International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING

ISSN:2147-6799 www.ijisae.org Original Research Paper

Effective Data Mining with Smart City Scheduling using Recurrent Encoder Neural Networks

Dr. Kavitha S. 1[1], Dr. Shobana R. *[2], Dr. Aruna K. [3] , Dr. Priyadharsini C. [4]

Submitted: 29/01/2024 **Revised**: 07/03/2024 **Accepted**: 15/03/2024

Abstract: Urban areas have drastically increased their population, which results in a shortage of resources like transportation, electricity, water, housing, public services, etc. Therefore, it is important to have a strategy for urban area improvement with the aid of a smart city, which is more apparent when using Wireless Sensor Networks. This study suggests a brand-new, machine learning-based technique for effective data mining for scheduling in smart cities. This works aims to initiate a data prediction method through Recurrent Neural Network, namely recurrent encoder neural netwoks (REncNN) reshold Denoising then remove and discover abstract features of sensory data. Hence predicted data are scheduled in multi-layer network design to contain sensor/device networks in the encoder block of the network. The simulation is done in MATLAB by picking the parameter such as Packet Delivery Ratio (PDR), throughput, network lifetime, Prediction rate, RMSE, RAE. As a result, the proposed REncNN achieves 89.84% of Packet Delivery Ratio, 95.34% of throughput, 81% of network lifetime, 80.94% of prediction rate, 43.12% of MAE, 44.32% of RMSE and 41.92% of RAE.

Keywords: *WSN, Data prediction, Internet on Things(IoT), scheduling, pre-processing, neural network, smart city.*

1. Introduction

Internet of Things (IoT) has advanced quickly in recent years, and WSN are now widely used because of their less energy consumption, versatility, and large deployment[1, 2]. These networks work by sensing, gathering, processing, and transmitting sensory data by cooperating between nodes[3]. The effectiveness of utilizing data prediction techniques to eliminate pointless data transmission enhances data collecting quality and lengthens network lifetime [4]. To forecast specific sensory data, present approaches typically rely on periodicity and redundancy and base their predictions on previous data, which leads to low prediction stability and biassed predictions [5]. Data recovery from lost data is aided by correlation of the sensory data. The temporal correlation, for instance, can be seen when the physical environment is constantly changing.

On the one hand, when the collection length is short enough[6], value of successive sensory data for a single node is often continuous. Instead, sensors are placed in similar physical or environmental settings, and the information they gather is typically spatially correlated [7]. The prediction method can assist end-users in anticipating periodic change of monitored object or area, making it possible to manage any potential risks [8]. Data pretreatment can enhance data quality by recovering part of sensory data that has been lost or converted to a distorted version compared to original value [9]. Deep learning has advanced considerably in recent years [10]. We chose to use deep neural network-based WSNs in this work due of these important factors. The following are the main contributions of this paper:

- Quantile Probability Distribution is used here to analyse the data from the smart environment for further scheduling.
- The recurrent encoder neural network (REncNN), trained to find the ideal next hop, forms the basis of the data prediction technique.

Paper is organized as: In section 1, background of wireless sensor networks,data prediction model and application of DL in data prediction are discussed along withthe contribution. In section 2, existing techniques for data prediction in wireless sensor networks are discussed. Section 3 gives proposed recurrent encoder neural netwoks (REncNN) with preprocessing, data analysis, and prediction. Graphs are produced and experimental analysis is completed in section 4. Section 5, which includes conclusion as well as future work, concludes essay.

2. Related works

Creating intellectual phenomena, such as a method for data evolution, is a component of data prediction. The following explanations of data prediction strategies help to achieve this goal: According to [13], the data collection process in IoT-enabled WSN is congested. A RNN based

[1] Assistant Professor, Auxilum College(Autonomous), Vellore-632 006. ORCID ID : 0000-0002-7874-1138
[2] Assistant Prpfessor, Auxilum College(Autonomous), Vellore-632 006. ORCID ID : 0000-0001-6463-5994
[3] Professor & Vice Principal, Manikan Ramaswami College of Arts & Science, Madurai.. ORCID ID : 0000-0002-5800-0340
[4] Assistant Professor, Karunya Institure of Technology and Sciences, Coimbatore.. ORCID ID : 0000-0002-4656-4878
** Corresponding Author Email: shobanar@auxiliumcollege.edu.in*

LSTM termed RNN-LSTM, which separates network into multiple layers as well as places them into SNs, is included in the provided DMM method in [14]. Adaptive Firefly Routing technique is used in conjunction with an unique routing method based on the Q-learning framework as well as Deep Extreme Learning Machines in [15]. In order to achieve an effective and autonomous calibration procedure, a novel Received Data Strength Indicator based on Indoor Ranging Model utilising Deep Learning (IRMDL) was developed in [16]. A SWSNM built on GAN method, a type of unsupervised learning technique, is introduced in [17]. A generator (G) network and a discriminator (D) network make up SWSNM. Proposed prediction model is utilised in the WSN-IoT smart city architecture shown below to address these problems.

3. System Models

The smart city architecture is considered with various applications such as smart healthcare, smart agriculture, smart transportation and smart education, where sensor data in a wireless network sense the data. Hence, it is forwarded to the WSN-IoT gateway in the network layer. The data prediction and scheduling take place in this layer. Hence, the optimized data is transmitted to WSN-IoT architecture, which then forward to user devices as given in the fig.1.

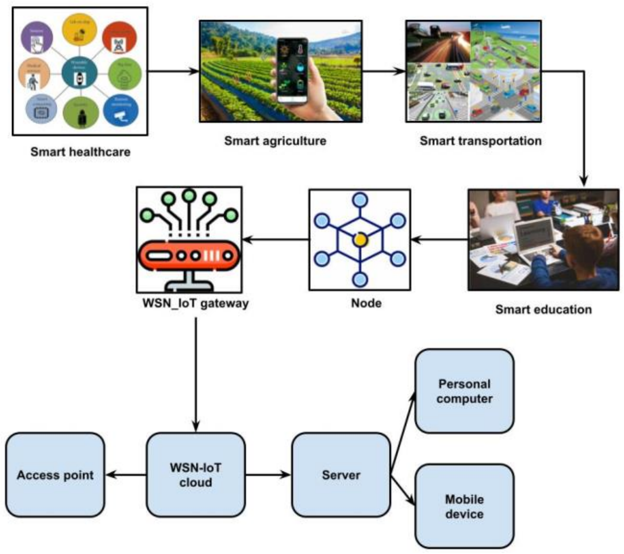

Fig. 1. Architecture of WSN-IoT for Data Prediction and Scheduling

3.1 Analyzing Data Quality by Quantile Probability Distribution (QDP)

The sensors are supposed to be unlabeled and fully distributed throughout the sensor network, which does not contain a fusion centre. Each node uses the edges to communicate with its neighbours. The greatest degree is dmax, while the degree dn indicates number of neighbours at node n. If there is at least one path between each pair of nodes in a graph, it is said to be linked. The wireless communication channel used by each node n to communicate inside neighbourhood Nn is tainted by random noise. Equation (1) illustrates how a Quantile Probability Distribution (QDP) produces the state ωn(i) in the following way:

$$\omega n(i) \rightarrow \theta p,\ \forall n \tag{1}$$

p equals n, the number of nodes. Let ωn(i) and ψn(i) stand for the quantile estimate state and an intermediate state variable, respectively, at iteration i. Based on the nearby measurement information xn for specified constant p, node n modifies its state ωn(i). The algorithm starts with a local update of intermediate variable ψn(i), which ωn(i) is then updated throughout the averaging phase. Equation (2) provides the local update step:

$$\psi n(i) = \omega n(i) - \propto (i)[u\omega n(i) - xn) - p] \tag{2}$$

where $ei \geq 0$ is a sequence with deterministic step size. Following that, as indicated in equation (3), the averaging step is carried out at node n:

$$\psi n(i+1) = \omega n(i) - \beta(i) \sum_{i \in N} \psi n(i) - (\psi n(i) + \pi(i)] \tag{3}$$

Where ψn (i) represents the state that is transferred from node I while being affected by communication random noise at node n, π(i)n represents set of node neighbours, and $\beta(i)$ is step-size that regulates the exchange rate between node n and its neighbours at any given time. Iterative procedure is used in the Quantile Probability Distribution (QDP) method. As demonstrated in equation (4), the step-size I must drop more quickly than the data for $i \rightarrow \infty$, to converge as α(i)

$$\omega(i+1) = \omega(i) - \eta(i)L\,\omega(i) - \alpha(i)y(i) - \alpha(i)y(i) - \pi(i) \tag{4}$$

3.2 Data Preprocessing using Dohono Threshold Denoising

Equation (5) illustrates how the fundamental noise model can be represented if the original data is assumed to be f(t) and the polluted noise data is assumed to be s(t):

$$s(t) = f(t) + \sigma e(t) \tag{5}$$

When the noise is represented by e(t) and the noise strength by σ. We typically assume that e(t) is Gaussian white noise.

$$Ti = \sigma \sqrt{2\log N} \tag{6}$$

Risk given by estimate is the one that is closest to lowest theoretical risk it as shown in equation (7).

$$k(u, v) = corr(u, v) * \frac{\sqrt{Pw(u)/Pcorr(v)}}{w(u,v)} \tag{7}$$

where, $Pw(u) = \sum_v w(u,v)^2$, '*' indicates convolution, $Pcorr(v) = \sum_v corr(u,v)^2$

The modulus values are more than one or less than one at various scales, depending on the various properties of random noise and coefficients of data. New specification h (u, v) should be defined as per equation (8):

$$h(u, v) = 1 - \ln|k(u, v)| \quad (8)$$

Thus, equation (9) shows how the new double threshold approach is set:

$$w(i.j) = \begin{cases} sgn(wj,k) * [wj,k] - \mu 1 e^{\mu 1 - \mu 2} & where, [wj,k] \geq \mu 1 \\ sgn(wj,k) * [wj,k] - \mu 1 e^{\mu 1 - wj,k} & where \mu 1 < \mu 2 \\ 0 & where [wj,k] < \mu 1 \end{cases} \quad (9)$$

where $\mu 2$ is the upper threshold and $\mu 1$ is the lower threshold, satisfying $\mu 1$ = k$\mu 2$ and 0<k<$[wj,k]$ function is continuous at both threshold points, $\mu 1 < [wj,k] < \mu 2$ as $[wj,k]$ increases as shown in fig. 2.

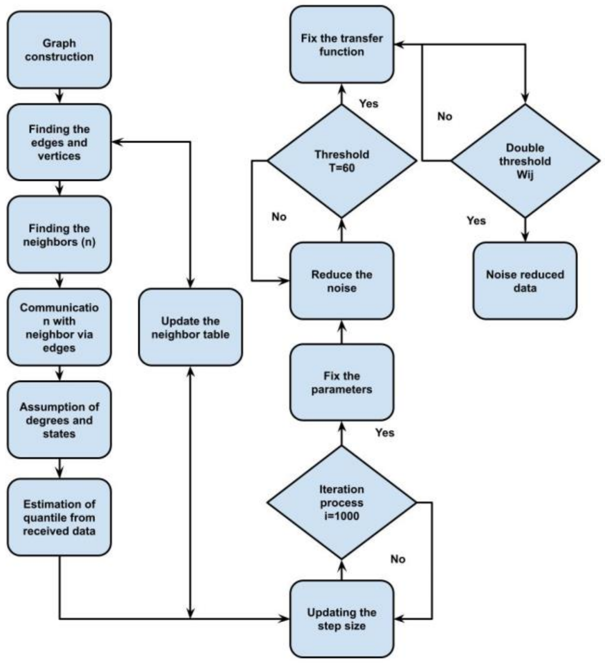

Fig. 2. Flow Chart for Improving Data Quality

3.3 Efficient Data Prediction and Scheduling

Recurrent encoder neural networks (REncNN) incorporate weighted-importance of input relevant series into consideration as compared to single attention encoder-decoder design, as seen in fig. 3.

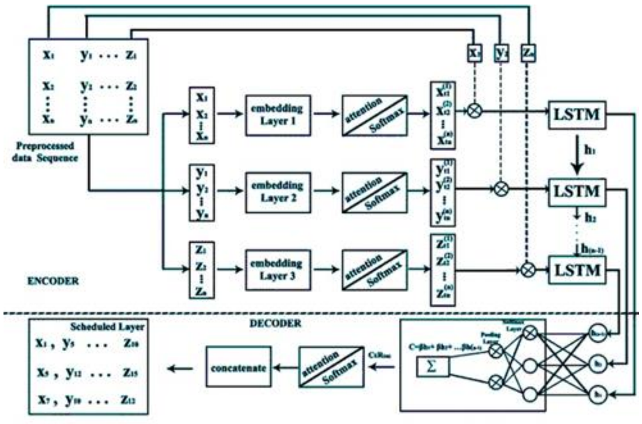

Fig. 3. Data Prediction and Scheduling using Recurrent Encoder Neural Netwoks (REncNN)

Finally, as stated in equation (10) the encoding procedure is updated:

Encode stage: $h_t = f1(x_t, h_{t-1})$ (10)

The equation (11) illustrates the transformation of each original component into a weighted one.

$$x_t = (\alpha^1_t x^1_t, \alpha^2_t x^2_t, \alpha^3_t x^3_{t......} \alpha^n_t x^n_t)^T \quad (11)$$

The hidden state ht-1 and entire kth relevant sequence $x^k = [x^k_1, x^k_2, . . . , x^k_T]$ in all time steps define the attention weight a^k_t. The second attention model used here uses a softmax normalisation and another completely linked network, as indicated in equations (12) and (13):

$$e^k_t = v^T_e \tanh(We[ht\text{-}1; x^k]), \qquad 1 \leq k \leq n \quad (12)$$

and

$$\alpha_t^{\,k} = e^{-xt}\sqrt{n-1} \quad (13)$$

Neuron i in layer m produces signal depicted in equation (14):

$$I_i^{(m)} = f_{ReLu}(b^{(m,i)} + \sum_j I_j^{(m-1)} W_j^{(m,i)}) \quad (14)$$

Here, $f_{ReLu}(.)$ Rectified Linear Unit (ReLU) with the activation function $f_{ReLu}(a) = \max(0, a)$. Here, two distinct output function types are taken into consideration. The softmax function is a typical output function in classification problems with K classes, as demonstrated in equations (15) and (16):

$$f_i = \frac{\exp(I_i^{(o)})}{\sum_j \exp(I_j^{(o)})} \quad (15)$$

$$I_i^{(o)} = b^{(o,i)} + \sum_{k=1}^{k} W_k^{(o,i)} I_k^{(N)} \quad (16)$$

We additionally take into account the logistic output function variation represented by equation (17):

$$f = a + (b-a)(1+\exp(b^{(o)} + \sum_j W_j^{(o)} I_j^{(N)})^{-1} \quad (17)$$

This generates a continuous output f with parameters $b^{(o)} and W^{(o)}$ that must fall within the range (a, b). The decoding process's equation is presented as illustrated in equation (18):

$$\text{Decode stage: } d_t = f2(c_t, y_t, d_{t-1}) \quad (18)$$

The Long Short Term Memory unit is primarily used because it can circumvent issue of vanishing gradients as well as more effectively record long-term interdependence of time series. The required result is produced by feature representation in the previous time step y_{T+1}, as illustrated in equation (19):

$$y_{T+1} = F(y1, y2 \ldots\ldots yT, x1, x2, \ldots.. xT, z1, z2, \ldots.. zT) \quad (19)$$

A multi-level feature fusion perspective might be used to explain the use of cT in final prediction phase. All of the embedded data from the encoder module is included in cT because it is weight-sum of (h1, h2,...,hT). Gradient range is maintained by this skip connection in a manner similar to that of res-block or dense-block.

Algorithm

Input- sensory data (D={x1,yz,z1,z2,y2,z2,…..xn,yn,zn}

Output- scheduled data

Learning (L)= L1 , L2, . . . , Ln;

start training (t) method

ht = Lt(D)

D0 = φ;

for i = 1, . . . , m:

zit = hi (xi)

end;

For every view

class cj ←Hj(x)

if

Hj(x)$=wi * hi(x)$

Hj(x)$= wi(j) * hi(x)$

Hj(x)= Hj(x)

End if

Train learner$_{view}$ *on Liter*

Allocate class probabilities for every $u_i \in U^{iter}$

For every class

Detect top class=Iter+1

Update method

4. Performance Analysis

Experimental result is carried out in MATLAB-19 software, and parameters utilized for analysis are Packet Delivery Ratio (PDR), throughput, network lifetime, Prediction rate, MAE, RMSE,and RAE. These specifications are compared with three states of art techniques such as RNN based LSTM called RNN-LSTM, Q-Deep Extreme Learning Mechanism (Q-DELM) and Indoor Ranging Model utilizing DL (IRMDL) with the proposed recurrent encoder neural netwoks (REncNN).The operational parameter of the network is shown in Table-1.

Table 1. : Operational Parameters

Operational Parameters	Values
Epochs	900
Hidden layers	7
Batch Size	16
Learning Rate Drop	1e-2
Batch normalization epsilon	10^{-4}
Initial Learn Rate	1e-2
Learn Rate Drop Period	40

Packet Delivery Ratio (PDR) - It is average ratio of the total packets received (R)successfully to the total packets originally sent (S)as shown in equation (20):

$$\text{PDR} = \sum_0^N \frac{R}{S} \quad (20)$$

Table 2 shows the comparison of Packet Delivery Ratio (PDR)between existing RNN-LSTM Q-DELM, IRMDL and proposed REncNN techniques.

Table 2. Comparison of Packet Delivery Ratio (PDR)

Number of Epochs	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN [Proposed]
100	85	85.2	85.4	89.2
300	85.6	87	94.9	89.9
500	86.2	89.2	96.3	90.6
700	87.3	91.8	98	91.5
900	89	93	99	93

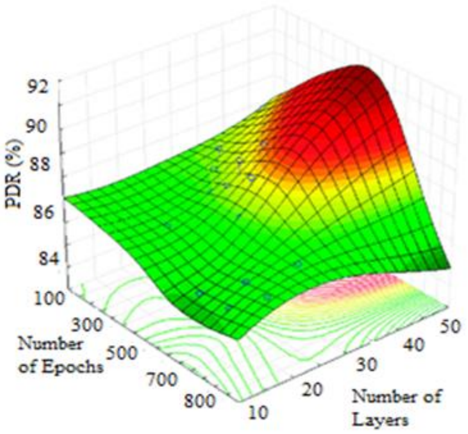

Fig. 4. Comparison of throughput

Fig. 5 gives comparison of throughputbetween existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis is number of epochs, Y-axis is number of layers utilized for analysis, and Z-axis is thethroughput in %. When compared, existing RNN-LSTM Q-DELM, IRMDLmethods achieve 80.82%,88.38% and 85.22%, while proposed REncNN method attains 95.34%, which is 15.52% better than RNN-LSTM,7.04% better than Q-DELM and 10.12% better than REncNN.

- **Network Lifetime (NL)-** Network lifespan is the maximum amount of time that all nodes in the network can last before one or more of them run out of energy. Equation (22) illustrates the formula as follows:

$$NL=\sum_{T=1}^{T} Etx(k,d) + \sum_{r=1}^{r} Erx(k) \quad (22)$$

Where,$Etx(k, d)$ is the transmitted energy betweennode k and d,$Erx(k)$ is the remaining energy at the destination side x(k)

Table 3 gives comparison of network lifetime between existing RNN-LSTM Q-DELM, IRMDL and proposed REncNN techniques.

Table 3. Comparison of Network Lifetime

Number of epochs	**RNN-LSTM[14]**	**Q-DELM[15]**	**IRMDL[16]**	**REncNN [proposed]**
100	62.1	63.2	64.2	76.7
300	62.5	63.3	65.6	80.5
500	64.6	64.5	66.7	81.8
700	65.8	67.8	68.9	84.6
900	67.5	68.4	69.1	86.4

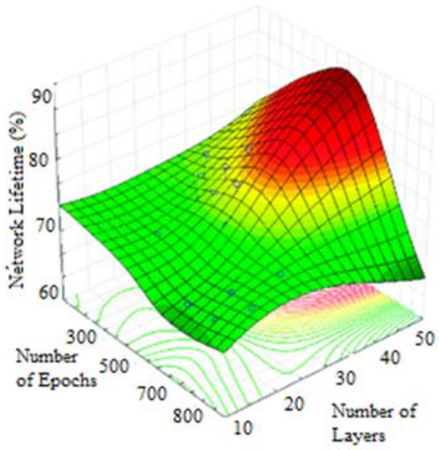

Fig. 5 Comparison of network lifetime between existing and proposed Technique

Fig. 5. Compares network lifetime between existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis is number of epochs, Y-axis indicates number of layers utilized for analysis, and Z-axis is network lifetime values in %. When compared, existing RNN-LSTM Q-DELM, IRMDLmethods achieve 63.3%,64.24% and 65.9%, while proposed REncNN method attains 81%, which is 19.1% better than RNN-LSTM,17.24% better than Q-DELM and 16.9% better than REncNN.

- **Prediction rate-**To produce a prediction rate, it is necessary to estimate the number of data and the residual standard deviation to provide a good estimate of the forecast standard deviation.

  Table 5 gives comparison of prediction rate between existing RNN-LSTM Q-DELM, IRMDL and proposed REncNN techniques.

Table 4. Comparison of Prediction Rate

Number of epochs	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN [proposed]
100	51.1	53.1	54.2	76.8
300	52.2	53.3	55.4	80.9
500	54.5	54.8	56.9	81.1
700	55.4	57.6	58.4	84.4
900	57.5	58.4	59.6	86.5

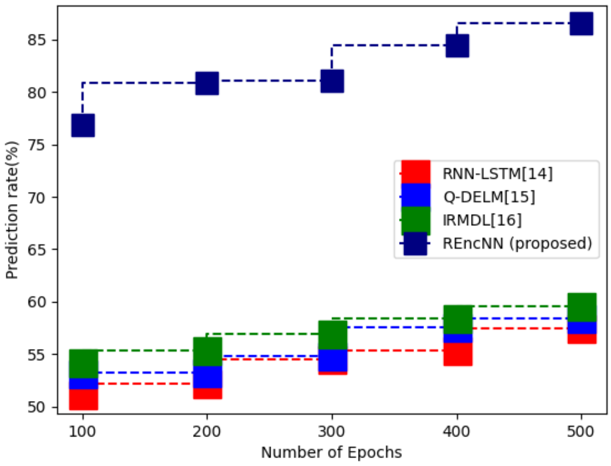

Fig. 6 Comparison of prediction rate

Fig. 6 gives comparison of prediction rate between existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis shows number of epochs. When compared, existing RNN-LSTM Q-DELM, IRMDLmethods achieve 53.14%,54.44% and 55.9%, while proposed REncNN method attains 80.94%, which is 27.54% better than RNN-LSTM,26.5% better than Q-DELM and 25.04% better than REncNN.

MAE- The error between two observations reflecting the same phenomenon is measured by this metric. Comparisons between projected and observed data, subsequent time and starting time. The formula is given in equation (23) as follows:

$$\text{MAE}= \sum_{i=1}^{n}(yi - xi) \qquad (23)$$

The new REncNN approach is compared to the existing RNN-LSTM Q-DELM, IRMDL, and existing RNN-LSTM methods in Table 6 in terms of MAE.

Table 5. Comparison of MAE

Number of epochs	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN[proposed]
100	66.4	44.2	43.1	41.1
300	70.9	45.3	43.3	42.2
500	71.1	46.9	44.8	44.1
700	74.4	48.4	47.4	45.7
900	76.5	49.6	48.4	47.5

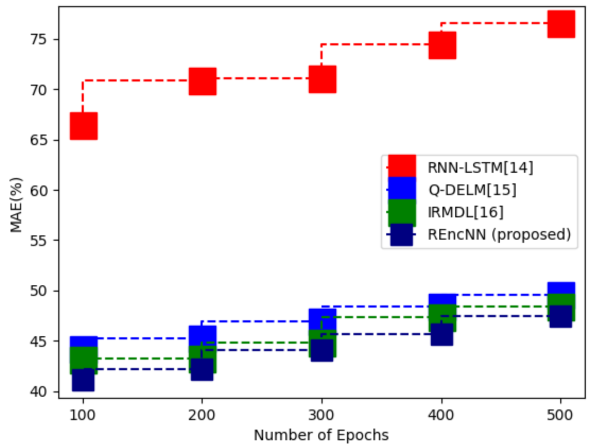

Fig. 7 Comparison of MAE

Fig. 7 shows comparison of MAE between existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis shows number of epochs, Y-axis is MAE values in %. When compared, existing RNN-LSTM Q-DELM, IRMDL technique achieve 70.86%,45.88% and 44.4%, while proposed REncNN method attains 43.12%, which is 27.24% better than RNN-LSTM,2.76% better than Q-DELM and 1.32% better than REncNN.

RMSE- It is a frequently employed statistic for contrasting values predicted by a model or evaluate with values actually observed. Equation (24) gives following formula:

$$\text{RMSE} = \sqrt{\sum_{t}^{T}(y'(t) - y(t)} \qquad (24)$$

Table 6 gives comparison of RMSE between existing RNN-LSTM Q-DELM, IRMDL methods and proposed REncNN techniques.

Table 6. Comparison of RMSE

Number of epochs	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN[proposed]
100	68.4	46.2	45.1	43.1
300	72.9	47.3	45.3	44.2
500	73.1	48.9	46.8	45.1
700	76.4	50.4	49.4	46.7
900	78.5	52.6	50.4	47.5

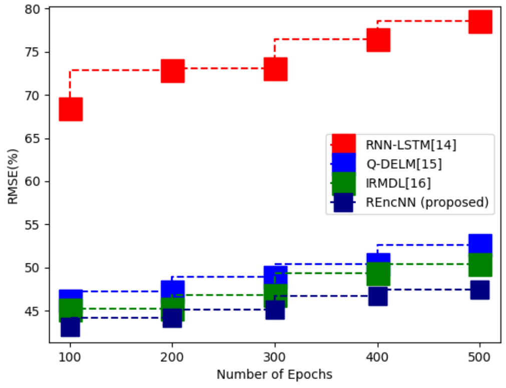

Fig. 8. Comparison of RMSE

Fig. 8 shows the comparison of RMSE between existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis is number of epochs, Y-axis is RMSE values obtained in %. When compared, existing RNN-LSTM Q-DELM, IRMDLmethods achieve 72.86%,47.88% and 46.4%, while proposed REncNN method achieves 44.32%, which is 28.54% better than RNN-LSTM,3.44% better than Q-DELM and 2.08% better than REncNN.

RAE- When comparing a mean error (also known as a residual) to errors produced by a simple or naive model, the relative absolute error is expressed as a ratio. The formula is given in equation (25) as follows:

$$\textbf{RAE} = \frac{\sum_{i=1}^{n}(pi-Ai)^2}{\sum_{i=1}^{n}Ai} \quad (25)$$

Table 8 gives comparison of RAE between existing RNN-LSTM Q-DELM, IRMDL techniques and proposed REncNN method.

Table 7. Comparison of RAE

Number of epochs	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN [proposed]
100	61.4	44.2	42.1	40.1
300	63.9	45.3	44.3	41.5
500	66.2	47.5	45.5	42.1
700	69.4	46.4	47.4	44.4
900	71.3	50.6	48.1	46.5

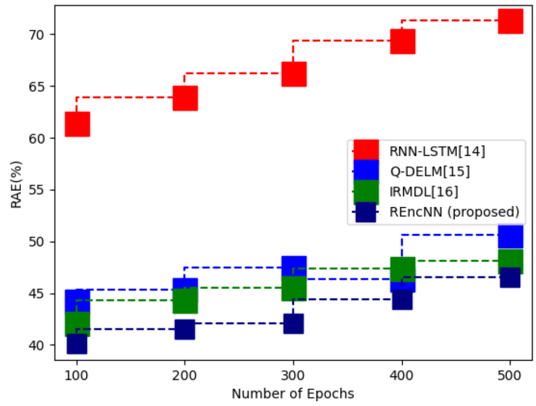

Fig. 9 Comparison of RAE

Fig. 9 gives comparison of RAE between existing RNN-LSTM Q-DELM, IRMDLmethods, and proposed REncNN technique where X-axis is number of epochs, Y-axis is RAE values obtained in percentage. When compared, existing RNN-LSTM Q-DELM, IRMDL methods achieve 65.44%,45.82% and 44.48%, while proposed REncNN method attains 41.92%, which is 28.54% better than RNN-LSTM,5.44% better than Q-DELM and 2.08% better than REncNN.

Table 8 shows the Overall comparison between existing RNN-LSTM Q-DELM, IRMDL and proposed REncNN techniques.

Table 8. Overall Cmparison between Proposed and Existing Techniques

Parameters	RNN-LSTM[14]	Q-DELM[15]	IRMDL[16]	REncNN [proposed]
PDR(%)	86.62	89.24	94.72	90.84
Throughput(%)	81.82	89.38	86.22	96.34
Network Lifetime(%)	64.3	65.24	66.9	82
Prediction Ration(%)	54.14	55.44	56.9	81.94
MAE(%)	71.86	46.88	45.4	44.12
RMSE(%)	73.86	48.88	47.4	45.32
RAE(%)	66.44	46.82	45.48	42.92

5. Conclusion

The wireless sensor network gathers sensory data from a variety of sensors based on nodes, which demonstrates the regional variations in a number of environmental parameters. In this study, we build a multi-feature method based on LSTM and a recurrent encoder NN (REncNN) to predict the various sensory inputs. First, to enhance quality of data, the Quantile Probability Distribution (QDP) and Dohono Threshold Denoising are employed. Then, Dual Encoding RNN is utilized to learn prediction features respectively. Finally, scheduling process is done to transfer the data.Parameter such as Packet Delivery Ratio (PDR), throughput, network lifetime, Prediction rate, RMSE, RAE and RAE by comparing with state-of-art methods such as RNN based LSTM called RNN-LSTM, Q-Deep Extreme Learning Mechanism (Q-DELM) and Indoor Ranging Model using Deep Learning (IRMDL). As a result, the proposed REncNN achieves89.84% of Packet Delivery Ratio, 95.34% of throughput, 81% of network lifetime, 80.94% of prediction rate, 43.12% of MAE, 44.32% of RMSE and 41.92% of RAE. The future work is to include a clustering process and secure routing protocol to reduce energy consumption in network and to improve the machine learning based predictive value analysis.

References

[1] Murugan, S., Jeyalaksshmi, S., Mahalakshmi, B., Suseendran, G., Jabeen, T. N., & Manikandan, R. (2020). Comparison of ACO and PSO algorithm using energy consumption and load balancing in emerging MANET and VANET infrastructure. *Journal of Critical Reviews*, *7*(9), 2020.

[2] Pandey, O.J.; Hegde, R.M. Low-latency and energy-balanced data transmission over cognitive small world WSN. IEEE Trans. Veh. Technol. 2018, 67, 7719–7733.

[3] Yan, R.; Sun, H.; Qian, Y. Energy-aware sensor node design with its application in wireless sensor networks. IEEE Trans. Instrum. Meas. 2013, 62, 1183–1191.

[4] Goel, S.; Passarella, A.; Imielinski, T. Using buddies to live longer in a boring world [sensor network protocol]. In Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOMW'06), Pisa, Italy, 13–17 March 2006; p. 5.

[5] Chanak, P., & Banerjee, I. (2020). Congestion free routing mechanism for IoT-enabled wireless sensor networks for smart healthcare applications. *IEEE Transactions on Consumer Electronics*, *66*(3), 223-232.

[6] Mohanty, S. N., Lydia, E. L., Elhoseny, M., Al Otaibi, M. M. G., & Shankar, K. (2020). Deep learning with LSTM based distributed data mining model for energy efficient wireless sensor networks. *Physical Communication*, *40*, 101097.

[7] Anitha, D., &Karthika, R. A. (2021). DEQLFER—A Deep Extreme Q-Learning Firefly Energy Efficient and high performance routing protocol for underwater communication. *Computer Communications*, *174*, 143-153.

[8] Guidara, A., Fersi, G., Jemaa, M. B., &Derbel, F. (2021). A new deep learning-based distance and position estimation model for range-based indoor localization systems. *Ad Hoc Networks*, *114*, 102445.

[9] Nasser, N., Khan, N., Karim, L., ElAttar, M., & Saleh, K. (2021). An efficient Time-sensitive data scheduling approach for Wireless Sensor Networks in smart cities. *Computer Communications*, *175*, 112-122.

[10] Yuan, Z., Hu, J., Wu, D., & Ban, X. (2020). A dual-attention recurrent neural network method for deep cone thickener underflow concentration prediction. *Sensors*, *20*(5), 1260.

[11] Zhang, Y.; Tian, Y.; Kong, Y.; Zhong, B.; Fu, Y. Residual Dense Network for Image Restoration. IEEE Trans. Pattern Anal. Mach. Intell. **2018**.

[12] Vijayan, V. P., Paul, B., & Chooralil, V. S. (2021). Energy Efficient Sustainable City Monitoring Using IoT Enabled Wireless Sensor Networks and Data Analytics. *Journal of Information Science & Engineering*, *37*(2).

[13] Balica, R. Ș. (2022). Machine and deep learning technologies, Wireless Sensor Networks, and virtual simulation algorithms in digital twin cities. *Geopolitics, History, and International Relations*, *14*(1), 59-74.

[14] Mohiddin, M., Kohli, R., Dutt, V. B. S., Dixit, P., & Michal, G. (2021). Energy-efficient enhancement for the prediction-based scheduling algorithm for the improvement of network lifetime in WSNs. *Wireless Communications and Mobile Computing*, *2021*.

[15] Nagarajan, S. M., Deverajan, G. G., Chatterjee, P., Alnumay, W., & Ghosh, U. (2021). Effective task scheduling algorithm with deep learning for Internet of Health Things (IoHT) in sustainable smart cities. *Sustainable Cities and Society*, *71*, 102945.

[16] Lăzăroiu, G., Andronie, M., Iatagan, M., Geamănu, M., Ștefănescu, R., & Dijmărescu, I. (2022). Deep Learning-Assisted Smart Process Planning, Robotic Wireless Sensor Networks, and Geospatial Big Data Management Algorithms in the Internet of Manufacturing Things. *ISPRS International Journal of Geo-Information*, *11*(5), 277.

[17] Dowlatshahi, M. B., Rafsanjani, M. K., & Gupta, B. B. (2021). An energy aware grouping memetic algorithm to schedule the sensing activity in WSNs-based IoT for smart cities. *Applied Soft Computing*, *108*, 107473.