Article

Deep Learning-Based Multiparametric Predictions for IoT

Muhammad Ateeq 1,2,*, Muhammad Khalil Afzal 2,*, Muhammad Naeem 3, Muhammad Shafiq 4,*, and Jin-Ghoo Choi 4,*

1 Department of Computer Science and IT, The Islamia University of Bahawalpur, Bahawalpur 63100, Pakistan; muhammad.ateeq@iub.edu.pk
2 Department of Computer Science, COMSATS University Islamabad Wah Campus, Wah Cantt 47040, Pakistan
3 Department of Electrical and Computer Engineering, COMSATS University Islamabad Wah Campus, Wah Cantt 47040, Pakistan; muhammadnaeem@gmail.com
4 Department of Information and Communication Engineering, Yeungnam University, Gyeongsan 38541, Korea
* Correspondence: khalilafzal@ciitwah.edu.pk (M.K.A.); shafiq@ynu.ac.kr (M.S.); jchoi@yu.ac.kr (J.-G.C.)

Received: 31 July 2020; Accepted: 16 September 2020; Published: 19 September 2020

Abstract: Wireless Sensor Networks (WSNs) and Internet of Things (IoT) often suffer from error-prone links when deployed in resource-constrained industrial environments. Reliability is a critical performance requirement of loss-sensitive applications, and Signal-to-Noise Ratio (SNR) is a key indicator of successful communications. In addition to the improvement of the physical layer through modulation and channel coding, machine learning offers adaptive solutions by configuring various communication parameters dynamically. In this paper, we apply a Deep Neural Network (DNN) to predict SNR and Packet Delivery Ratio (PDR). Analysis results based on a real dataset show that the DNN can predict SNR and PDR at the accuracy of up to 96% and 98%, respectively, even when trained with very small fraction (≤10%) of data. Moreover, a common subset of features turns out to be useful in predicting both SNR and PDR so as to encourage considering both metrics jointly. We may control the transmission power in the dynamic and adaptive manner when we have predictable SNR and PDR, and thus fulfill the reliability requirements with energy conservation. This can help in achieving sustainable design for the communication system.

Keywords: Internet of Things; wireless sensor networks; signal-to-noise ratio; packet delivery ratio; power conservation; deep neural networks

1. Introduction

IEEE 802.15.4 is one of the most popular communication standards for Wireless Sensor Networks (WSNs) and Internet of Things (IoT) [1]. Indeed, the standard accounts for more than 50% deployments of WSNs [2]. Prominent applications of IEEE 802.15.4-based WSNs and IoT cover smart factory, smart home, monitoring (area, health, etc.), sensing (earth, environment, etc.), and smart grid [3–5]. In these applications, Quality of Service (QoS) is the increasingly important and challenging requirement for sustainable system design [6–8]. Sustainability is the leading factor for societal development, and communication is a key application in this context [9]. There has been a growing number of studies in the field of WSNs and IoT that focus on sustainability in particular. IoT is conceived to be pivotal for achieving sustainable development in all walks of life [10]. Moreover, sustainability has been focused in WSN/IoT application domains like transportation [11], civil infrastructure [12], agriculture [13], sports [14], and water and air quality monitoring [15,16].
In traditional WSNs, the primary design factor was energy conservation. However, modern deployment scenarios of WSNs pose various conflicting requirements on the QoS, which yields multiple complicated optimization problems [17]. In this context, data-driven solutions such as machine learning are attracting huge attention from both academia and industry [18,19]. Reliability is an important requirement in diverse applications with the Packet Delivery Ratio (PDR) being an apt metric. It is well known that the strength of the received signal directly impacts the probability of successful transmissions, so Signal-to-Noise-Ratio (SNR) is critical for reliable communications [20]. There are numerous studies on PDR in relation to the SNR [20,21].

In the physical layer, extensive efforts have been carried out to improve SNR and reliability through innovative modulation and coding schemes, and they have achieved significant gains [22,23]. However, there remains a margin for fine-tuning the wider communication stack parameters from all layers, e.g., transmission power, packet size, etc. [24]. The availability of performance data and sophisticated machine learning algorithms encourage using these diverse parameters influential for signal strength and reliability to seek an adaptive design for the parameters. In IoT networks, it has been in focus to adopt intelligent techniques to improve QoS [25]. It has been demonstrated that we can predict PDR, energy consumption, and delay accurately by deep learning [26,27]. The previous literature motivates us to study the possibility of fine-tuning the various communication parameters based on the state-of-the-art learning algorithms. We summarize the contributions of this paper as follows.

- Adaptive transmission power control to ensure adequate SNR that maximizes PDR without wastage of energy.
- Analysis of correlations of diverse communication parameters with SNR and PDR.
- Adaptive system design that fine-tunes the configurations under evolving circumstances, considering performance constraints. This can complement the gains from physical layer modulation and coding schemes, and help achieve a sustainable design.
- Groundwork of a fully cognitive framework for adaptive QoS in WSNs and IoT.

The rest of this paper is organized as follows. We survey the related works in Section 2. Section 3 details the dataset used and describes the relationship between communication parameters and metrics of interest (i.e., SNR and PDR). We illustrate the adopted neural network model in Section 4 and present the prediction results in Section 5. Finally, Section 6 concludes the paper.

2. Related Works

Data-driven techniques are gaining popularity in predicting the QoS of wireless communications. We can find some of the QoS prediction methods in [18,19,26–28]. Tao [18] predicted the success probability of the next frame based on the received signal strength indicator, link quality indicator, SNR, and packet reception ratio. Ironically, the values recorded at the receiver-side are used as input to predict the transmission success at the sender-side. Kulin [28] used a neural network to predict packet loss ratio. The input features for the neural network includes inter-packet interval, number of nodes, and the number of received packets and erroneous packets. Both the number of received packets and erroneous packets are counted on the receiver-side, so predicting another receiver-side metric based on these quantities is not intuitive and beneficial. However, inter-packet arrival and number of nodes are useful in this context. Ayhan [19] used a neural network to predict transmission power level, lifetime, and inter-node distance from the data of preconfigured simulation scenarios. In order to predict any one of these metrics, the other two were used as input, resulting in a circularly dependent design.

The studies discussed show that it is not easy to predict the QoS exactly without depending on receiver-side values. However, in [26,27], the authors adopted deep learning to predict PDR, energy consumption, and delay from the diverse parameter configurations without any receiver-side values. We extend their effort here by predicting the SNR based on ~50,000 configurations with 7 key stack parameters and rich per-packet information.
3. Dataset and Analysis

We use a public IEEE 802.15.4 performance dataset, collected for almost 50 thousand configurations of 7 preconfigured stack parameters from different layers [29]. These preconfigured parameters include Inter-Arrival Time (IAT), Packet Size (PS), maximum Queue Size (QS), Maximum Transmissions (MT), Retry Delay (RD), Transmission Power (TP), and Distance (DT). For each configuration, 300 transmissions were made. The second set of parameters consists of information collected for each packet and includes Over-Flow (OF), Actual Queue Size (AQS), and Actual Transmission number (AT). OF indicates if there was a buffer overflow while attempting to transmit the current packet. AQS represents the current size of the queue, whereas AT indicates the actual transmission attempt count. These values were recorded for each of the 300 transmissions for all of 50 thousand configurations. Both sets of parameters provide sufficient variability in the resulting metric like SNR and are used as input features to the learning algorithm. We summarize these parameters in Table 1.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Parameter</th>
<th>Type</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application</td>
<td>Inter-Arrival Time: IAT (ms)</td>
<td>preconfigured</td>
<td>10,15,20,25,30,35,40,50</td>
</tr>
<tr>
<td></td>
<td>Packet Size: PS (bytes)</td>
<td>preconfigured</td>
<td>20, 35, 50, 65, 80, 95, 110</td>
</tr>
<tr>
<td>MAC</td>
<td>max Queue Size: QS</td>
<td>preconfigured</td>
<td>1,30,60</td>
</tr>
<tr>
<td></td>
<td>Actual Queue Size: AQS</td>
<td>per-packet</td>
<td>varies between 0–60</td>
</tr>
<tr>
<td></td>
<td>Over Flow: OF</td>
<td>per-packet</td>
<td>0, 1</td>
</tr>
<tr>
<td></td>
<td>Max Transmissions: MT</td>
<td>preconfigured</td>
<td>1, 3, 5</td>
</tr>
<tr>
<td></td>
<td>Actual Transmissions: AT</td>
<td>per-packet</td>
<td>varies between 0–5</td>
</tr>
<tr>
<td></td>
<td>Retry Delay: RD (ms)</td>
<td>preconfigured</td>
<td>30, 60</td>
</tr>
<tr>
<td>PHY</td>
<td>Transmission Power: TP</td>
<td>preconfigured</td>
<td>3,7,11,15,19,23,27,31</td>
</tr>
<tr>
<td></td>
<td>Distance: DT (m)</td>
<td>preconfigured</td>
<td>10, 15, 20, 25, 30, 35</td>
</tr>
</tbody>
</table>

Before predicting the metrics, we first analyze the relationship among various communication parameters (listed in Table 1) and the performance metrics, i.e., PDR and SNR. We use box-plots and scatterplots for the representation and discuss the analysis results in the following.

3.1. Packet Delivery Ratio (PDR)

Relationships of PDR with a single parameter and multiple parameters are shown in Figure 1. We use the box-plots to represent the relationship with each parameter and the 3-dimensional scatter-plots to represent the relationship with multiple parameters.

3.1.1. Effects of Individual Parameters

Figure 1a–e describes the relationship of PDR with IAT, TP, DT, AT, and AQS, respectively. The PDR tends to increase with the increase in IAT (Figure 1a,b), and AT (Figure 1d). The exception arises in the case of TP (Figure 1b) when the value of the TP level goes beyond 23, resulting in too much power for smooth communication. On the other hand, with the increase in the values of DT (Figure 1c) and AQS (Figure 1e), PDR tends to fall. As seen in Figure 1a–e, it suffices to conclude that the variations in the values of different communication parameters do have a relationship with the QoS metric under study. Therefore, it presents an interesting case for testing the potential of using these parameters for predictive QoS performance.

3.1.2. Effects of Multiple Parameters

Figure 1f,g highlights the joint effect of OF/AQS and OF/IAT on PDR, respectively. The relation of AQS/OF on PDR is shown in Figure 1f: PDR increases and OF decreases as AQS rises. A similar trend in PDR and OF is observed as IAT increases (Figure 1g). The overall trends observed in Figure 1f,g reaffirm and further strengthen the findings from Figure 1a–e.
Figure 1. Relationship between communication parameters and packet delivery ratio.
3.2. Signal-to-Noise Ratio (SNR)

Relationships of SNR with a single parameter and multiple parameters are shown in Figure 2. Similar to PDR, for the representation of relationships with each parameter, box-plots are used, whereas, the relationship with multiple parameters are shown by scatterplots.

![Signal-to-Noise Ratio (SNR)](image)

Figure 2. Relationship between communication parameters and signal-to-noise ratio.

3.2.1. Effects of Individual Parameters

Figure 2a–d describes the relationship of SNR with IAT, TP, DT, and AQS, respectively. The SNR tends to rise with the increase in IAT (Figure 2a) and TP (Figure 2b) values. The rise in IAT indicates lesser traffic and hence potentially improving channel conditions, whereas TP directly impacts SNR.
However, when the TP level is too high (23 and beyond), the SNR tends to deteriorate slightly. This phenomenon is observed for PDR as well. The SNR tends to fall with the increase in DT (Figure 2c) values as DT directly impacts SNR. Interestingly, SNR falls with an increase in AQS (Figure 2d) potentially due to more traffic volume caused by increase packet rate and heavy losses.

3.2.2. Effects of Multiple Parameters

Figure 2e,f highlight the joint effect of various combinations of TP, DT, and PS on SNR. Intuitively, SNR shows a direct proportion to TP and inverse to DT (Figure 2e). Similarly, the joint impact of TP and PS on SNR reflects that PS is also in an inverse proportion to SNR (Figure 2f). Overall, the findings from the effect of multiple parameters are affirmative of the effect observed for individual parameters.

Based on the analysis unleashed above, it is encouraging to test the potential of the variations in the communication parameters for predicting performance metrics (PDR and SNR in this case). In the following text, we describe the prediction model and discuss the results.

4. Prediction Model

Deep Neural Network

In order to capture the relationship between diverse parameters and target performance metrics, we adopt a Deep Neural Network (DNN) with 10 hidden layers, 64 neurons per layer, and a learning rate of 0.001. The activation function used at hidden layers is the rectified linear unit. The hyperparameter values are chosen empirically. We illustrate the overall flow of the prediction model in Figure 3. We aim to predict SNR and PDR based on about 50 thousand combinations of various preconfigured and per-packet input parameters. We can represent the data as the following input matrix,

\[
\begin{bmatrix}
    f_{1,1} & f_{1,2} & \ldots & f_{1,m} & \text{snr}_1 & \text{pdr}_1 \\
    f_{2,1} & f_{2,2} & \ldots & f_{2,m} & \text{snr}_2 & \text{pdr}_2 \\
    \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    f_{n,1} & f_{n,2} & \ldots & f_{n,m} & \text{snr}_n & \text{pdr}_n \\
\end{bmatrix},
\]

where \( f \) is a feature, \( \text{snr} \) and \( \text{pdr} \) are the prediction targets, and \( n \) and \( m \) are the number of rows and columns, respectively. The data are split into training set (50%), cross-validation set (20%), and testing set (30%).

![Figure 3. Prediction model for Packet Delivery Ratio (PDR) and Signal-to-Noise Ratio (SNR).](image-url)
Computation in a DNN is carried out using forward and backward propagations. During forward propagation at layer $j$, we multiply the input $I^{[j-1]}$ with the weight $W^{[j]}$ and add the regression bias $c^{[j]}$, which results in the output $O^{[j]}$. The activation function $g(\cdot)$ takes $O^{[j]}$ as the input and produces the next layer’s input $I^{[j]}$:

$$O^{[j]} = W^{[j]} \cdot I^{[j-1]} + c^{[j]},$$  \hspace{1cm} (2)

$$I^{[j]} = g^{[j]}(O^{[j]}),$$  \hspace{1cm} (3)

The next step known as backward propagation computes the rate of change in order to fine-tune the learning:

$$dO^{[j]} = dI^{[j]} \times g^{[j]}(O^{[j]}),$$  \hspace{1cm} (4)

$$dW^{[j]} = \frac{1}{n_t} (dO^{[j]} \cdot I^{[j]T}),$$  \hspace{1cm} (5)

$$dc^{[j]} = \frac{1}{n_t} \sum_{i=1}^{n_t} O^{[j]},$$  \hspace{1cm} (6)

$$dI^{[j-1]} = W^{[j]T} \cdot dO^{[j]},$$  \hspace{1cm} (7)

where $g^{[j]}(\cdot)$ represents the derivative of activation function $g(\cdot)$, and $n_t$ is the number of training examples. In order to minimize the prediction error, the weights and regression bias are updated iteratively through subtracting their respective derivatives using gradient descent:

$$W^{[j]} = W^{[j]} - \alpha \times dW^{[j]},$$  \hspace{1cm} (8)

$$c^{[j]} = c^{[j]} - \alpha \times dc^{[j]}.$$  \hspace{1cm} (9)

We used Root Mean Square Error (RMSE), Mean Bias Error (MBE), Mean Absolute Error (MAE), and Normalized MAE (NMAE) for the evaluation of training models. The term MBE instead of ME is used since it indicates the bias caused by cancellation of positive and negative errors. The prediction error is calculated using

$$Err_i = X_{i\text{actual}} - X_{i\text{predicted}},$$  \hspace{1cm} (10)

where $i$ represents the index of current example, and $X_{i\text{actual}}$ and $X_{i\text{predicted}}$ represent the actual and predicted values for the $i$th sample in training data, respectively. This $Err_i$ is used to compute the RMSE, MBE, MAE, and NMAE as follows.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n_t} (Err_i)^2}{n_t}},$$  \hspace{1cm} (11)

$$MBE = \frac{\sum_{i=1}^{n_t} Err_i}{n_t},$$  \hspace{1cm} (12)

$$MAE = \frac{\sum_{i=1}^{n_t} |Err_i|}{n_t},$$  \hspace{1cm} (13)

$$NMAE = \frac{\sum_{i=1}^{n_t} |Err_i|}{n_t \times (X_{\text{max}} - X_{\text{min}})},$$  \hspace{1cm} (14)

where $n_t$ represents the total number of samples in the test data. The RMSE computes the sum of the squares of differences of prediction errors on all data samples divided by the total number of samples. The square root is used to obtain the normalized value representative of the range of predicted output. MBE represents the average of prediction errors for all samples, MAE calculates the average
of absolute values of prediction errors for all samples, whereas NMAE normalizes the MAE over the range of values.

5. Analysis Results

In this section, we first present the prediction error for each individual feature and their combinations complemented with a correlation matrix to highlight the correlations among features and prediction targets (Figure 4d). The correlation values represented are calculated based on Pearson correlations. We then analyze the effect of dimensionality (fraction of data used for training the DNN model and the number of epochs) on prediction error. The section is concluded with a statistical description of the relationship of the TP with PDR and SNR.

![Image of Figure 4](image-url)

**Figure 4.** Prediction results: (a) Individual features results for SNR; (b) Features combination results for PDR; (c) Features combination results for SNR; (d) Correlation matrix showing relationships among features and Actual as well as Predicted values of PDR and SNR.

5.1. Impact of Features

RMSE, MAE, and MBE for SNR against each of the 10 features are shown in Figure 4a. RMSE is higher than MAE, thus indicating the variance of the frequency distribution of the error magnitudes. A value close to 0 for MBE indicates a relatively even spread of positive and negative errors. Prediction results for PDR against individual features have already been presented in [26]. DNN was trained for the different combinations of features to predict both metrics. The best prediction results were achieved against a combination of five features for both PDR and SNR. In case of PDR, these features included OF, AQS, AT, TP, and DT. Figure 2b shows the prediction results in terms of RMSE, MAE, MBE, and NMAE for PDR against OF, AQS, AT, TP, and DT. In case of SNR, the five
features that yield the best prediction result include OF, AQS, IAT, TP, and DT. The RMSE, MAE, MBE, and NMAE are plotted in Figure 2c for SNR against OF, AQS, IAT, TP, and DT. The NMAE is 0.0235 and 0.0395 for PDR and SNR, respectively which indicates good prediction accuracy. It appears that four out of five features for predicting PDR and SNR are the same. The correlation of these best features with the actual and predicted values of both PDR and SNR are shown as a correlation matrix in Figure 4d. It is apparent that OF, DT, TP, and AQS have a strong and similar correlation with Actual and Predicted values of both PDR (APDR and PPDR) and SNR (ASNR and PSNR). Correlations between AT and SNR as well as between IAT and PDR were not calculated as these features (i.e., AT and IAT) were not used in the prediction of these (i.e., SNR and PDR) quantities, respectively. In addition, the correlation is evident among actual and predicted values, and between PDR and SNR. This reaffirms the role of the features in prediction accuracies.

5.2. Impact of Dimensionality

There are two aspects that define dimensionality in our context: One is the fraction of data used for training the DNN, and the other is the number of iterations (epochs) used. Figure 5 shows the effect of dimensionality on training progress. Specifically, DNN is trained with 1%, 3%, 5%, 10%, 20%, and 30% data to predict PDR (Figure 5a) and SNR (Figure 5b). The maximum number of epochs is set to 2000. It is apparent that learning improves with increasing the fraction of training data and the number of epochs. However, there are interesting and important observations made. It is apparent from Figure 5b that even when 1% data is used to train the DNN model, the RMSE decreases to a value under 3.0. When 10% or more data is used for training DNN, the RMSE either does not improve after 500 epochs or the improvement is statistically insignificant. The minimum NMAE achieved for 1, 3, 5, 10, 20, and 30% fraction of training data is 0.070, 0.058, 0.052, 0.048, 0.045, and 0.044, respectively. Thus, the accuracy in terms of NMAE when the DNN model is trained with only 1% data is 93%. Although these results are from one specific dataset and cannot be broadly generalized, it is still encouraging that a very small fraction of data can yield an accuracy which is practical.

5.3. Adaptive Transmission Power Control

Predictable SNR and PDR present an interesting case for adaptive TP control. The 10th, 30th, 50th, and 70th percentiles for the actual and predicted values of both PDR and SNR are presented in Table 2 for all TP levels. It is evident that a TP level of 15 achieves almost maximum PDR. Although both metrics remain stable for TP levels 17 and 23, beyond 23 PDR tends to drop a little. The same relationship between PDR, SNR, and TP is shown in Figure 6. It can be concluded from both Table 2 and Figure 6 that TP can be dynamically adjusted in accordance with the target values for PDR and SNR minimizing energy consumption and maximizing reliability.
Table 2. Effect of different Transmission Power (TP) levels on actual as well as predicted values of PDR and SNR.

<table>
<thead>
<tr>
<th>TP</th>
<th>PDR 10th Percentile</th>
<th>PDR 30th Percentile</th>
<th>PDR 50th Percentile</th>
<th>PDR 70th Percentile</th>
<th>SNR 10th Percentile</th>
<th>SNR 30th Percentile</th>
<th>SNR 50th Percentile</th>
<th>SNR 70th Percentile</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Actual</td>
<td>Predicted</td>
<td>Actual</td>
<td>Predicted</td>
<td>Actual</td>
<td>Predicted</td>
<td>Actual</td>
<td>Predicted</td>
</tr>
<tr>
<td>3</td>
<td>0.00</td>
<td>0.06</td>
<td>0.29</td>
<td>0.34</td>
<td>0.59</td>
<td>0.57</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>7</td>
<td>0.23</td>
<td>0.32</td>
<td>0.78</td>
<td>0.77</td>
<td>0.96</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>11</td>
<td>0.65</td>
<td>0.62</td>
<td>0.93</td>
<td>0.91</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>15</td>
<td>0.70</td>
<td>0.69</td>
<td>0.97</td>
<td>0.97</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>19</td>
<td>0.70</td>
<td>0.70</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>23</td>
<td>0.70</td>
<td>0.71</td>
<td>0.98</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>27</td>
<td>0.65</td>
<td>0.66</td>
<td>0.93</td>
<td>0.91</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>31</td>
<td>0.61</td>
<td>0.64</td>
<td>0.90</td>
<td>0.88</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Figure 6. Relationship of TP with predicted PDR and SNR.
6. Conclusions and Future Work

In this work, we predict the SNR and PDR of IEEE 802.15.2-based WSNs by deep learning, in order to improve reliability with energy conservation through dynamic transmission power control. We train the deep neural network using the data on different stack parameters. We report the prediction results for each individual parameter and for the best combination of parameters as well. The correlations strengthen the proposition that deep learning can play a critical role in implementing the adaptive wireless system. We train the deep neural network by different fractions of data and various numbers of epochs. Our analysis reveals that the prediction results are accurate even when the models are trained for a very small fraction (≤10%) of data. We also present the statistical demonstration on the potential of dynamic transmission power control. The overall results encourage implementing a prototype of this system considering aspects like model training frequency, performance data collection, management, and dissemination. It further motivates us to study the effect of wider parameter configurations on the performance of WSNs and IoT networks in diverse deployment scenarios. Furthermore, the current study serves as a foundation work towards the design of a cognitive framework for adaptive QoS management in WSNs and IoT.

Author Contributions: M.A. proposed the main idea of research, performed the experimentation and wrote the manuscript. M.K.A. and M.N. guided with the design of experiment, provided critical feedback, and helped in shaping the research, presentation of the results and the manuscript. M.S. and J.-G.C. organized and refined the manuscript.

Funding: This research was supported by the Basic Science Research Program through the National Research Foundation (NRF) of Korea funded by the Ministry of Education under Grant 2018R1D1A1B07048948, and in part by the National Research Program for Universities, Higher Education Commission, Pakistan No.: 7522/Balochistan/NRPU/R&D/HEC/2017.

Conflicts of Interest: The authors declare no conflicts of interest.

References


