Name: Océane SOULIER

Name of your Level 1: Morjane SAIDANI

Paper title: Predictive business monitoring via generative adversarial nets: the case of next event prediction

Source: Google scholars

Keywords specific to the paper: business process monitoring, deep learning

Summary of the main contributions (use text paragraphs, tables and if necessary, figures):  
The paper proposes an adaptation of an adversarial training framework already existing called Generative Adversarial Nets (GANs) for predictive process monitoring and next event prediction. The goal is to predict “the most likely next event by determining both its label and its timestamp”. The framework involves two neural networks (a generator and discriminator) that play a minmax game during training. The generator learns to generate predictions that are indistinguishable from the ground truth, while the discriminator tries to distinguish real from generated predictions. At the end, they reach the Nash equilibrium, which is considered as the optimal solution since none of the players can find a better strategy to maximize its gains and minimize the ones of the other player.

The approach is shown to outperform the accuracy prediction (systematically higher than each baseline), to have a better generalization of the datasets, to be less sensitive to the number of training sequences (meaning it eliminates the need to have a large amount of them) and not to suffer from accuracy fluctuations.

According to the paper, one of the challenges with existing deep learning approaches (see below) for next event prediction is the limited size of real-life event logs available for training. To counter this, the generator now learns how to fool the discriminator with the input data distribution, meaning with a set of given data it is now able to generate much more and fool the discriminator to a point that it is not able to distinguish real prefixes from fake ones.

Overall tremendously good resultats, the authors emphasize the fact that their idea needs further experiments and analysis. Indeed, the optimum solution (thus Nash equilibrium) is not guaranteed. However, the worst case scenario is still the best case scenario in a conventional training (no adversarial process), meaning the method does not appear to be risky. According to them, using another architecture may be a first idea of investigation.

AI model used (e.g. Neural network, etc.):

The problem of next event prediction has been addressed with different neural networks. Neural network is a method in artificial intelligence that teaches computers to process data in a way that is inspired by the human brain[[1]](#footnote-0).

Several of them are mentioned, such as Recurrent Neural Networks (RNNs), Long-Short-Tem Memory (LSTM) and Convolutional Neural Networks (CNNs). All of them were used to predict the next event label (but not the timestamp, meaning they appear to be lacunars). Generative Adversarial Nets (GANs) and Deep Neural Networks (DNNs) are also mentioned.

Definitions, as given in the paper:

* DNN are machine learning models that aim at learning feature hierarchies at multiple levels of abstractions. It learns directly from data.
* RNNs are machine learning models exploiting the notion of parameter sharing that employs a single set of parameters for different parts of a model. It has main issues with catastrophic forgetting (the models forgets the learned patterns) and optimization stability (the optimization does not converge).
* GANs is the framework seen above, with two players called generator and discriminator. GANs provide substantial advantages compared to the two others models. It’s a more robust model.

Both LSTM and CNN require loads of labeled training data in order to be able to function. It means that it requires a large number of parameters to be anticipated and estimated. They are quoted as models used in previous works, seen as incomplete or at least subject to improvement.

RNN, however, is the basis of the novel framework and helps GANs to run correctly. RNN models the two players (the generator and the discriminator). LSTM represents the architecture of the structure.

Supported by a software application? (If yes, provide more details):

The approach was supported by Python 3.6 via PyTorch 1.2.0

1. [What is a Neural Network?](https://aws.amazon.com/what-is/neural-network/?nc1=h_ls) [↑](#footnote-ref-0)