Sequential Ice Hockey Events Generation using Generative Adversarial Network

Md Fahim Sikder

IDA, Linköping University, Sweden
md.fahim.sikder@liu.se

Abstract. We have generated events and coordinates that lead to a hockey goal in this project. Swedish Hockey League data from season 2020-21 for twenty matches provided by Sportlogiq were used that contain event data of those matches. We used TimeGAN to generate event data. TimeGAN learns the distribution of the original time-series data and creates a synthetic version of it. After that, we showed which events led to a goal from the synthetic data. We used principal component analysis (PCA) plots to show the original and synthetic data distributions as a qualitative evaluation. Also, we have used a sequence prediction model to test the synthetic data quantitatively. We compared the synthetic data quality with another two GAN models.

Keywords: Ice Hockey · Coordinates Generation · Machine Learning · Time Series Generation · Event Generation · TimeGAN

1 Introduction

It is always a point of interest for a hockey team to know or learn what steps they can take that will lead to a particular outcome. The outcome can be a goal, successful zone entry, or analyze the whole game situation. Determining player performance, if they are playing up to their potential, and fixing some gaps in the strategies all can be possible by analyzing game event data. It is also possible to predict the match's outcome by going through the past events data. These events data can be tabular in format or can be images or videos. Different techniques have been used to extract meaningful features depending on the data. Such as: classifying puck possession events using computer vision techniques [7], predicting player actions [5], even creating risk prediction model that can identify if the player has concussion or not [3].

In this project, we deal with tabular data that is time-series in nature. We intend to use this tabular time-series data and generate a series of events that lead to a particular outcome, in our case, generate events and coordinates that lead to a hockey goal. This would be interesting to the team analytics. For generating the events, we have used Time-Series Generative Adversarial Network (TimeGAN) [8] and created synthetic data. These synthetic data follow the same distribution as the original data. We have evaluated the synthetic data using the PCA plot. Besides testing qualitatively, we tested this synthetic data on a sequence prediction model. After generating the data, we searched for goal events from the synthetic dataset and showed five previous events and their coordinates before the goal events. Finally, we plotted the coordinates and events to show this visually. We have also compared the TimeGAN with another two GAN models. This is the first work that has used synthetic data to generate hockey events to our best knowledge.

Rest of the paper follows this sequence. Background of this project can be found on section 2, problem formulation on section 3, methodology and results on section 4, 5 respectively. Finally we have concluded the discussion in section 6.

The code of this project can be found here [1].

2 Background

Generative Adversarial Network or GAN is part of the vast deep learning field specializing in generating data. GAN consists of two neural network architectures, namely, generator and discriminator. The generator tries to generate fake data from noise, and the discriminator tries to distinguish between the real and fake data. After the training, the generator can generate synthetic data that follows the same distribution of original data. However, it is not easy to train a GAN. Different training improvement techniques have been proposed in recent years. GAN architecture differs based on the domain, and it has been most successful in the computer vision area. GAN architectures have been proposed to handle the time-series generation in recent years. The volatile nature of time series makes it challenging to synthesize them. Time-Series Generative Adversarial Network or TimeGAN is one of the GAN architectures that can synthesize time-series data. In TimeGAN's core, Long Short Term Memory (LSTM) [4] has been used to learn the pattern of time-series data.

LSTM is a variant of Recurrent Neural Network that works best with sequential data and is free from its predecessor's gradient vanishing and exploding problem. It uses three gates (forget, input, and output gate) to remember a longer sequence length.

3 Problem Formulation

Given a series of events $S_{1:T}$, we need to understand the pattern and then need to generate events that will lead to a particular event. In our case, we want to find the previous N steps of events that leads to *goal*, $p(S_{event_{[T-N:T-1]}}|S_{event=goal_{[T]}})$. To achieve this outcome and understand the pattern of training Dataset, D, we need to learn the density $\hat{p}(S_{1:T})$ that can approximate $p(S_{1:T})$.

For learning the density we will use Generative Adversarial Network (GAN). So, the objective is,

$$\min_{\hat{p}} D\big(p(S_{1:T})||\hat{p}(S_{1:T})\big) \tag{1}$$

[1] `https://github.com/fahim-sikder/event-generation-ice-hockey`

4 Methodology

GAN usually has two neural network architectures, generator, and discriminator. However, TimeGAN has four components: Encoder, Decoder, Generator, and Discriminator. At the beginning of the training, the encoder and decoder take the original data and encode it into a latent space. Then, the generator and discriminator operate within the embedding space to create sequential synthetic data. Figure 1 shows the overall architecture of the TimeGAN.

TimeGAN uses three types of loss functions. Reconstruction losses were used in the auto-encoding phase that oversees the accurate reconstruction of the original data. Supervised and Unsupervised loss were used to train the generator and discriminator parts.

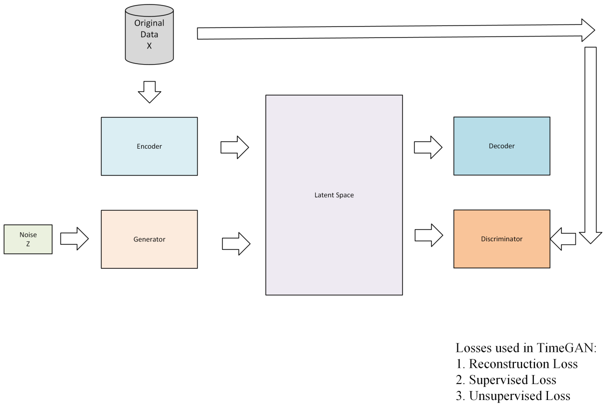


Fig. 1: TimeGAN Architecture

4.1 Data Processing

Sportlogiq provided the data used in this project as a part of the competition with permission from the Swedish Hockey League, which contains event data from 20 matches during the 2020-21 session. Each row in the data represents a segment of a hockey match that contains different information like the id of the game, player id, event name such as pass, goal, carry, coordinates. Our goal in this project is to generate events and coordinates, so we have not used all the features. Therefore, we only used seven features and every 30 sequences before the goal. After selecting the features, class labels were converted into numbers and then were scaled from $0-1$. We have then converted the whole dataset into chunks of 24 sequence lengths.

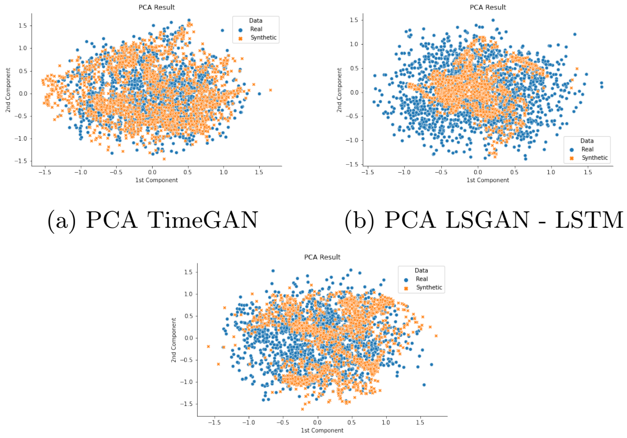
4.2 Approach

After pre-processing the data, we fed it to the TimeGAN and generated the synthetic data. For the implementation of the TimeGAN, we have used a python package called ydata-synthetic [2]. We have searched for the goal event and taken the five previous events from the goal event from the synthetic data. As the synthetic data follows the distribution of the original data, we should be able to find goals in the synthetic data. Finally, we plot the data into a hockey rink using another python library called hockey_rink [1]. Besides, TimeGAN, we also implemented two GAN models using LSGAN [6] where LSTM and GRU were used as its core (generator, discriminator). We have trained these three models for 20k epochs.

5 Results and Discussions

Our main contribution to this project is the idea of the usage of synthetic data to find a particular event and the events leading to that. For example, we have shown the events that lead to a hockey goal. More importantly, this is a generalized solution, so we can use any targeted particular outcome and find the series of events leading to that.

Figure 2 shows the PCA plots of three GAN models. Each dot represents a sequence in the dataset, blue dots represent the real data, and orange dots represent the synthetic data.



(a) PCA TimeGAN (b) PCA LSGAN - LSTM

(c) PCA LSGAN - GRU

Fig. 2: PCA Plots of three GAN models

We can see that TimeGAN's PCA plot is overlapping with the real data, that represents TimeGAN learns the distribution better than the other two models.

Besides PCA plots, we have also evaluated the generated data using a sequence prediction task. For example, given a 23 sequence, predict the next sequence. To test this, we have created a Recurrent Neural Network based model (GRU) that we have trained with the synthetic data and tested with the original test data (we did not use this to train the GAN models). We used mean squared error and mean squared log error to evaluate the sequence prediction task. Table 1 shows that TimeGAN achieved lower errors than the other two models.

Table 1: Comparison of Three GAN models on sequence predicting task

Models	MAE	MRLE
TimeGAN	**0.246165**	**0.053882**
LSGAN-LSTM	0.2999977	0.062845
LSGAN-GRU	0.293644	0.071429

Using the TimeGAN, we have sampled more than 140k event data and plotted the goal in a heatmap using hockey_rink library. Here we have shown in figure 3 the heat-map of goals from synthetic data and original data.

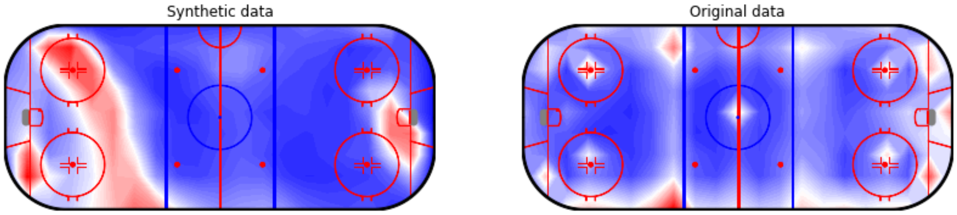
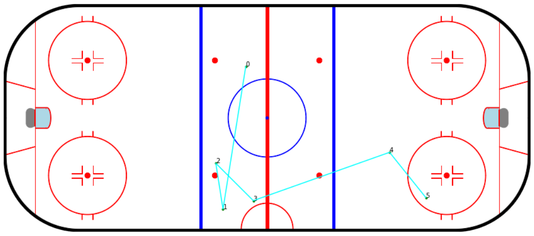


Fig. 3: Heatmap of Goal position, synthetic vs original data

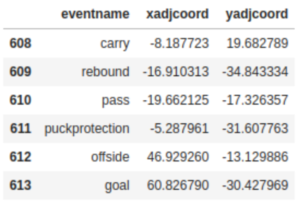
Also, we have shown how our generated events leads to a goal looks like in the hockey field in figure 4. Here we have shown the previous five events before the goal.

6 Conclusions

In this project, we have used synthetic data to find out the series of events that leads to a particular outcome. First, we used TimeGAN to generate the synthetic data, then evaluated this using PCA plots and sequence prediction tasks. Finally, we showed a heatmap of generated goals and a particular example of how the synthetic data looks and plotted the coordinate in a hockey rink.



(a) Synthetic Goal



	eventname	xadjcoord	yadjcoord
608	carry	-8.187723	19.682789
609	rebound	-16.910313	-34.843334
610	pass	-19.662125	-17.326357
611	puckprotection	-5.287961	-31.607763
612	offside	46.929260	-13.129886
613	goal	60.826790	-30.427969

(b) Synthetic Goal Events

Fig. 4: Synthetic goal plot and events

References

1. Hockey rink. `https://github.com/the-bucketless/hockey_rink`, accessed: 2022-05-14
2. ydata-synthetic. `https://github.com/ydataai/ydata-synthetic`, accessed: 2022-05-14
3. Bruce, J.M., Echemendia, R.J., Meeuwisse, W., Hutchison, M.G., Aubry, M., Comper, P.: Development of a risk prediction model among professional hockey players with visible signs of concussion. British journal of sports medicine **52**(17), 1143–1148 (2018)
4. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
5. Li, F.: Description, analysis and prediction of player actions in selected hockey game situations. Ph.D. thesis, University of British Columbia (2004)
6. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE international conference on computer vision. pp. 2794–2802 (2017)
7. Tora, M.R., Chen, J., Little, J.J.: Classification of puck possession events in ice hockey. In: 2017 IEEE conference on computer vision and pattern recognition workshops (CVPRW). pp. 147–154. IEEE (2017)
8. Yoon, J., Jarrett, D., Van der Schaar, M.: Time-series generative adversarial networks. Advances in Neural Information Processing Systems **32** (2019)