Enhancing Wireless Data Transmission: A GAN-based Approach for Time Series Data Restoration

Daejin Han
Dept. of Computer Science and Engineering
Kongju National University
Cheonan, Korea
djhan8733@gmail.com

Woongsoo Na
Dept. of Software
Kongju National University
Cheonan, Korea
wsna@kongju.ac.kr

***Abstract*—Recent technological advancements have led to a significant increase in the use of wireless communication for data transmission, particularly in sensor networks and device-to-device communications, necessitating not only rapid but also reliable data exchange. In conventional wireless communication, if a packet is lost in transmission due to retransmission, it is retransmitted. In this case, the subsequent packets will wait for them so that the order of the packets is not reversed, which is called Head of Line (HOL) Blocking. This problem can lead to overall transmission delays and losses, and is one of the reasons for inconsistent data transfer times. In this paper, we propose a technique to significantly reduce the retransmission rate of the sender and data loss by retransmission using Generative Adversarial Net (GAN). To evaluate the performance of this study, we applied our technique to a portion of SolarCube solar data assuming that it was corrupted, such as NaN processing, value change, and sign change, and the average recovery rate was 92.36%, demonstrating that it is possible to detect and recover losses in the data transmission process. This research is expected to have applications in data loss detection and recovery during transmission of time series data.**

***Index Terms*—Wireless Communication, Data Transmission, Time Series Data, Data Preprocessing, Data Restoration**

I. INTRODUCTION

Recent advances in computing technology have increased the speed of wireless communication and improved the performance of sensors that collect data, resulting in an increasing number of services using wireless networks. However, compared to wired communication, the speed of data transmission in wireless communication is not consistent due to various external factors or physical factors such as obstacles and weather conditions, and this leads to possible data loss. As a result, various techniques are proposed to address the problems caused by time lags across data [1].

In order to minimize the time delay in the transmission process, this paper proposes a technique to reduce the retransmission rate by detecting the loss in the transmission process of data and restoring the lost data using GAN, which performs inspection and restoration based on training data. [2].

II. RELATED WORK

GAN is a learning method using a Generator Model [3] (G Model) and a Discriminator Model (D Model). This uses the concept of "competition" in the GAN framework, where the D Model learns to determine whether the input sample data is the sample data generated by the G Model or the actual training data distribution. At this time, the G Model creates an authentic model to deceive the D Model based on the training data, and conversely, the D Model judges whether the input data is true or false to improve its ability through adversarial learning [4].

GANs can determine whether input data is correct based on fake data and training data, and can be used for restoration to recover missing and corrupted data [5]. GAN's restoration technology also allows for adversarial learning of the G and D models, where the G model restores the corrupted data to resemble the actual data and the D model determines if the G model is restored to the point where it resembles the original.

III. DATASET AND GAN MODELING

A. DataSet for Training GAN

The purpose of this paper is to apply the collected solar data to loss restoration in wireless communication, and to take advantage of the advantages of GANs specialized in loss restoration, we preprocess the data set of this study, which is a two-dimensional time series data [6].

Table 1 describes the Solar Cube solar data used as training data for this study. The data is a time series and was recorded in hourly increments from October 2015 to March 2023. Before using the data for the study, we preprocessed the data to ensure that there are no outliers or unnecessary data in the measured data. Table.1 describes the parameters of the training data used in this study, and Fig.1 is a graph of solar data representing the measured values of those parameters over time.

When checking the distribution of the solar data in Fig.1, we can see a significant decrease in DC voltage (V) in some

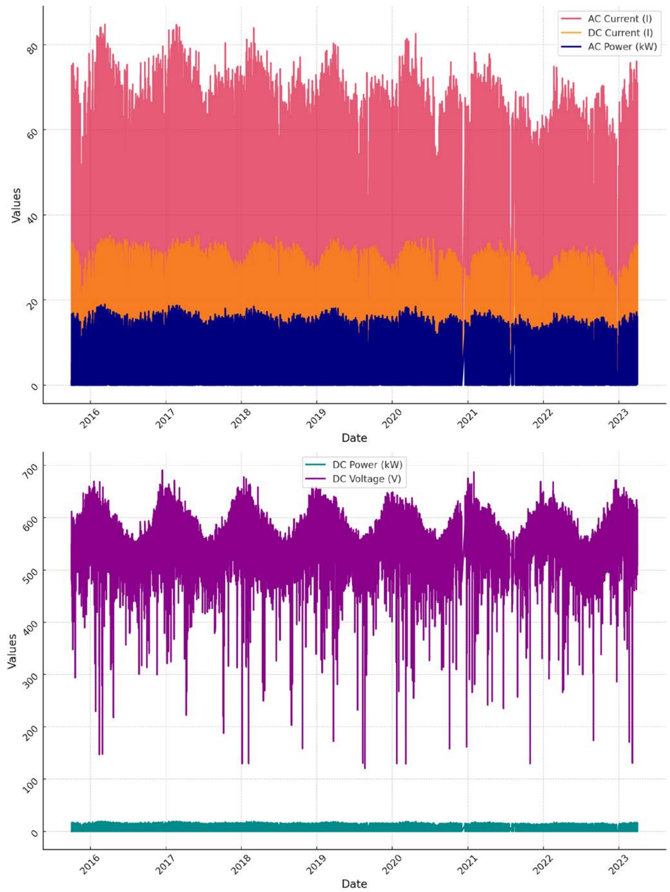


Fig. 1. Solar Power Generation Dataset

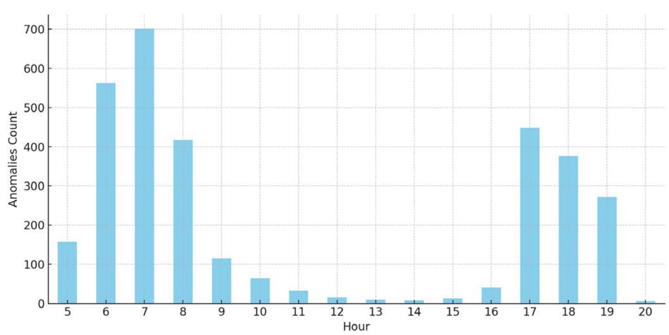


Fig. 2. DC Voltage(V) distribution by time zone

parts, and we first checked whether it is an outlier or a natural value. Fig.2 is a graph showing the time of day when values of DC Voltage (V) deviate significantly from the average value. When we checked this, we found that this was largely the case at 6∼8 and 17∼19. It can be seen that the solar power generation time of the solar cube, which is the source of data for this study, ranges from 5∼19 o'clock depending on the season, with less power generation at the beginning and end of the day and a large variation from day to day, resulting in a momentary decrease in the value of DC Voltage (V) at that time. Therefore, we believe that the dataset is a natural

TABLE I
DESCRIPTION OF SOLAR POWER GENERATION DATA PARAMETERS

Column	Description	Units
Date	Time and date of data recording	Date/Time
AC Voltage	Voltage output of the solar panels	V
AC Current	Current output of the solar panels	A
DC Power	Power received by the inverter	kW
DC Voltage	Voltage received by the inverter	V
DC Current	Current received by the inverter	A

phenomenon and used it as a training dataset.

B. GAN Modeling

This section describes the modeling of a GAN that will learn the data before experimenting with techniques that allow the sender to recover from losses without retransmissions, making it similar to the original data. The training data used in this study is a two-dimensional time series graph, allowing us to apply the benefits of GANs from Section 2.

First, we need to normalize the data, and since the scale difference between the measurements in the prepared data is large and the percentage of outliers is low, we used the Min-Max normalization method. This is a feature scaling method that changes the feature value between 0 and 1, which reduces the overfitting phenomenon due to the size of the value and makes the model perform better.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (1)$$

The key to modeling is to use GANs to generate time series data. The GAN model consists of two main components: a Generator and a Discriminator. This was chosen after conducting the following experiments to build up the model layers and use the appropriate activation function as the two elements progress. Fig.3 shows the change in loss per epoch after applying LeakyReLU, ReLU, sigmoid, tanh, and ELU to this learning model. study [7], [8].

$$sigmoid(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

$$tanh(x) = 2\sigma(2x) - 1 \quad (3)$$

$$ReLU(x) = max(0, x) \quad (4)$$

$$LeakyReLU(x) = max(0.01x, x) \quad (5)$$

$$ELU(x) = \begin{cases} x & \text{if } x > 0, \\ \alpha(e^x - 1) & \text{if } x \leq 0. \end{cases} \quad (6)$$

The key principle of GANs lies in the competitive nature of its architecture, where the fake data produced by the generator fools the discriminator well and conversely the discriminator learns to make better judgments, is ideal and can determine that a model is not suitable if it becomes too strong early on in training or if its performance evaluation shows large deviations. Therefore, among the activation functions tested, we can see that LeakyReLU and ReLU outcompete each other longer than other activation functions and learn the model well,

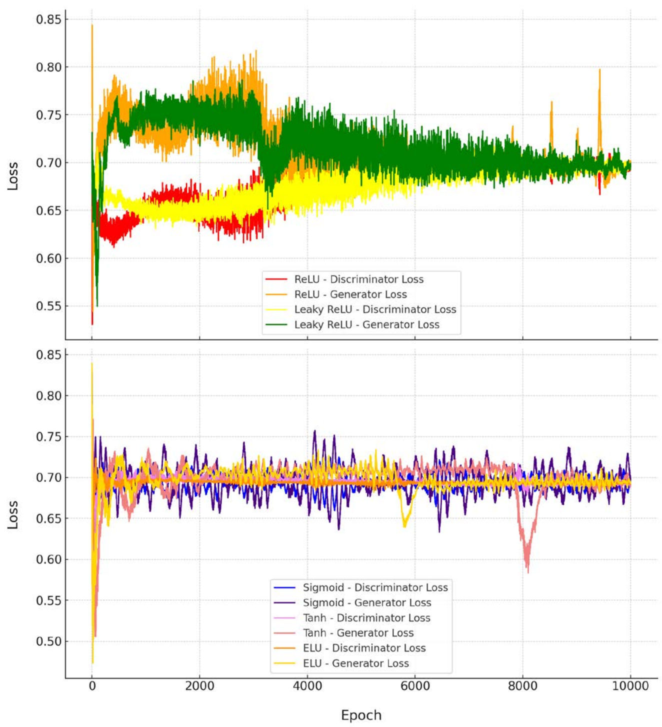


Fig. 3. Performance Comparision of Activation Fuction

and furthermore, LeakyReLU does not show significant partial loss instability compared to ReLU, and we finally selected LeakyReLU for stable learning.

- Generator takes a noise vector in latent space as input and generates time series data that mimics the solar data used as training data. The model consists of two Dense layers, and the activation function used is the LeakyReLU function. The first Dense layer takes the noise vector from the latent space as input, maps it to a higher dimensional feature space, and applies a nonlinear transformation using LeakyReLU. The output layer then uses the tanh function to reconstruct the output in the form of time series data.
- The Discriminator is composed of two layers, the Flatten layer and the Dense layer. We start by converting the time series data to a one-dimensional vector in the Flatten layer to make it easier for the Discriminator to process. The Dense layer is then used to determine whether the input time series data is real or fake. The model was also run with LeakyReLU, and in the end, the output layer uses a sigmoid function to determine that the closer it is to 1, the more authentic the input data is.

IV. Performance Evaluation

Based on the modeling in Section 3 of this thesis, the model was trained using the train set and then evaluated using the test set. For the evaluation analysis of the model, the following losses were assumed for restoration due to data loss in wireless communication, which is an application situation.

- ”Data Missing” means that the data was lost in transit, and we assume that a portion of the test set is NaN, meaning that some of the columns at that time are missing. We did this for 30% of the test set.
- ”Data Corrupted” assumes that no existing data is missing due to data loss, but that various tampering has occurred, such as broken values or positive values having negative values. We did this for 30% of the test set.
- ”Data Inconsistent” is an inconsistency in the data, which assumes that if the data sent was an integer, it was also an integer when it was received, but with a different value, or that there was an outlier that didn’t fit at that time. We did this for 40% of the test set. This is a relatively common type of data loss, and we’ve weighted it heavily toward the more common types.

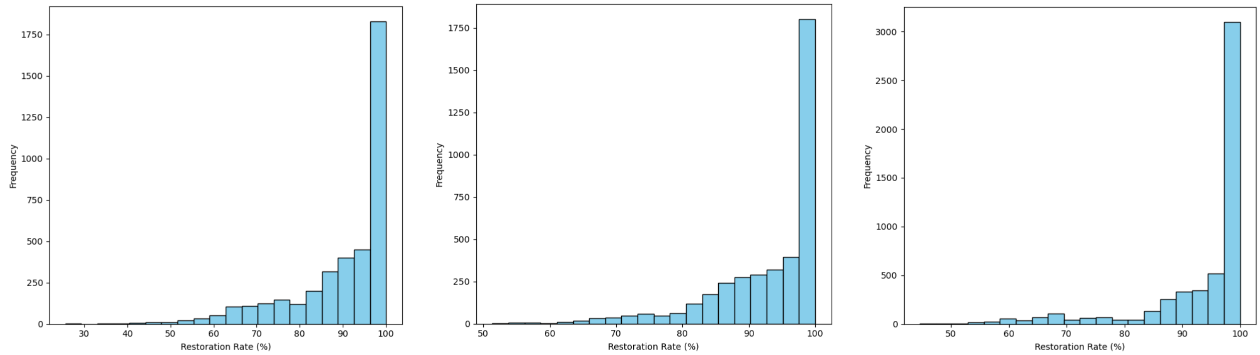
We conducted experimental evaluation of restoration for Data Missing, Data Corrupted, and Data Inconsistent under the above loss assumptions. Fig.5 shows the breakdown of each loss type for Mean Squared Error (MSE) and Mean Absolute Error (MAE), the numerical evaluation metrics selected for the experimental evaluation.

First, MSE is the squared average of the difference between the predicted and actual values, which is used in this study to measure how close the overall reconstructed data is to the original data and to assess the sensitivity to large errors that may occur during the reconstruction process. The MSE measurements resulted in 0.09903, 0.04752, and 0.061595, respectively, and were analyzed as follows. Among the three types of data loss, Data Missing was the highest when analyzed by MSE, indicating that NaN data was more difficult to restore than other types. This indicates that the error in restoration was relatively large. On the other hand, Data Corrupted was the lowest, indicating that the sign of the value changed during data tampering, and the error in restoration was relatively low, which is a significant result compared to other loss types.

MAE is an index that averages the absolute difference between the reconstructed value and the true value and was used to calculate the true reconstruction rate and to evaluate the technique proposed in this study. The MAE measurements were 0.08038, 0.052488, and 0.048361, respectively, and the analysis results are as follows. In terms of MAE measurement results, the overall result is relatively high for the Data Missing part, such as MSE, but the MAE of Data Inconsistent is similar to Data Corrupted, so you can see that the evaluation of the restoration is good overall when the actual restoration is performed due to the characteristics of MAE that are relatively less affected by errors, and you can see that the error rate of some data is large.

Fig.4 is a graph showing the distribution of restoration rates after restoring each type of loss, where the x-axis is the restoration rate (%) and the y-axis is the number of restored data corresponding to that restoration rate.

$$RestorationRate = 1 - MAE \tag{7}$$



(a) Data Missing (b) Data Corrupted (c) Data Inconsistent

Fig. 4. Data Restoration Rates by Type

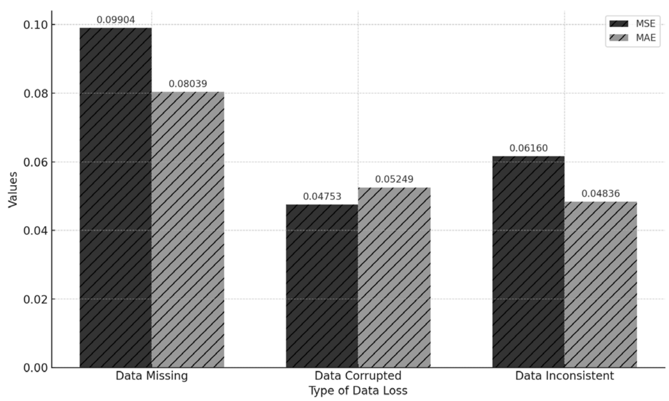


Fig. 5. MSE and MAE Results by Data Loss Type

In our analysis, we found that Data Missing sometimes has a recovery rate as low as 30%, with an average performance of 70 to 90%. Data Corrupted shows a minimum of 50% recovery rate and overall better performance. Data Inconsistent has the best restoration rate of the data loss types assumed in this study, with a distribution that stays around 90% on average.

The average restoration rates are 90.23%, 93.12%, and 93.73%, respectively. As the GAN-based modeling in this study shows a restoration rate of more than 90% for each type of data loss, we can confirm that the two-dimensional time series data, which is the solar data used in this study, is applicable to the restoration of GANs, and we can show the applicability of how to reduce the retransmission rate of the transmitter based on the detection of lost data in the data transmission process and the significant restoration results.

V. Conclusion

In this study, we proposed a recovery method for time series data loss in wireless communication by applying GAN. The proposed method showed an average restoration rate of 92.36% for the missing, tampered, and lost scenarios of solar data, and demonstrated its applicability to loss detection and recovery of real-world situation. The implication of this study is that GAN has a high performance guarantee and applicability for the application of time series data restoration, and it shows excellent performance in determining whether the received data is lost or not. However, the restoration rate and deviation from it are different depending on the type of loss, so it is possible to improve the performance based on the difference in GAN modeling suitable for each, or to apply it to various loss situations in real life, although virtual simulation of loss was conducted in this study, so higher performance improvement and applicability can be expected.

Acknowledgment

This work was supported by the Priority Research Centers Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2019R1A6A1A03032988) and in part by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2023- RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation.

References

[1] E Ryu, M Ahn, J Park, J Lim, D Seong, and J Yoo. A congestion control scheme for minimizing data-loss in wireless multimedia sensor networks. *Journal of the KIISE: Computing Practices and Letters*, 19(5):278–282, 2013.

[2] Wenzhong Li, Han Zhang, Shaohua Gao, Chaojing Xue, Xiaoliang Wang, and Sanglu Lu. Smartcc: A reinforcement learning approach for multipath tcp congestion control in heterogeneous networks. *IEEE Journal on Selected Areas in Communications*, 37(11):2621–2633, 2019.

[3] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A Bharath. Generative adversarial networks: An overview. *IEEE signal processing magazine*, 35(1):53–65, 2018.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

[5] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2536–2544, 2016.

[6] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, and See-Kiong Ng. Mad-gan: Multivariate anomaly detection for time series data with generative adversarial networks. In *International conference on artificial neural networks*, pages 703–716. Springer, 2019.

[7] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. Activation functions in neural networks. *Towards Data Sci*, 6(12):310–316, 2017.
[8] Chigozie Nwankpa, Winifred Ijomah, Anthony Gachagan, and Stephen Marshall. Activation functions: Comparison of trends in practice and research for deep learning. *arXiv preprint arXiv:1811.03378*, 2018.