

Synthetic Data Generation of Surgical Drills Using Physics-Constrained GAN

Tessa E. Livingston, Joshua A. Blaney, Suresh S. Muknahallipatna

Department of Electrical Engineering and Computer Science, University of Wyoming, Laramie, WY, USA

Email: tliving7@uwyo.edu, jblaney2527@gmail.com, sureshm@uwyo.edu

How to cite this paper: Livingston, T.E., Blaney, J.A. and Muknahallipatna, S.S. (2026) Synthetic Data Generation of Surgical Drills Using Physics-Constrained GAN. *Journal of Intelligent Learning Systems and Applications*, 18, 235-254.
<https://doi.org/10.4236/jilsa.2026.183015>

Received: March 28, 2026

Accepted: July 13, 2026

Published: July 16, 2026

Copyright © 2026 by author(s) and Scientific Research Publishing Inc.
This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).
<http://creativecommons.org/licenses/by/4.0/>



Open Access

Abstract

Class imbalance in time series data occurs when some classes have far fewer training samples than others. Training a neural network classifier on such an imbalanced dataset results in bias and poor performance. In this paper, we focus on the class imbalance issue in a surgical bone-drilling dataset, where the drill traverses certain bone regions faster than others, leading to far fewer recorded samples in those regions. We propose a physics-constrained Generative Adversarial Network (P-GAN) that generates synthetic time-series data for underrepresented classes. Unlike standard generative models, the proposed P-GAN enforces physics constraints during the generation of synthetic samples. These constraints include ensuring depth and force values are within sensor-valid ranges and that depth sequences progress monotonically. Furthermore, these constraints enforce that the synthetic samples behave like real drill measurements. We evaluate the proposed method on the drill dataset by adding 150 synthetic sequences per underrepresented class and measure the effect on the model's performance. Results on a drill dataset show consistent improvements in F1 scores for both GRU and CNN-based classifiers, suggesting that physical constraints are essential for generating useful synthetic data on small, domain-specific datasets.

Keywords

Synthetic Data Generation, Time Series, Data Augmentation, Imbalanced Data, Generative Adversarial Networks, Medical Data Analysis, TS Classification

1. Introduction

Computer-assisted navigation (CaN) systems have been assisting surgeons in their practice for approximately 30 years. At the same time, modern surgical drills

include sensors that measure force, pressure, depth, and torque. Integrating sensor data into the CaN system improves drill localization within the bone, enabling surgeons to determine the drill's position and penetration depth during procedures. This integration has the potential to improve surgical outcomes in orthopedic procedures by providing surgeons with real-time instrument localization during operations. In open reduction internal fixation (ORIF) surgery for distal radius fractures, the surgeon must drill into the bone and place screws with precision; errors in drill depth estimation can lead to tendon irritation, tendon rupture, or soft-tissue damage. Despite these technological advances, accurately classifying drill position from sensor data remains challenging.

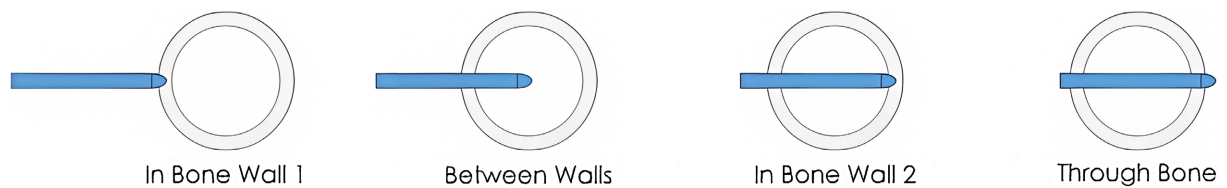


Figure 1. Simplified illustration of drilling through a bone.

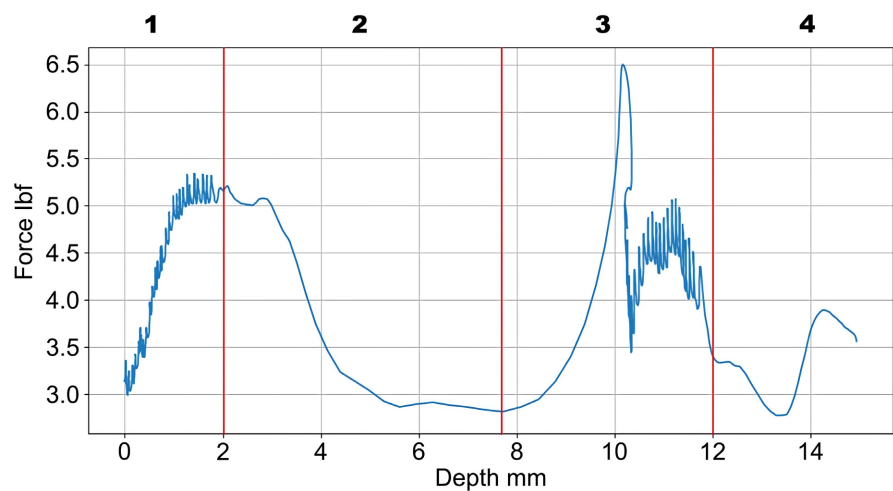


Figure 2. Example of drill distance and force through bone.

Figure 1 and **Figure 2** provide an illustration of drilling through the bone to install screws and plates used to set a distal radius fracture and demonstrate an ideal sequence of data instances collected from a depth sensor and a force sensor during bone drilling. In this context, “ideal” denotes low noise and ease of interpretation through visual analysis. Region 1 corresponds to drilling on bone wall 1; Region 2 corresponds to drilling between bone walls; Region 3 corresponds to drilling on bone wall 2; and Region 4 corresponds to drilling beyond the bone.

Blaney and Muknahallipatna [1] demonstrated that machine learning (ML) models such as long short-term memory (LSTM) [2], gated recurrent unit (GRU) [3], residual network [4], and inception network [5] architectures can perform automatic instrument localization through individual data instance classification

(IDIC). They used time-series data collected from depth and force sensors embedded in the surgical drill to train an ML model to localize the drill as a classification problem. Their evaluation revealed that all architectures achieved high classification accuracy for the majority of classes, *i.e.*, region 1 and region 3, corresponding to the two bone walls. However, performance on the minority classes was substantially lower, *i.e.*, region 2 (between bone walls) and region 4 (through bone).

They identified that the structural class imbalance in the training dataset was the root cause of the poor performance. Because the drill's sensors sample at a constant rate, the number of recorded data instances per region is proportional to the time the drill spends in that region. The drill traverses the inter-wall and post-bone regions significantly faster than it does the bone walls themselves, resulting in a dataset in which classes/regions 2 and 4 are substantially underrepresented. Although they identified synthetic data generation and augmentation as promising approaches to address this imbalance, existing methods for generating both statistically and physically realistic synthetic time-series data for this surgical application remain underdeveloped. This gap highlights the need for domain-specific generative models capable of producing valid synthetic sequences.

This paper directly addresses the class imbalance issue by proposing a physics-constrained Generative Adversarial Network (P-GAN) for synthetic time-series augmentation of underrepresented classes. Existing synthetic data generation methods applied to this dataset tend to yield sequences that are statistically valid but physically invalid (e.g., non-monotonic depth profiles or force values outside the sensor's operating range). Such synthetic data is inconsistent with real drill measurements and corrupts the dataset. The proposed P-GAN method explicitly incorporates domain-specific physical constraints into both the generator's output layer and its training objective. Specifically, P-GAN enforces monotonically increasing depth sequences to reflect the continuous advancement of the drill and constrains both depth and force to empirically determined sensor-valid ranges. These explicit constraints ensure that generated samples adhere to the actual physical properties of drilling processes, resulting in synthetic data that is both physically consistent and more effective for training a better classifier model.

This work makes the following contributions:

- Proposes a physics-constrained generative adversarial network (P-GAN) for synthetic data generation and augments the synthetic data into class-imbalanced medical time series. The model is built using the SeriesGAN architecture [6] with three modifications: hard output constraints, dropout regularization, and differentiable physics penalty terms.
- A study to understand the effect of P-GAN augmentation on the GRU and CNN classifiers from Blaney and Muknahallipatna [1], demonstrating consistent F1 improvements on minority classes without degrading performance on majority classes.

The remainder of the paper is organized as follows. Section 2 reviews related research on time series classification and approaches for addressing class imbalance through synthetic data generation. Section 3 details the proposed algorithm, its implementation, and the synthetic data generation process. Section 4 describes the drill dataset and the evaluation metrics used. Section 5 presents the key results and the impact on downstream classification performance. Finally, Sections 6 and 7 discuss the study's limitations and outline directions for future research.

2. Related Works

Time series classification (TSC) is more challenging than single-sample classification. Ismail Fawaz *et al.* [7] argued that TSC is fundamentally a data-mining task and reviewed various deep neural network architectures. Following this, Blaney and Muknahallipatna [1] studied various standard TSC architectures, which include recurrent networks such as Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU), as well as convolutional architectures adapted from image processing, including residual networks [8] and inception networks [5] [9]. Specifically, they evaluated individual-data instance classification (IDIC) on the surgical-drill dataset. They established GRU-based models as the strongest baseline, and identified class imbalance as one of the main obstacles to further performance gains. This work builds on the IDIC pipeline and focuses on improving the classifier's performance by minimizing class imbalance through generative data augmentation.

Generative Adversarial Networks (GANs) have been widely applied to generate synthetic time-series data. The TimeGAN framework by Yoon *et al.* [10] demonstrated that adversarial training could produce synthetic data sequences that preserve both feature distributions and temporal dynamics. Several approaches have been proposed to address the issues with the TimeGAN framework. Wasserstein GAN, proposed by Arjovsky *et al.* [11], improved training stability by addressing mode collapse; the Sig-Wasserstein GAN by Ni *et al.* [12] introduced signature-based discriminators to better match distributions; and SeriesGAN by EskandariNasab *et al.* [6] further improved performance across multiple benchmark datasets by combining adversarial learning with an autoregressive component.

At the same time, diffusion models have also emerged as an alternative for generating synthetic time-series data. Lin *et al.* [13] provided a survey of various diffusion models used in time series applications. Specifically, Coletta *et al.* [14] proposed GuidedDiffTime, which frames constrained time-series generation as a differentiable optimization problem, allowing domain-specific constraints to be imposed at inference time. Sikder *et al.* [15] proposed TransFusion, a method that combines diffusion models with transformer architectures to generate long, high-fidelity sequences. Furthermore, pure transformer architectures have also been explored for generating synthetic time series data. Srinivasan and Knottenbelt [16] proposed TsTGAN, which integrates the transformer self-attention mechanism within a GAN framework to better capture long-range temporal dependencies.

In addition to GANs and diffusion models, variational autoencoders (VAEs) have been used to generate synthetic time-series data. Kingma and Welling [17] proposed the Variational Autoencoder (VAE), a generative framework that learns a structured latent space by jointly optimizing a reconstruction objective and a Kullback-Leibler divergence regularization term. While VAEs have since been applied to synthetic time-series generation, they do not incorporate domain-specific physical constraints when generating samples.

While the above works explore methods for generating synthetic time series data, synthetic data for time series augmentation to address class imbalance has been less explored in classification contexts. Fawaz *et al.* [8] demonstrated that Dynamic Time Warping-based augmentation improves classification accuracy for deep residual networks. However, interpolation-based augmentation does not increase the diversity of training examples as generative models can, particularly for minority classes with few samples. More critically, interpolation methods provide no mechanism for enforcing physical constraints.

The reviewed literature shows that existing generative models for time series synthesis are designed for general-purpose datasets and do not incorporate domain-specific physical constraints into the generation process. At the same time, augmentation methods that respect domain properties (Fawaz *et al.* [8]) lack the generative capacity to capture the class distributions. For the surgical drill dataset, having statistically valid but physically invalid synthetic data does not improve the downstream classifier's performance; rather, it degrades it. Our work (P-GAN) addresses this gap by adding physical constraints directly into the training objective, producing synthetic sequences that are statistically representative of the class and physically consistent with the drilling process.

3. Methodology

This section describes the proposed P-GAN method for generating synthetic time series data to address class imbalance in the surgical drill dataset. P-GAN is a physics-constrained Generative Adversarial Network built on the SeriesGAN architecture. We propose and study three modifications to the SeriesGAN architecture to adapt it specifically to the drill dataset. Starting with dropout regularization to prevent overfitting, constraints to enforce valid physical ranges of sensor data, and physics-augmented loss terms to penalize physically invalid sequences. The core idea is that SeriesGAN provides the necessary statistical validity, and the physics constraints provide the necessary physical validity for the sample sequence. Together, these modifications ensure that the physical properties of the drilling process are preserved and that the generated synthetic data samples can augment missing samples in underrepresented classes.

3.1. P-GAN Base Architecture

P-GAN is built on the SeriesGAN framework [6], which employs a dual-network adversarial design. The generator is a recurrent neural network based on Gated

Recurrent Units (GRUs), which captures temporal dependencies within the sequence. The discriminator is a 1D convolutional network that classifies a given sequence as real or synthetic. The generator takes a noise tensor $Z \in \mathbb{R}^{B \times T \times D}$ as input and produces a synthetic sequence of the same shape, where B is the batch size, T is the sequence length, and D is the feature dimension. The discriminator is trained to distinguish real sequences from generated ones, while the generator is trained to produce sequences that the discriminator cannot distinguish from real data.

The SeriesGAN architecture was designed for general-purpose time series and produces unconstrained output values. To adapt SeriesGAN to the surgical drill dataset, enforce physical constraints, and improve training stability, we make the following three modifications:

3.1.1. Modification 1—Dropout Regularization

The base SeriesGAN generator consists of four stacked GRU layers that capture sequence dependencies while avoiding vanishing gradients. A dropout wrapper is applied to each GRU layer with an output keep probability of 80%. Interleaving dropout within the recurrent stack prevents any single hidden unit from dominating the learned dynamics, reducing run-to-run variance and mitigating overfitting. This is particularly important for the drill dataset, where the minority classes have few training examples, and the generator is at risk of memorizing rather than generalizing the class distribution.

3.1.2. Modification 2—Hard Output Constraints

The GRU stack feeds into a dense projection layer of size two, corresponding to the depth and force features, producing unconstrained real values $r \in \mathbb{R}^2$. To enforce sensor-valid output ranges, a sigmoid activation is applied to squash the raw output to the unit interval given by Equation (1).

$$[h]\tilde{r} = \sigma(r) \in (0,1)^2 \quad (1)$$

The squashed output is then rescaled to the physical operating range of the drill sensors using a linear transformation given by Equation (2).

$$[h]\hat{x} = \tilde{r}(m_{\max} - m_{\min}) + m_{\min} \quad (2)$$

where $m_{\min} = [0, 0]$ and $m_{\max} = [60 \text{ mm}, 22 \text{ lbf}]$, corresponding to the minimum and maximum depth and force values observed across all drilling sequences in this dataset. These bounds are distinct from the force and depth sensors' full hardware ranges used for normalization in Section 4; the generator is constrained to the physically achievable range of possible surgical procedures rather than the sensor's theoretical limits. $m_{\min}$ and $m_{\max}$ were selected to effectively address the observed depth limits of surgical procedures in the dataset.

This ensures that every generated sequence falls within the physically valid range of the depth and force sensors, regardless of the generator's internal state. The full constrained generator procedure is presented in **Algorithm 1**.

Algorithm 1 Time-Series Generator: Constrained

Input: Noise tensor $Z \in \mathbb{R}^{B \times T \times D}$, sequence length T , dropout keep-prob. p , number of layers L , hidden dim. D

Output: Generated series constrained $\in \mathbb{R}^{B \times T \times D}$

cells $\leftarrow \{\}$

for $i = 1$ to L **do**

$c \leftarrow \text{GRUCell}(D, \text{activation} = \tanh)$

$c \leftarrow \text{DropoutWrapper}(c, \text{output_keep_prob} = p)$

append c to **cells**

end for

multi_cell $\leftarrow \text{MultiRNNCell}(\text{cells})$

$(G, _) \leftarrow \text{dynamic_rnn}(\text{multi_cell}, Z, \text{sequence_length} = T)$

raw $\leftarrow \text{Dense}(G, \text{units} = D)$

squashed $\leftarrow \sigma(\text{raw})$

min_vals $\leftarrow [0, 0]$, **max_vals** $\leftarrow [60, 22]$

reshape **min_vals**, **max_vals** to $[1, 1, D]$

constrained $\leftarrow \text{squashed} \times (\text{max_vals} - \text{min_vals}) + \text{min_vals}$

return **constrained**

3.1.3. Modification 3—Physics-Constrained Generator Loss

We add two differentiable physics penalty terms into the generator loss alongside the standard adversarial loss L_{GAN} . This allows us to penalize the physically invalid drill sequences generated by the network. The two physics penalty terms are:

- **Bound-violation loss:** which penalizes any generated depth value that falls outside the defined physical limits given by Equation (3). Where d denotes a depth value in mm, $d_{\max}$ and $d_{\min}$ are the maximum and minimum permissible depths respectively, and $\hat{d}_t$ is the generated depth at time t .

$$L_{\text{bound}} = \mathbb{E}_t \left[\max(0, \hat{d}_t - d_{\max}) + \max(0, d_{\min} - \hat{d}_t) \right]. \quad (3)$$

- **Monotonicity loss:** which penalizes any decrease in depth between consecutive time steps, reflecting the physical constraint that the drill advances continuously and does not move backward during a drilling sequence given by Equation (4).

$$L_{\text{mono}} = \mathbb{E}_t \left[\max(0, \hat{d}_t - \hat{d}_{t+1}) \right]. \quad (4)$$

Finally, the total generator loss, which combines the adversarial loss with both physics penalties and is weighted by λ_1 and λ_2 , is given in Equation (5).

$$L_G(\theta) = L_{\text{GAN}}(\theta) + \lambda_1 L_{\text{bound}}(\theta) + \lambda_2 L_{\text{mono}}(\theta). \quad (5)$$

where θ denotes the trainable parameters of the generator network.

To prevent training instability during the early stages of optimization, the penalty weights are annealed linearly from zero to their maximum values over the first half of training iterations. The λ term is given by Equation (6).

$$\lambda_i(k) = \lambda_{i,\max} \frac{k}{K-1}, \quad k = 0, 1, \dots, K-1, \quad i \in \{1, 2\}. \quad (6)$$

The full physics-constrained training algorithm is presented in **Algorithm 2**.

Algorithm 2 Physics-Augmented Generator Training**Require:** Training data $\{(X, T)\}$, total epochs K , batch size B Physical limits $d_{\min}, d_{\max}$; max weights $\lambda_{1,\max}, \lambda_{2,\max}$

```

1: for  $k = 0$  to  $K - 1$  do
2:    $\lambda_1 \leftarrow \lambda_{1,\max} \frac{k}{K-1}$ 
3:    $\lambda_2 \leftarrow \lambda_{2,\max} \frac{k}{K-1}$ 
4:   for each batch  $(X_b, T_b)$  of size  $B$  do
5:      $Z_b \leftarrow \text{sample noise of shape } (B, \cdot)$ 
6:      $\hat{X}_b \leftarrow G(Z_b; \theta_G)$ 
7:      $L_{\text{GAN}} \leftarrow -\mathbb{E}_b[D(\hat{X}_b)]$ 
8:      $L_{\text{bound}} \leftarrow \mathbb{E}_{t,b}[\max(0, \hat{d}_{t,b} - d_{\max}) + \max(0, d_{\min} - \hat{d}_{t,b})]$ 
9:      $L_{\text{mono}} \leftarrow \mathbb{E}_{t,b}[\max(0, \hat{d}_{t,b} - \hat{d}_{t+1,b})]$ 
10:     $L_G \leftarrow L_{\text{GAN}} + \lambda_1 L_{\text{bound}} + \lambda_2 L_{\text{mono}}$ 
11:     $\theta_G \leftarrow \theta_G - \eta \nabla_{\theta_G} L_G$ 
12:   end for
13:   for each batch  $(X_b, T_b)$  do
14:     Update discriminator  $\theta_D$  via its own loss
15:   end for
16: end for

```

3.2. Implementation Details

P-GAN is implemented in TensorFlow. The generator consists of four GRU layers with a hidden dimension of $D = 2$, followed by the constrained output layer. All models are trained with a batch size of 128 and a constant learning rate of 0.0001 using the Adam optimizer. The physics penalty weights are annealed to their maximum values, $\lambda_1 = 1.0$ and $\lambda_2 = 0.5$, over the first half of the training epochs.

4. Dataset and Evaluation Metrics**4.1. Drill Dataset**

The raw drill dataset was collected from a medical drilling experiment and consists of two primary measurements: the force (in lbf) applied by the drill and the drill depth (in mm). These measurements were continuously recorded at high temporal resolution during drilling using the attached sensors at a constant sampling rate.

After acquiring the raw data, feature normalization was applied by scaling the raw values to the range $[0, 1]$ using the sensor's maximum depth capacity (600 mm) and the largest force value observed in the dataset (22 lbf). Note that while 600 mm represents the sensor's full hardware range, the actual drilling depth observed across all sequences is bounded by 60 mm, which is used as the generator's hard output constraint in Section 3. **Figure 1** provides the visual context that illustrates a simplified view of the bone drilling process, with four distinct classes corresponding to different drilling depths: in bone wall 1 (class 1), between walls (class 2), in bone wall 2 (class 3), and through bone (class 4).

During preprocessing, the data was partitioned into 1341 series, yielding 1,375,999 samples. Partitioning was guided by label confidence. The domain ex-

pert assigned the highest confidence labels, which were then split into the validation and test sets. The training set was labeled accordingly. The resulting distribution of data instances across the four classes is as follows: 28.54% for class 1 (in bone wall 1), 12.15% for class 2 (between walls), 41.04% for class 3 (in bone wall 2), and 18.26% for class 4 (through bone).

The imbalanced nature of the training dataset, particularly the under-representation of classes 2 and 4, hinders the downstream task of training a neural network classifier. Although theoretical expectations suggest that force values should be higher when the drill contacts the bone and lower when drilling elsewhere, this relationship is not consistently observed, further emphasizing the need for synthetic data generation to balance the dataset.

4.2. Evaluation Metrics

We utilized the following two metrics to evaluate the performance of the generated synthetic samples.

4.2.1. Discriminative Score

The discriminative score is a quantitative metric that measures how well synthetic data can be distinguished from real data. To compute this score, a post hoc time-series classification model (a Long Short-Term Memory (LSTM) network) is trained to distinguish between real and generated sequences. The classifier's error on the held-out test set reflects the similarity between the two distributions. Here, a lower error rate, adjusted to an optimal value of zero, implies that the synthetic data is nearly indistinguishable from the original data. This metric serves as a proxy for measuring the generative model's ability to accurately mimic the real data distribution. The calculation for this metric is given by Equation (7), where accuracy is the post-hoc classifier's classification accuracy.

$$\text{Discriminative Score} = |0.5 - \text{accuracy}| \quad (7)$$

4.2.2. Mean Absolute Error (MAE)

To assess whether the generated data preserves the temporal dependencies, the MAE is calculated. For this evaluation, an LSTM model is trained on the synthetic dataset for one-step-ahead prediction. The trained model is then tested on the real dataset, and forecasting performance is measured as the MAE, Equation (8) between the actual values (y_i) and the predicted values ($\hat{y}_i$). N is the total number of samples considered in the evaluation. A low MAE indicates that the synthetic data retains sufficient predictive information, making it beneficial for downstream time-series prediction applications.

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (8)$$

5. Results

This section covers the results obtained. First, we present the results obtained re-

garding the quality of the generated synthetic data. Second, we augment the generated data with the drill dataset and study the classifier's performance. Furthermore, we perform an ablation study to understand the effects of three modifications and present the incremental improvement from each.

5.1. Modification-Based Performance

The performance improvements based on the three modifications are shown in **Table 1**. Lower values are ideal for both metrics: a discriminative score near zero indicates that synthetic sequences are statistically indistinguishable from real ones, and a lower MAE indicates that temporal structure is preserved. The percentage changes in **Table 1** follow this convention, where a negative percentage change indicates a lower, and thus more ideal.

Table 1. SeriesGAN results over all model variations with drill dataset: Discriminative, predictive score, and percentage change in performance.

Model	Discriminative Score (% Change)	MAE (% Change)
Vanilla	0.2813 ± 0.0699 (0% Baseline)	0.2651 ± 0.0011 (0% Baseline)
+Constrained	0.1941 ± 0.1256 (−31.0%)	0.3029 ± 0.0065 (+12.5%)
+Dropout	0.2374 ± 0.0209 (−15.6%)	0.236 ± 0.0086 (−11.0%)
+Physics	0.2084 ± 0.0639 (−25.9%)	0.2257 ± 0.0149 (−14.9%)

Adding hard output constraints improved the discriminative score substantially (−31.0%), indicating that bounding depth and force to sensor-valid ranges produce sequences that more closely resemble real data. However, the MAE increased (+12.5%), suggesting the constrained generator overfits to the valid range boundaries rather than capturing the underlying temporal dynamics. The subsequent addition of dropout corrected this overfitting: discriminative score and MAE improved relative to the constrained-only configuration, recovering predictive quality while retaining most of the discriminative gain. Finally, incorporating physics-augmented loss terms further improved both metrics, yielding the best overall MAE (0.2257) and a discriminative score (0.2084) well below the vanilla baseline. These findings highlight that constraint enforcement, regularization, and physics-informed training are complementary, as each addresses a distinct failure mode, and all three are necessary for the full benefit.

5.2. Ablation Study

To evaluate the individual contributions of the various novel additions to the SeriesGAN model, a more detailed ablation study was conducted. Several versions of the SeriesGAN model were evaluated, each incorporating different combinations of model components: hard output constraints to enforce domain-specific bounds, dropout to prevent overfitting, and physics-informed loss functions to penalize physically implausible sequences.

Table 2. SeriesGAN results over all model variations with drill dataset: Discriminative and predictive score.

Model	Discriminative Score	MAE
Vanilla	0.2813 ± 0.0699	0.2651 ± 0.0011
Constrained Only	0.1941 ± 0.1256	0.3029 ± 0.0065
Dropout Only	0.3266 ± 0.045	0.2524 ± 0.0003
Physics Only	0.0645 ± 0.0498	0.2165 ± 0.0009
Physics + Constrained	0.1443 ± 0.0895	0.2292 ± 0.0003
Physics + Dropout	0.2655 ± 0.1136	0.2803 ± 0.0041

The performance of all model combinations is summarized in **Table 2**. The discriminative score measures how indistinguishable synthetic data is from real data (lower is better, with zero being ideal), while the MAE assesses how well the synthetic data's temporal structure transfers to real-data prediction tasks. The Vanilla model serves as the baseline.

Constrained Only: Applying hard output constraints alone produced the second-best discriminative score (0.1941), a meaningful improvement over the vanilla baseline (0.2813). However, the MAE increased from 0.2651 to 0.3029, indicating that while constraining outputs to sensor-valid ranges makes sequences look more realistic to a discriminator, it does not ensure that the underlying temporal dynamics are faithfully reproduced. The generator learns to stay within physical bounds but tends to overfit to the boundary region rather than generalizing the class distribution.

Dropout Only: Adding dropout without any physical constraints increased the discriminative score from 0.2813 to 0.3266, meaning synthetic sequences became easier to distinguish from real data. Dropout alone, without any guidance about valid sensor ranges, reduces the generator's capacity to reproduce fine-grained signal characteristics. The MAE decreased slightly (0.2651 to 0.2524), consistent with dropout's role as a regularizer that discourages memorization and improves generalization of temporal patterns.

Physics Only: The physics-augmented loss alone produced the lowest discriminative score in the entire ablation (0.0645), indicating that penalizing monotonicity violations and bound violations is the single most powerful driver of statistical realism in this dataset. The MAE also decreased relative to the baseline (0.2165). However, this configuration exhibited high variance and training instability in practice: without the sigmoid-clamping of hard output constraints to anchor the generator's output range, the physics penalty alone is insufficient to stabilize training across runs. This motivates combining the physics loss with at least one structural modification.

Physics + Constrained: Combining the physics loss with hard output constraints improved the discriminative score substantially (0.1443) and produced a low MAE (0.2292). Relative to Physics Only, the discriminative score has in-

creased, which reflects the trade-off introduced by sigmoid clamping: the hard constraints stabilize training but restrict the generator's output distribution, slightly reducing its ability to match the full range of real sequences.

Physics + Dropout: Pairing the physics loss with dropout, but without hard output constraints, yielded a discriminative score of 0.2655 and the worst MAE among physics-containing configurations (0.2803). Without the sigmoid clamping to bound outputs, dropout alone is insufficient to prevent the generator from producing out-of-range values, which degrades both metrics relative to the full P-GAN configuration.

Overall, the ablation demonstrates that all three modifications are complementary. Hard output constraints provide training stability and range validity; dropout prevents overfitting on the small minority-class training sets; and the physics-augmented loss directly encodes domain knowledge about drilling dynamics. The complete P-GAN, incorporating all three, achieves the best balance of discriminative realism and MAE, and avoids the instability observed in the Physics Only configuration.

5.3. Effect of Synthetic Data on Classification

5.3.1. Experimental Setup

For the downstream classifier, we adopt the GRU-based model without modification from Blaney and Muknahallipatna [1], which evaluated several deep models for time-series individual data-instance classification on the same drill dataset and reported GRU as the best-performing baseline among the tested architectures. Consistent with that work, we train one-vs-rest (OvR) binary classifiers for each of the four drill regions (classes 1 - 4) using a univariate input (Depth in mm), window length $T = 100$, batch size 128, and a decision threshold of 0.5.

All OvR heads use the same optimizer and early stopping settings as the baseline in [1], with a batch size of 128. To isolate the effect of synthetic augmentation, we evaluate at a fixed decision threshold of 0.5 for every head and report Accuracy (Acc.), Precision (Prec.), Recall (Rec.), and F1 on the held-out test set. The univariate Depth-only input is justified by prior findings that bivariate (Depth + Force) did not improve accuracy relative to Depth alone on this dataset [1]. The results in Tables 3-7 show the results in which synthetic data was added only to

Table 3. Performance of RNN aggregated across runs. # Synth. = Number of Synthetic Samples, Acc. = Accuracy, Prec. = Precision, Rec. = Recall.

Class	# Synth.	Acc.	Prec.	Rec.	F1
1	0	97.42%	99.49%	97.14%	98.30%
2	0	21.48%	39.84%	94.83%	56.11%
2	150	55.30%	53.95%	96.24%	69.14%
3	0	88.75%	98.15%	82.87%	89.87%
4	0	77.95%	81.47%	98.62%	89.22%
4	150	84.26%	81.46%	98.62%	92.30%

Table 4. Performance of CNN aggregated across runs. # Synth. = Number of Synthetic Samples, Acc. = Accuracy, Prec. = Precision, Rec. = Recall.

Class	# Synth.	Acc.	Prec.	Rec.	F1
1	0	96.44%	97.57%	97.78%	97.67%
2	0	35.70%	85.58%	99.14%	92.00%
2	150	83.32%	97.18%	99.06%	92.74%
3	0	90.62%	97.03%	85.39%	90.84%
4	0	59.00%	94.37%	56.28%	70.51%
4	150	85.20%	98.17%	92.83%	95.43%

the original dataset for cases in which classes 2 and 4 were trained.

Next, the dataset was augmented with synthetic data for all training runs. A total of 300 synthetic sequences were added, 150 belonging to class 1, and the remaining 150 belonging to class 3.

Table 5. Performance of RNN aggregated across runs all with synthetic data. # Synth. = Number of Synthetic Samples, Acc. = Accuracy, Prec. = Precision, Rec. = Recall.

Feature Set	Class	Acc.	Prec.	Rec.	F1
Bivariate	1	96.82%	97.09%	98.82%	97.95%
	2	37.74%	1.00%	92.81%	96.27%
	3	87.78%	76.62%	30.27%	43.40%
	4	92.36%	41.31%	97.92%	58.10%
Depth (Univariate)	1	97.41%	97.61%	99.13%	98.36%
	2	92.06%	1.00%	92.81%	96.27%
	3	89.47%	96.18%	83.98%	89.67%
	4	93.50%	96.80%	94.25%	95.51%
Force (Univariate)	1	85.10%	1.00%	75.03%	85.74%
	2	92.06%	1.00%	92.813%	96.27%
	3	51.77%	99.99%	50.82%	67.39%
	4	80.94%	99.95%	81.463%	89.762%

Table 6. Performance of CNN aggregated across runs with synthetic data. # Synth. = Number of Synthetic Samples, Acc. = Accuracy, Prec. = Precision, Rec. = Recall.

Feature Set	Class	Acc.	Prec.	Rec.	F1
Bivariate	1	90.40%	94.02%	99.29%	96.59%
	2	61.45%	66.09%	95.18%	77.99%
	3	83.78%	87.62%	79.86%	83.56%
	4	93.07%	99.64%	91.72%	95.51%

Continued

Depth (Univariate)	1	96.18%	97.16%	97.89%	97.52%
	2	77.74%	84.26%	99.78%	91.37%
	3	93.60%	98.03%	89.17%	93.40%
	4	78.28%	87.47%	92.01%	89.68%
Force (Univariate)	1	72.87%	78.74%	85.83%	82.13%
	2	84.22%	97.13%	92.75%	94.89%
	3	62.41%	86.69%	57.93%	69.54%
	4	66.48%	80.97%	80.70%	80.83%

Table 7. Performance of ResNet aggregated across runs with synthetic data. # Synth. = Number of Synthetic Samples, Acc. = Accuracy, Prec. = Precision, Rec. = Recall.

Feature Set	Class	Acc.	Prec.	Rec.	F1
Bivariate	1	95.39%	99.97%	79.27%	88.42%
	2	54.88%	93.16%	92.43%	92.80%
	3	72.03%	99.23%	51.52%	67.83%
	4	86.39%	98.19%	85.07%	91.15%
Depth (Univariate)	1	96.31%	1.00%	75.03%	85.74%
	2	62.92%	95.81%	97.51%	96.65%
	3	89.28%	99.61%	11.96%	21.36%
	4	92.39%	98.61%	93.35%	95.91%
Force (Univariate)	1	63.91%	80.37%	81.80%	81.07%
	2	68.37%	88.65%	92.62%	90.60%
	3	51.77%	99.69%	50.82%	67.32%
	4	76.89%	96.80%	81.38%	88.42%

5.3.2. Data Splits and Augmentation Protocol

Following [1], the original 2023 validation and test partitions are kept fixed to avoid leakage. Synthetic sequences are added only to the training fold of targeted classes. Unless otherwise noted, all synthetic sequences use the same window length ($T = 100$), normalization (μ, σ) , and preprocessing as the baseline, and respect the physical constraints described in the Method section. In the one-vs-rest (OvR) setup, class k receives synthetic positives when training the k th binary head; negatives are drawn from the union of the remaining (real) classes. Validation and test folds contain no synthetic data.

5.3.3. Data Imbalance Addressed by Synthetic Data Augmentation

The statistics of the data balance of the original dataset is shown in Table 8 and the same analysis for the synthetically augmented dataset is shown in Table 9. Each table reports the per-class sample counts (# Samples) and the resulting percentage share

$$\% \text{Samples}(k) = \frac{\# \text{Samples in class } k}{\sum_{c \in \{0,1,2,3\}} \# \text{Samples in class } c} \times 100.$$

Table 8 shows the baseline distribution before augmentation; **Table 9** shows the distribution after adding synthetic sequences *only* to the training fold and *only* for the targeted classes (2 and 4). Validation and test folds remain unchanged (no synthetic data), so their class balances are identical to the original. The number of synthetic sequences per minority class was selected based on downstream classifier performance. Preliminary experiments evaluated augmentation counts of 50, 100, 150, 200, and 300 sequences per minority class. Counts below 150 produced insufficient improvement in minority class F1, while counts above 150 consistently led to high precision paired with low recall in the OvR classifiers, indicating that the synthetic advantages began to dominate the training and pushed the decision boundary too aggressively. The value of 150 sequences per class was therefore selected as the point that maximizes F1 improvement without introducing this precision-recall imbalance. A systematic sensitivity analysis across a finer range of augmentation counts, with visualization of the precision-recall tradeoff curve, is identified as an area for future work.

Table 8. Class sample data balance for original dataset.

Class	# Samples	% Samples
1	396,885	28.43%
2	190,946	13.98%
3	493,849	35.37%
4	284,415	20.82%

Table 9. Class sample data balance for dataset augmented with synthetic data.

Class	# Samples	% Samples
1	396,885	28.43%
2	205,946	14.75%
3	493,849	35.37%
4	299,415	21.45%

As intended, the augmentation increases Classes 1 and 3 by +15,000 samples each (from 190,946 $\rightarrow$ 205,946 and 284,415 $\rightarrow$ 299,415, respectively), while Classes 1 and 3 are held fixed. In percentage terms, Class 2 rises from 13.98% to 14.75% (+0.77%) and Class 4 from 20.82% to 21.45% (+0.63%), narrowing their gaps to the majority class (Class 3 at 35.37%) without altering the relative prevalence of Classes 1 and 3. Overall, this shifts the training distribution toward a more balanced mix for the classes of interest, while preserving the integrity of the validation/test splits for unbiased evaluation.

5.4. Analysis

5.4.1. RNN

Table 3 summarizes the GRU (RNN) outcomes. Classes 1 and 3 were augmented; classes 0 and 2 were not.

For Class 2, adding 150 synthetic positives increases Acc. from 21.48% to 55.30% (+33.82 points) and F1 from 56.11% to 69.14% (+13.03). Precision rises from 39.84% to 53.95% (+14.11) while Recall remains high (94.83% $\rightarrow$ 96.24%, +1.41). This pattern indicates the synthetic sequences primarily reduce false positives without sacrificing sensitivity.

For Class 4, 150 synthetic samples improve Acc. from 77.95% to 84.26% (+6.31) and F1 from 89.22% to 92.30% (+3.08). Recall is already saturated at 98.62% and remains unchanged, while precision is essentially stable (81.47% $\rightarrow$ 81.46%). Here, augmentation trims residual errors against the majority of negatives.

Classes 1 and 3, trained only on real data, already achieve strong performance (Class 1 F1 98.30%; Class 3 F1 89.87%), suggesting limited necessity for augmentation with the currently available dataset.

5.4.2. CNN

Table 4 shows the 1D-CNN results under the same protocol.

For Class 2, augmentation yields a large accuracy gain (35.70% $\rightarrow$ 83.32%, +47.62). Precision improves (85.58% $\rightarrow$ 97.18%), Recall remains near ceiling (99.14% $\rightarrow$ 99.06%), and F1 nudges upward (92.00% $\rightarrow$ 92.74%). The substantial increase in accuracy reflects fewer false positives in the OvR framing.

For Class 4, augmentation produces the largest overall benefit: Accuracy climbs from 59.00% to 85.20% (+26.20), Recall from 56.28% to 92.83% (+36.55), Precision from 94.37% to 98.17%, and F1 from 70.51% to 95.43% (+24.92). Synthetic transitions likely broaden coverage of difficult boundary dynamics that previously drove false negatives.

Classes 1 and 3 remain strong without augmentation (Class 1 F1 97.67%; Class 3 F1 90.84%), consistent with the RNN observations.

5.4.3. Cross-Model Comparison and Takeaways

Across both backbones, adding synthetic positives to the underrepresented heads (Classes 2 and 4) consistently improves F1 and overall accuracy, with the CNN showing larger gains, especially via recall recovery for Class 4. The RNN gains are steadier but smaller, largely through precision improvements for Class 2. When synthetic positives are too narrow or overly “clean”, OvR heads can become conservative (high precision, low recall). The physics-aware generation used here appears to expand coverage without producing implausible artifacts, which aligns with the observed recall gains.

Operating point matters: a fixed 0.5 threshold is convenient for comparability, but per-class threshold tuning on validation data typically yields further improvements (e.g., maximizing F1 or meeting a minimum recall). We keep thresholds fixed to isolate the effects of augmentation.

5.4.4. Error Analysis

Residual errors concentrate near boundaries (e.g., wall entry/exit) where short, low-SNR fluctuations blur class transitions. Without augmentation, CNN Class 4 exhibits many false negatives; introducing synthetic transitions substantially recovers recall. For RNN Class 2, remaining errors are mostly false positives, suggesting partial overlap between genuine positives and specific negative contexts. Additional diversity in synthetic depth/force trajectories, or mild label smoothing for synthetic samples, may reduce overconfident spikes.

In summary, physics-constrained synthetic augmentation improves minority-class recognition for both RNNs and CNNs while preserving strong performance in majority regimes. The largest gains occur where synthetic data expands the coverage of transitional dynamics (Class 4), producing substantial recall and F1 improvements, especially for the CNN. These findings support the use of domain-constrained synthetic data to remedy class imbalance in drill-depth time-series classification.

6. Limitations

Although synthetic data generation methods, including GANs, diffusion models, and transformers, have demonstrated substantial promise in mitigating class imbalance and enhancing model performance, they are not without limitations. One of the primary challenges is ensuring that the generated synthetic data is highly representative of real-world data, particularly in complex domains such as medical applications. Despite the refined constraints and loss functions applied to the generative models, overfitting to specific patterns within the training data may occur, leading to synthetic sequences that do not fully capture the variability observed in real-world scenarios. Additionally, these models require significant computational resources and fine-tuning, especially when dealing with high-dimensional time series data. The need for large amounts of real data for effective model training also remains a challenge, as even state-of-the-art techniques may not generalize well when faced with significantly different real-world distributions. Furthermore, while synthetic data can augment existing datasets, it is not a perfect substitute for actual labeled data, especially in domains requiring high precision and expert validation.

7. Future Work

Future research can build on this study by exploring more sophisticated hybrid models that combine the strengths of different generative approaches. For instance, integrating transformers with GANs and diffusion models could enable more robust modeling of complex temporal dependencies and reduce overfitting in certain configurations. Another promising direction involves the exploration of few-shot learning methods to reduce the need for extensive real data, making synthetic data generation more efficient and accessible. A closer examination of the quality of synthetic data, particularly through domain-specific validation by medical professionals, could lead to further refinements in ensuring the clinical

applicability of the generated sequences. Finally, integrating real-time data augmentation during training could enable adaptive methods to handle changing distributions over time. A promising alternative direction is the use of physics-informed neural networks (PINNs), which embed the governing physical laws (e.g., drilling rate dynamics or torque constraints) directly into the neural network training. This could allow the model to enforce physical realism implicitly through differential equation solvers rather than relying on explicit penalty terms. Such models may generalize better with less data and reduce computational overhead associated with tuning complex loss functions.

8. Conclusion

In this paper, we investigated the use of advanced generative models, including GANs, P-GANS, diffusion models, and transformer architectures, for augmenting imbalanced time series datasets in medical drilling applications. The results demonstrate that synthetic data generation can significantly enhance classification performance by balancing the class distribution and capturing the temporal dynamics inherent in the data. Modifications to SeriesGAN, such as the inclusion of dropout and physics-informed constraints, resulted in improved performance over the baseline model, with the drill dataset exhibiting comparable results to standard benchmarks. While promising, challenges such as overfitting and the need for substantial computational resources remain, highlighting areas for future improvements. Preliminary experiments that add physics-constrained synthetic sequences only to the training fold yield consistent qualitative gains in the one-vs-rest classifier, most notably for the minority/transitional regimes (Classes 2 and 4). The augmentation appears to curb over-confident false positives in Class 2 and recover boundary-related false negatives in Class 4, while leaving Classes 1 and 3 largely unchanged and preserving the integrity of the fixed validation/test splits. These trends motivate future work to scale the augmentation (e.g., mixing ratios and transition diversity), tune per-class operating points and calibration, and study robustness under domain shift so that the observed improvements translate into stable, clinically meaningful performance. Overall, the findings suggest that synthetic data generation offers a viable solution for addressing imbalanced datasets in critical applications like medical diagnostics. Moreover, this framework can be extended to broader domains such as construction drilling, blood flow modeling, or other biomedical signal generation tasks where physical laws govern system dynamics.

Acknowledgements

We express our sincere thanks to the domain experts for providing help with labeling the datasets and insights about medical procedures.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

- [1] Blaney, J.A. and Muknahallipatna, S.S. (2024) An Evaluation of Deep Learning Models for Classifying Time Series Individual Data Instances. *Journal of Computer and Communications*, **12**, 187-206. <https://doi.org/10.4236/jcc.2024.1211014>
- [2] Hochreiter, S. and Schmidhuber, J. (1997) Long Short-Term Memory. *Neural Computation*, **9**, 1735-1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [3] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H. and Bengio, Y. (2014) Learning Phrase Representations Using RNN Encoder-Decoder for Statistical Machine Translation. arXiv: 1406.1078.
- [4] He, K.M., Zhang, X.Y., Ren, S.Q. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 770-778. <https://doi.org/10.1109/CVPR.2016.90>
- [5] Szegedy, C., Liu, W., Jia, Y.Q., Sermanet, P., Reed, S., Anguelov, D., et al. (2015) Going Deeper with Convolutions. 2015 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 7-12 June 2015, 1-9. <https://doi.org/10.1109/cvpr.2015.7298594>
- [6] EskandariNasab, M., Hamdi, S.M. and Boubrahimi, S.F. (2024) SeriesGAN: Time Series Generation via Adversarial and Autoregressive Learning. 2024 *IEEE International Conference on Big Data (BigData)*, Washington, 15-18 December 2024, 860-869. <https://doi.org/10.1109/bigdata62323.2024.10825115>
- [7] Ismail Fawaz, H., Forestier, G., Weber, J., Idoumghar, L. and Muller, P. (2019) Deep Learning for Time Series Classification: A Review. *Data Mining and Knowledge Discovery*, **33**, 917-963. <https://doi.org/10.1007/s10618-019-00619-1>
- [8] Fawaz, H.I., Forestier, G., Weber, J., Idoumghar, L. and Muller, P.A. (2018) Data Augmentation Using Synthetic Data for Time Series Classification with Deep Residual Networks. arXiv:1808.02455.
- [9] Ismail Fawaz, H., Lucas, B., Forestier, G., Pelletier, C., Schmidt, D.F., Weber, J., et al. (2020) InceptionTime: Finding AlexNet for Time Series Classification. *Data Mining and Knowledge Discovery*, **34**, 1936-1962. <https://doi.org/10.1007/s10618-020-00710-y>
- [10] Yoon, J.S., Jarrett, D. and Van der Schaar, M. (2019) Time-Series Generative Adversarial Networks. 33rd *Conference on Neural Information Processing Systems (NeurIPS 2019)*, Vancouver, 8-14 December 2019, 5508-5518.
- [11] Arjovsky, M., Chintala, S. and Bottou, L. (2017) Wasserstein GAN. arXiv: 1701.07875.
- [12] Ni, H., Szpruch, L., Wiese, M., Liao, S. and Xiao, B. (2020) Conditional Sig-Wasserstein Gans for Time Series Generation. *SSRN Electronic Journal*, **34**, 622-670. <https://doi.org/10.2139/ssrn.3623086>
- [13] Lin, L.Q., Li, Z.K., Li, R.K., Li, X.L. and Gao, J.B. (2023) Diffusion Models for Time Series Applications: A Survey. *Engineering Information Technology & Electronic Engineering*, **25**, 19-41. <https://doi.org/10.1631/FITEE.2300310>
- [14] Coletta, A., Gopalakrishnan, S., Borrajo, D. and Vyetrenko, S. (2023) On the Constrained Time-Series Generation Problem. *Advances in Neural Information Processing Systems* 36, New Orleans, 10-16 December 2023, 61048-61059. <https://doi.org/10.52202/075280-2668>
- [15] Sikder, M.F., Ramachandranpillai, R. and Heintz, F. (2024) Transfusion: Generating Long, High Fidelity Time Series Using Diffusion Models with Transformers. *Machine*

Learning with Applications, **20**, Article ID: 100652.

<https://doi.org/10.1016/j.mlwa.2025.100652>

- [16] Srinivasan, P. and Knottenbelt, W.J. (2022) Time-Series Transformer Generative Adversarial Networks. arXiv: 2205.11164.
- [17] Kingma, D.P. and Welling, M. (2022) Auto-Encoding Variational Bayes. arXiv: 1312.6114.