Enabling a GAN-Based Model to Produce Strong Long-Range Forecasts

A technical paper prepared for presentation at SCTE TechExpo24

Wei Cai
Senior Lead Data Scientist
Cox Communications, Inc.
wei.cai@cox.com

Mohsin Afridi
Lead Netowrk Engineer
Cox Cmmunications, Inc
mohsin.afridi@cox.com

Table of Contents

Title **Page Number**

List of Figures

Title **Page Number**

List of Tables

Title **Page Number**

1. Introduction

The question of generating accurate forecasts in a long memory (or long range) process has attracted much attention with telecom traffic data as it is crucial to formulate capacity planning and budget allocation in a cost-effective manner.

However, there has been a growing awareness of a variety of difficulties to implement long-range forecasting using telecom time series data. Firstly, insufficiency of telecom traffic data has posed challenges to effectively execute sophisticated statistical models and machine learning (ML) models [1]. Furthermore, irregular patterns in network time series data made conventional outlier detection method difficult to detect, which might introduce noise in the forecast, and hence greatly affect forecast accuracy. Lastly, the predictions made using the state-of-the-art statistical models or highly supervised machine learning models tend to experience error propagation and lose accuracy as the prediction time horizon expands. It is less likely for those models to correctly extrapolate the special characteristics of network time series data, particularly when small-scale historical data is presented as the learning dataset.

In order to address the issues highlighted earlier, literature has introduced a relatively recent approach that involves using a Generative Adversarial Network (GAN) architecture to generate a soft representation for both the short- and long-term dependencies in the time series. The GAN architecture was initially proposed by Goodfellow et al [2]. Originally, GANs were primarily designed for processing picture data. Since their introduction, substantial progress has been achieved in expanding their capabilities, and they are extensively employed in various tasks such as text generation, audio signal generation, spectral data generation, tabular data generation, and time series data generation [3][4][5].

Nonetheless, as far as we are aware, GAN has been focused less on temporal time series data. Consequently, there has not been much research done on how to use GAN to improve long-range forecasting.

In this paper, we evaluate the performance of GAN in time series forecasting and propose a hybrid forecasting strategy of incorporating GAN into a long horizon forecasting process using telecom traffic data.

The rest of this paper is organized as follows. In Section 2, we examine previous studies that used GAN algorithms and provide an introduction on one conventional deep learning model RNN as well as two most popular GAN algorithms: Wasserstein GAN-GRU and Wasserstein GAN-GP. In Section 3, after presenting sample data used in this paper, we proceed to compare the performance of GAN with RNN. We outline the process of utilizing GAN to generate synthetic time series and explore the feasibility of employing GAN to long-range prediction. Finally, Section 4 contains concluding remarks, and identifying specific areas for further research.

2. Related Work

Although the complete spectrum of scenarios employing GANs to forecast time series data is still being studied, numerous studies have explored the potential of utilizing GANs to overcome the scarcity of data during model training and improve model performance. Patil et al. [5] employed attention mechanisms and principles of Conditional Generative Adversarial Networks (CGAN) and successfully tackled the issue of having a limited and well-documented dataset of chest X-ray (CXR) images. They used these techniques to create synthetic images that closely mimic real medical images. Researchers concluded that deep learning models, trained on an augmented dataset, outperformed other models, especially in the context of having a small size training data.

Similar work has been done on breast ultrasound images. Lennart et al. [3] applied GAN to generate high quality realistic breast ultrasound synthetic images to address limited training data. The study revealed that GANs can effectively generate synthetic ultrasound images that are both high quality and exhibit a wide range of variations close to real images. This, in turn, enhances the classification accuracy of Convolutional Neural Networks (CNNs) and thus offers a valuable advantage in computer-aided diagnostics.

On the other hand, certain studies have effectively implemented the GAN framework inside a temporal context. The initial implementation of the GAN framework on sequential data, known as C-RNN-GAN, utilized Long Short-Term Memory (LSTM) networks for both the generator and discriminator components. Data is generated at regular intervals by using a noise vector and the data generated from the preceding time step as inputs [4]. Moreover, scholars have suggested using GAN-based methods to produce various forms of time-series data to improve data quality and optimize the performance of forecasting models. Liu et al. (2022) presented a new forecasting approach called Generative Forecasting (GenF), which utilizes a GAN to produce synthetic data for future time periods. The synthetic data, along with the actual data, are subsequently utilized to provide projections over longer time horizons. In the conducted studies, researchers reported a substantial improvement in prediction accuracy [6].

While the usefulness of GAN in time series forecasting has been shown, there is a lack of examples illustrating its applicability in network data, particularly in predicting the volume demands of network traffic. The study undertaken by Naveed et al. (2022) is one of the few studies that have employed GANs to analyze network data. A comparative analysis was performed on two generative models, TimeGAN and DoppelGANAger as well as a deep learning auto-regressive model called PAR. The comparison was done using real mobile network datasets. Based on their research, they observed that GAN-generated values were not only effective in substituting missing data in a time-series data, but they also discovered that GAN-based structures performed better than the auto-regressive technique.

While there is less research on the effectiveness of GANs in long-range forecasting, our objective in this study is to propose a hybrid framework for long-range forecasting that incorporates GAN's synthetic data to potentially enhance the accuracy of long-term predictions. Specifically, we evaluate the similarity between real data and synthetic data predicted by a GAN. We compare GAN models with a RNN model for time series forecasting. The goal is to determine the appropriate architectural designs for making long-range forecasts using GAN-based synthetic data.

3. Learning Methods

This section presents an overview of various models for forecasting temporal time series, along with a detailed discussion of the specific methodologies employed in this study.

3.1. Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN), a member of the neural network family, is well-suited to capture long-term dependencies of time series, and widely used framework in the fields of time series forecasting. A classical RNN is constructed by a sequence of an input layer, hidden layer and an output layer. The connectivity between different layers allows the model to learn patterns and trends in time series data. Many to many RNN architecture was employed in this study, as shown in Fig. 1, which takes a sequence of time series inputs ending at time=t, and produces a sequence of outputs starting at time=t+1.

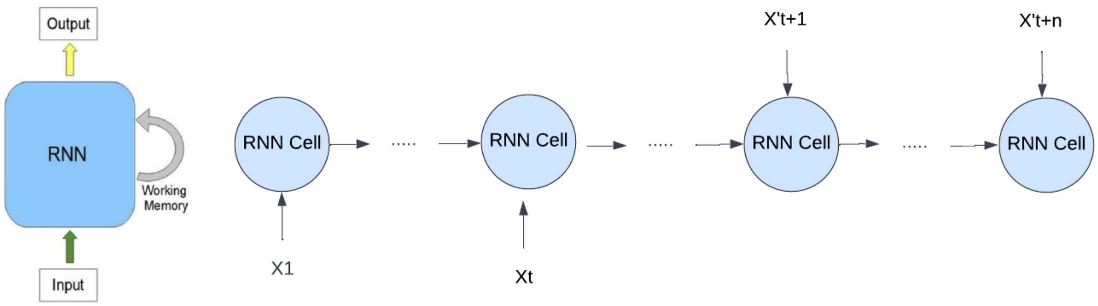

Figure 1– Simple Many to Many RNN Structure

3.2. GAN-based Methods

3.2.1. Basic GAN

The GAN algorithm was first introduced in 2013 [10]. The core of the GAN is composed of two multilayer CNNs or fully connected neural networks, referred to as the generator (*G*) and discriminator (*D*), which act as two competing agents. The *G* network tries to model a noise vector z to fit the probability distribution of the real data and create fake data, whereas the *D* strives to distinguish the synthetic data and the real data. In a well-trained GAN, the training process concludes and the model reaches convergence when the *G* generates synthetic instances to a degree where the *D* will find it difficult to differentiate between data from the synthetic dataset and the original dataset. In other words, the two networks are engaged in a two-player min-max game where they strive to reach a point called Nash equilibrium, where the *D* cannot distinguish between the real data and the generated data anymore.

Mathematically, the process of a standard GAN algorithm is expressed in (1) as:

$$\min(G) \max V\,(D, G) = E_{x\,\sim Pdata(x)}[\log(D(x)] + E_{z\,\sim P_{z\,(z)})}[1 - \log(G(z)] \qquad (1)$$

In this equation, x represents the input data, log(D(x)) is the projected output of the discriminator for x_i, whereas log(D(G(z)) is the output of the discriminator for the data generated by the GAN, denoted as G(z). The objective of the equation is to maximize the discriminator network to correctly identify generated data from real data.

A simple GAN architecture to produce high-quality image data can be illustrated, as shown in Fig.2.

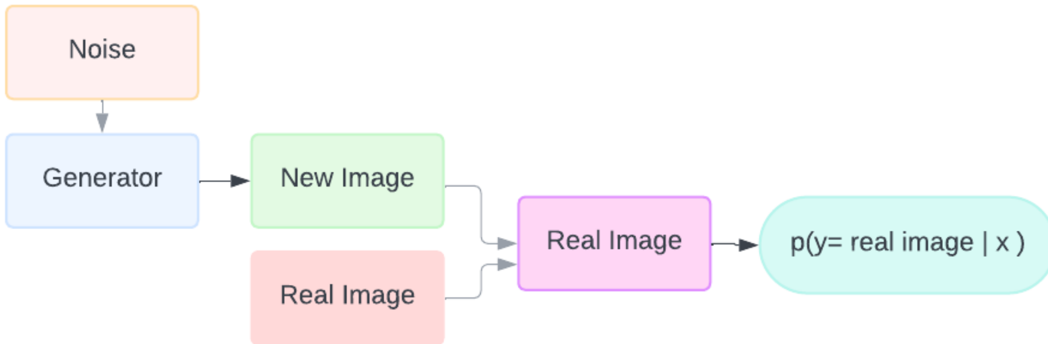

Figure 2 – Simple GAN Structure

Although the concept of the GAN is exciting, and promises many applications, such as producing visual or video content. Several studies have shown that one of the biggest problems with GANs is that they are hard to train and can have problems like overfitting, mode-collapsing (where they only pick samples from one class in the data), and training instability due to vanishing gradient issues.

To get around these challenges and ensure stable GAN training, other types of GANs have been developed, including the Conditional Wasserstein GAN (CWGAN) and the Wasserstein GAN (WGAN). In the subsequent parts, we will demonstrate the functioning of two algorithms in WGAN: WGAN-GP [7] and WGAN-GRU, which proved efficient in many complex GAN applications.

3.2.2. Wasserstein GAN with Gradient Penalty (WGAN-GP)

The WGAN was first proposed in [8], and it distinguishes itself from the standard GAN in several aspects. Unlike the standard GAN, WGAN and its variations use the Wasserstein distance to minimize the loss of discriminator function, which provides a smoother gradient everywhere. Furthermore, WGAN generator parameters are updated after training the discriminator multiple times, rather than updating them after every discriminator update as in a standard GAN, which is related to the stability and convergence of the training process. Finally, without the use of sigmoid activation in the final layer of the WGAN discriminator, the output of the WGAN discriminator spans between negative infinity ($-\infty$) and positive infinity (∞), instead of the typical range of 0 and 1. The deviation from 1 has the potential drawback of introducing instability during the training process.

To address this issue, the gradient penalty was introduced into the WGAN's discriminator to penalize the discriminator if the gradient norm deviates from 1 [9]. The inclusion of this penalty term improved the quality of the general samples by the GAN and it improved model training stability. The WGAN-GP proved superior to the traditional WGAN by introducing the gradient penalty to penalize the discriminator if the gradient norm deviates from 1.

3.2.3. Wasserstein GAN with Gated Recurrent Unit (WGAN-GRU)

Another improved variant of WGAN is WGAN-GRU, which combines the strengths of both WGAN and GRU. The gated recurrent unit (GRU) is a gated recurrent neural network (RNN), which is characterized by a small number of parameters and a relatively simpler training process. GRU-based WGAN models can achieve better learning outcomes from sequential data than other RNNs while using a WGAN-based network to distinguish between real and generated samples.

4. Methodologies

This section begins by presenting a summary of the datasets utilized in this study. Subsequently, we compare the performance of GAN-based models with RNN on two sets of datasets used in this study. Next, we propose GAN-models for long-range forecasting and provide a comprehensive explanation of how we have put this suggested paradigm into implementation.

4.1. Datasets

For implementing machine learning or deep learning-based models, it is desirable to use actual data from a real network. For a time series traffic forecasting task using neural network-based models, a dataset that consists of daily traffic load that spans from 2018 to Jun 2024 would be appropriate for model training and testing in relation to data size.

To judge how well our selected generation methods work for network time series data of different behaviors, two datasets with different patterns in seasonality and trend: D1 and D2, were used in this study.

Figures 3 (a) and (b) illustrate the D1 and D2, respectively. D1 exhibits a more irregular pattern, particularly during the COVID-19 period, whereas D2 demonstrates a more stationary pattern.

These two datasets were employed to evaluate the performance of four models: RNN, GAN, WGAN-GP and WGAN-GRU. For each dataset, a small amount (i.e., less than 5%) of missing values are imputed using adjacent non-missing values. After replacing missing values with linear interpolation for each time series, we adjust and rescale values of target variables to [0,1] for data normalization. The data preparation also includes dividng the collected time series datasets into two sets: training and testing, where the testing dataset is applied to find the optimal parameters, and the optimal time series length.

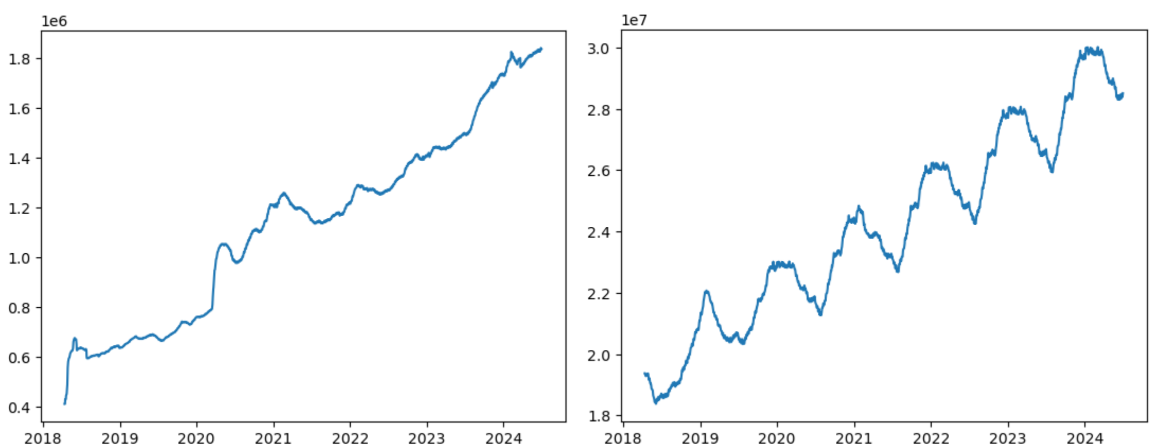

Figure 3 - These two datasets were employed to evaluate the performance of four models: RNN, GAN, WGAN-GP

4.2. Comparison Criteria

Root Mean Squared Error: To evaluate the accuracy of the methods on D1 and D2, the Root Mean Squared Error (RMSE) is selected. RMSE is a frequently used measure of the differences between forecast values and actual values. RMSE is considered as a proper measure of accuracy, see (2)

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{((predicted\ value\ (i) - actual\ value\ (i))^2}{n}} \tag{2}$$

Mean Absolute Percentage Error (MAPE) is a measure of accuracy of a method for constructing fitted time series values in trend estimation. It usually expresses accuracy as a percentage, and is defined in (3)

$$\text{MAPE} = \frac{1}{n}\sum_{i=1}^{n} \left| \frac{predicted\ value\ (i) - actual\ value\ (i)}{n} \right| * 100 \tag{3}$$

Where n denotes the number of data points in the sequence.

4.3. WGAN Model Setting

To find the most appropriate GAN model for the proposed long-range forecasting, different GAN-based methods have been investigated with the focus on comparing WGAN models (i.e., WGAN-GP and WGARN-GRU) with a basic GAN.

In WGAN models used in this study, CNN is used in the network structure of the generator and discriminator of WGAN-GRU. CNN is one of the best DL models for its ability to handle time series. In contrast, WGAN-GP in this study uses a straightforward feedforward architecture for the generator, without the recurrent structure of the GRU layers.

The main difference between WGAN-GP and WGAN-GRU is that in WGAN-GP, the generator typically uses a convolutional neural network, whereas WGAN-GRU is the use of a RNN, specifically the GRU (Gated Recurrent Unit) in the generator. In this study, the WGAN-GRU uses a generator that consists of three stacked GRU layers, which allows the WGAN-GRU to capture and model the sequential data.

Before training DL models, some hyper-parameters need to be fixed, like optimizer, learning rate, batch size, and number of epochs. These hyper-parameters could impact the model performance and learning speed.

To assess the prediction for time series, we used the following training hyper-parameters of WGAN models, as summarized in Table 1.

Table 1 - Sample Hyper-parameters in WGANs

Parameters	WGAN-GP	WGAN-GRU
Batch size	128	128
Learning rate	0.000115	0.000164
Num of epochs	300	300
Critic iterations	5	
Weight clip	0.01	

4.4. Results

Tables 2 and 3 showcase the superior performance of WGAN-GRU compared to three other models in predicting network traffic demand across datasets with varying characteristics.

On the D1 dataset, the Mean Absolute Percentage Errors (MAPEs) for the WGAN-GRU model are 0.36% on the training dataset and 0.47% on the testing dataset. The RNN model has MAPEs of 1.04% on the training data and 0.67% on the testing data. The WGAN-GP model, on the other hand, has MAPEs of 4.33% on the training data and 1.33% on the testing data. The basic GAN exhibited the poorest performance with MAPEs of 3.05% on the training data and 11.10% on the testing data. GAN evidently shows overfitting issues inherent in its architecture.

Similar observations could be obtained with the D2 dataset. On the D2 dataset, the MAPEs for the WGAN-GRU model are 0.36% on the training dataset and 0.47% on the testing dataset, which means that this model performs extremely well on a more stationary time series dataset. The RNN model has MAPEs of 1.38% on the training data and 0.52% on the testing data, which consistently performs well. The WGAN-GP model, on the other hand, has MAPEs of 1.93% on the training data and 1.33% on the testing data. The basic GAN with the same tuning parameters (i.e., learning rate and epoch etc.) failed to converge. Consequently, we can conclude that it is not appropriate to employ the basic GAN for forecasting long-term network time series.

Nevertheless, it is intriguing to see that the performance of WGAN-GRU is quite similar to that of RNN. The potential of WGAN-GRU to address common GAN issues such as overfitting, as well as its applicability in generating longer and more intricate time series, is evident. However, we have to admit the performance of WGAN-GRU is not as stable as RNN. This is particularly relevant as D1 exhibits more intricate patterns compared to D2. To select the appropriate model, it is crucial to consider that time-series data can exhibit various trends and patterns. Therefore, any reliable generative models must demonstrate consistent performance in order to be valuable. Therefore, it can be inferred that WGAN-GRU could be the preferable choice over other GANs for generating synesthetic data. For the advantage of model convergency and training stability, RNN or LSTM can be the recommended base model for long-range forecasting model on the original data augmented with GAN-based synthetic data.

Table 2 - D1 Model Performance Comparison

	RNN	GAN	WGAN-GP	WGAN-GRU
Training RMSE	11,691	32,983	48,300	16,898
Testing RMSE	12,877	211,648	36,211	69,940
Training MAPE	1.04%	3.05%	4.33%	1.70%
Testing MAPE	0.67%	11.10%	1.76%	3.36%

Table 3 - D2 Model Performance Comparison

	RNN	GAN	WGAN-GP	WGAN-GRU
Training RMSE	149,646	Failed to converge	500,097	104,948
Testing RMSE	179,632	Failed to converge	429,172	158,431
Training MAPE	1.38%	Failed to converge	1.93%	0.36%
Testing RMSE	0.52%	Failed to converge	1.33%	0.47%

5. Conclusion

In this paper, a GAN-based long-range time series forecast model has been developed with a focus on using WGAN models to augment network time series and further improve long-range forecasting. The model incorporates WGAN algorithms to generate synthetic data during the data processing stage, that would be coupled with real time series to augment data scale.

In conclusion, based on the analysis and model performance comparison, we propose a hybrid long-range forecasting model that integrates the RNN with GAN-based models for improved prediction of network traffic, as illustrated in Fig. 3.

The proposed method consists of two phases. The first phase is related to the WGAN to generate synthetic time series data and couple them with the original data, while the second phase is related to using more classical RNN or a special kind of recurrent neural network LSTM to train and forecast on the augmented dataset. The core of the model is to implement a WGAN-GRU or WGAN-GP to generate synthetic data that discriminators cannot easily distinguish from real data. In this proposed framework, the WGAN-GRU or WGAN-GP component plays a vital role in improving the training process by creating synthetic data and augmenting the limited training dataset to better handle long-range forecasting and improve long-horizontal forecast accuracy. The main process of the model can be summarized as shown in Fig. 4:

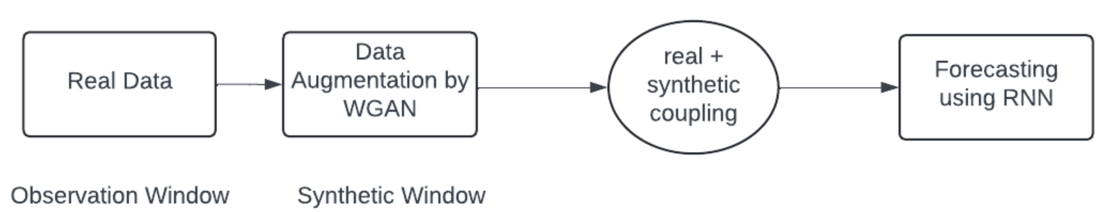

Figure 4 – Proposed Long-range Forecasting Method: RNN-WGAN

After synthetic data are coupled with the original data, the model utilizes RNN or LSTM to effectively leverage the sequential nature of time series data and capture temporal patterns that may be crucial for a long-term prediction. It is reasonable to assume that by coupling the RNN model with the GAN model, better results can be expected for long memory time series forecasting.

Abbreviations

CNN	Convolutional Neural Network
DL	Deep Learning
GAN	Generative Adversarial Network
GRU	Gated Recurrent Unit network
LSTM	Long Short-Term Memory Network
MAPE	Mean Absolute Percentage Error
PAR	Predictive Auto-Regressive
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
WGAN-GP	Wasserstein GAN with Gradient Penalty
WGAN-GRU	Wasserstein GAN with Gated Recurrent Unit

Bibliography & References

[1] G. Barlacchi, M. De Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, A multi-source dataset of urban life in the city of Milan and the Province of Trentino, Scientific Data, vol. 2, no. 1, pp. 1–15, Dec. 2015.

[2] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. (2014) Generative adversarial networks, Available from: http://arxiv.org/abs/1406.2661. Cited 2023 Feb 27

[3] Lennart et al. GANs for generation of synthetic ultrasound images from small datasets, Current Directions in Biomedical Engineering, 2023, 8(1):17-20.

[4] Yoon J, Jarrett D, van der Schaar M. Time-series generative adversarial networks.

In: Wallach H, Larochelle H, Beygelzimer A, Alché-Buc F d', Fox E, Garnett R, editors. Advances in Neural Information Processing Systems Curran Associates, Inc.. 2019. Available from: https://proceedings.neurips.cc/paper/2019/fle/c9efe5f26cd17ba6216bbe2a7d26d490-Paper.pdf

[5] Maack, Lennart ; Holstein, Lennart ; Schlaefer, Alexander, GANs for generation of synthetic ultrasound images from small datasets Current directions in biomedical engineering, 2022-07, Vol.8 (1), p.17-20.

[6] Patil, Prakash ; Deokate, Sarika T ; Bhoite, Amol ; Prusty, Sashikanta ; Patil, Abhijeet Bholaji ; Mange, Purva, GAN-Enhanced Medical Image Synthesis: Augmenting CXR Data for Disease Diagnosis and Improving Deep Learning Performance, Journal of Electrical Systems, 2023, Vol.19 (3), p.53-61

[7] Liu, Shiyu ; Ghosh, Rohan ; Motani, Mehul, Towards Better Long-range Time Series Forecasting using Generative Forecasting, arXiv.org, 2022-12

[8] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, Improved training of Wasserstein GANs, in Proc. Adv. Neural Inf. Process. Syst., vol. 30, 2017, pp. 1–11.

[9] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in Proc. Int. Conf. Mach. Learn., 2017, pp. 214–223.

[10] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved Wasserstein GANs, Aquat. Procedia. (2017), https://doi.org/10.1016/j.aqpro.2013.07.003.

List of Figures

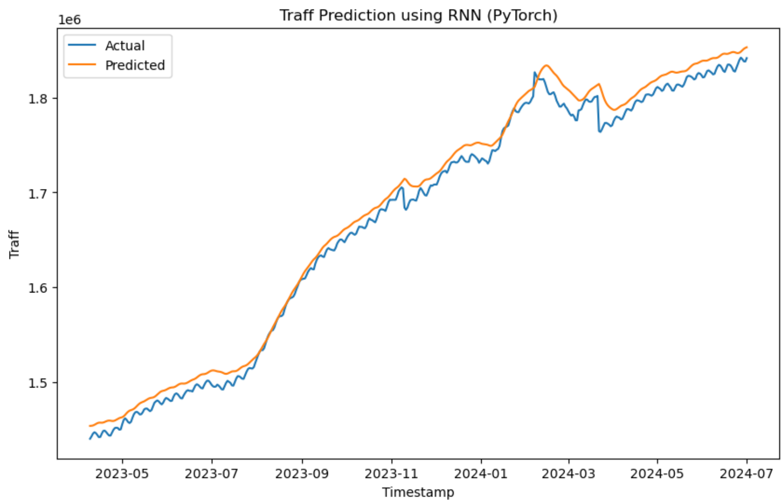

Figure 5 – D1 RNN Actual vs Predicted

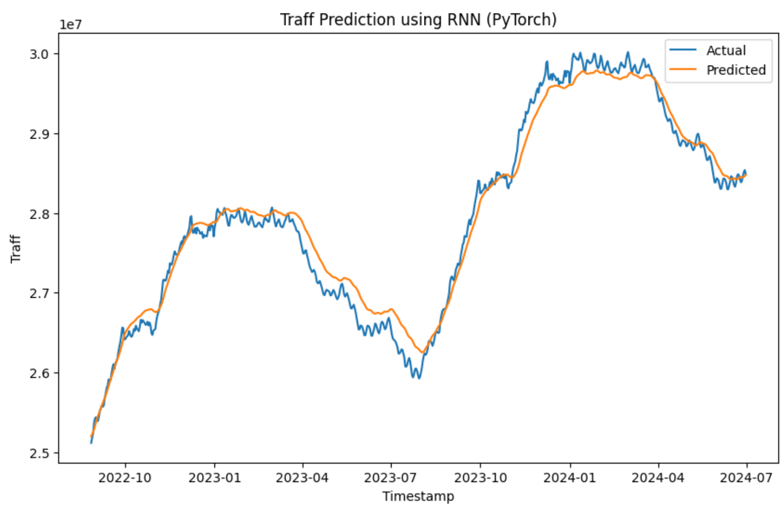

Figure 6– D2 RNN Actual vs Predicted

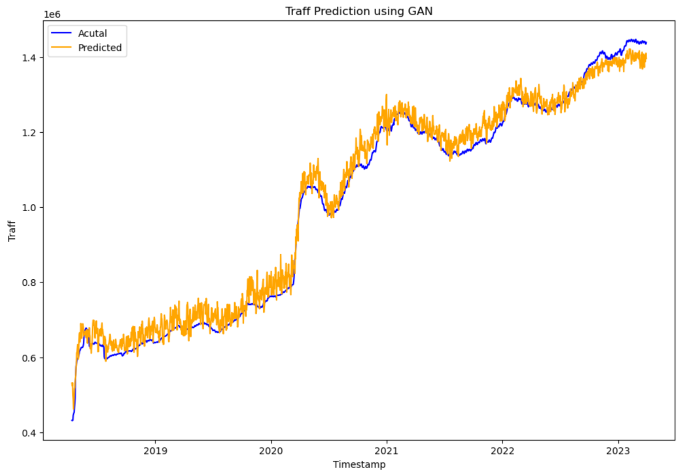

Figure 7 – D1 GAN Actual vs Predicted

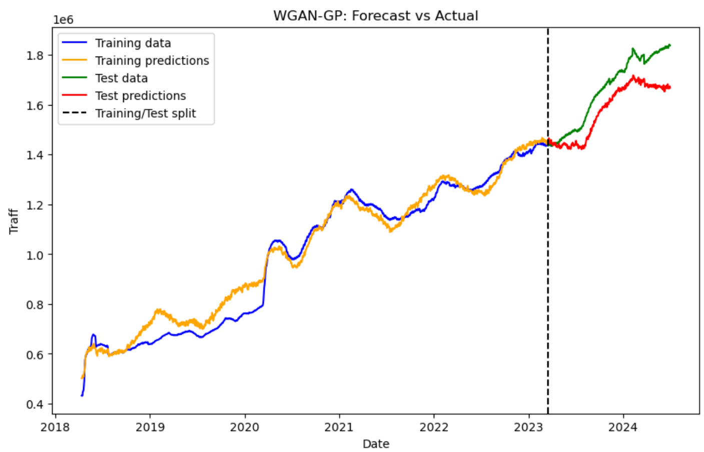

Figure 8 – D1 WGAN-GP Actual vs Predicted

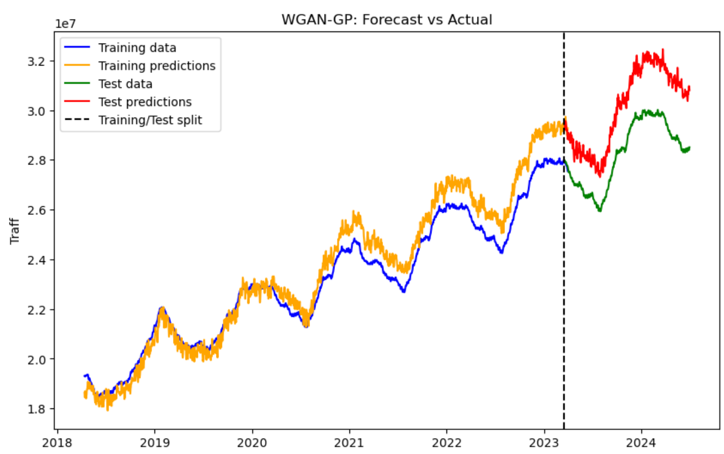

Figure 9 – D2 WGAN-GP Actual vs Predicted

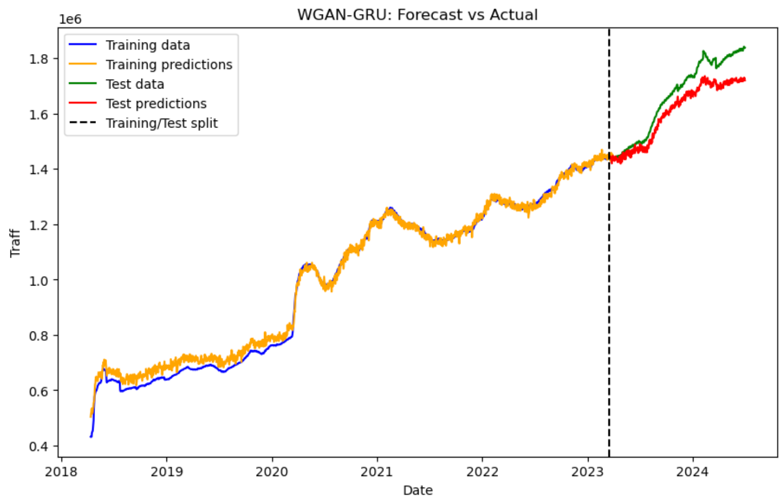

Figure 10 – D1 WGAN-GRU Actual vs Predicted

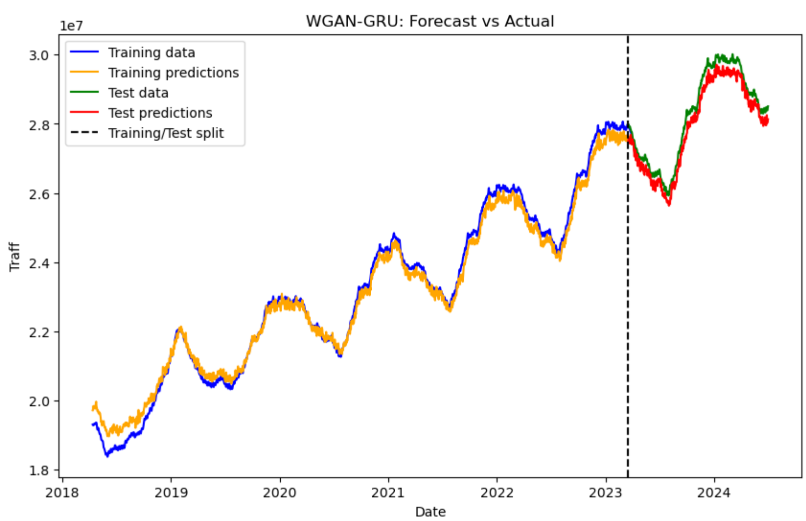

Figure 11 – D2 WGAN-GRU Actual vs Predicted