumulative threshold. They signify the application's requirement for data accuracy and energy efficiency. To determine the optimum prediction accuracy, we have conducted simulations:

- with $\delta = 0.5$, which means high data accuracy and low energy efficiency
- with $\delta = 1$, which means low data accuracy and high energy efficiency

5.1 Estimation of the two-prediction vector length:

As discussed in section 3.2, that each SNs within a cluster will construct two vectors of the same period (length): one holds the sequence of real sensed reading ($RDV_{SN}^i$), and the other stores the values which will be used to forecast the sensor readings in CHs and SNs ($FDV_{SN}^i$). Each CH will also build an equivalent vector for its CMN as $FDV_{CH}^i$. Also, for $i^{th}$ SN the $FDV_{SN}^i\neq FDV_{SN}^i$. The real sensed readings are denoted.
as \( R = r_1[t_1], r_2[t_2], ..., r_n[t_n] \), where \( n \) represents the vector length required for performing data prediction by the NQR algorithm. Generally, the more the length of the observed sequence, the more accurate the predictions will be. One issue with extended vector length will be more energy will be needed and result in more complexity in result calculation. Therefore, to estimate the optimum \( n \), the prediction accuracy is computed by varying \( n \) from 1 to 10.

We have arbitrarily chosen five data sub-sequence of 50 continuous readings from the real sensor data. Then we implemented the NQR algorithm on the chosen five data sub-sequence by varying \( n \) from 2 to 10 and evaluated the average prediction accuracy with each \( n \). Finally, we determine the rate of prediction accuracy for increasing the value of \( n \). We plotted the prediction accuracy with both \( \delta = 0.5 \) (high data accuracy) and with \( \delta = 1 \) (high data accuracy).

As described in Fig. 3, by fine-tuning vector length \( n \), the prediction accuracy reaches its maximum value at \( n = 5 \), so we have also selected it as prediction vector length for performing experiments with NQR Algorithm.

\[
\text{Communication Overhead} = 1 - \left( \frac{\sum_{i=0}^{n} r_{n+1}[r_{n+1}]}{D} \right) \times 100
\]

Here \( \sum_{i=0}^{n} r_{n+1}[r_{n+1}] \) is the total positive predictions count obtained in \( t \) time by ten data-gathering rounds. As already discussed, the CMN does not have to send real sensed data \( RDV_{ij}^n \) to the respective CH when the prediction threshold \( \delta \) is inbound. Therefore, the data transmission and its energy are preserved. As the prediction threshold \( \delta \) reduces, there will be more data transmission, and so the prediction accuracy increases and energy ingesting increases. The reason for this is that the low value
of prediction threshold $\delta$ will give a low success rate for prediction. Then again, more negative predictions will result in more real sensed data transmission. Furthermore, on increasing the prediction vector length, $n = 10$, the complexity in calculation rises markedly.

Also, the positive prediction based on the large data sub-sequence can hold weighted cumulative error and can recurrently exceed the prediction threshold. Subsequently, more data transmissions will take place, resulting in additional transmission overhead.

Fig 4: Assessment of Energy ingesting overhead of NQR Algorithm when the prediction vector length is $n=5$, and 10 and threshold ($\delta$) varies from 0.1 to 1.0.

Fig 5: Assessment of communication overhead of NQR Algorithm when threshold ($\delta$) varies from 0.1 to 1.0

5.3 Assessment of total positive prediction count when threshold varies from 0.1 to 1.0
The comparative study of the NQR Algorithm, which reduces the energy consumption and delivers a greater number of positive predictions than ELR [22] and TDPA [10], is illustrated in Fig. 4 and Fig. 5.

Fig 6: Assessment of positive predictions when threshold ($\delta$) varies from 0.1 to 1.0.

TDPA gives more positive predictions and achieves better than ELR with the combined advantage of data aggregation, CH rotation, and data predictions. However, the calculation model of TDPA is difficult to understand and brings additional overhead in data prediction cycles. NQR gives the highest positive predictions due to a simple and lightweight data prediction model that competently decreases discrepancies in continuous predictions rounds, therefore reduces overall overheads. Furthermore, the proposed NQR uses probabilistic-based EESCT to avoid unnecessary CH change in each communication cycle, thus improving the WSN lifetime drastically.

5.4 Assessment of predicted data quality when threshold varies from 0.1 to 1.0

In WSN, data accuracy is an important application factor as users benefit from it in various applications. We have used Mean Absolute Deviation ($MAD$) to determine the predicted data quality at the BS. It is denoted as follows in Eq. (29)

$$MAD = \frac{1}{D} \times \sum_{i=1}^{D} |r_i - p_i|$$

(29)

Where $r_i$ is real sensed reading, $p_i$ is predicted value, and $D$ is total sensed value.
The ELR algorithm probably uses the hidden law of periodicity to evaluate prediction, but it structures dependent and cannot to scaled for wide applications. Furthermore, it can cause a cumulative error, and TDPA integrates data aggregation, CH rotation, and data predictions to simultaneously achieve high data accuracy and energy efficiency.

Correspondingly, TDPA more data value to perform data prediction and has relatively higher algorithmic complexity. This enlightens the high $MAD$ achieved with the variation in the threshold, as emphasized in Fig. 7. Overall, the NQR algorithm brings improved performance proficiency with a sensible prediction threshold.

NQR delivers enhanced proficiency with a controlled threshold, preferably when the threshold is less than 0.5. Henceforth, with simple, lightweight, less complex, and negligible storage overhead is taken by prediction data vector ($FDV_{SN}^t$), the NQR model algorithm data duplicity while promising data accurateness.

5.5 Heterogeneity Test:

As already discussed, the proposed NQR algorithm can be integrated with both homogeneous and heterogeneous WSNs. In heterogeneous WSN, few SNs are equipped with different initial energy levels. To test the heterogeneity, the SNs are divided as normal SN and advance SN. The normal SNs have the initial energy of $E_0$. In advanced SN, $m$ number of advanced SNs have $a$ times more energy as compared to the normal SNs. So, there are $mN$ advanced SNs with the initial energy of $E_0(1 + a)$. For experiments, the percentage of advanced SNs is kept as $m = 0.1, 0.2, 0.3$ and 0.4 for different scenarios, and the parameter is fixed at $a = 3$, which means the advanced SNs have three times more energy as compared to the normal SNs. The simulation results also demonstrate that the network lifetime is improved when heterogeneity is introduced. Fig. 8 validates the impact of heterogeneity in network lifetime.
6. Conclusion

This Two Vector Data-Prediction Model is considered an important data-driven technique for energy-efficient Data Aggregation in Wireless Sensor Networks. By minimizing the number of data transmission cycles that potentially send duplicate data, these strategies help to conserve the WSN's energy. This work proposed a simple, lightweight, scalable, and structure-free which can be easily merged with various data aggregation algorithms. It works with both homogeneous and heterogeneous WSNs. NQR used a two-vector data prediction model, which delivers high prediction precision with smaller historical data and without much knowledge about the data distribution. NQR algorithm generates an approximation of future time series and calculates its prediction threshold. The SN will not participate in the current cycle of data transmission if the prediction threshold is within a predefined threshold, thus conserving transmission energy. Otherwise, the SN will transit its real sensed reading to the CH. Experimentation outcomes validate the NQR algorithm as it accomplishes higher prediction correctness, a greater number of positive predictions and reduces energy exhaustion by avoiding data transmission in a positive prediction cycle. Experiments have also shown that NQR outperforms the TDPA and ELR algorithms on the observed parameters.

REFERENCES

[1] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. IEEE Communications magazine, 40(8), 102-114.

[2] Yick, J., Mukherjee, B., & Ghosal, D. (2008). Wireless sensor
network survey. *Computer networks*, *52*(12), 2292-2330.

[3] Jain, K., & Bhola, A. (2018). Data aggregation design goals for monitoring data in wireless sensor networks. *Journal of Network Security Computer Networks*, *4*(3), 1-9.

[4] Pourghelbeh, B., & Navimipour, N. J. (2017). Data aggregation mechanisms in the Internet of things: A systematic review of the literature and recommendations for future research. *Journal of Network and Computer Applications*, *97*, 23-34.

[5] Gupta, K., & Sikka, V. (2015). Design issues and challenges in wireless sensor networks. *International Journal of Computer Applications*, *112*(4), 0975-8887.

[6] Matos, T. B., Brayner, A., & Maia, J. E. B. (2010, March). Towards in-network data prediction in wireless sensor networks. In *Proceedings of the 2010 ACM Symposium on Applied Computing* (pp. 592-596).

[7] Krishna, G., Singh, S. K., Singh, J. P., & Kumar, P. (2018, May). Energy conservation through data prediction in wireless sensor networks. In *Proceedings of 3rd International Conference on Internet of Things and Connected Technologies (ICIoTCT)* (pp. 26-27).

[8] Agarwal, A., Gupta, K., & Yadav, K. P. (2016, March). A novel energy efficiency protocol for WSN based on optimal chain routing. In *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 368-373). IEEE.

[9] Wei, G., Ling, Y., Guo, B., Xiao, B., & Vasilakos, A. V. (2011). Prediction-based data aggregation in wireless sensor networks: Combining grey model and Kalman Filter. *Computer Communications*, *34*(6), 793-802.

[10] Sinha, A., & Lobiyal, D. K. (2015). Prediction models for energy efficient data aggregation in wireless sensor network. *Wireless Personal Communications*, *84*(2), 1325-1343.

[11] Avinash, R. A., Janardhan, H. R., Adiga, S., Vijeth, B., Manjunath, S., Jayashree, S., & Shivashankarappa, N. (2015, December). Data prediction in wireless sensor networks using Kalman filter. In *2015 International Conference on Smart Sensors and Systems (IC-SSS)* (pp. 1-4). IEEE.

[12] Raza, U., Camerra, A., Murphy, A. L., Palpanas, T., & Picco, G. P. (2015). Practical data prediction for real-world wireless sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, *27*(8), 2231-2244.
[13] Yue, Y., Li, J., Fan, H., Qin, Q., Gu, L., & Du, L. (2017). Fault prediction based on the kernel function for ribbon wireless sensor networks. *Wireless Personal Communications*, 97(3), 3277-3292.

[14] Wang, T., & Wang, M. (2018). Communication network time series prediction algorithm based on big data method. *Wireless Personal Communications*, 102(2), 1041-1056.

[15] Rajesh, G., & Chaturvedi, A. (2019, December). Correlation Analysis of Multimodal Sensor Data in Environmental Sensor Networks. In *2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)* (pp. 1-6). IEEE.

[16] Cheng, H., Xie, Z., Shi, Y., & Xiong, N. (2019). Multistep data prediction in wireless sensor networks based on one-dimensional CNN and bidirectional LSTM. *IEEE Access*, 7, 117883-117896.

[17] Jain, K., & Kumar, A. (2020). An energy-efficient prediction model for data aggregation in sensor network. *Journal of Ambient Intelligence and Humanized Computing*, 11(11), 5205-5216.

[18] JayaSingh, R. (2020). Augmented Data Prediction Efficiency for Wireless Sensor Network Application by AI-ML Technology. In *Deep Learning Strategies for Security Enhancement in Wireless Sensor Networks* (pp. 330-348). IGI Global.

[19] Jain, K., & Kumar, A. (2020). Energy-Efficient Data-Aggregation Technique for Correlated Spatial and Temporal Data in Cluster-Based Sensor Networks. *International Journal of Business Data Communications and Networking (IJBDCN)*, 16(2), 53-68.

[20] El-Sayed, W. M., El-Bakry, H. M., & El-Sayed, S. M. (2020). Integrated data reduction model in wireless sensor networks. *Applied Computing and Informatics*.

[21] Nels, S. N., & Singh, J. A. P. (2021). Hierarchical Fractional Quantized Kernel Least mean Square Filter in Wireless Sensor Network for Data Aggregation. *Wireless Personal Communications*, 1-22.

[22] Jain, K., Agarwal, A., & Kumar, A. (2021). A novel data prediction technique based on correlation for data reduction in sensor networks. In *Proceedings of International Conference on Artifi-
cial Intelligence and Applications (pp. 595-606). Springer, Singapore.

[23] Heinzelman, W. R., Chandrakasan, A., & Balakrishnan, H. (2000, January). Energy-efficient communication protocol for wireless microsensor networks. In Proceedings of the 33rd annual Hawaii international conference on system sciences (pp. 10-pp). IEEE.

[24] Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions on wireless communications, 1(4), 660-670.

[25] Issariyakul, T., & Hossain, E. (2009). Introduction to network simulator 2 (NS2). In Introduction to network simulator NS2 (pp. 1-18). Springer, Boston, MA.