Research Article

A Novel Machine Language-Driven Data Aggregation Approach to Predict Data Redundancy in IoT-Connected Wireless Sensor Networks

G. C. Jagan¹ and P. Jesu Jayarin²

¹Faculty of Electronics Engineering, Sathyabama Institute of Science and Technology, Chennai, India
²Institute of Information Technology, Department of Data Analytics, Saveetha School of Engineering, SIMATS, Chennai-602105, Tamil Nadu, India

Correspondence should be addressed to G. C. Jagan; jagan11gc@gmail.com

Received 30 June 2022; Accepted 20 September 2022; Published 18 October 2022

Academic Editor: Yan Huang

Copyright © 2022 G. C. Jagan and P. Jesu Jayarin. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Real world data aggregation and delivery in Internet of Things (IoT) technology are essential to predict and retrieve target data in short time so that the end user feels no delay but ensures a high quality of information. In addition to habitat monitoring and disaster management, these networks have a wide range of other uses, including security and military operations. The processing capabilities of sensor nodes are restricted due to the fact that they have a limited battery life and hence a modest size and processing capacity. WSNs are also susceptible to failure as a result of the limited battery power available. In WSNs, data aggregation is practiced as an energy efficient strategy to reduce computing and transmission latency. It is because of sensor node distribution density that shares the same data at a time data redundancy comes to exist. It is possible to reduce redundancy by adopting a suitable machine learning algorithm while executing the data aggregation process. Researchers are still chasing behind algorithms and modeling strategies effectively to ease the process of developing an effective and acceptable data aggregation strategy from existing wireless sensor network (WSN) models. A three stage framework is proposed for an efficient data aggregation mechanism, and the stages are Modified LEACH, extreme learning machines (ELM), adaptive Kalman filter, and Bi-LSTM. This experiment result shows better performance than the existing methods.

1. Introduction

Today, the world is experiencing a tremendous use of digital data due to which the growth in data collection and distribution processes has been rising in a rapid pace which in turn leads to be a vital decisive factor in developing IoT system architectures [1]. It is evident that the amount of data collected from various digital communication infrastructures gets doubled at the rate of two times a year. Several efficient data collection models are developed to upgrade the performance of many IoT-assisted sensing applications. The sensor systems perform data acquisition which forms the first part of data collection process to provide better services to users. It is known that there are several numbers of sensors deployed in wireless sensor networks to perform data collection from industrial and other natural or artificial environments. The main goal of sensor deployment is to sense data of intent from a hostile area. As these sensing mechanisms consume significant energy, different network-based data processing mechanisms are considered to develop sensor centric IoT applications. Such data aggregation techniques in IoT-enabled networks conserve energy with network longevity.

WSN finds its applications in areas such as home automation, monitoring different types of surroundings, healthcare, and industrial control to mention a few. The communication between constituent sensors will lie in a short range only [2, 3], wherein the sensor nodes have limited bandwidth and other associated resources. The sensor works on collecting signals from sources such as light,
temperature, and heat and passes it to data conversion units called microcontrollers. Overall communication effectiveness of sensor nodes would be dependent on type of data aggregation techniques employed. Data aggregation is carried out with few objectives so that it reduces energy consumption along with other resource utilizations. The network lifetime would also increase when the data aggregation algorithm is carefully chosen [4]. Data aggregation is preferred in cases where multiple sensor nodes are in operation to fetch signals on a same parameter under a high node density scenario.

In this research work, energy minimization is carried out in two stages such that at first, data prediction and, secondly, a statistical prediction modeling are performed out, respectively. Data prediction is done in an IoT network to predict future data coming from all live nodes. A given IoT network contains a sensing node known as aggregator to collect data and broadcast it to remaining nodes. The aggregator node sends only the required amount of data instead of sending all received data after processing it using suitable data precipitation techniques. Besides, a sufficient data reduction is further achieved during the second phase using a statistical data prediction model to identify neighboring nodes which periodically generate data.

Figure 1 portrays various applications of IoT, where data aggregation plays a major role in a large scale. Thus, this manuscript focuses on developing efficient data aggregation mechanisms for IoT-enabled machines available in industries.

2. Literature Survey

WSN is prominently used in IoT, to collect environmental data because of its large-scale deployment, low cost, and low-energy consumption. It is time-consuming and difficult to reduce the amount of data gathered and transferred across a network without affecting data integrity. Literature [5] proposes a two-stage model in which the first stage is for effective data collection by unmanned aerial vehicle (UAV) where as the next stage is NP hard maximization problem to model the full or partial collection of data by hovering of the UAVs. This proposed two stage model tries to maximize data collection with minimal energy consumption.

Researchers on [6] proposed an Energy-Efficient Data Aggregation Mechanism (EEDAM) to save energy at the cluster level. Edge computing is used to give on-demand trusted services to IoT devices with the least amount of delay, and blockchain is incorporated into a cloud server for verifying the edge in order to provide secure services to IoT devices with the least amount of delay. In work [7], author presents methods on how to effectively deal with the data veracity issue that arises due to the existence of misbehaving nodes, outliers, missing readings, and redundancy in the raw IoT sensor data by using a data aggregation technique. The data aggregation methodology is intended for use with extremely uncertain raw IoT sensor data acquired by device to device connection, as opposed to other methods.

In [8], authors presented a review paper for aggregation which proposes LTE-WLAN aggregation (LWA) that is implemented using a Software-Defined Networking (SDN) based technique to manage aggregation across LTE and WLAN Access Points eliminating excess connection attempts thus serving users only with essential services. The genetic algorithm is used to pick the best WLAN access point. This reduces the traffic demand in licensed spectrum and increases the UE throughput. Another paper [9] describes several energy efficient data aggregation techniques employed in sensor networks.

In [10], authors reported a review article which covers a detailed analysis of methodical analysis of data aggregation in WSN. Here, they discussed about challenges in data aggregation and various methods and tools used. In [11], a survey is found on various data aggregation models which used machine learning techniques and finally, authors proposed a novel priority-based data aggregation (PbDA) technique which is machine learning based on confrontation emergency situations. Again, in [12], research work proposes an adaptive event differential privacy (Re-ADP) system, and all the collected sensor information at different timings may be protected sequentially through an unlimited stream of data in real time, without compromising performance. They are meant to provide aggregated data to cloud storage; it may reduce the processing load on cloud storage servers, enhance communication efficiency, and preserve the privacy of data sent to them.

Another unique data aggregation approach with network clustering and an extreme learning machine (ELM) that effectively removes unwanted and erroneous data is identified in [13]. The instability during training phase of ELM is tackled using basis functions. All the sensed data are preprocessed to filter noise using Kalman filter before delivering data to a particular CH. This supports to accuracy improvements. Again, an energy-efficient data aggregation strategy is proposed in [14] in which IoT nodes encode sensor input into a binary format before routing. Next, the data is compressed at the edge node and then pumped into the IoT cloud through the shortest route. An accurate data aggregation and prediction model improves performance of the cloud.

An energy efficient LEACH protocol is introduced [15, 16] to improve network routing by reducing resource consumption. Clustering points out an appropriate cluster head to reschedule TDMA slots of a particular sensor node and to balance all sensors data transmission such that each node sends the same quantity of data. This reduces energy consumption of nodes and hence increases life of the network.

The research contributions in [17] focus on theoretical analysis of extreme learning machine (ELM) to achieve improvements in the context of stability, efficiency, and accuracy of WSNs. The work present in [18] aims at fast traffic forecasting to predict vehicle count that is anticipated in the successive time period in a traffic junction. During uncertain signal transmissions, adaptive Kalman filters offer reasonable prediction intervals according to empirical evidence, along with a better adaptability in a variable traffic scenario. It is inferred that during sensitivity analysis, the adaptive Kalman filter performance is stable as its memory capacity increases.
In [19], a CLSTM-based model is proposed to precipitate the nowcasting issues. It uses both input and prediction output as spatiotemporal sequences. The resultant correlations outperform FC-LSTM and operational ROVER algorithm. Research reports in [20] used an LSTM network enhanced with word embeddings which is already trained using a significant number of Twitter message samples. It applied tokenization, word normalization, segmentation, and spell correction to optimize the identification of the most significant words.

Study in work [21] reveals that the research work implements aggregate sales forecasting using the deep learning algorithm ConvLSTM, which is developed by the University of Michigan. When looking at the sales forecasting, it might take geographical correlations between neighboring shops into consideration. It is discovered that the proposed ASFC method reduces errors and improves prediction quality. It is the goal of this effort to develop aggregate sales forecasting using the deep learning algorithm ConvLSTM.

Research in [22] uses a new prediction approach to develop sensor-connected IoT applications. Bi-LSTM and 1-D CNN are used to extract characteristics with distinct features, resulting in one-step prediction. After recursively combining previous data with new prediction findings, a multistep prediction model is arrived that significantly improved the performance.

An IoT system for Wireless Medical Sensor Network (WMSN) is proposed [23] to monitor medical data transmitted to central storages. The work deals with offering highly secured data aggregation and then transmit to desired locations. This prevents the intrusion of undesired users. The current schemes are significantly complex due to the use of complex product functions to generate batch keys. Thus, it makes the systems experience high computational complexity and large memory utilization. Authors presented a new lightweight Secure Aggregation and Transmission Scheme (SATS) that has a low complex EXOR logic to find out the batch key by eliminating the tedious multiplication steps. In addition, the work includes Aggregator Node Receiving Message Algorithm (ARMA) for effective data aggregation. This collective approach is found to be one of the preferred choices for SATS to mitigate security threats, viz., denial of service, man in the middle, and reply kind of attacks in a given IoT-WSN scenario. Simulations in NS2 show that the proposed SATS presents a lightweight type of data transmission, minimal computations, and allied communication costs along with an improved memory size and low energy consumption.

As the security of WSN data transmission is a key factor to determine the quality of service, there are data aggregation (DA) schemes [24] framed with a suitable security mechanism to offer safe and reliable data delivery. This presents a review of secure data aggregation (SDA) focusing only on security threats.

This work [25] presents an energy aware and secured data aggregation algorithm. End results show that the proposed approach preserves nodal energy significantly besides achieving a prolonged network lifetime. Data aggregation becomes very effective in large scale of WSNs wherein a huge volume of data is involved and out of which only a particular size is useful, whereas the remaining are said to be redundant [26]. An increased redundancy will decrease the system performance in the context of additional computational overhead and unnecessary transmission besides memory wastage. Data aggregation aims at data mining where only the useful data is precipitated to ensure data transmission with better consistency, accuracy, and efficiency. Data mining plays a pivotal role in wireless sensor networks combined with Internet of Things to achieve remote data communication. A new redundancy checking approach is
proposed in this work that performs better redundant data mining compared to other counterparts. Table 1 illustrates the comparison of some existing methods.

3. Proposed Method

With the help of WSN, we have developed an effective data aggregation technique for utilization prediction in IoT operated machines. To predict the following three stages, Modified LEACH, extreme learning machines (ELM), adaptive Kalman filter, and Bi-LSTM are used as shown in Figure 2.

3.1. M-LEACH. The LEACH method requires few nodes to assign as cluster heads which are more distant from the BS than they are, in order for the process to work. The sensor nodes transmit their sensed data to the centralized access points. Extra transmissions are communications that squander the energy of the network and are thus referred to as such. The suggested protocol operates over a large area, similar to that of a wide area network but simultaneously reducing the complexity of communication and the complexity of time management. All sensor nodes are deployed into a scalable distributed cluster environment in accordance with the suggested protocol, and the region is divided into various numbers of clusters. All the clusters are assigned with fixed cluster nodes. Any individual node should be attached with any of the cluster available in the network. Cluster heads are selected based on the modified LEACH algorithm. Only CHs will communicate with base stations.

Steps to be followed in modified LEACH:

Step 1. Deployment of sensor nodes.

Step 2. Formation of cluster.

Step 3. Calculate cluster head threshold for all nodes.

Step 4. Check the threshold value. If threshold is not higher, then the node is not a cluster head (CH) else proper selection of cluster head (CH).

Step 5. CH waits for join request messages.

Step 6. Broadcast a message from one node to all CHs.

Step 7. Modify TDMA schedule duration based on the largest cluster and send its cluster members.

Step 8. Sensor nodes send sensed data to its CH.

Step 9. Data aggregation on CH.

Step 10. Sent the data to extreme learning machines (ELM).

3.2. Extreme Learning Machines (ELM). The LEACH output is fed into an ELM to eliminate excess and error prone data. As illustrated in Figure 3, the ELM is a feed forward neural network with two stages of learning. The projection stage is nontrainable, and the input weights are chosen at random. No iterative calculation is required. This feature reduces computing time for training the model, but random selection of biases and weights causes prediction instability. To overcome ELM’s flaw, a Mahalanobis distance-based radial basis function (MDRBF) is suggested to be integrated with ELM’s network.

3.3. Adaptive Kalman Filter (AKF). In the field of data fusion, AKF is one of the most often utilized approaches. It decreases the amount of noise in the data and provides an accurate approximation of the state vector containing

The following is the formula for the ELM network with k hidden nodes:

\[ a_j(c) = \sum_{j=1}^{k} f^j(w_j, c) + b_j, \]  

where \( w \) is the weight and \( b \) is the bias.

\[ d(c_j) = \sum_{j=1}^{k} (w_j a_j) + b'_j, \]

\[ Xw' = d, \]  

\[ X \rightarrow \text{output of the hidden layers.} \]  

The following equation is solved by least-square fitting:

\[ F = \min \left( Xw' - y_j \right). \]  

\[ TD \ (Training \ Data) = \{(c_j, d_j), (j = 1, 2, 3, \ldots, n), \]  

\( c_j \rightarrow \text{input data}, \ d_j \rightarrow \text{target data.} \]

\[ X = \left[ \begin{array}{ccc}
\alpha(w_1 c_1 + a_1) & \cdots & \alpha(w_n c_1 + a_k) \\
\vdots & \ddots & \vdots \\
\alpha(w_1 c_n + b_k) & \cdots & \alpha(w_n c_n + b_k)
\end{array} \right], \]  

\[ Xw' = d, \]  

\[ \alpha \rightarrow \text{mdrbf} \]
Table 1: Reference comparison table.

<table>
<thead>
<tr>
<th>Author details</th>
<th>Journal details</th>
<th>Observation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Huo et al. [12]</td>
<td>Wireless Communications and Mobile Computing (2018)</td>
<td>A real-time stream data aggregation framework with adaptive-event differential privacy is proposed. Authors used privacy protection and smart grouping based on K-means clustering. Results shown outperform the existing works.</td>
</tr>
</tbody>
</table>

Figure 2: Three stages of data aggregation and prediction.

Figure 3: Structure of ELM network.
Figure 4: Structure of one cell.

Figure 5: Structure of Bi-LSTM.

Figure 6: Sensor arrangement (courtesy: Intel Lab Berkeley).
Figure 7: Experimental setup.

Figure 8: OPNET-network environment setup.
valuable information. It has been widely used for a variety of applications, including estimate, tracking, and sensor fusion.

Step for AKF is given below:

**Step 1.** Find the prior state estimation error covariance

$$\hat{S}_{k|k-1} = \hat{\sigma}_{k-1|k-1} \hat{G}^t + R,$$

$$\hat{G}_{k|k-1} = \hat{\sigma}_{k-1|k-1} \hat{G}^t + R,$$

$$\hat{G}_{k|k-1} : \text{Prior State Estimation error covariance.}$$

**Step 2.** Compute the errors

$$E_k = Z_k - Y_k^T \hat{S}_{k|k-1},$$

$$E_k : \text{Observation of errors.}$$

**Step 3.** Update observation process covariance matrix $C_k$

$$\hat{E} = \frac{1}{M} \sum_{i=1}^{M} E_{k,i+1},$$

$$\hat{E} : \text{Average observation errors,}$$

$$C_k = \frac{1}{M} \sum_{i=1}^{M} \left( (E_{k,i+1} - \hat{E})(E_{k,i+1} - \hat{E})^t - \frac{M - 1}{M} Y_{k,i+1} \hat{G}_{k,i+1|k-1} \hat{G}^t + R \right),$$

where $M$ is the AKF memory size.

**Step 4.** Compute the gain of the AKF

$$KG_k = \frac{\hat{G}_{k|k-1} Z_k}{Z_k^T \hat{G}_{k|k-1} Z_k + C_k},$$

$$KG_k : \text{Kalman gain at time } k.$$
**Table 2: Accuracy in %**

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>11:07 am to 12:07 pm (60 minutes divided into each 6 minutes)</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDP [5]</td>
<td>97.105 97.635 97.14 96.59 96.39 96.2 98.005 96.46 96.135 97.26</td>
<td>96.892</td>
</tr>
<tr>
<td>IDAD2DC [7]</td>
<td>94.965 94.295 93.76 94.79 94.675 94.52 95.52 95.8 95.895 96.635</td>
<td></td>
</tr>
<tr>
<td>READP [12]</td>
<td>96.195 94.7 95.835 96.085 96.03 96.225 95.8 95.895 95.8</td>
<td></td>
</tr>
<tr>
<td>SDNAELWA [8]</td>
<td>95.91 96.15 96.715 96.55 97.055 95.77 96.415 97.305 97.195 97.27</td>
<td></td>
</tr>
<tr>
<td>MLELMAKF</td>
<td>98.18 98.395 97.79 98.125 98.54 97.925 97.93 98.32 97.6</td>
<td>97.725</td>
</tr>
</tbody>
</table>

**Figure 10: Error rate.**

**Step 5.** Estimate the posterior state and its covariance error

\[
\hat{S}_{jk} = \hat{S}_{j-1,k} + KG_i E_k,
\]

\[
\tilde{G}_{jk} = (1 - KG_k V_i^T)^2 \tilde{G}_{j-1},
\]  

\[G_{jk} \text{ : Posterior State Estimation error covariance.} \]

**Step 6.** State estimation error computation

\[
b_i = \hat{S}_{jk} - \Phi \tilde{S}_{j-1,k-1} \]

\[b_i \text{ : system state estimation errors.} \]

**Step 7.** Update state process covariance matrix \(Q_k\)

\[
\bar{b} = \frac{1}{M} \sum_{i=1}^{M} b_{k-i+1},
\]

\[\bar{b} \text{ : average system state estimation errors,} \]

\[Q_k = \frac{1}{M} \sum_{i=1}^{M} \left( (\hat{h}_{k-i+1} - \bar{b})^T (\hat{h}_{k-i+1} - \bar{b}) \right) - \frac{1}{M} \sum_{i=1}^{M} \tilde{z}_{k-i+1} \tilde{z}_{k-i+1}^T \]

**3.4. Bidirectional LSTM.** RNN is a type of conventional LSTM approach with its module contains a singular neuronal structure to represent the human brain. In LSTM, the module is made up of cells that each have three gates. These two modules are organized into a chain structure. The three gates of a cell are named as input, hidden, and output, respectively, as illustrated in the following Figure 4.

The mathematical models for these gates mentioned are as follows. The input is defined as

\[x(t) = \delta(W_jt(t) + V_j h(t-1) + a_i),\]

in which \(h(t-1)\) denotes previous gate, \(i(t)\) is the current input cell, \(\delta\) is the sigmoid function, and \(W_j\) and \(V_i\) are the weights of the input gates.

\[g(t) = \delta(W_jit(t) + V_j h(t-1) + a_j),\]

where \(g(t)\) is the forget gate in the cell, and \(W_j\) and \(V_j\) are the weights of the forget gates.

\[\tilde{z}(t) = \text{tanh} \left( W_j t + V_j h(t-1) + a_i \right),\]

where \(\tilde{z}(t)\) will update the memory unit, which will update the alternate information.

\[z(t) = g(t) \star z(t-1) + x(t) \star \tilde{z}(t),\]

where \(z(t)\) will update the cell.

Information from the forgotten gate is merged with the updated information which results in a new state, where \(W_j\) and \(V_j\) are weights of forgotten gates and updated information, respectively, \(\star \) is the Hadamard product, and a new alternative state is

\[\text{Output}(t) = \delta(W_o x(t) + V_o h(t-1) + a_o),\]

\[h(t) = \text{output}(t) \star \text{tanh} \left( z(t) \right),\]

where \(W_o\) and \(V_o\) are the weights of the output gate.

Here, equations (18) and (19) are used to calculate the output gate.
The input data is processed by both forward and backward layers using activation functions, and the final output is created as a result of this processing as illustrated in Figure 5.

4. Experiment Setup

This work uses the Intel Indoor dataset [27] which comprises four types of data that are acquired using 54 nodes from Intel Research Lab, Berkeley, as shown in Figure 6. The data is divided into four categories, viz., temperature, humidity, light, and voltage, respectively.

Mica2Dot types of sensors are used in this setup. The sensor board collects timely topological data along with temperature, light, humidity, and voltage information in every 31 seconds such that on one sensor per 31 seconds, the data was gathered from a tiny database developed on a TinyOS.

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDP [5]</td>
<td>2.895</td>
<td>2.365</td>
<td>2.86</td>
<td>3.41</td>
<td>3.61</td>
<td>3.8</td>
<td>1.995</td>
<td>3.54</td>
<td>3.865</td>
<td>2.74</td>
<td>3.108</td>
</tr>
<tr>
<td>IDAD2DC [7]</td>
<td>5.035</td>
<td>5.705</td>
<td>6.24</td>
<td>5.21</td>
<td>5.325</td>
<td>5.17</td>
<td>5.695</td>
<td>4.825</td>
<td>5.4</td>
<td>3.96</td>
<td>5.2565</td>
</tr>
<tr>
<td>MLELMKAF</td>
<td>1.82</td>
<td>1.605</td>
<td>2.21</td>
<td>1.875</td>
<td>1.46</td>
<td>2.07</td>
<td>2.07</td>
<td>1.68</td>
<td>2.4</td>
<td>2.275</td>
<td>1.947</td>
</tr>
</tbody>
</table>

Figure 11: Precision.
This dataset contains almost 2.3 million recorded values received from sensor outputs. Compressed file size is 34 MB, whereas the uncompressed file has 150 MB size. Dataset is divided and taken into account for every 6 minutes. The model will analyze and provide results for every one hour. Figures 7 and 8 show the network environment setup in OPNET.

5. Results and Discussion

In this work, the following performance metrics are analyzed and proved that the propose method is the better one when compared with existing works which are EEDP, IDAD2DC, READP, and SDNAELWA.
Figure 9 and Table 2 show the percentage of accuracy calculated from the given method and is compared with other existing methods. The average accuracy percentage of the proposed MLELMAKF method is 98.053. The average percentage difference of the proposed method (MLELMAKF) is improved by 1.19%, 3.43%, 2.44%, and 1.45 than the existing methods like EEDP, IDAD2DC, READP, and SDNAELWA, respectively.

Figure 10 and Table 3 show the percentage of error rate of the proposed method with the comparison of the four existing methods. The average error rate percentage of the proposed method is 1.947. The average percentage difference of the proposed method (MLELMAKF) is improved by 45.93%, 91.88%, 75.59%, and 53.42% than the existing methods like EEDP, IDAD2DC, READP, and SDNAELWA, respectively.
Figure 11 and Table 4 show the precision comparisons as percentage. The average precision percentage of the proposed method is 98.053. The average percentage difference of the proposed method (MLELMAKF) is improved by 0.838%, 2.98%, 1.72%, and 1.59% than the existing methods like EEDP, IDAD2DC, READP, and SDNAELWA, respectively.

Figure 12 and Table 5 show the percentage of sensitivity of the proposed method with the comparison of the four existing methods. The average sensitivity percentage of the proposed method is 98.053. The average percentage difference of the proposed method (MLELMAKF) is improved by 1.53%, 3.85%, 3.12%, and 1.33% than the existing methods.
like EEDP, IDAD2DC, READP, and SDNAELWA, respectively.

Figure 13 and Table 6 show the percentage of specificity of the proposed method with the comparison of the four existing methods. The average specificity percentage of the proposed method is 97.66. The average percentage difference of the proposed method (MLELMAKF) is improved by 0.85%, 3.00%, 1.76%, and 1.57% than the existing methods like EEDP, IDAD2DC, READP, and SDNAELWA, respectively.

Figure 14 and Table 7 show the percent F-score comparisons. The average specificity percentage of the proposed method is 98.04. The average percentage difference of the proposed method (MLELMAKF) is improved by 1.18%,
3.42%, 2.42%, and 1.46% than the existing methods like EEDP, IDAD2DC, READP, and SDNAELWA, respectively. Figure 15 and Table 8 show the throughput comparisons with four existing methods from which it is inferred that the proposed method delivers 1358 kbps packets which is far better compared to the values 84, 218,187, and 126 of their respective EEDP, IDAD2DC, READP, and SDNAELWA counterparts.

Figure 16 and Table 9 show that the latency of the proposed method is very low when compared to other existing methods.
Table 9: Latency (ms).

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDP [5]</td>
<td>44</td>
<td>52</td>
<td>44</td>
<td>40</td>
<td>53</td>
<td>46</td>
<td>48</td>
<td>45</td>
<td>54</td>
<td>47</td>
<td>47.3</td>
</tr>
<tr>
<td>IDAD2DC [7]</td>
<td>98</td>
<td>97</td>
<td>91</td>
<td>93</td>
<td>85</td>
<td>89</td>
<td>88</td>
<td>98</td>
<td>78</td>
<td>82</td>
<td>89.9</td>
</tr>
<tr>
<td>READP [12]</td>
<td>79</td>
<td>89</td>
<td>74</td>
<td>77</td>
<td>77</td>
<td>88</td>
<td>80</td>
<td>88</td>
<td>70</td>
<td>85</td>
<td>80.7</td>
</tr>
<tr>
<td>SDNAELWA [8]</td>
<td>50</td>
<td>58</td>
<td>68</td>
<td>56</td>
<td>58</td>
<td>60</td>
<td>55</td>
<td>60</td>
<td>67</td>
<td>69</td>
<td>60.1</td>
</tr>
<tr>
<td>MLELMAKF</td>
<td>25</td>
<td>24</td>
<td>15</td>
<td>16</td>
<td>13</td>
<td>21</td>
<td>12</td>
<td>14</td>
<td>15</td>
<td>21</td>
<td>17.6</td>
</tr>
</tbody>
</table>

Figure 17: End-to-end delay.
Table 10: End to-end delay (ms).

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDP [5]</td>
<td>234</td>
<td>272</td>
<td>262</td>
<td>216</td>
<td>269</td>
<td>255</td>
<td>276</td>
<td>260</td>
<td>274</td>
<td>263</td>
<td>258.1</td>
</tr>
<tr>
<td>IDAD2DC [7]</td>
<td>495</td>
<td>509</td>
<td>471</td>
<td>477</td>
<td>469</td>
<td>475</td>
<td>444</td>
<td>518</td>
<td>423</td>
<td>447</td>
<td>472.8</td>
</tr>
<tr>
<td>READP [12]</td>
<td>410</td>
<td>453</td>
<td>376</td>
<td>397</td>
<td>412</td>
<td>470</td>
<td>427</td>
<td>458</td>
<td>383</td>
<td>450</td>
<td>423.6</td>
</tr>
<tr>
<td>SDNAELWA [8]</td>
<td>260</td>
<td>329</td>
<td>381</td>
<td>303</td>
<td>309</td>
<td>324</td>
<td>298</td>
<td>335</td>
<td>369</td>
<td>371</td>
<td>327.9</td>
</tr>
<tr>
<td>MLELMAKF</td>
<td>169</td>
<td>152</td>
<td>103</td>
<td>102</td>
<td>75</td>
<td>113</td>
<td>64</td>
<td>113</td>
<td>83</td>
<td>114</td>
<td>108.8</td>
</tr>
</tbody>
</table>

Figure 18: Average processing time.
Table 11: Average processing time (ms).

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDAD2DC [7]</td>
<td>881</td>
<td>885</td>
<td>847</td>
<td>835</td>
<td>833</td>
<td>826</td>
<td>796</td>
<td>880</td>
<td>761</td>
<td>814</td>
<td>835.8</td>
</tr>
<tr>
<td>READP [12]</td>
<td>754</td>
<td>816</td>
<td>712</td>
<td>762</td>
<td>750</td>
<td>844</td>
<td>795</td>
<td>820</td>
<td>716</td>
<td>800</td>
<td>776.9</td>
</tr>
<tr>
<td>SDNAELWA [8]</td>
<td>585</td>
<td>673</td>
<td>730</td>
<td>647</td>
<td>638</td>
<td>660</td>
<td>635</td>
<td>680</td>
<td>703</td>
<td>708</td>
<td>665.9</td>
</tr>
<tr>
<td>MLELMAKF</td>
<td>471</td>
<td>465</td>
<td>403</td>
<td>405</td>
<td>363</td>
<td>410</td>
<td>352</td>
<td>391</td>
<td>362</td>
<td>424</td>
<td>404.6</td>
</tr>
</tbody>
</table>

Figure 19: Average energy consumption.
Both Figure 17 and Table 10 show that the end-to-end delay is very low comparatively. Figure 18 and Table 11 reveal that the average processing time is found to be very low when compared with four of the existing methods. Figure 19 and Table 12 show that average energy consumption is significantly reduced when compared with the existing methods.

6. Conclusion

We conclude that the proposed approach significantly contributes to the development of efficient neural network architecture to perform data aggregation and prediction from the IoT enabled services in an industrial background which employs a typical wireless sensor network. After comparisons across the four existing methods, the proposed prediction model exhibits a significant performance improvement across all metrics of a neural network due to the inclusion of the three steps, viz., ELM-based redundancy removal, AKF-based noise removal, and Bi-LSTM-based prediction, respectively. The proposed approach MLELMaKF has an average accuracy rate of 98.053% which positively differs by 1.19%, 3.43%, 2.44%, and 1.45% compared to current methods, viz., EEDP, IDAD2DC, READP, and SDNAELWA, respectively. Additionally, other performance metrics such as average latency, end-to-end delay, average processing time, and average energy consumption and throughput are obtained as 17.6 ms, 114 ms, 404 ms, and 1570.9 nJ and 1358 kbps, respectively. It is evident from Results that the proposed data aggregation scheme outperforms existing EEDP, IDAD2DC, READP, and SDNAELWA methods significantly. This research can be taken to next levels by adding up the number of layers in ELM and Bi-LSTM stages such that the prediction accuracy of both redundant and error data becomes better with considerable reduction in the number of computations.

Data Availability

The data used to support the finding of this research are accessed from http://db.csail.mit.edu/labdata/labdata.html.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References


Table 12: Average energy consumption (μJ).

<table>
<thead>
<tr>
<th>Time stamp</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>Average</th>
</tr>
</thead>
<tbody>
<tr>
<td>EEDP [5]</td>
<td>1627</td>
<td>1633</td>
<td>1623</td>
<td>1605</td>
<td>1623</td>
<td>1616</td>
<td>1632</td>
<td>1613</td>
<td>1628</td>
<td>1608</td>
<td>1620.8</td>
</tr>
<tr>
<td>IDAD2DC [7]</td>
<td>1833</td>
<td>1821</td>
<td>1825</td>
<td>1819</td>
<td>1811</td>
<td>1807</td>
<td>1824</td>
<td>1804</td>
<td>1819</td>
<td>1828</td>
<td>1819.1</td>
</tr>
<tr>
<td>READP [12]</td>
<td>1686</td>
<td>1653</td>
<td>1681</td>
<td>1674</td>
<td>1671</td>
<td>1678</td>
<td>1676</td>
<td>1682</td>
<td>1686</td>
<td>1655</td>
<td>1674.2</td>
</tr>
<tr>
<td>SDNAELWA [8]</td>
<td>2138</td>
<td>2107</td>
<td>2130</td>
<td>2134</td>
<td>2120</td>
<td>2112</td>
<td>2110</td>
<td>2102</td>
<td>2131</td>
<td>2118</td>
<td>2120.2</td>
</tr>
<tr>
<td>MLELMaKF</td>
<td>1561</td>
<td>1565</td>
<td>1572</td>
<td>1569</td>
<td>1579</td>
<td>1553</td>
<td>1584</td>
<td>1583</td>
<td>1562</td>
<td>1581</td>
<td>1570.9</td>
</tr>
</tbody>
</table>


