Synthetic 5G Traffic Generation: A Machine Learning Approach

Karsten Cedric van der Deijl[1]
Supervisor(s): Nitinder Mohan[1], **Marco Colocrese**[1]
[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 23, 2025

Name of the student: Karsten Cedric van der Deijl
Final project course: CSE3000 Research Project
Thesis committee: Nitinder Mohan, Marco Colocrese, Guohao Lan

Synthetic 5G Traffic Generation: A Machine Learning Approach

Karsten Cedric van der Deijl
Delft University of Technology
Delft, The Netherlands

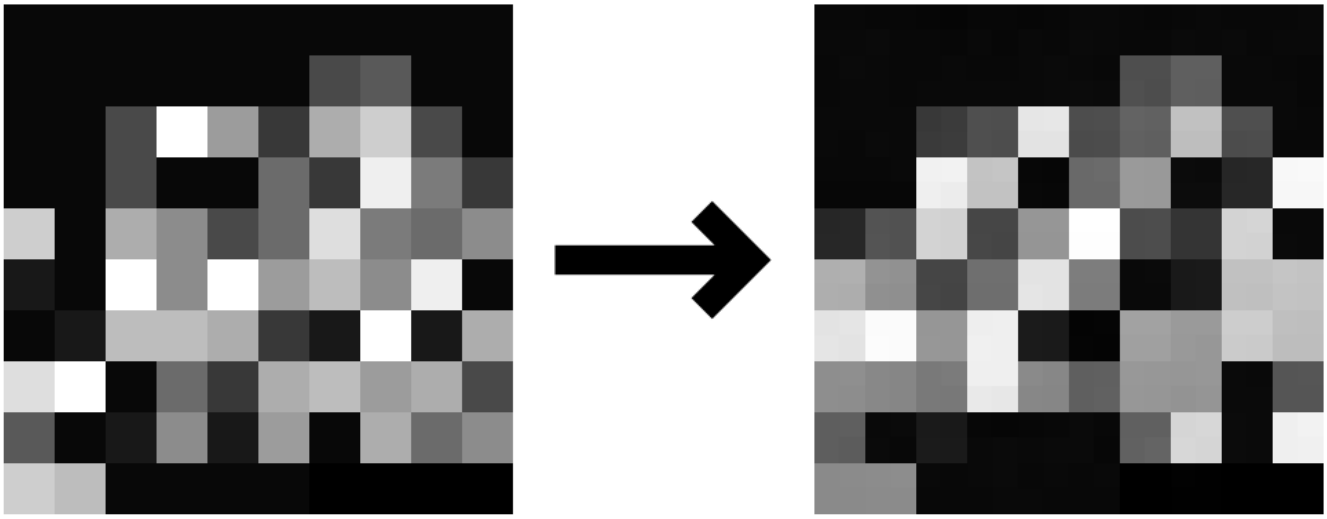

Figure 1: Left: Real Encoded Packet → Right: Generated Encoded Packet
Encoded through the method desribed in [1]

Abstract

Academic research in 5G networking faces a lack of accessible, realistic packet-level datasets, limiting innovation and reproducibility. This paper evaluates two state-of-the-art machine learning approaches, PAC-GAN and TabularARGN, for generating synthetic 5G TCP/IP packet headers. Using a real 5G packet-capture dataset, we adapt both models to include inter-packet timing and rigorously assess them on protocol validity, marginal distribution alignment, and joint distribution fidelity. Results show that PAC-GAN produces highly valid and statistically faithful synthetic packets, effectively modeling complex header dependencies and temporal patterns. While TabularARGN ensures strict protocol compliance, it struggles to capture higher-order correlations and traffic diversity. Our findings establish convolutional generative models like PAC-GAN as practical tools for producing realistic, protocol-compliant synthetic 5G traffic, broadening access to datasets for benchmarking and security testing.

1 Introduction

The lack of accessible, high-fidelity traffic datasets fundamentally constrains academic research into mobile communications infrastructure. Network operators and equipment vendors typically treat network traces as proprietary, and privacy concerns further restrict public sharing. As a result, students and researchers often cannot access realistic workloads, impeding efforts to benchmark performance, validate new protocols, or develop effective capacity-planning strategies under authentic conditions.

Existing synthetic traffic-generation tools offer limited relief. While they can produce artificial datasets, these tools frequently fail to capture the complex temporal and structural patterns in real-world 5G network traffic. This shortfall means that generated data may not accurately reflect the nuanced dependencies and distributions that characterize actual network behavior, limiting their utility for rigorous simulation and evaluation.

To address these barriers, we have honed in on and systematically evaluated the capabilities of two advanced deep generative models, PAC-GAN and TabularARGN, to synthesize individual 5G packet headers. We aim to generate synthetic packet data that closely preserves the statistical properties and structural nuances observed in real network traffic by focusing on these targeted machine learning approaches. This enables the creation of diverse, realistic packet-level datasets on demand, reducing reliance on scarce proprietary data. By democratizing access to high-fidelity synthetic network traffic, our approach supports more rigorous and reproducible research in network benchmarking, capacity planning, and stress-testing, ultimately empowering both academia and industry to advance the state of the art in 5G and future network technologies.

1.1 Related Works

The generation of synthetic network traffic has traditionally relied on tools and models that emphasize aggregate flow statistics and bandwidth patterns, rather than the synthesis of individual, protocol-compliant packets. Conventional generators such as TRex[1], for example, are widely used for producing high-throughput network traffic in testing environments. However, these tools typically operate at fixed packets-per-second (pps) and bits-per-second (bps) rates, and lack the ability to reproduce the nuanced temporal variability and packet size distributions observed in real-world traffic. As a result, they fall short in capturing the structural and temporal diversity necessary for realistic simulation and benchmarking.

In parallel, a significant body of academic work has focused on flow- and session-level synthetic data generation. These approaches generally model high-level properties, such as connection patterns, session durations, or aggregate bandwidth, using machine learning techniques including GANs, VAEs, and Bayesian networks. While effective for reproducing statistical trends at the flow level, these models do not address the technical challenges of generating individual packets with realistic protocol headers and inter-field dependencies, which are essential for fine-grained simulation and protocol evaluation.

A comprehensive review by Schoen et al. [2] highlights this gap, noting that most prior works in the field are limited to flow-based or trace-based generation and do not provide solutions for the direct synthesis of structurally valid, protocol-compliant packets. This persistent focus on aggregate metrics underscores the need for new approaches capable of generating realistic packet-level data that faithfully reflects both the structural and temporal characteristics of real network traffic. The field of synthetic network traffic generation has historically focused on aggregate flow statistics and bandwidth modeling, rather than packet-level synthesis. As highlighted by Schoen et al. [2], most existing approaches generate synthetic data at the flow or session level, capturing high-level statistical properties but lacking the capability to generate realistic packet-wise traffic data. Their comprehensive survey demonstrates that while a variety of machine learning methods have been applied to trace and flow generation, the direct synthesis of protocol-compliant, structurally valid packets remains a significant research gap.

1.1.1 Flow and Trace Generation. Schoen et al. [2] systematically reviewed the landscape of synthetic traffic generation and found that the majority of prior works focus on reproducing aggregate metrics such as connection patterns, session durations, or bandwidth utilization. These models often employ GANs, VAEs, or Bayesian networks to generate flow-level traces, but do not address the technical challenges of generating individual packets with realistic protocol headers and field dependencies. As a result, they are limited in their ability to support fine-grained simulation, benchmarking, or protocol testing scenarios that require packet-level fidelity.

1.1.2 Direct Packet Generation. To address these limitations, recent research has explored generative models capable of synthesizing individual packets. Three models stand out as the closest to achieving true packet-level generation: PAC-GAN [1], PacketCGAN [3], and PcapGAN [4]. All three leverage GAN-based architectures to model the distribution of real network packets and generate protocol-compliant outputs.

- **PAC-GAN** [1] introduces a convolutional GAN framework that spatially encodes raw packet bytes into 20×20 matrices, allowing the model to learn both local field dependencies and global packet structure. This enables the generation of protocol-compliant packets that can be transmitted over real networks and elicit valid responses from servers, demonstrating practical utility for simulation and testing. PAC-GAN's architectural novelty, particularly its use of CNNs and nibble-based encoding, makes it a strong baseline for evaluating structural and statistical fidelity in synthetic 5G header synthesis.
- **PacketCGAN** [3] extends the GAN paradigm with conditional labels, enabling targeted synthesis of specific traffic classes and addressing class imbalance, which is especially relevant for rare encrypted flows in security applications. By incorporating attention mechanisms, PacketCGAN enhances class-specific feature replication, producing more faithful representations of underrepresented packet types.
- **PcapGAN** [4] focuses on sequential packet synthesis by employing GANs trained on entire pcap sessions, aiming to capture the structural and temporal patterns present in real network conversations. By generating sequences of packets that reflect the flow and session-level ordering observed in actual traffic, PcapGAN produces synthetic traces that better approximate the dynamics of client-server exchanges.

These models collectively represent the state-of-the-art in direct packet synthesis. Among them, PAC-GAN is selected as the primary baseline in this work due to its demonstrated technical validity and architectural innovations, making it an ideal target for evaluation and comparison in the context of 5G traffic generation.

1.1.3 Tabular Generation. An alternative approach to packet synthesis is to represent packet headers as tabular data, where each protocol field corresponds to a column. This enables the use of advanced tabular data generators, which can model the joint distribution of header fields without explicit protocol knowledge. A collection of state-of-the-art generators in the literature are presented here.

- **TabularARGN** [5] pioneers any-order auto-regressive density estimation for mixed-type tabular data, dynamically shuffling column order during training to learn all conditional dependencies across features. With support for variable-length sequences and built-in privacy mechanisms, TabularARGN achieves superior fidelity and efficiency on benchmarks, making it a promising candidate for representing packet headers as structured rows and columns in 5G traffic synthesis. Notably, Tiwald et al. demonstrate that TabularARGN consistently outperforms other leading tabular data generators, including GAN-, VAE-, and diffusion-based models, across a diverse set of real-world

[1] trex-tgn.cisco.com

and synthetic datasets, highlighting its robustness and generalization capabilities in capturing complex data distributions.

- **TVAE** [6] applies variational autoencoder principles to mixed-type tabular data, converting heterogeneous features into normalized vectors for probabilistic sampling. While TVAE offers uncertainty quantification and robust univariate fidelity, its performance on discrete distributions is limited compared to auto-regressive methods.
- **CTGAN** [6] employs mode-specific normalization and class-conditional sampling to handle mixed continuous and categorical features in tabular data. CTGAN excels at matching marginal distributions of individual fields but requires careful hyperparameter tuning to avoid mode collapse.
- **REaLTabFormer** [7] leverages transformer architectures with self-attention to model intricate relationships within tabular datasets, achieving high-fidelity synthetic samples on structured benchmarks. Despite its computational demands, the model's ability to capture long-range feature interactions offers insight into advanced sequence modeling approaches.

While none of these models were explicitly designed for packet synthesis, the tabular representation of packet headers allows for their direct application. TabularARGN, in particular, is highlighted for its ability to enable high-fidelity, protocol-compliant header generation by modeling each field as a column in a structured dataset.

In summary, the current landscape of synthetic network traffic generation is characterized by a divide between high-level flow modeling and emerging packet-level synthesis. This work bridges the gap by systematically evaluating both direct packet generation models (with a focus on PAC-GAN) and advanced tabular data generators (with a focus on TabularARGN), providing a comprehensive assessment of their suitability for realistic 5G packet header generation.

1.2 Research Question

The research question posed by this paper is as follows:

How can machine learning techniques be used to generate synthetic 5G network traffic? What ML techniques are most suitable for this task? The subquestions that hone in on the core of the question are as follows:

- *What are the existing ML-based methods for synthetic traffic generation?*
- *How do the methods compare in terms of fidelity and ease of integration?*

2 Contributions

The contributions of this paper are as follows.

2.1 Packet-Level Generation

As discussed in Section 1.1, most existing synthetic network data generation approaches either focus on aggregate flow statistics or generate entire packet traces using machine learning, aiming to reproduce statistical properties over time. Meanwhile, conventional traffic generators such as TRex produce streams of realistic packet headers but typically rely on flat or fixed inter-packet timing, lacking the nuanced variance and structural diversity seen in real network traffic. As a result, the direct generation of individual, high-fidelity packet headers that capture structural and temporal characteristics remains a largely unexplored research area, which we aim to address.

2.2 Inter-Packet Timing Modeling

This study introduces a new approach by explicitly modeling inter-packet timing dynamics within the generative process. PAC-GAN and TabularARGN are adapted to represent and synthesize relative time deltas between packets as part of the header generation task. By integrating timing as a core feature, the models can capture temporal characteristics essential for realistic and temporally consistent synthetic network traffic, a capability not addressed in previous network traffic generation studies.

2.3 Evaluation Framework

To comprehensively assess generative performance, we introduce and apply a robust evaluation framework that quantifies protocol validity, marginal distribution alignment, and joint distribution fidelity to comprehensively assess generative performance using established statistical metrics. This multi-faceted approach enables a holistic and quantitative analysis of how faithfully both individual field distributions and complex cross-field dependencies in the synthetic data align with those observed in real network traffic.

sectionMethodology

2.4 Dataset

The experiments leverage the full packet-capture dataset collected by Coldwell et al. [8]. This dataset consists of a variety of traffic collected on a simulation 5G network. The packets are captured at multiple interfaces placed throughout the simulation network. As shown in figure 2. For the purposes of this paper, only the tcp and ip-headers captured on interface *eno1* are utilized. At this point in the network pure ip-packets appear without any 5G (gprs) specific tunneling.

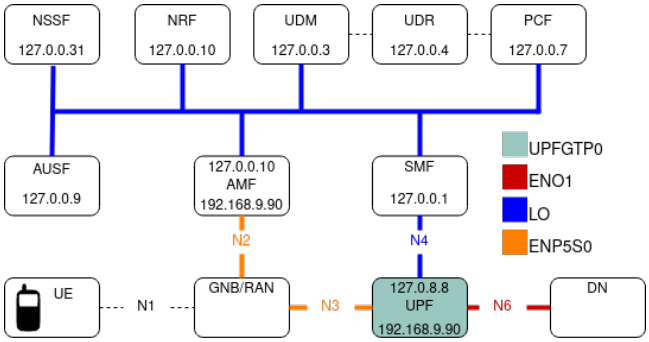

Figure 2: Simulation Network Diagram [8]

A 10% test and 90% training set split was utilized. Resulting in a test set of *267,928* rows and a train set of *2,411,323* rows

In the pre-processing stage, the raw packet captures are parsed using scapy [9], resulting in a full tabular representation of the packet headers. This tabular representation contains all ip and tcp fields as well as the timestamp associated with the packet and a derivative field, timedelta, which is computed as the time since the previous packet for every row. An analysis of mode prevelance is presented in Table 1. Based on these mode prevalences, several fields are identified to possess extremely low to no variance and are therefore candidates for omission from tabular generation and the evaluation stage. As highlighted in the table, these fields are: all *ip.flags*, *ip.tos*, *ip.version*, *ip.proto*, *ip.ihl*, *tcp.flags.ACK*, *-CWR*, *-URG*, *-ECE*, *tcp.dataofs*, *tcp.urgptr*, and *tcp.reserved*. Additionally, neither *ip.options* nor *tcp.options* are considered, as their variable length nature is incompatible with the architecture of the tabular generation approach. Checksums (*ip.chksum*, *tcp.chksum*) aren't considered since this paper only concerns headers and their computation inherently relies on full packet content. Lastly, absolute packet timestamps are not generated as they hold little relevance for this study; instead, the derivative field, *timedelta*, is utilized as it provides a more meaningful representation of inter-packet timing dynamics.

Field	Mode Value	Mode %	Non-Mode %
timestamp	2022-06-06 19:25:30	0.00%	100.00%
timedelta	0.00	9.17%	90.83%
ip.chksum	21112	0.16%	99.84%
ip.dst	192.168.70.215	46.39%	53.61%
ip.flags.DF	True	90.82%	9.18%
ip.flags.MF	False	100.00%	0.00%
ip.flags.RF	False	100.00%	0.00%
ip.frag	0	100.00%	0.00%
ip.id	0	1.66%	98.34%
ip.ihl	5	100.00%	0.00%
ip.len	52	29.32%	70.68%
ip.options	*[empty bytes]*	100.00%	0.00%
ip.proto	6	100.00%	0.00%
ip.src	192.168.70.215	53.60%	46.40%
ip.tos	0	91.17%	8.83%
ip.ttl	63	53.05%	46.95%
ip.version	4	100.00%	0.00%
tcp.ack	0	1.81%	98.19%
tcp.chksum	63545	0.18%	99.82%
tcp.dataofs	8	79.92%	20.08%
tcp.dport	443	53.32%	46.68%
tcp.flags.ACK	True	98.19%	1.81%
tcp.flags.CWR	False	100.00%	0.00%
tcp.flags.ECE	False	100.00%	0.00%
tcp.flags.FIN	False	99.01%	0.99%
tcp.flags.PSH	True	51.33%	48.67%
tcp.flags.RST	False	98.68%	1.32%
tcp.flags.SYN	False	98.98%	1.02%
tcp.flags.URG	False	100.00%	0.00%
tcp.options	*[empty bytes]*	18.52%	81.48%
tcp.reserved	0	100.00%	0.00%
tcp.seq	4013659489	0.08%	99.92%
tcp.sport	443	46.08%	53.92%
tcp.urgptr	0	100.00%	0.00%
tcp.window	502	16.69%	83.31%

Table 1: Mode Prevelance Over All Fields in the Dataset Spanning *2,679,251 Rows*

2.5 TabularARGN

In this work, we utilize the official TabularARGN implementation [5] to generate synthetic packet headers from tabular data representations, which are derived through pre-processing as described in section 2.4. The model operates by encoding all protocol fields as categorical sub-columns, where categorical features are consolidated by grouping rare values, numeric features are discretized into percentile bins, and datetime fields are decomposed into granular components. Each sub-column is mapped to an embedding vector, with dimensionality determined by feature cardinality. During training, TabularARGN randomly permutes feature order in each batch and applies permutation masks, enabling any-order conditioning and robust estimation of conditional distributions. Masked embeddings are processed through feed-forward regressor blocks. The training objective is the sum of categorical cross-entropy losses across all sub-columns, with teacher forcing, early stopping on a validation split, learning-rate decay, and checkpointing to ensure convergence. This approach guarantees protocol validity by construction, as only protocol-compliant values are sampled for each field. Extensions for sequential data and multi-table scenarios are supported via an LSTM-based history encoder and a feed-forward context processor, respectively, enabling conditional generation across related tables [5].

2.6 PAC-GAN

The PAC-GAN model is implemented following the original architecture described by Cheng et al. [1], with adaptations to focus on header and timing synthesis. Each packet is represented as a 48-byte vector, composed of an 8-byte nanosecond-precision *timedelta* and 40 bytes of concatenated ip and tcp headers. This vector is encoded into a 20×20 matrix by splitting bytes into nibbles, arranging them row-major, and duplicating each nibble in a 2×2 block, preserving sequential structure and enabling convolutional processing. As shown in figure 3: the generator receives a 64-dimensional Gaussian noise vector, which is upsampled via two dense layers of 1024 and 6400 units after which it is reshaped into a $5 \times 5 \times 256$ tensor. This is followed by two deconvolution layers (with 4×4 kernels each as well as 64 and 32 filter sizes, respectively). A final 3×3 convolutional layer with a filter-size of 1 and LeakyReLU activation produces the $20 \times 20 \times 1$ output matrix.

The discriminator mirrors this structure, as illustrated in figure 4. It accepts a $20 \times 20 \times 1$ input and processing it through two convolutional layers (using 4×4 kernels with 64 and 128 filter-sizes, respectively), followed by flattening and a dense layer of 256 units, and a final linear output.

[2]Created with PlotNeuralNet [10]
[3]Created with PlotNeuralNet [10]

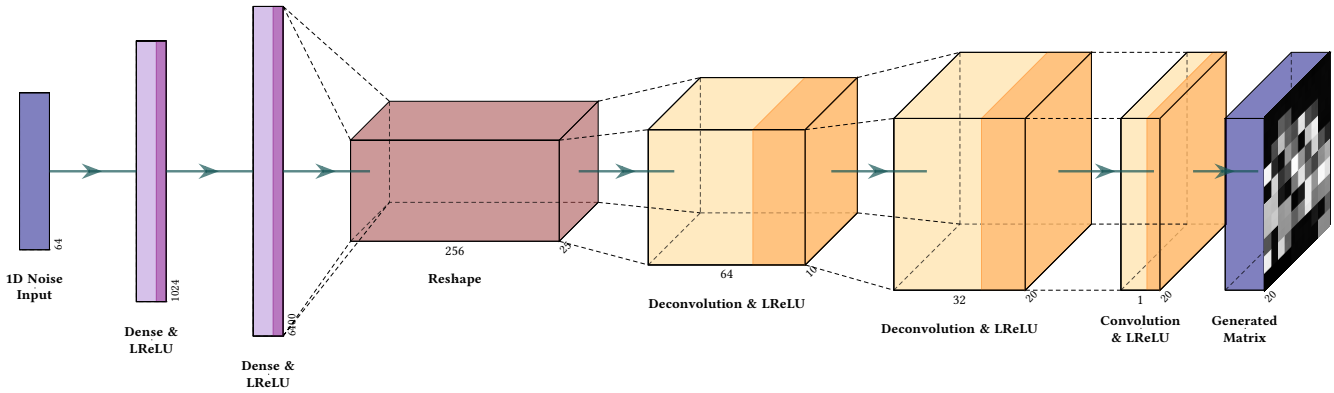

Figure 3: Generator Architecture[2]

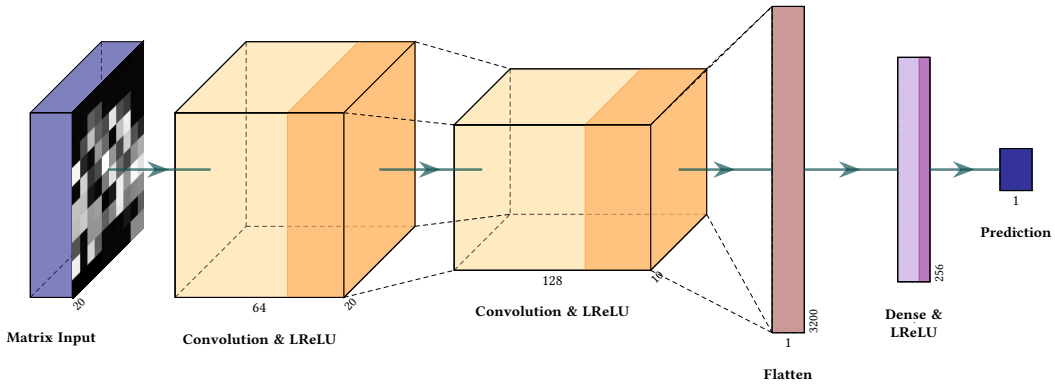

Figure 4: Discriminator Architecture[3]

LeakyReLU activations are applied after each layer except the final output. similarly, a stride of 2 is used for each convolutional or deconvolutional layer. The model is trained using the Wasserstein GAN loss with a gradient penalty coefficient of 10. Adam optimizers are used for both generator and discriminator, with a learning rate of 0.001 and $\beta_1 = 0.5$. Synthetic samples are decoded by reversing the nibble duplication and matrix packing, enabling direct evaluation of protocol validity and statistical fidelity.

3 Evaluation Strategy

A rigorous, quantitative evaluation strategy is adopted to assess the realism and utility of the synthetic packet header data generated by TabularARGN and PAC-GAN. The evaluation is designed to determine how closely the generated samples replicate the statistical, structural, and temporal properties of real 5G network traffic, thereby addressing the central research question: Can generative models produce synthetic packet headers that are statistically and functionally faithful to genuine data?

Both models are trained using the same dataset split and are tasked with generating synthetic samples equivalent in size to the test set. The evaluation process then compares these synthetic samples to the real data using a suite of metrics selected to capture key aspects of fidelity, diversity, and validity.

3.1 Categorical & Numeric Fields

For the purposes of evaluation and analysis, a distinction was made between categorical and numeric columns throughout the evaluation metrics. Categorical fields are those fields that are either inherently non-numeric, such as flags, or those where numeric distance holds no meaning, such as tcp ports. Resulting in the categorical fields: *ip.id*, *tcp.sport*, *tcp.dport*, *tcp.flags.FIN*, *-SYN*, *-RST*, *-PSH & -ACK*. And numeric fields: *ip.ttl*, *ip.len*, *tcp.window*, *timedelta*, *ip.src*, *ip.dst*, *tcp.seq & tcp.ack*. The non-obvious choice of treating ip addresses as numerical fields was made because their 32-bit unsigned integer representation preserves the logical structure and ordering of the address space, enabling the use of quantitative distance-based metrics to meaningfully capture distributional differences and similarities between real and synthetic samples.

3.2 Evaluation Metrics

A selection of evaluation metrics will be used to evaluate the performance of PAC-GAN compared to the TabularARGN model. The performance is evaluated in three major dimensions: **Validity**, **Marginal alignment**, and **joint distribution alignment**.

- **Validity** metrics provide an insight into how well the models can adapt to the requirements and constraints associated

with protocol formats. Providing an avenue through which the feasibility of real packet generation can be evaluated.
- **Marginal alignment** metrics describe how much the overarching field-wise distribution aligns among the real and generated data sets. Thus indicating the quality and fidelity of the packet generation process.
- **Joint distribution alignment** metrics evaluate the extent to which models capture higher-order dependencies spanning and intersecting multiple fields, characterizing the alignment of cross-field dependencies between real and test datasets.

3.2.1 Packet Validity metric. The packet validity is quantified by calculating the packet validity ratio. The ratio of generated packet headers adhering to protocol specifications:

$$\frac{\#\text{Valid\_Packets}}{\#Generated\_Packets}$$

It is computed by utilizing the scapy library [9] to parse generated packets, where all successfully parsed packets are considered valid. Prior to this computation, default values are overwritten for fields that aren't relevant for generation as discussed in Section **??**. The default values used are equivalent to the mode of each respective field.

This method can be applied directly for the PAC-GAN model as it generates byte-wise packet representations. The TabularARGN output is first concatenated into a raw byte-wise packet form from its field-wise tabular representation.

3.2.2 Marginal Distribution Metrics. We employ three complementary measures to evaluate per-field alignment between real and synthetic datasets, each capturing a different aspect of marginal similarity for both discrete and continuous features.

- **Univariate Similarity.** This metric was introduced by Tiwald et al. [5], who proposed computing the average L_1 similarity between real and synthetic histograms. Numeric fields are divided into deciles, and categorical fields are reduced to the ten most frequent categories. Scores close to 1 indicate that individual feature distributions are closely matched, while lower values highlight the existence of fields where the model exhibits mode collapse or distributional drift.
- **JSD for Categorical and EMD for Numeric Fields.** Following the approach of Schoen et al. [2], Jensen-Shannon Divergence (JSD) is used to assess the similarity between categorical distributions, while Earth Mover's Distance (EMD) is applied to numeric fields. JSD quantifies the divergence between real and synthetic categorical distributions by averaging their Kullback-Leibler divergences to a midpoint distribution:

$$\mathrm{JSD}(P \parallel Q) = \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M), \quad M = \frac{P+Q}{2} \tag{1}$$

For numeric features, EMD measures the "earth-moving" cost required to align real and synthetic distributions by integrating the absolute differences between their cumulative distribution functions:

$$\mathrm{EMD}(P, Q) = \int_{-\infty}^{\infty} |F_P(x) - F_Q(x)|\, dx \tag{2}$$

This combination allows for sensitive detection of both categorical distribution shifts and subtle numeric discrepancies, ensuring that both types of marginal statistics are faithfully preserved in the synthetic data.

3.2.3 Joint Distribution Metrics. To capture dependencies that span multiple header fields, both pairwise and manifold-based approaches are employed, following the approaches in the literature [2, 5, 11].

(1) **Bivariate Similarity.** Tiwald et al. [5] proposed constructing normalized contingency tables for every feature pair, using the same binning strategies as in the univariate metric, and computing the average L_1 similarity between real and synthetic tables. High scores indicate that the model faithfully preserves pairwise relationships, such as those between packet size and flag settings, while lower scores reveal unrealistic or implausible field combinations.
(2) **Manifold-Based Metrics.** Schoen et al. [2] adopt the metrics introduced by Naeem et al. [11] to perform the joint distribution analysis for their synthetic traffic traces analysis. A set of four manifold-based metrics that provide a thorough assessment of how synthetic samples inhabit the real-data feature space:
   - *Coverage* verifies that most real samples have at least one synthetic neighbor within a chosen radius, exposing mode dropping when coverage is low and confirming comprehensive span when high.
   - *Recall* measures the fraction of real samples covered by the synthetic manifold, reflecting the breadth of generative diversity. High recall denotes full representation of real traffic patterns.
   - *Density* measures how many real points fall within neighborhoods around synthetic samples, indicating whether generated points focus on dense, realistic regions or drift into sparse areas.
   - *Precision* calculates the fraction of synthetic points that lie within the real-data support, directly assessing the realism of generated packets; high precision means few outliers.

   Interpreting coverage alongside density distinguishes overdispersed generation (high coverage, low density) from mode collapse (high density, low coverage), while the balance between precision and recall reveals whether the model is conservative (high precision, low recall) or expansive (high recall, low precision) in its generation behavior.

4 Results and Discussion

The results of the experiment provide a comprehensive comparison between PAC-GAN and TabularARGN for the task of synthetic ip & tcp packet header generation. Both models were trained until their respective loss curves stabilized, as shown in Figures 5 & 6. For PAC-GAN, the generator and discriminator losses converged smoothly after approximately 150 epochs, with minimal divergence between training and validation curves. This stability indicates

that the adversarial training process was well-regularized, avoiding both underfitting and the notorious mode collapse or oscillatory behavior sometimes observed in GAN-based models [12]. Similarly, TabularARGN reached a plateau in training and validation loss within 20 epochs, demonstrating rapid convergence and suggesting that the model had fully captured the conditional dependencies present in the tabular representation of packet headers. This rapid convergence displayed by TabularARGN is one of the primary benefits touted by its authors and is demonstrably accurate, even though its generation performance may not match expectations. The absence of significant overfitting or loss spikes in both models further confirms that the chosen architectures and training regimes were well-suited to the data and task complexity.

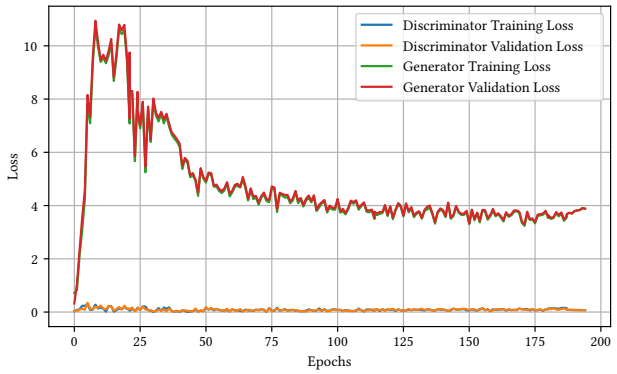

Figure 5: Loss Over Training Epochs for PAC-GAN

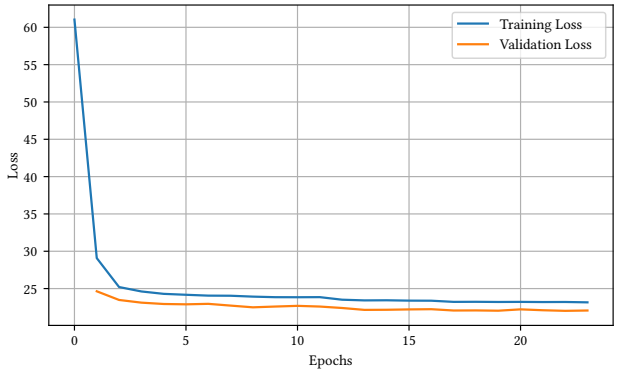

Figure 6: Loss Over Training Epochs for TabularARGN

Regarding the quantitative evaluation, Table 2 summarizes the key metrics. TabularARGN achieves perfect validity (100%), a direct consequence of its enforced data-type constraints and categorical encoding. By design, this model cannot generate packets with invalid field values or structurally inconsistent headers, as each feature is discretized and sampled from a set of protocol-compliant options. This property makes TabularARGN exceptionally robust for scenarios where strict protocol adherence is essential. In contrast, PAC-GAN attains a slightly lower validity score (96.89%), reflecting its byte-level generative process: while it learns the joint distribution of header fields through convolutional encodings, rare invalid byte combinations may still occur. Nevertheless, the high validity rate demonstrates that PAC-GAN's adversarial learning is highly effective at internalizing the structural rules of packet headers.

Metric	PAC-GAN	TabularARGN
Validity ↑	96.89%	100.00%
EMD ↓	0.0191	0.0121
JSD ↓	0.2281	0.2404
Univariate ↑	0.8215	0.7069
Bi-Variate ↑	0.6651	0.4814
Coverage ↑	0.1786	0.0015
Recall ↑	0.9843	0.9986
Density ↑	0.2316	0.0027
Precision ↑	0.5318	0.0073

Table 2: Evaluation Metrics Results

Beyond validity, the remaining metrics provide deeper insight into the generative fidelity of both models. PAC-GAN outperforms TabularARGN in univariate and bivariate similarity, indicating that PAC-GAN more accurately preserves both individual feature distributions and pairwise relationships, which is critical for generating headers that are plausible in isolation and structurally coherent as a whole. The convolutional architecture of PAC-GAN, which processes headers as spatial matrices, likely contributes to its superior modeling of inter-field dependencies—an advantage that is less pronounced in the permutation-invariant, column-wise training of TabularARGN.

Manifold-based metrics further distinguish the two approaches. PAC-GAN achieves substantially higher coverage and density, meaning its synthetic samples span a broader region of the real data manifold and cluster more closely around high-density areas. Precision and recall metrics reinforce this observation: PAC-GAN balances a high recall with a much higher precision than TabularARGN, indicating that it generates a diverse set of realistic headers without drifting into implausible or outlier regions. In contrast, TabularARGN's extremely low coverage, density, and precision suggest that while it samples exhaustively from valid field combinations, it fails to capture the joint structure and diversity of real packet distributions, resulting in over-dispersed and less realistic outputs.

Interestingly, TabularARGN's best comparative performance is in Earth Mover's Distance (EMD), with a lower score on continuous features. This is likely due to its binning and discretization of numeric fields, which smooths out the distribution and simplifies alignment with real data. However, this marginal advantage does not compensate for its deficiencies in joint and structural metrics, especially for applications where the interplay between multiple header fields is crucial.

In summary, these results highlight the importance of architectural inductive bias in generative modeling for structured data.

While TabularARGN's tabular approach guarantees protocol validity and is well-suited to applications prioritizing strict compliance, it lacks the capacity to model complex, higher-order dependencies. PAC-GAN, on the other hand, leverages convolutional encodings to achieve a more faithful reproduction of both the marginal and joint statistics of real 5G packet headers, producing synthetic data that is not only valid but also structurally and statistically realistic. This balance makes PAC-GAN a compelling choice for scenarios requiring both protocol adherence and high-fidelity traffic simulation.

5 Responsible Research

5.1 Ethical Implications

The generation of synthetic network traffic data raises significant ethical considerations regarding privacy preservation, data governance, and potential misuse. While the original dataset from Coldwell et al. was collected in a controlled simulation environment, the synthetic generation of realistic packet headers introduces privacy risks if applied to real network traces containing personally identifiable information or sensitive communication patterns. This approach mitigates these concerns by focusing exclusively on protocol-level header fields rather than payload content, generating valid and useful packets up to layer 4 of the OSI stack while significantly limiting the potential for exposing sensitive source data.

The generation of synthetic 5G network data holds significant promise for democratizing access to high-fidelity datasets, enabling researchers in resource-constrained institutions or regions to conduct rigorous analyses without relying on proprietary data. By providing synthetic alternatives that preserve statistical properties of real traffic, this approach aims to level the playing field for academic innovation in network optimization, security testing, and protocol development.

5.2 Reproducibility

Reproducibility represents a cornerstone of this research methodology, with all experimental parameters, model configurations, and evaluation metrics documented to enable independent verification of results. The TabularARGN implementation was utilized directly from its published GitHub repository without modification, ensuring that results reflect the model's intended performance characteristics rather than implementation-specific optimizations. For PAC-GAN, the model architecture was reconstructed following the original specifications by Cheng et al., with all hyperparameters and training procedures explicitly documented in the methodology section. The dataset preprocessing pipeline, including feature selection criteria and statistical transformations, has been thoroughly described to support independent replication.

6 Conclusion and Future Work

6.1 Conclusion

The results of this study demonstrate that PAC-GAN almost always outperforms TabularARGN in the generation of synthetic 5G packet headers. PAC-GAN achieves high validity (96.89%), strong marginal alignment, and superior joint distribution fidelity, making its outputs suitable for real-world applications that require realistic and protocol-compliant traffic. The convolutional encoding of packet fields in PAC-GAN enables robust modeling of inter-field dependencies, resulting in synthetic headers that closely resemble genuine 5G traffic patterns. Thus answering the research question: How can machine learning techniques be used to generate synthetic 5G network traffic? What ML techniques are most suitable for this task? PAC-GAN is the clear choice and is capable of generating synthetic 5G network traffic with very little modification.

6.2 Future Work

Several promising directions for future research are identified:

- **Explore More Models** Future work should explore the performance of additional generative models, such as PacketCGAN, PcapGAN, TVAE, CTGAN, and REaLTabFormerm, within the established evaluation framework. This would provide a more holistic perspective on the state-of-the-art in synthetic packet generation and help clarify the strengths and limitations of each approach.
- **Integration of Additional Protocol Headers** Extending the synthesis task to include 5G specific headers (such as GPRS tunneling and NAS messages) could yield valuable insights and enable the generation of complete 5G packets. This would require an adaptation of the evaluation metrics to accommodate the multi-layered nature of such packets. However, the current PAC-GAN model, with its byte-level representation, is well-suited to handle these extensions, potentially enabling the generation of complete 5G packets, including GPRS headers.
- **Privacy-Preserving Generation:** Investigating privacy-preserving mechanisms, such as differential privacy, within generative models would ensure that synthetic traces do not leak sensitive information from real network captures, supporting ethical data sharing and regulatory compliance.

By pursuing these directions, future work can further establish synthetic traffic generation as a reliable tool for network research, benchmarking, and security testing in the evolving 5G landscape.

References

[1] Adriel Cheng. 2019. PAC-GAN: Packet Generation of Network Traffic using Generative Adversarial Networks. In *2019 IEEE 10th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*. 0728–0734. doi:10.1109/IEMCON.2019.8936224 ISSN: 2644-3163.

[2] Adrien Schoen, Gregory Blanc, Pierre-François Gimenez, Yufei Han, Frédéric Majorczyk, and Ludovic Me. 2024. A Tale of Two Methods: Unveiling the Limitations of GAN and the Rise of Bayesian Networks for Synthetic Network Traffic Generation. In *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*. 273–286. doi:10.1109/EuroSPW61312.2024.00036 ISSN: 2768-0657.

[3] Pan Wang, Shuhang Li, Feng Ye, Zixuan Wang, and Moxuan Zhang. 2019. PacketCGAN: Exploratory Study of Class Imbalance for Encrypted Traffic Classification Using CGAN. doi:10.48550/arXiv.1911.12046 arXiv:1911.12046 [cs].

[4] Baik Dowoo, Yujin Jung, and Changhee Choi. 2019. PcapGAN: Packet Capture File Generator by Style-Based Generative Adversarial Networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. 1149–1154. doi:10.1109/ICMLA.2019.00191

[5] Paul Tiwald, Ivona Krchova, Andrey Sidorenko, Mariana Vargas Vieyra, Mario Scriminaci, and Michael Platzer. 2025. TabularARGN: A Flexible and Efficient Auto-Regressive Framework for Generating High-Fidelity Synthetic Data. doi:10.48550/arXiv.2501.12012 arXiv:2501.12012 [cs].

[6] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling Tabular data using Conditional GAN. doi:10.48550/arXiv.1907.00503 arXiv:1907.00503 [cs].

[7] Aivin V. Solatorio and Olivier Dupriez. 2023. REaLTabFormer: Generating Realistic Relational and Tabular Data using Transformers. doi:10.48550/arXiv.2302.02041 arXiv:2302.02041 [cs].

[8] Cooper Coldwell, Denver Conger, Edward Goodell, Brendan Jacobson, Bryton Petersen, Damon Spencer, Matthew Anderson, and Matthew Sgambati. 2022. Machine Learning 5G Attack Detection in Programmable Logic. In *2022 IEEE Globecom Workshops (GC Wkshps)*. 1365–1370. doi:10.11578/dc.20220811.1

[9] Philippe Biondi and the Scapy community. 2022. *Scapy*. https://scapy.net/

[10] Haris Iqbal. 2018. HarisIqbal88/PlotNeuralNet v1.0.0. doi:10.5281/zenodo.2526396

[11] Muhammad Ferjad Naeem, Seong Joon Oh, Youngjung Uh, Yunjey Choi, and Jaejun Yoo. 2020. Reliable Fidelity and Diversity Metrics for Generative Models. doi:10.48550/arXiv.2002.09797 arXiv:2002.09797 [cs].

[12] Youssef Kossale, Mohammed Airaj, and Aziz Darouichi. 2022. Mode Collapse in Generative Adversarial Networks: An Overview. In *2022 8th International Conference on Optimization and Applications (ICOA)*. 1–6. doi:10.1109/ICOA55659.2022.9934291 ISSN: 2768-6388.