A Query Details

On each datasets, we conduct two queries. Each query targets specific metrics within its respective dataset, offering diverse insights into time-series behavior, statistical characteristics, or cumulative values across different IoT domains.

The first two queries, Q1 and Q2 are conducted on the dataset Weather Forecast. Q1 calculates the average temperature when the minimum temperature is above 20 degrees Celsius, and the result is converted to Fahrenheit. Q2 calculates the average wind level and wind direction, grouped by 1-minute intervals within the specified time range. The results represent wind characteristics over short time intervals. Q3 and Q4 are conducted on the AMPds dataset. Q3 providing statistical insights into the variation and dispersion of current values within the dataset, while Q4 calculates the total apparent energy (S) in kilovolt-ampere hours (kVAh) and the total real power (P) in kilowatt-hours (kWh), providing cumulative energy usage metrics. Q5 and Q6 are conducted on the Smart Grid dataset. Q5 indicates daily average plug energy usage. And Q6 calculates cumulative daily household energy usage. Q7 and Q8 are conducted on the Linear Road dataset. Q7 offers insights into the variability of direction data over daily intervals for a specific lane, while Q8 calculates the average speed, grouped by 1-day intervals within the specified time range, representing the average daily speed of vehicles. The last two queries, Q9 and Q10, are conducted on the Computer Monitor dataset. Q9 computes the variance of cpu usage, grouped by 1-day intervals within the specified time range. It highlights daily variations in CPU utilization. And Q10 reveals daily fluctuations in task priority levels.

B IoT-benchmark Data Generation Algorithm

Algorithm 1 describes the data generation strategy. For parameter details, please refer to Table 1.

Table 1: Parameters Description of Data Generator

Notation	Data Features
μ_v	Mean of values
μ_d	Mean of deltas
θ_d	Variance of deltas
γ	Repetition Rate
η	Increase rate
l	Series length

C Proof of Proposition 4.7

The proof of Proposition 4.7 is as follows.

Proof. Let $Cost\left(I_u\left(c\right)\right)$ denote the cost of the traditional query process of a compressed database, i.e. decompressing first, then restore and query on uncompressed data, where

$$\begin{aligned}
&\mathrm{Cost}(I_u\left(c\right)) = Cost\left(\left(Q_u \circ R_u \circ U\right)\left(c\right)\right) = Cost\left(Q_u\left(R_u\left(U\left(c\right)\right)\right)\right)\\
&= \mathrm{Cost}\left(\left\{u: \begin{array}{l} u_1 \leftarrow op_1\left(u_0\right), u_2 \leftarrow op_2\left(u_1\right), \\ \ldots, u \leftarrow op_n\left(u_{n-1}\right) \end{array} \middle| \begin{array}{c} i \in \{1,\ldots,n\} \\ op_i \in \Pi \end{array} \right\}\right)\\
&\quad + \mathrm{Cost}(R_u\left(u\right)) + Cost\left(U\left(c\right)\right)\\
&= \mathrm{Cost}(U\left(c\right)) + Cost\left(R_u\left(u\right)\right) + Cost\left(\left\{op_1, op_2, \ldots, op_{n-1}\right\}\left(u_1\right)\right).
\end{aligned}$$

Algorithm 1: Numerical Data Generator [65]

Input: μ_v, μ_d, θ_d, γ, η, length n
Output: TS
$DS \leftarrow$ empty_list();
while $|DS| < n$ **do**
  $isRepeat \leftarrow$ random_index(γ);
  if *isRepeat* **then**
    $repeat\_len \leftarrow$ random$(8, T)$;
    DS.append$(0,$ repeat_len$)$;
  else
    $isPositive \leftarrow$ random_index(η);
    $delta \leftarrow 0$;
    if *isPositive* **then**
      while $\delta \leq 0$ **do**
        $\delta \leftarrow$ random_gauss(μ_d, θ_d);
      end
    else
      while $\delta \geq 0$ **do**
        $\delta \leftarrow$ random_gauss(μ_d, θ_d);
      end
    end
    DS.append(δ);
  end
end
$TS \leftarrow$ prefix_sum(DS);
TS.zoom(μ_v);
return TS;

Let $Cost\left(I_c\left(c\right)\right)$ denote the cost of the partial homomorphic query process of a compressed database, where

$$\begin{aligned}
&\mathrm{Cost}(I_c\left(c\right)) = Cost\left(\left(I'_u \circ Q_c \circ R_c\right)\left(c\right)\right) = Cost\left(I'_u\left(Q_c\left(R_c\left(c\right)\right)\right)\right)\\
&= \mathrm{Cost}(I'_u\left(c_j\right)) + Cost\left(Q_c\left(c_0\right)\right) + Cost\left(R_c\left(c\right)\right)\\
&= \mathrm{Cost}\left(\left\{u: \begin{array}{l} u_j \leftarrow op_j\left(u_{j-1}\right), u_{j+1} \leftarrow op_{j+1}\left(u_j\right), \\ \ldots, u \leftarrow op_n\left(u_{n-1}\right) \end{array} \middle| \begin{array}{c} i \in \{j,\ldots,n\} \\ op_i \in \Pi \end{array} \right\}\right)\\
&\quad + \mathrm{Cost}\left(\left\{c_j: \begin{array}{l} c_1 \leftarrow op'_1\left(c_0\right), c_2 \leftarrow op'_2\left(c_1\right), \\ \ldots, c_j \leftarrow op'_j\left(c_{j-1}\right) \end{array} \middle| \begin{array}{c} i \in \{1,\ldots,j\} \\ op'_i \in \Theta \end{array} \right\}\right)\\
&\quad + \mathrm{Cost}(U\left(c_j\right)) + Cost\left(R_c\left(c\right)\right)\\
&= \mathrm{Cost}(R_c\left(c\right)) + Cost\left(\left\{op'_1, op'_2, ..op'_j\right\}\left(c_0\right)\right) + Cost\left(U\left(c_j\right)\right)\\
&\quad + \mathrm{Cost}(\{op_{j+1}, op_{j+2}, \ldots, op_{n-1}\}\left(u_{j-1}\right)),
\end{aligned}$$

with $\varphi\left(u_i\right) = c_i, j \in \{1...n-1\}$. Then we have

$$\begin{aligned}
&\mathrm{Cost}(I_u\left(c\right)) - Cost\left(I_c\left(c\right)\right)\\
&= \mathrm{Cost}(U\left(c\right)) + Cost\left(R_u\left(u\right)\right) + Cost\left(\left\{op_1, op_2, \ldots, op_{n-1}\right\}\left(u_0\right)\right)\\
&\quad - \left(Cost\left(R_c\left(c\right)\right) + Cost\left(\left\{op'_1, op'_2, ..op'_j\right\}\left(c_0\right)\right) + Cost\left(U\left(c_j\right)\right)\right.\\
&\quad \left. + \mathrm{Cost}(\{op_{j+1}, op_{j+2}, \ldots, op_{n-1}\}\left(u_{j-1}\right))\right)\\
&= \left(Cost\left(U\left(c\right)\right) - Cost\left(U\left(c_j\right)\right)\right) + \left(Cost\left(R_u\left(u\right)\right) - Cost\left(R_c\left(c\right)\right)\right)
\end{aligned}$$

$$+\left(Cost\left(\{op_1, op_2, \ldots, op_{n-1}\}\left(u_0\right)\right) - Cost\left(\{op'_1, op'_2, ..op'_j\}\left(c_0\right)\right)\right.$$
$$\text{-Cost}\left(\{op_{j+1}, op_{j+2}, \ldots, op_{n-1}\}\left(u_{j-1}\right)\right)$$

$$= \left(Cost\left(U\left(c\right)\right) - Cost\left(U\left(c_j\right)\right)\right) + \left(Cost\left(R_u\left(u\right)\right) - Cost\left(R_c\left(c\right)\right)\right)$$
$$+\left(Cost\left(\left\{op_1, op_2, \ldots, op_j\right\}\left(u_0\right)\right) - Cost\left(\left\{op'_1, op'_2, ..op'_j\right\}\left(c_0\right)\right)\right)$$

Note that when $j = n-1$, query Q is a fully homomorphic query.

By Lemma 4.6, $Size\left(c\right) \geq Size\left(c_j\right)$, thus, $Cost\left(U\left(c\right)\right) \geq Cost\left(U\left(c_j\right)\right)$. And by Definition 4.5, we have

$$Cost\left(\left\{op_1, op_2, \ldots, op_j\right\}\left(u_0\right)\right) \geq Cost\left(\left\{op'_1, op'_2, \ldots, op'_j\right\}\left(c_0\right)\right).$$

And by Definition 4.4, we have

$$Cost\left(R_u\left(u\right)\right) \geq Cost\left(R_c\left(c\right)\right)$$

Thus, we have $Cost\left(I_u\left(c\right)\right) - Cost\left(I_c\left(c\right)\right) \geq 0$, i.e.,

$$Cost\left(I_c\left(c\right)\right) \geq Cost\left(I_c\left(c\right)\right).$$

□