Financial Inclusion: New Algorithms

Karim BEGUIR, Co-Founder & CEO, InstaDeep

kb@instadeep.com
@kbeguir

DSN Data Science Nigeria

InstaDeep™

Google Developers Experts

Financial Risk Analytics

GANs: a history

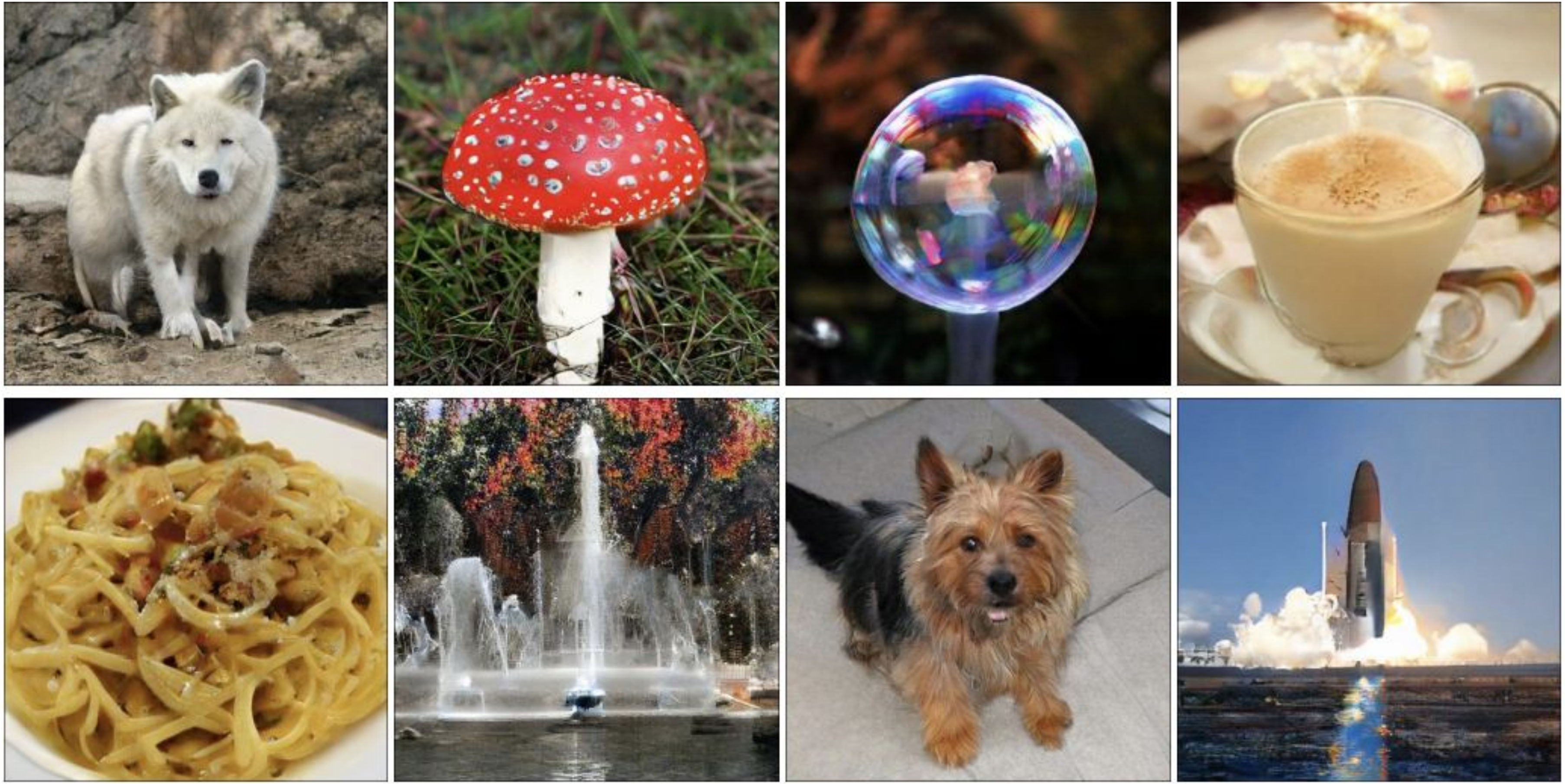
- June 2014: Ian Goodfellow publishes seminal GAN paper

- 2015: Radford’s DCGAN (“Deep Convolution GAN”) generates images

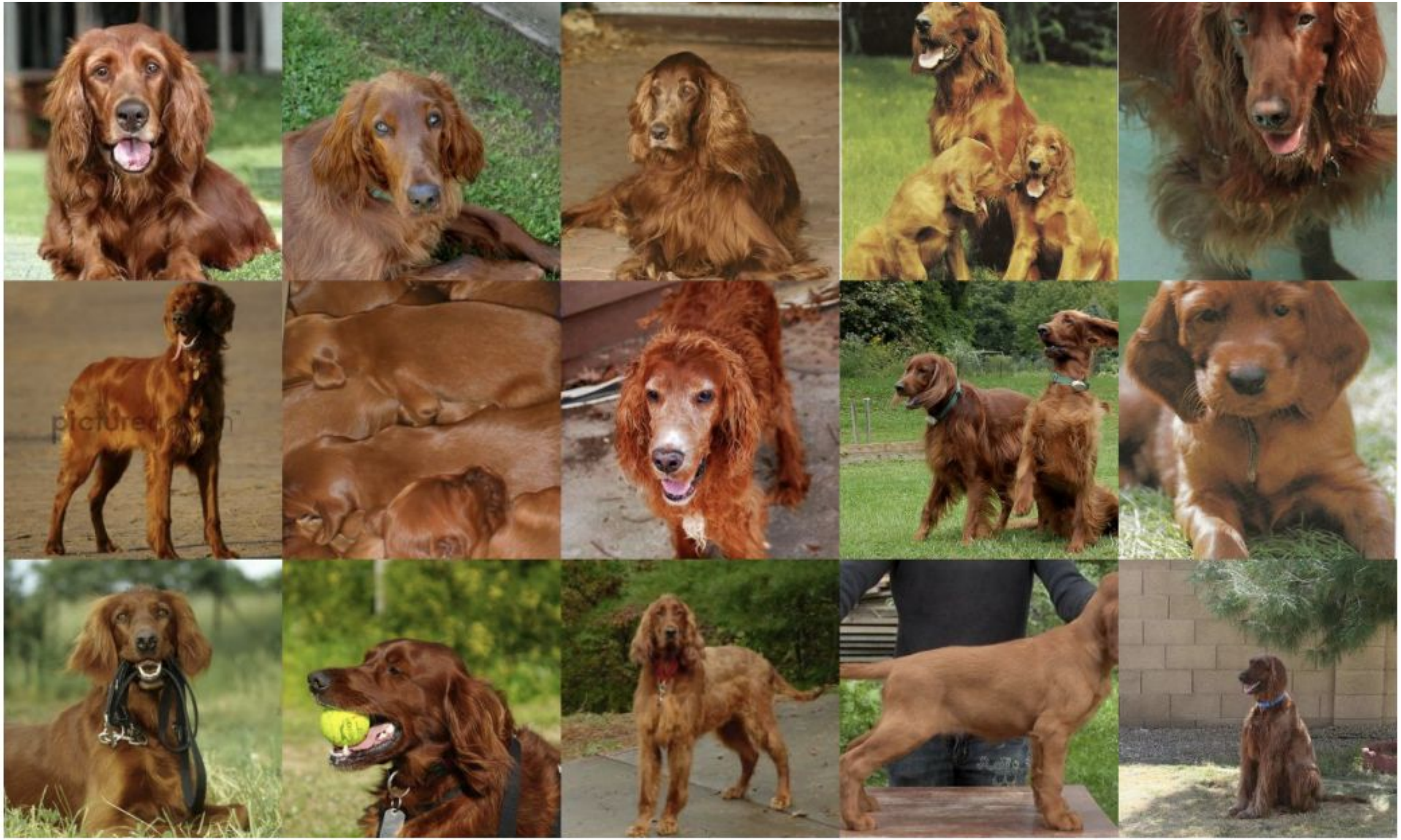
- July 2016: Yann LeCun (head of FB’s AI Research) says *“GANs are the most interesting idea of the last 10 years of Machine Learning”*

- Dec 2017: high resolution images without conditionality

- Sep 2018: high resolution images conditional on categories



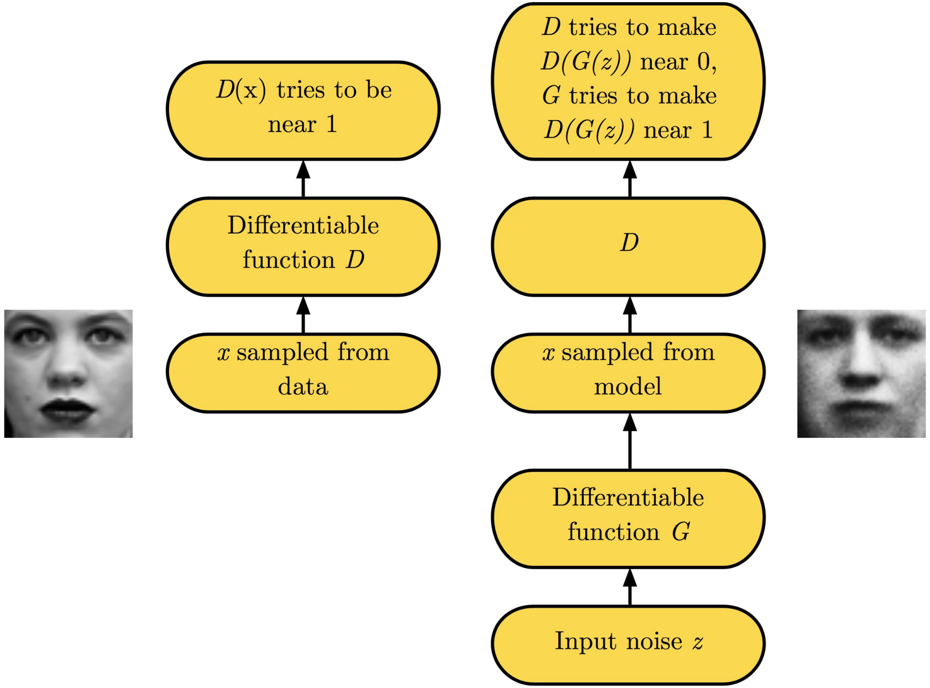
(Large Scale GAN training for high fidelity image synthesis, Bruck et al. 2018)



(Large Scale GAN training for high fidelity image synthesis, Bruck et al. 2018)

How it works

GANs use two neural networks, the generator G and the discriminator D. G creates fake photos and tries to fool D that they are real data. D tries to get it right.



D(x) tries to be near 1

Differentiable function D

x sampled from data

D tries to make $D(G(z))$ near 0, G tries to make $D(G(z))$ near 1

D

x sampled from model

Differentiable function G

Input noise z

How it works

Mathematically, this can be expressed as a zero-sum game where G and D try to minimize the following cost functions:

$$J^{(D)} = -\frac{1}{2}\mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}} \log D(\boldsymbol{x}) - \frac{1}{2}\mathbb{E}_{\boldsymbol{z}} \log\left(1 - D\left(G(\boldsymbol{z})\right)\right)$$
$$J^{(G)} = -J^{(D)}$$

Ian Goodfellow proved that a unique equilibrium exists where the G will generate fake sample photos so good that the discriminator (and humans) can’t distinguish them from real ones.

Incredibly, you can build an algorithm where **you start from nothing**, G and D refine themselves iteratively and sample quality keeps improving.

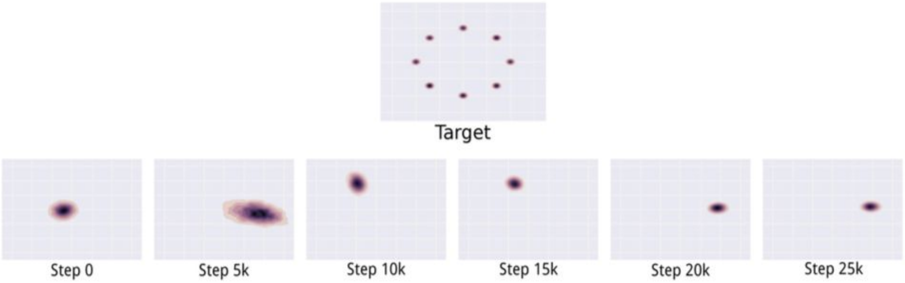
In practice, it’s possible to build samples of great quality **“out of thin air”**!

GANs for financial applications

Cutting-edge AI research has focused on multiple uses of GANs, mostly images and and video but relatively few articles on GANs for finance. This is due to limitations:

- Unstable training: the model might not converge at all.

- Mode collapse (the inability to generate differentiated samples) is an issue,

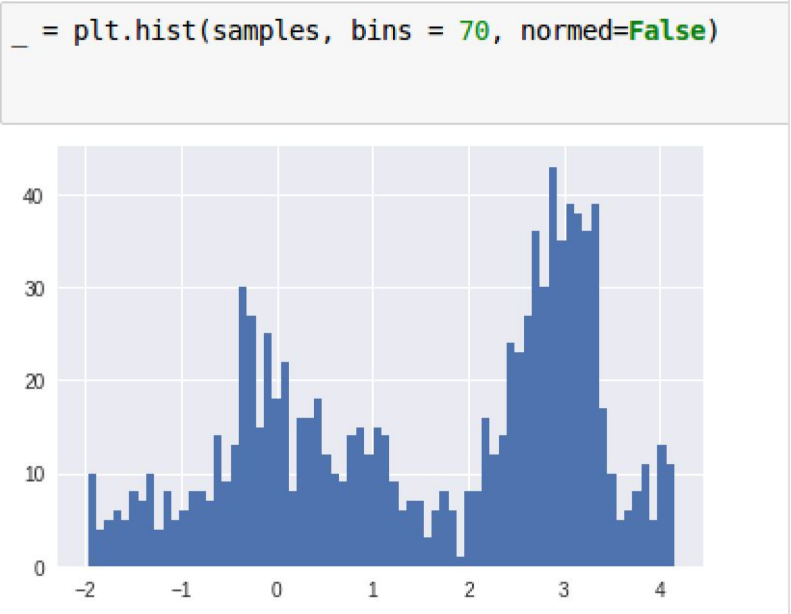


Lack of a convergence metric: in all previous examples you manually assess convergence, but on time series human intuition is limited and you need a metric

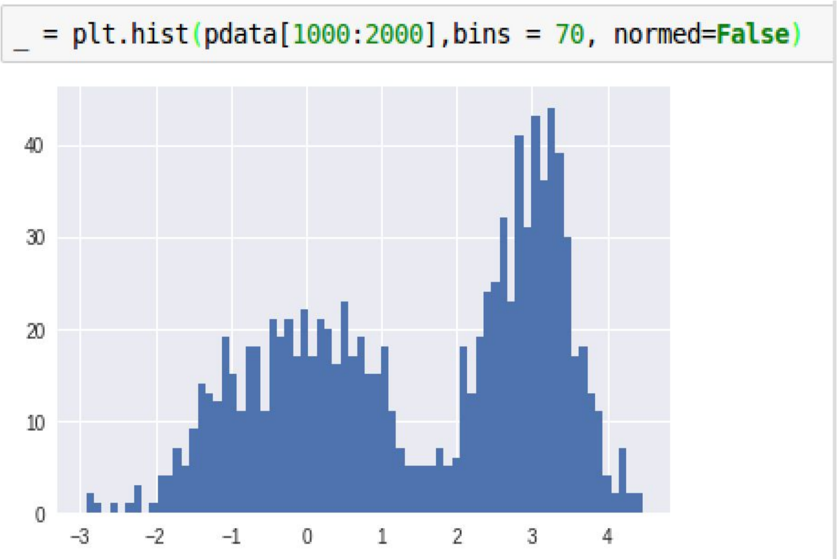
GANs for financial applications

Build a data set of samples a one-dimensional mixture of gaussians (here an equi-probable N(3,0.5) and a N(0,2).

Give the raw data without any context to our generative model. It rebuilds it!



```
_ = plt.hist(samples, bins = 70, normed=False)
```



```
_ = plt.hist(pdata[1000:2000],bins = 70, normed=False)
```

GANs for financial applications

Original Correlation Matrix

$$\begin{bmatrix}
1 & 0.90 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.90 & 1 & 0.40 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.40 & 1 & 0.60 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.60 & 1 & 0.20 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.20 & 1 & 0.90 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.90 & 1 & 0.45 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.45 & 1 & 0.80 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.80 & 1 & 0.70 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.70 & 1 & 0.30 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.30 & 1
\end{bmatrix}$$

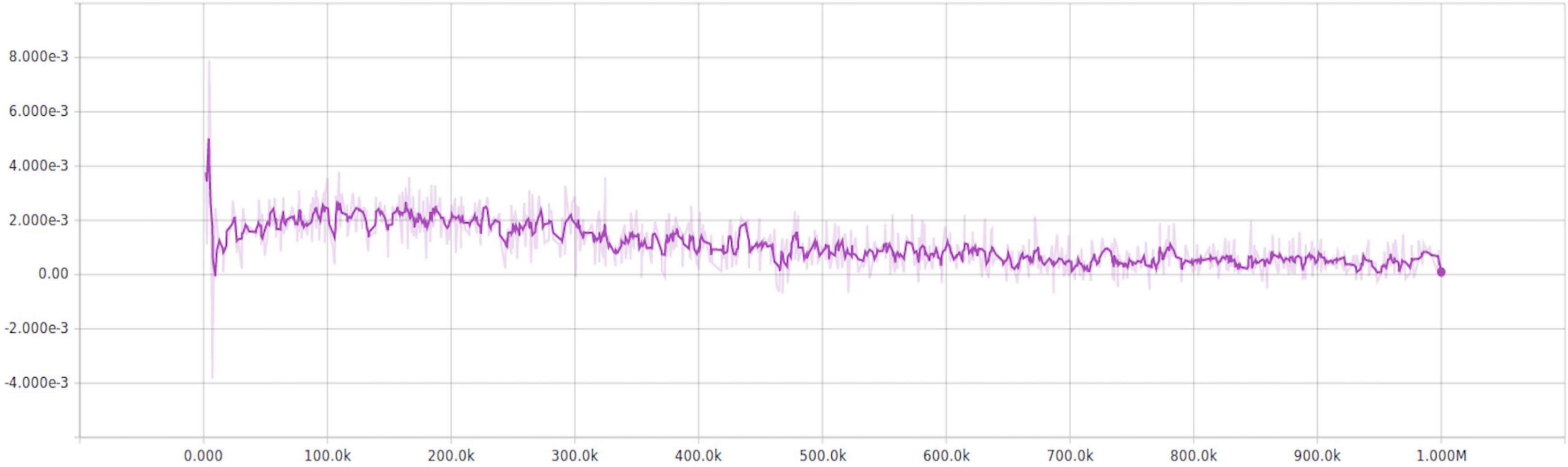
Rebuilt Correlation Matrix

$$\begin{bmatrix}
1 & 0.87 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0.87 & 1 & 0.39 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0.39 & 1 & 0.53 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0.53 & 1 & 0.19 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.19 & 1 & 0.87 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0.87 & 1 & 0.44 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0.44 & 1 & 0.76 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0.76 & 1 & 0.65 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.65 & 1 & 0.30 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.30 & 1
\end{bmatrix}$$

GANs for financial applications

Key insight: a metric to assess convergence **quantitatively**.

Solving the issue of convergence assessment for non-visual/manually checkable samples and and opens up the way to financial/predictive analytics applications



Results on real stocks

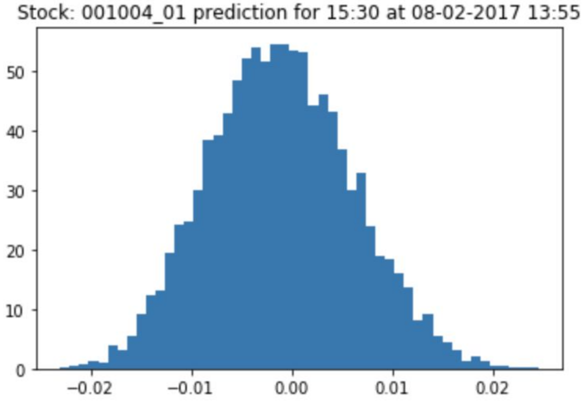
Original data: S&P500 stocks

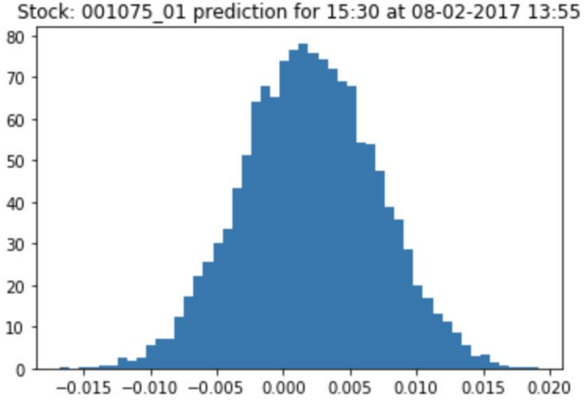
Period: 2010-2018

Conditional model of hourly returns

Generator models cross-correlations too

Approach can work on **any financial times series**





Stock: 001078_01 prediction for 15:30 at 08-02-2017 13:55

Financial Applications

Predictive analytics and sampling of variables of interest that are **not-historical** and **model-free** yet measurably “realistic” to the historical dataset.

Advanced scenario/strategies risk exploration tools to optimize advanced risk/reward in a non-gaussian, model-independent framework.

Optimal execution strategies for stocks or for correlated assets with smart use of historical data.

Cutting-edge portfolio allocation more accurate than efficient frontier-type tools. This uses advanced AI methods developed by AI research teams, and not familiar to traditional financial professionals.

Time Series Analysis

Problem to solve:

Deep Learning has been popular in computer vision, NLP and speech processing etc.

But when it comes to **financial data** and especially **time series**, the use of **traditional machine learning** techniques is still **dominant** and we rarely find real life applications of deep learning in tabular and time series data.

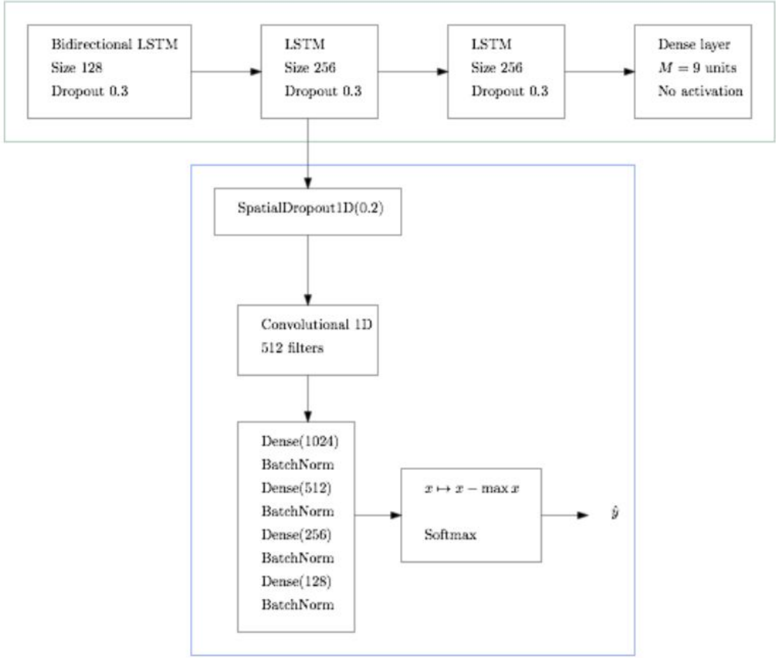
For general inference and classification tasks, machine learning practitioners tend to rely on heavy models combined with handmade feature engineering that are **time consuming** and require a lot of **experience**.

In a search for a more automated way to tackle this kind of classification, we propose a **new method** using deep learning techniques, and in particular RNNs.

Model

The training procedure is made of three steps. First, an asymmetric autoencoder is trained. The loss function is the square root mean square error (sRMSE):

$$\mathrm{sRMSE}_j = \sqrt{\sum_{i=1}^{N} \left(x_i^{(j)} - p_i^{(j)}\right)^2}, \quad 1 \leq j \leq L$$



Bidirectional LSTM
Size 128
Dropout 0.3

LSTM
Size 256
Dropout 0.3

LSTM
Size 256
Dropout 0.3

Dense layer
$M = 9$ units
No activation

SpatialDropout1D(0.2)

Convolutional 1D
512 filters

Dense(1024)
BatchNorm
Dense(512)
BatchNorm
Dense(256)
BatchNorm
Dense(128)
BatchNorm

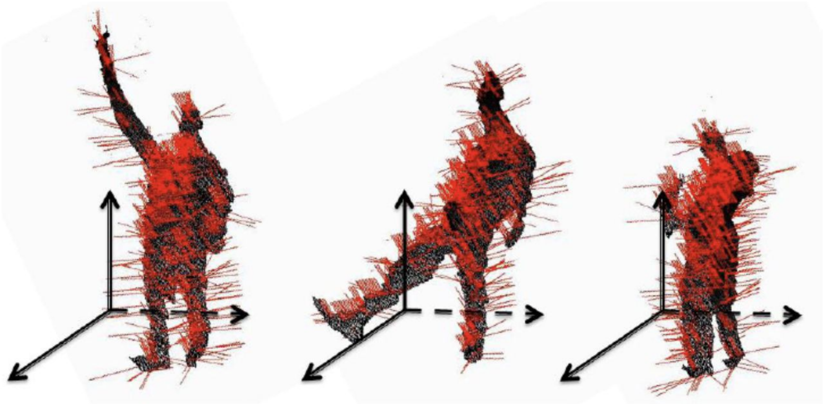
$x \mapsto x - \max x$
Softmax

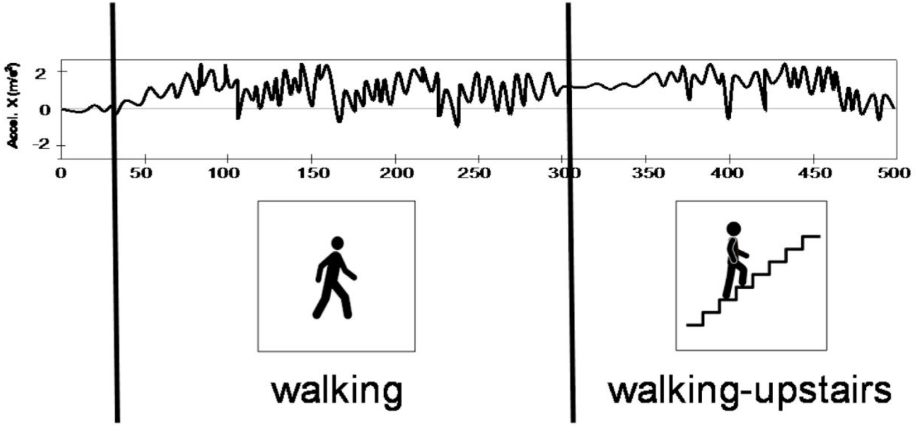
$\hat{y}$

Finally, this encoded information is used as the input of a multi-layer conv net trained with a classical cross-entropy loss.

Data

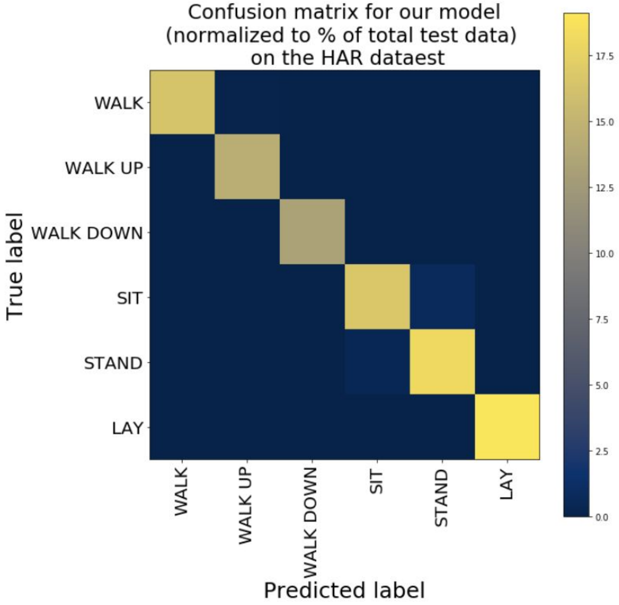
We work with the Human Activity Recognition (HAR) and Occupancy Detection (OD) datasets as well a financial customer activity dataset (classified).

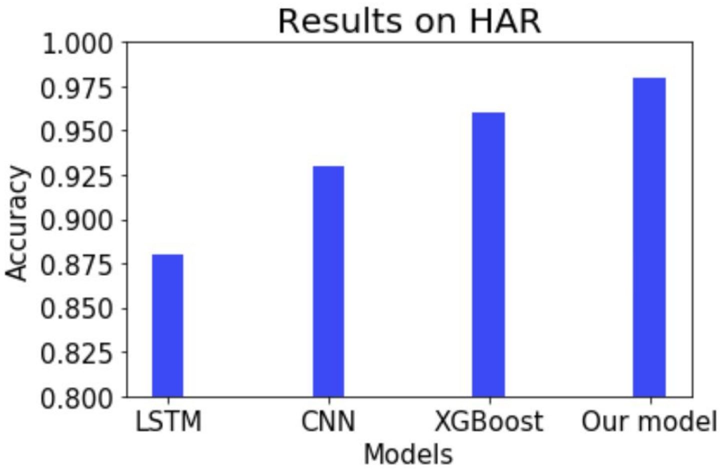




Results

On the HAR dataset, we get **98% accuracy** on the test set, vs. 96% for an XGboost ML model with hand-engineered features. Simple LSTM networks do not exceed 92% of accuracy.

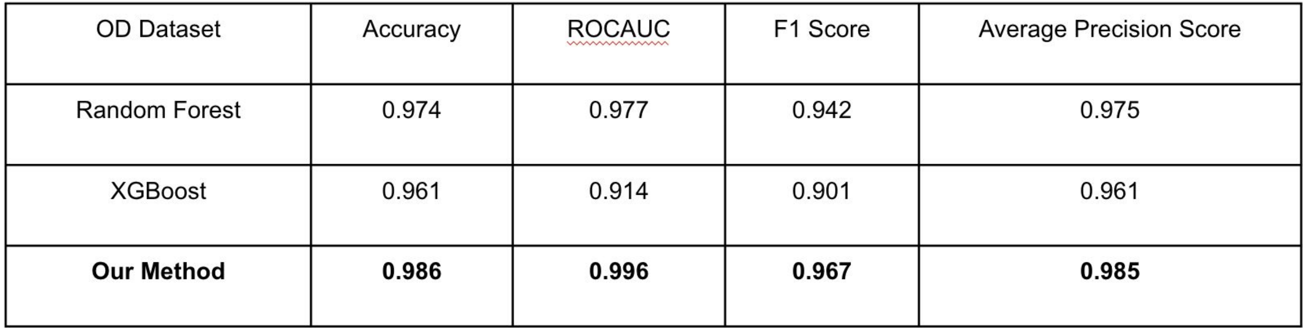




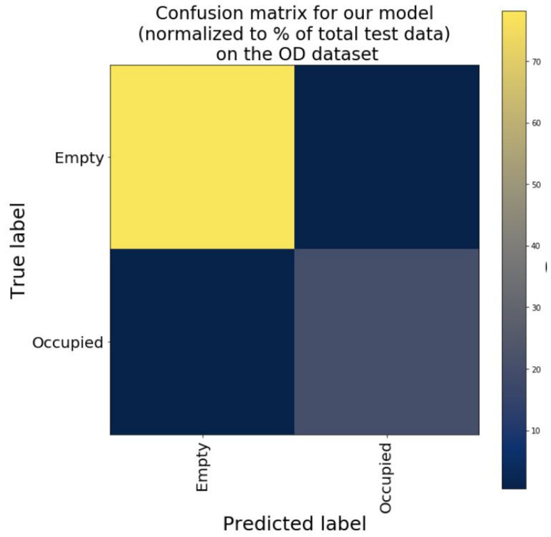
InstaDeep™

Results

For the OD dataset, two classes. Goal is to detect whether there is a person in the room. The measured features are temperature, humidity, light and CO2.



OD Dataset	Accuracy	ROCAUC	F1 Score	Average Precision Score
Random Forest	0.974	0.977	0.942	0.975
XGBoost	0.961	0.914	0.901	0.961
Our Method	**0.986**	**0.996**	**0.967**	**0.985**



We obtain qualitatively similar results predicting customer activity on the financial dataset. This has Multiple applications, from improved client management, credit scoring for financial inclusion and other.

Conclusion

- ML is delivering **results** in speech, vision, NLP, chess, design, healthcare etc.
- We present two recent AI algorithms with promising financial applications
- Fundamental advances in AI are powering new, improved algorithms for financial applications and financial inclusion. Stay tuned!

Contact: kb@instadeep.com @kbeguir

DSN Data Science Nigeria

InstaDeep™

Google Developers Experts