

A Generative Approach for Production-Aware Industrial Network Traffic Modeling

Alessandro Lieto*, Qi Liao*, Christian Bauer†

*Nokia Bell Labs, Stuttgart, Germany

†Trumpf, Ditzingen, Germany

E-Mails: alessandro.lieto, qi.liao@nokia-bell-labs.com, christian.bauer@trumpf.com

Abstract—The new wave of digitization induced by Industry 4.0 calls for ubiquitous and reliable connectivity to perform and automate industrial operations. 5G networks can afford the extreme requirements of heterogeneous vertical applications, but the lack of real data and realistic traffic statistics poses many challenges for the optimization and configuration of the network for industrial environments. In this paper, we investigate the network traffic data generated from a laser cutting machine deployed in a Trumpf factory in Germany. We analyze the traffic statistics, capture the dependencies between the internal states of the machine, and model the network traffic as a production state dependent stochastic process. The two-step model is proposed as follows: first, we model the production process as a multi-state semi Markov process, then we learn the conditional distributions of the production state dependent packet interarrival time and packet size with generative models. We compare the performance of various generative models including variational autoencoder (VAE), conditional variational autoencoder (CVAE), and generative adversarial network (GAN). The numerical results show a good approximation of the traffic arrival statistics depending on the production state. Among all generative models, CVAE provides in general the best performance in terms of the smallest Kullback-Leibler divergence.

I. INTRODUCTION

The industrial network infrastructures are facing a radical transformation to accommodate the communication needs of today’s manufacturing technologies, Industrial Internet of Things (IIoT) applications and flexible shopfloor designs. The current industrial networks are designed for static configuration and specific applications, and they are, therefore, insufficient to cover the future needs posed by Industry 4.0 [1]. 5G systems and beyond are a natural fit for serving the evolving needs of future industrial environments, by providing the dynamicity and flexibility that is expected to be supported in the near future [2]. However, there is a lack of knowledge about the characteristics of industrial systems, and the typical configuration of cellular systems tailored to consumer applications is not able to meet the demanding needs of industrial applications. Especially for network planning, it is critical to know and quantify the amount of data to be transmitted, the statistics of data packets, the endpoints of the communication and many other details that require the direct observation from brownfield installations [3].

In general, the existing literature presents simplified findings from measurements performed over non-industrial networks or semi-artificial system setups. For example, there is a lot of literature focusing on the analysis and design of traffic in

Supervisory Control and Data Acquisition (SCADA) networks, including datasets [4] open to access for researchers. Among others, [5] analyzes the traffic patterns and models the interarrival time and the correlation between packet types generated in SCADA networks, limiting their observation to the industrial protocols applied therein. However, their evaluation is based on emulated traffic in test labs and not exemplary of realistic applications running on devices in a working industrial environment, i.e. not being able to address the complexity of the real industrial systems [3]. Indeed, the validation of results with realistic assumptions is very challenging for researchers in this area due to the unavailability of both production and network traffic data from industrial systems. Some works, such as [3], [6], analyze the differences between available theoretical and empirical traffic models with real brownfield installations, showing that there is a significant gap in the literature, especially for traffic estimation and generation. They provide some interesting findings, but do not share data or models that can be used to guide the research in this field. While [3] mainly focuses on the methodology for best practice in traffic estimation and modeling, the authors in [6] provide an empirical comparison between the well-known analytical models suggested from standardization bodies, such as 3rd generation partnership project (3GPP) and the 5G-Alliance for Connected Industries and Automation (5G-ACIA) [7], [8], and real data traffic collected from various factories. Their analysis focuses on the statistics of interarrival packet time, packet size and burstiness, showing that the real traffic is significantly more heterogeneous than the suggested models from 3GPP, with only limited match concerning the periodic traffic, and with high over-estimation with respect to the non-periodic traffic. Along the same lines, the methodology proposed in [9] considers the whole complexity of an industrial systems, when collecting all the aggregated traffic in an industrial network. Although this approach allows for higher complexity and gives insights for the design of a communication systems considering the overall network data traffic, it does not provide enough elements for repeatability. This is the most important point that we address in this paper. The contributions of the paper can be summarized as follows:

- First, we analyze the peculiar behavior of some industrial devices from measurements collected in a Trumpf factory. Based on the operations of the machines, we can identify

specific production dependent states of the machines and model their transitions as a semi-Markov process.

- Then, we apply generative models for reproducing the statistics of data packets, focusing in particular on the interarrival time and its dependency on packet size.
- Finally, we compare different models that can be used for reproducing the traffic pattern of the machine for each of its production state. We also grant access to those models and share them publicly with the research community for validation and testing in various 5G use cases, e.g., to model a digital twin of an industrial network [10]. Although our models cannot generalize for any industrial environment, it still provides a reference point for realistic industrial installations and can be used to generate more realistic instances for e.g. industrial network simulators [11].

The rest of the paper is structured as follows. In Section II, we describe the datasets and the characteristics of the traffic flows captured from laser cutting machines in a Trumpf factory. The methodology for the generative models is discussed in Section III, while in Section IV we illustrate our findings and analyze the performance of the applied models. Finally, Section V concludes the paper with a summary of the findings and discussion for the applications of the obtained models in industrial network setups.

II. DATA COLLECTION AND DATASET DESCRIPTION

We consider an exemplary industrial setup of a Trumpf factory where a laser cutting machine is deployed in a shop floor and communicates with a central management system over an industrial fixed network. All the information that have been exchanged between the machine and the management system are collected in a log file in .csv format, which represents the dataset of our evaluation. We filter the original dataset to capture only the entries needed for analyzing the statistical properties of the traffic generated by the industrial machine and to identify the specific characteristics of the machines, with its life-cycle, operational duty-cycle and status information to correlate the statistics of the traffic to the production state of the machine. The filtered dataset contains the following list of entries.

- *Processed Time* identifies the date and time of the day when a specific message has been sent by the machine. The time granularity is of millisecond scale, which is appropriate for the estimation of packet interarrival time.
- *Data.id* specifies the type of message sent by the machine. It is an ID used by the owner of the machine to recognize the messages that have been exchanged. We can use this information to identify control messages that are sent by the machine to inform about its current operational state.
- *Data.value* represents the content of the message. The content of the message is out of the scope of our evaluation, however some information can be useful to identify specific status information of the machines. In particular, we extract together with the *Data.id* the information

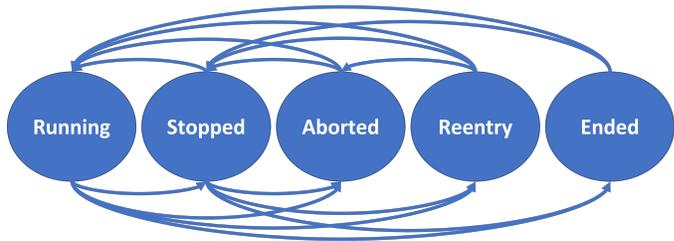


Fig. 1: A Markov chain of state transition of the machine

	<i>Running</i>	<i>Reentry</i>	<i>Stopped</i>	<i>Aborted</i>	<i>Ended</i>
<i>Running</i>	0	4	296	17	151
<i>Reentry</i>	51	0	36	21	0
<i>Stopped</i>	198	52	0	2	15
<i>Aborted</i>	9	0	31	0	0
<i>Ended</i>	63	0	103	0	0

TABLE I: The number of the total transitions among states

that identifies what is the status of the machine. An example of how to use such information is described in the following subsection.

- *Data.payload* defines the size of each packet that has been transmitted by the device. The computation of the payload is rather simple, since we apply binary conversion to the content of *Data.value* and we extract the dimensions in bytes. Although this calculation does not represent exactly the packet size over the network interface (since header and encryption are not considered in the calculation), it still provides coarse estimation of the amount of data to be transmitted. The payload has been then quantized in multiple of 32 bytes to align with the typical minimum packet size of 3GPP standards.

A. Production state transition of the laser cutting machine

As mentioned in the previous subsection, it is possible to derive the specific operational state of the machine from the available dataset. Thus, we aim to extract the production state of the machine, and, based on the state, to infer the statistical property of the network traffic (in terms of traffic pattern and distribution). From the dataset, we could retrieve a subset of meaningful states to be used for evaluation. We will refer to these states with the following terminology: [*Running*, *Reentry*, *Stopped*, *Aborted*, *Ended*]. The machine visits each of these states with a certain frequency, remains in each state for an interval of time, the so-called *sojourn time*, and then transitions to a new state. For this reason, we can represent the state transition of the machine as a Markov chain. An example is given in Fig. 1. As one can see from the figure, when the machine is in a *Running* state, it can move to any of the other states, but this does not apply for all the states. For example, from an *Aborted* state, it is only possible to transit to *Running* and *Stopped*, but not to the other two states. To give a more comprehensive understanding of those transitions, we summarize in Table I the number of total transitions according to the collected dataset for any of those states. From the available dataset, the most visited states are

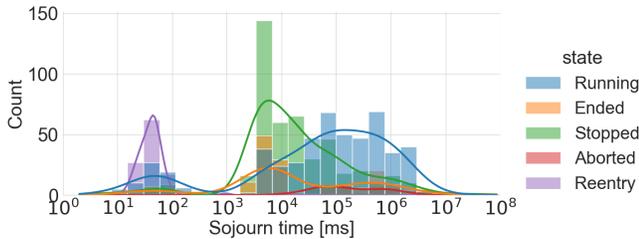


Fig. 2: The distribution of sojourn time of the machine per state

the *Running* state and the *Stopped* state, since the machine most likely meets a stopping criterion after completing some task when running. While from Table I one can retrieve the probability of transition among the states, another important aspect is to determine how long the machine stays in that state. This result is plotted in Fig. 2, where the *sojourn time* is shown in a logarithmic scale for readability purpose. Moreover, the logarithmic scale also helps understand the order of magnitude of such scale. As one can notice from the figure, the range of values can be very large, from few milliseconds up to minutes or even hours. Especially, the *Reentry* state differs from other states, with rather a short sojourn time, while the *Running* state is the one which lasts a longer time. By merging the transition state probability of Table I together with the sojourn time probability distribution of Fig. 2, one can empirically derive the continuous-time Markov chain to model the operational duty-cycle of the machine.

III. MODELING THE PRODUCTION-AWARE NETWORK TRAFFIC PROCESS

To infer the network traffic process depending on the production status, first we model the production process as a semi-Markov process (SMP), then obtain the production state dependent network traffic statistics, including the distribution of packet interarrival time and its correlation with packet size.

A. Modeling the Production Process

As shown in Fig. 1, the production process has five states, *Running*, *Reentry*, *Stopped*, *Aborted* and *Ended*, denoted by 1, 2, 3, 4, 5, respectively. A classical method to model such a multi-state machine is Markov process, which assumes that the sojourn time for each state follows the exponential distribution. However, in real-world applications, this assumption usually does not hold strictly. Thus, we consider to model the multi-state process of the machine as an SMP [12], which relaxes the exponential assumption and allows for random sojourn time.

To define an SMP for the production process, we consider a stochastic process $\{s_t\}_{t \geq 0}$, where $s_t \in \mathcal{S} := \{1, 2, 3, 4, 5\}$ denotes the state at time t .

- The *transition probability* from state $i \in \mathcal{S}$ to state $j \in \mathcal{S}$ is denoted as p_{ij} , i.e., whenever the process is in state $i \geq 0$, the probability to enter in state j is p_{ij} . The transition matrix is denoted as $\mathbf{P} = [p_{ij}]$ with zero diagonal entries. Note that this definition of transition

matrix does not include the information about the time of transition.

- The *probability density function of jumping time*, i.e., the time needed before jumping from state i to state j follows a distribution denoted by $F_{ij}(t)$.
- The *distribution of sojourn time*, i.e., the time that the process spends in state i before making a transition conditioned on the next state, is given by

$$H_i(t) = \sum_{j \in \mathcal{S}} p_{ij} F_{ij}(t). \quad (1)$$

From the collected data, we can extract the statistics of the transition probability (e.g., from Table I) and the distribution of the jumping and sojourn time (e.g., as shown in Fig. 2).

B. Modeling the Production Dependent Network Traffic

Given a production state, we aim to model the network traffic process conditioned to a given state by approximating the distributions of packet interarrival time. The state-of-the-art works often assume that the packet arrival follows a Poisson process, composing a limitation that the interarrival times are exponentially distributed with mean $1/\lambda$, where λ is the arrival rate [13]. However, a number of studies have shown that for both local-area and wide-area network traffic, the distribution of packet arrival clearly differs from exponential [14]. In align with these works, we examine the traffic modeling empirically for the industrial campus network, and observe that the distribution vastly differs from Poisson process. In fact, the traffic bursts appear on a very wide range of time scales. Thus, we propose to learn the distributions of traffic arrivals with deep generative methods, such that the pre-trained models can be loaded to automatically generate production state dependent network traffic in the industrial environments.

1) *Three Generative Models to Compare*: We consider the following three generative models to compare for the industrial traffic generation, as shown in Fig. 3.

Variational autoencoder (VAE) [15]: In VAE the input data \mathbf{x} is sampled from a parametrized distribution (a.k.a., the prior). The encoder maps \mathbf{x} to \mathbf{z} in a latent space, and the encoder and decoder are jointly trained to minimize a reconstruction error between the reconstructed \mathbf{x}' and the real input \mathbf{x} in terms of Kullback–Leibler (KL) divergence [16] between the parametric posterior and the true posterior.

Conditional variational autoencoder (CVAE) [17]: CVAE was developed based on the VAE architecture, with the only difference that CVAE inserts label information c in the latent space to force a deterministic constrained representation (conditioned by c) of the learned data. In our model, the label information c is the encoded categorical production state, e.g., by encoding the production state s_t with one hot encoding.

Generative adversarial network (GAN) [18]: The most popular generative model GAN simultaneously trains a generative model and a discriminative model. The generative model synthesizes samples \mathbf{x}' from the latent variables \mathbf{z} , while the discriminative model differentiates between the real sample \mathbf{x} and the synthesized sample \mathbf{x}' and returns a binary output y to classify both the real data and the fake data from the generator.

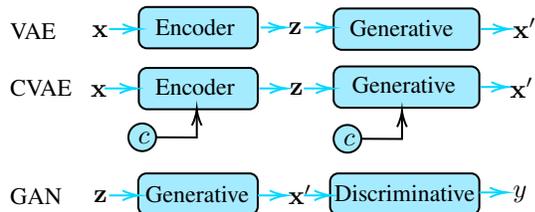


Fig. 3: Comparison among three generative models, where x and x' are the input and reconstructed/generated samples respectively, z is the latent vector, y is a binary output which represents whether it is a real or synthesized sample, and c is the condition of the production state.

2) *Two Traffic Generation Schemes*: Given a production state $S \in \mathcal{S}$, the objective is to generate the industrial network traffic based on the conditional distribution of packet interarrival time T and packet size L . We compare the following two schemes.

- **Independent Learning**: Assuming that the packet interarrival time and the packet size are independent, for each state $S \in \mathcal{S}$, we have

$$\Pr((T, L)|s_t = S) = \Pr(T|s_t = S) \cdot \Pr(L|s_t = S).$$

Thus, we can learn $\Pr(T|s_t = S)$ independently from $\Pr(L|s_t = S)$ with the previously introduced generative models. This means, for each state S , we train a single model for the distribution of packet interarrival time.

- **Joint Learning**: For each state $S \in \mathcal{S}$, we learn a joint two-dimensional distribution $\Pr((T, L)|s_t = S)$. Such model captures the correlation between packet arrival time and packet size, if the correlation exists. This allows us to train one model under each production state, while the model has higher dimension of the input (taking both T and L into account), compared with the independent learning scheme.

C. Workflow of Generating Industrial Network Traffic

After learning the production process as an SMP, as described in Section III-A, and pretraining generative models for the distributions of production state dependent traffic interarrival time, we can generate the industrial traffic following the steps provided in Algorithm 1.

Algorithm 1 Generating the industrial network traffic

Require: Inferred \mathbf{P} , $\{F_{ij}\}$, and $\Pr((T, L)|S)$, $S \in \mathcal{S}$

Require: Initial index of jump $n = 0$; initial state $s_n = 1$

Require: Maximum number of state jumps $N^{(\max)}$

- 1: **while** $n \leq N^{(\max)}$ **do**
 - 2: Generate next state s_{n+1} based on \mathbf{P}
 - 3: Generate jumping time based on $F_{s_n, s_{n+1}}$
 - 4: Before jump to s_{n+1} , generate T and L based on the pretrained generative models of $\Pr((T, L)|s_n)$.
 - 5: $n \leftarrow n + 1$ when it is the time to jump to state s_{n+1}
 - 6: **end while**
-

IV. PERFORMANCE EVALUATION

In this section we show the performance of the generative models discussed in Section III applied to the dataset illustrated in Section II. In particular, we consider two setups for the generation of the traffic:

- 1) We use VAE and GAN architectures to learn traffic distribution for each production state of the machine independently and compare it with the empirical distribution of the real data. This results in multiple per-state models, which is helpful for the evaluation of the behavior of the industrial environment in the specific states.
- 2) We use CVAE architecture to learn the state-dependent traffic distribution jointly for all states by adding the production state as a priori information. This leads to a single model that generates the distribution during the whole operational duty-cycle of the machine.

The two approaches give flexibility to apply the generative models in different contexts, depending on the need. In particular, we apply i) the GAN and VAE on a per-state dataset, where each dataset is extracted by the original dataset and filtered according to the production state of the machine. In contrast, for CVAE we consider ii) the production state of the machine as a priori information given to the neural networks as the label condition c . The main difference between the two approaches is that in i) we obtain multiple models, one for each production state, while in ii) we have a single model that can generate the per-state traffic patterns for any production state. For all three generative models, we consider the same neural network architecture and hyper-parameter tuning. The models of encoder and decoder for VAE and CVAE, as well as generator and discriminator for the GAN, are all built up with 5 hidden layers, with the number of neurons (32, 64, 32, 16, 32). In all schemes we adopt a learning rate of 0.001, and the loss functions are built upon the binary cross entropy, with the addition of the Kullback-Leibler (KL) loss for the VAE and CVAE. The latent dimensions for interarrival time distribution only and for the joint distribution of interarrival time and packet size are 2 and 4, respectively. The training is performed on the 70% of the whole dataset, which is shuffled and normalized after being rescaled on a logarithmic scale, with batch size of 32, for a total of 500 epochs. The numbers of samples for each state are not homogeneous, resulting in a total of ca. 15000 of samples for *Running* state and only ca. 500 for *Aborted* and *Reentry* states. For such reason, the testing is performed on variable number of samples for each state, but with a sufficient number to draw meaningful distributions.

Below, we show the experimental results when generating

- the distribution of interarrival time of packets only;
- the joint distribution of interarrival time and packet size.

A. Distribution of interarrival time of packets

One of the most important features for traffic estimation and generation is the interarrival time between packets. This parameter refers to the interval time between two consecutive

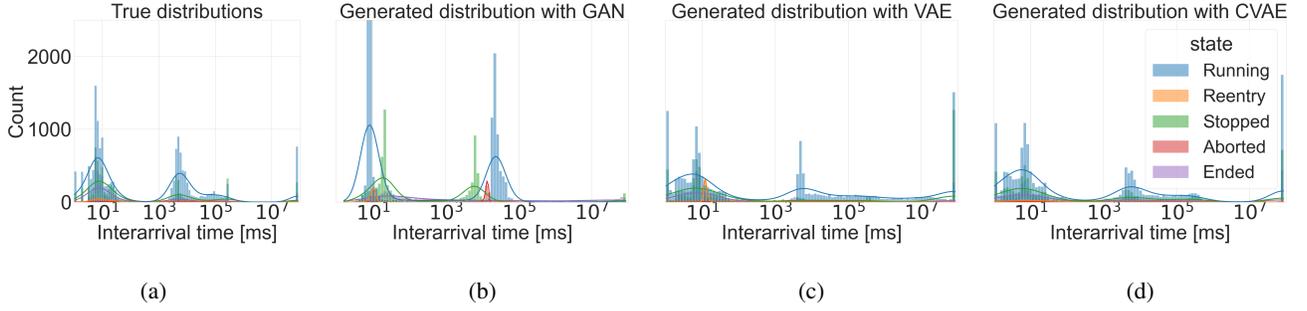


Fig. 4: Comparison of distributions for all states

transmissions and it is typically varying depending on the specific applications generating the data traffic. We focus on the interarrival time since it is also a fundamental unit of measure for critical applications, where the network must ensure the correct delivery of the packets in the range of few milliseconds. Therefore, understanding the interarrival time of packets gives an important information for scheduling operations and how to handle priorities among packets. We analyze the different characteristics of the interarrival time for each state, as defined in Section II, and compare the generated distributions with the empirical distribution computed from the original samples. Namely, we compare the original distribution in Fig. 4(a) with the ones generated with GAN, VAE and CVAE in Figs. 4(b), 4(c) and 4(d), respectively. In particular, the shaded bars show the histograms obtained from the distributions, while the solid lines show the approximation of the continuous distributions generated by the seaborn modules of *histplot*. For each figure, we compare the generated distribution with the empirical distribution to analyze the properties and similarities between the plotted samples. Moreover, the x-axis is plotted in logarithmic scale for the ease of visualization, since the range of the samples is quite large, from few milliseconds till minutes of interarrival time. However, the logarithmic scale gives a good understanding of the order of magnitude, showing that, for example, in the *Running* state it is possible to identify three main peaks, the first in the order of tens of milliseconds, the second in the order of tens of seconds, while the third one in the order of minutes. When comparing the results for each state, one can appreciate that VAE and CVAE generate, in general, distributions very close to the real data. In contrast, GAN seems unable to intercept all the peaks of the original distributions, in particular when observing the samples of interarrival times in the order of minutes. By looking at individual states, one can also notice how some states cannot be retrieved properly, both by GAN and VAE, due to the limited amount of data for the training. This is particularly true for the *Aborted* and *Reentry* states, which are the states less visited by the machine and, therefore, produce limited amount of samples. However, CVAE also gives a good approximation for those states, resulting in good accuracy for each state. To give a more comprehensive comparison among the proposed models, we also compare the KL divergence for each state in Table II. The values considering the interarrival

time only are referred to as VAE 1D, CVAE 1D, GAN 1D. As already shown from the plotted distributions, the CVAE obtains in general better results, providing similar values for each state, differently from VAE and GAN. However, VAE has comparable (if not better) performance for most of the states with enough samples for the training. Finally, the CVAE is able to generalize in a single model the distributions for all states, given also the high similarity of those distributions among some states.

B. Joint distribution of interarrival time and packet size

In this section, we analyze the joint distribution of interarrival time and packet size by comparing the results of the 2D distributions generated by the three models with the empirical distribution. For the sake of space and clarity, we only show a 3D plot of the 2D distribution for the *Running* state in Fig. 5, while referring at the KL divergence values in Table II, computed for the interarrival time distribution only, for generic comparison and discussion. The plots in the figure show the bars in the different colors to further highlight the difference in number of counts in the obtained histogram. We apply the logarithmic scale at both x-axis and y-axis for the interarrival time (in ms) and packet size (in bytes), respectively. One can notice that most of the packets for *Running* state are transmitted with short interarrival time (of few milliseconds) and have small packet sizes, while only few transmissions are established with large packet sizes. As for the 1D distribution, the VAE and CVAE outperforms GAN, showing more similarities in the distribution, confirmed by the lower values in the KL divergence in Table II (comparing the values of VAE 2D, CVAE 2D, GAN 2D). By comparing the values in Table II, one can deduce that interarrival time and packet size are not strongly correlated, given that the performance of the joint distribution do not outperform the one obtained with the interarrival time only. We remark that the scope of this paper is to give insights about the traffic pattern in industrial network and to share the generative models discussed for reproducing the realistic environments. The provided generative models can be used by the entire research community to build more realistic simulators and emulators for the industrial 5G use cases. The pre-trained model can be found in [10].

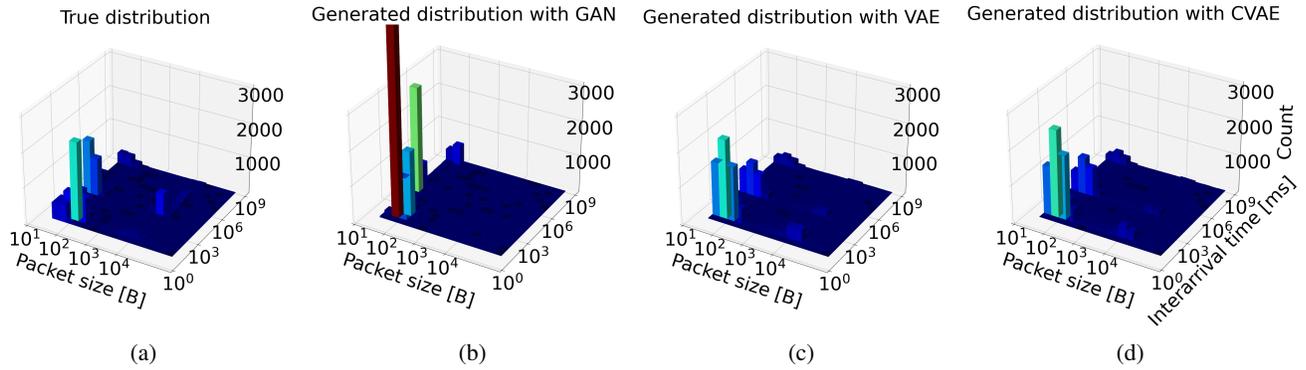


Fig. 5: Comparison of the joint 2D distribution for the *Running* state

	<i>Running</i>	<i>Reentry</i>	<i>Stopped</i>	<i>Aborted</i>	<i>Ended</i>
VAE 1D	0.40	4.27	0.52	0.50	0.34
CVAE 1D	0.26	0.38	0.34	0.47	0.36
GAN 1D	2.54	2.25	1.23	4.43	1.33
VAE 2D	0.33	0.35	0.49	0.78	0.35
CVAE 2D	0.28	0.95	0.36	0.69	0.38
GAN 2D	1.02	2.28	2.08	4.43	1.74

TABLE II: Model comparison based on KL divergence

V. CONCLUSION

In this paper we analyze the traffic patterns of a laser cutting machine deployed in a Trumpf factory. From the collected dataset, we can identify the different production states of the machine and extract the production state dependent statistics of the industrial network traffic. Specifically, we show that the production state process can be represented as a semi-Markov process, by modeling the transition probabilities and the sojourn time for each state. Moreover, we propose different architectures for generating traffic models that can mimic the statistics of the production dataset. VAE and GAN are considered for generating per-state traffic models, while CVAE is considered as a generic model to generate the traffic patterns for every state, by adding the conditional information of the state to the generator. We compare the performance of the different architectures and observe that CVAE reproduces samples closer to the empirical distribution computed from real data when considering only the statistics of the interarrival time, while VAE outperforms the other models for the joint distribution of interarrival time and packet size. The obtained models are also made publicly available to allow researchers to reproduce more realistic industrial environments for their research activities. The integration of those models in an industrial network simulator will be part of the future works.

ACKNOWLEDGMENT

This work was supported by the German Federal Ministry of Education and Research (BMBF) project KICK.

REFERENCES

[1] S. K. Rao and R. Prasad, "Impact of 5g technologies on industry 4.0," *Wireless personal communications*, vol. 100, no. 1, pp. 145–159, 2018.
[2] I. Malanchini *et al.*, "Leveraging machine learning for industrial wireless communications," *arXiv preprint arXiv:2105.02051*, 2021.

[3] M. Lavassani, J. Akerberg, and M. Björkman, "From brown-field to future industrial networks, a case study," *Applied Sciences*, vol. 11, p. 3231, 04 2021.
[4] NETRESEC, "Capture files from 4SICS Geek Lounge," Last visit 2022-06. [Online]. Available: <https://www.netresec.com/?page=PCAP4SICS>
[5] C.-Y. Lin and S. Nadjm-Tehrani, "Understanding IEC-60870-5-104 traffic patterns in SCADA networks," in *Proceedings of the 4th ACM Workshop on Cyber-Physical System Security*, 2018, p. 51–60.
[6] R. S. Mogensen, I. Rodriguez, G. Berardinelli, G. Pocovi, and T. Kolding, "Empirical IIoT Data Traffic Analysis and Comparison to 3GPP 5G Models," in *IEEE 94th Vehicular Technology Conference (VTC2021-Fall)*, 2021.
[7] 3GPP, "Service requirements for cyber-physical control applications in vertical domains, Release 17," 3rd Generation Partnership Project (3GPP), Technical Specification Group Services and Systems Aspects 22.104, 2020, version 17.4.0.
[8] 5G-ACIA, "A 5G traffic model for industrial use cases," 5G Alliance for Connected Industries and Automation, Tech. Rep., Nov. 2019.
[9] M. Lavassani, J. Åkerberg, and M. Björkman, "Modeling and profiling of aggregated industrial network traffic," *Applied Sciences*, vol. 12, no. 2, 2022.
[10] A. Lieto, Q. Liao, and C. Bauer, "Generative models for production-aware industrial network traffic modeling." [Online]. Available: <https://github.com/liaoq-blcr/genIndustrialNetTraffic>
[11] "KICK Project." [Online]. Available: <https://kick-project.de/en/>
[12] S. M. Ross, J. J. Kelly, R. J. Sullivan, W. J. Perry, D. Mercer, R. M. Davis, T. D. Washburn, E. V. Sager, J. B. Boyce, and V. L. Bristow, *Stochastic processes*. Wiley New York, 1996, vol. 2.
[13] V. S. Frost and B. Melamed, "Traffic modeling for telecommunications networks," *IEEE Communications Magazine*, vol. 32, no. 3, pp. 70–81, 1994.
[14] V. Paxson and S. Floyd, "Wide area traffic: the failure of poisson modeling," *IEEE/ACM Transactions on networking*, vol. 3, no. 3, pp. 226–244, 1995.
[15] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[16] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79 – 86, 1951.
[17] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Advances in neural information processing systems*, vol. 28, 2015.
[18] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.