# Deploying Machine Learning on the Oilfield: From the Labs to the Edge

Fahd Saghir – Industry Solutions Manager (Upstream O&G)
Matthieu Boujonnier – Architect / Data Scientist

Webinars

## SPE Webinars are Sponsored by:

SPE Foundation

# Poll Question

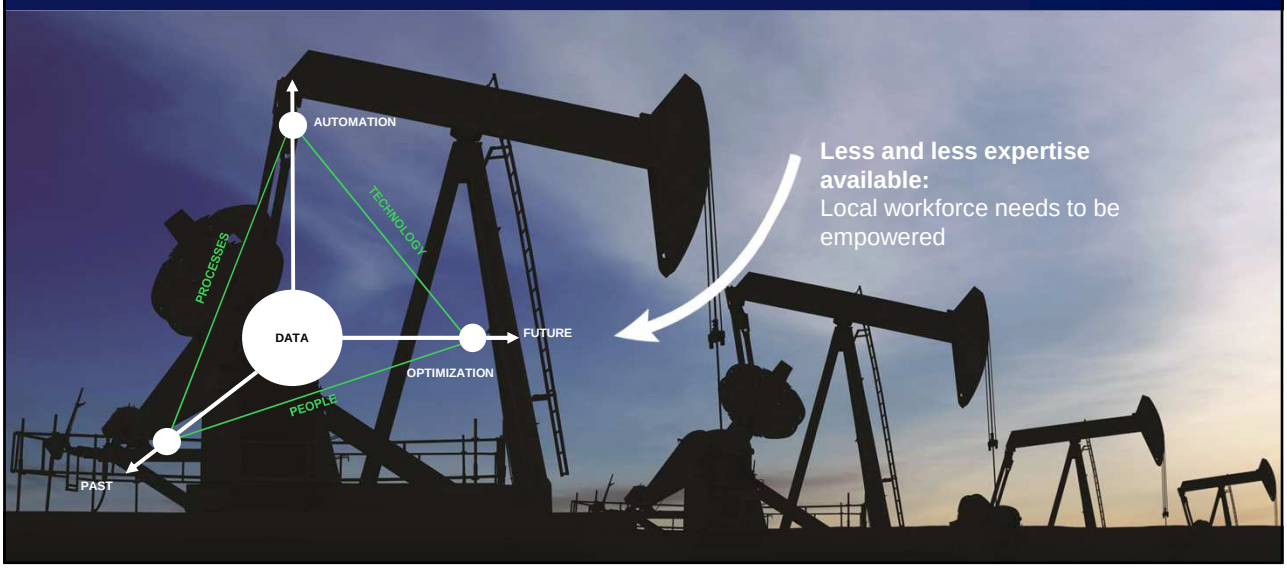
Please select your area of work/expertise:

# Poll Question

What is your experience with Data Science and Machine Learning applications?

# The Digital Oilfield Anatomy

Webinars

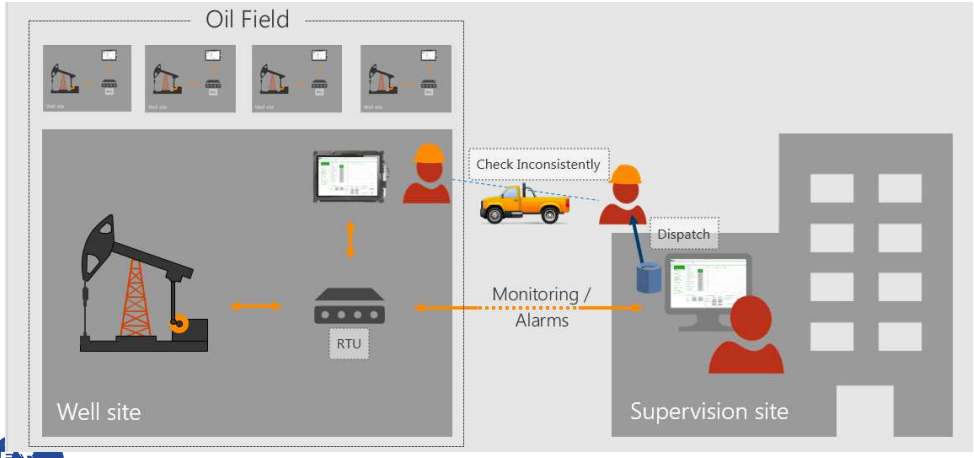



# Rod pump: facts

Webinars


| Downtime | 10 BOE/D | 250 BOE/D |
|---|---|---|
| 1 hour | $ 27 | $677 |
| 1 day | $648 | $16,250 |
| 1 week | $4,536 | $113,750 |

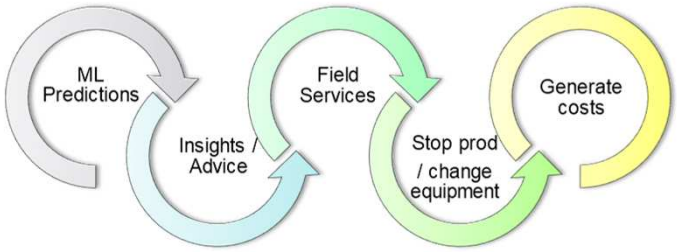
# Today SCADA-based solution



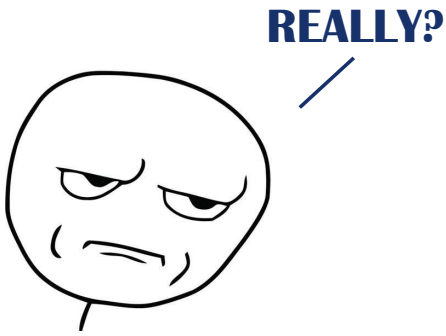

# Challenges

- Ensure Trust
  - **How can our customers trust ML predictions?**
- Extend model
  - **Labeled « expert » data is rare, how to ensure that our models will work for any « new » pump?**

# Customer's confidence is key

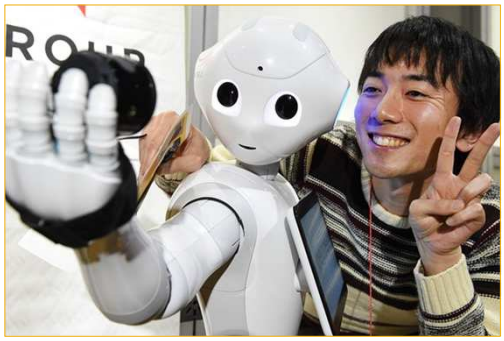




# Customer's confidence is key

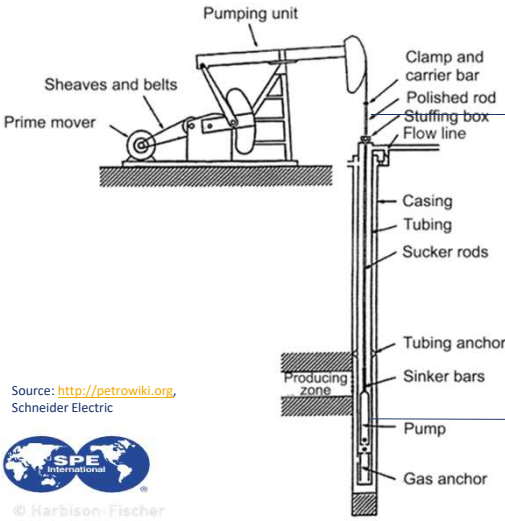



# Customer's confidence is key

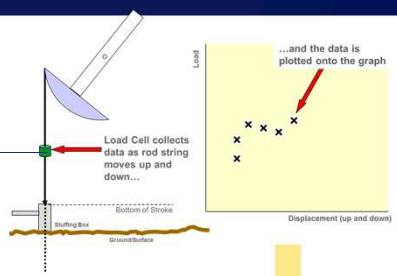

Let's do like humans do !



# A bit of mechanics and mathematics

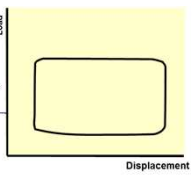



Source: http://petrowiki.org, Schneider Electric

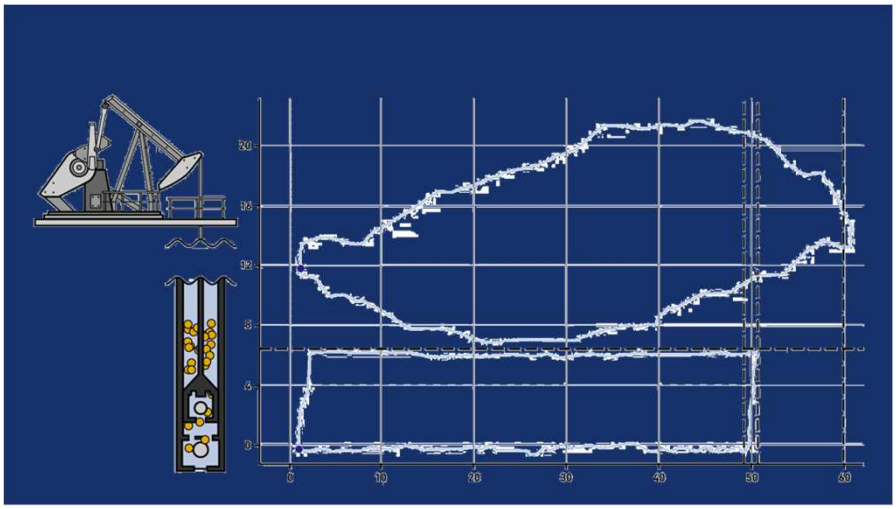



$$\partial^2 u/\partial t^2(x,t) = (a^2)\partial^2 u/\partial x^2(x,t) - (c)\partial u/\partial t(x,t)$$

The same data, after correcting for rod stretching, flexing, and stress wave propagation, looks like this, giving a clear view of what's happening at the *bottom* of the well ("downhole" card)

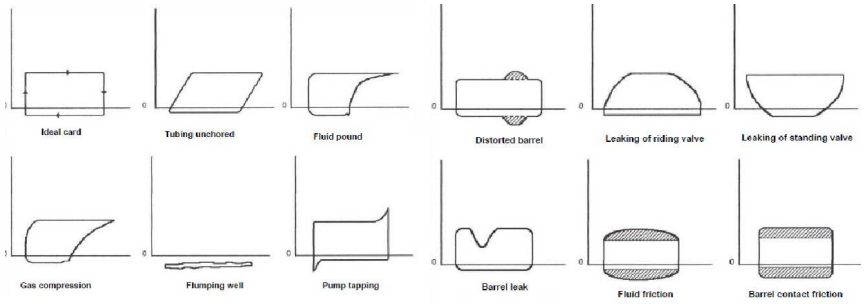


# Easier with a little animation…



# Human expertise

- Experts look at charts of failure patterns
- Experts use mostly their eyes, which interpret the image to a failure (but also look at the data)




Simplified Dynacards showing a failure pattern

# The way Machine Learning usually works

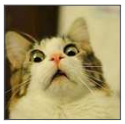
Webinars

1)Training: build model

data → Model ← label

Cat

2)Inference: use model



data → Model → label

Cat (0.86)
You (0.14)
(after this session)

# In real life

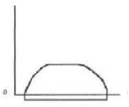
Webinars

1)Training

Data
Data
→ Model ← Label
Label

It's very challenging to obtain very large amount of labeled data

2)Inference



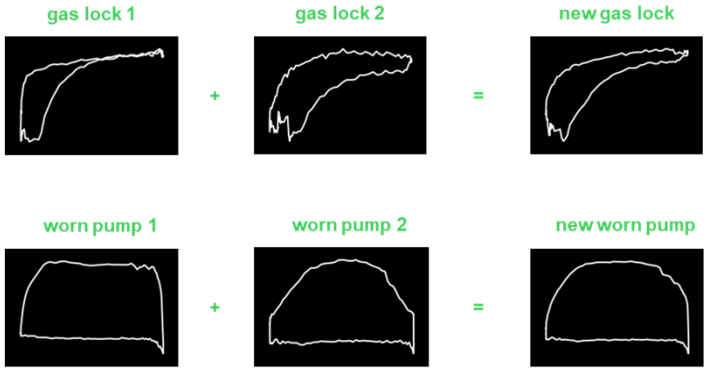
data → Model → label

# Data augmentation

Using existing tagged cards to increase the size of the training dataset

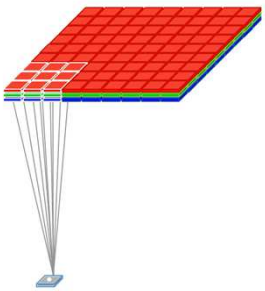
One way to get around a lack of data is to augment the dataset. The model will often be more robust and can even be simpler due to a better training set. This may prevent overfitting as well.

Pick images of the same class and combine them to get a new one:

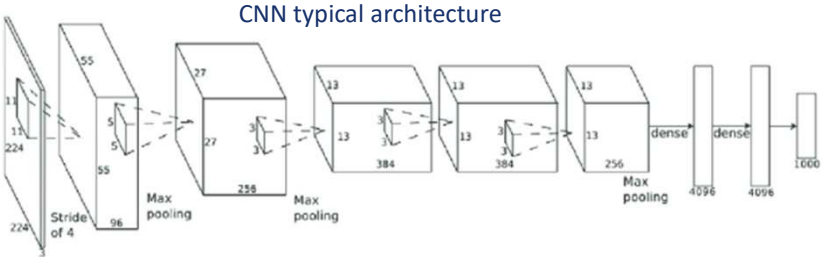


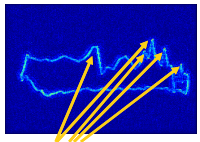

# Dynacards can be considered as an image

## Convolutional Neural Networks

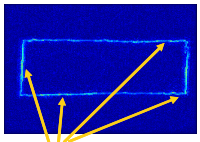


CNN typical architecture



Model: "*Those parts of the dynacard indicate mostly to me that this pump is grinding*"

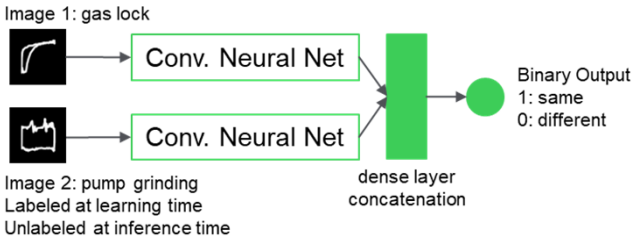


Model: "Mostly *because of those parts of the dynacard I think this pump works perfectly well*"

# Not sure what kind of cat it is but…not a dog!

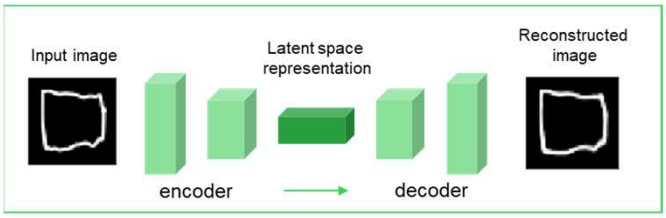
Siamese Network

When humans look at an object, they recognize its class *not only* because it's *similar* to some object but also because it's *different* from some objects.

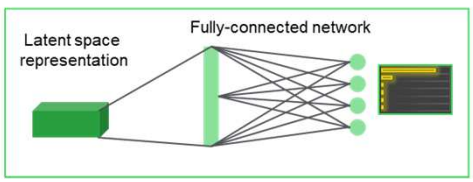



- Training is done by comparing each image against all the images in dataset and checking if the class is the same or different
- Data set is "augmented" by the combinations of pairs of all the images available

# Not enough labeled data? Use autoencoders!



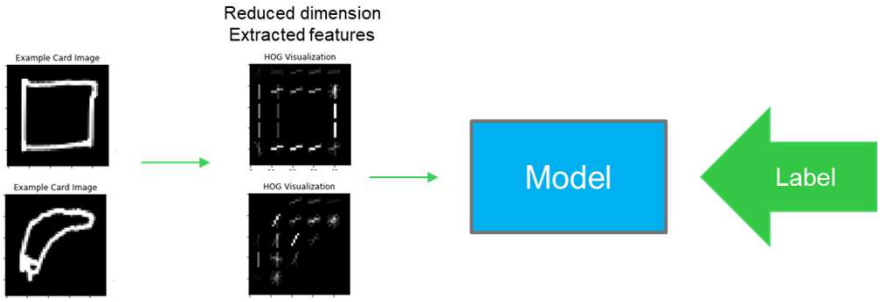

Self-supervised model - trained without labels!



# Dynacards are shapes, right?

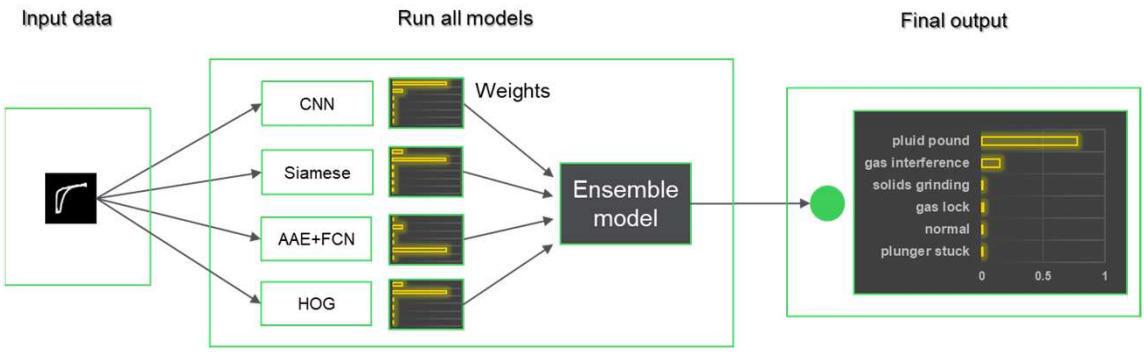
Histogram of Oriented Gradients (HOG)

- Simplify image by extracting new features:
- Gradients (x and y derivatives) of an image are large around contours (regions of abrupt intensity changes) and we know that contours contain a lot more information about shape than flat regions especially in our application!

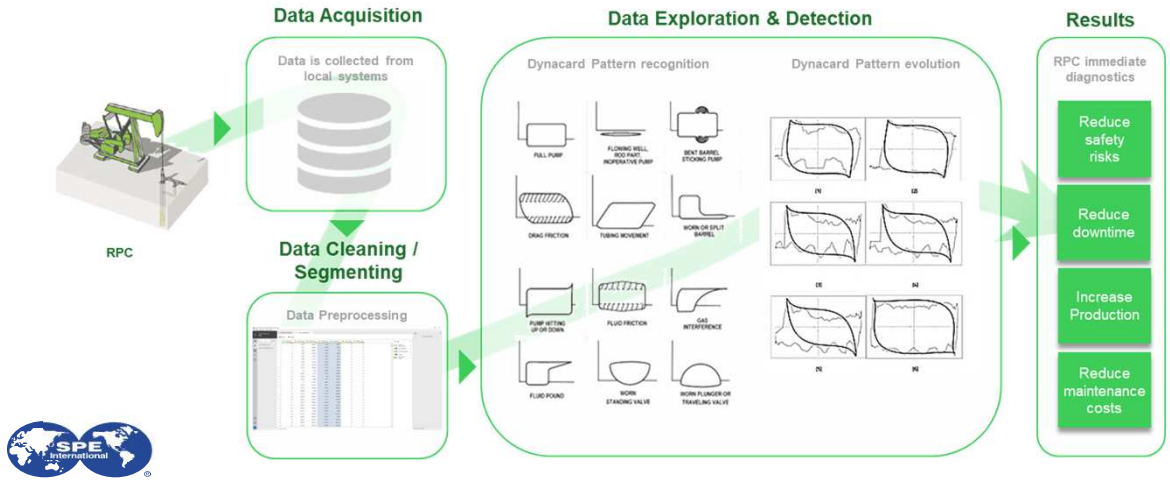



# Increase your odds! Use ensemble of models!

Instead of having one model combine many of them and make our task a team work!



# From the labs to a solution

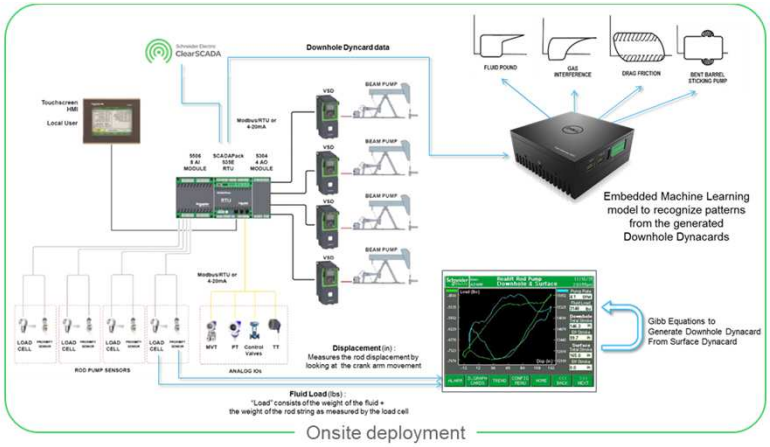



# Solution Architecture

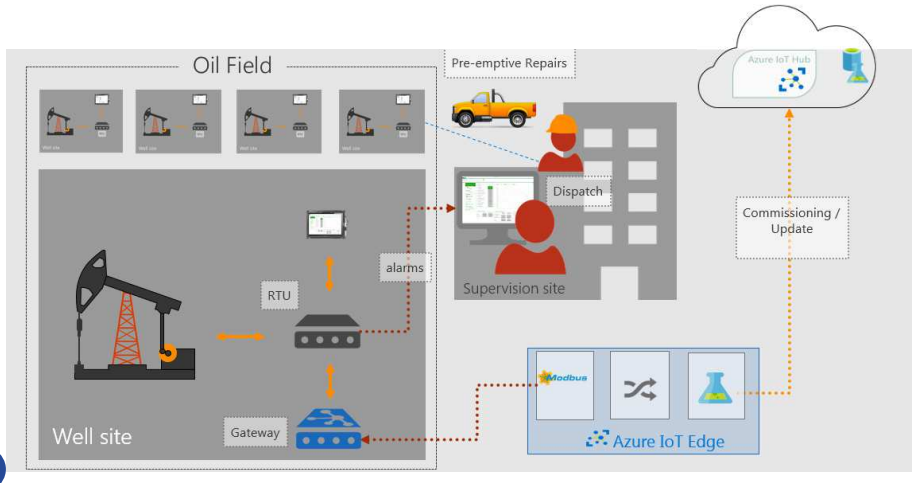
The solution is deployed in **harsh** environments where

- Internet connectivity is **unreliable and expensive**
- **Low bandwidth**
- Customers require high **data privacy** and confidentiality
- **Critical** systems are installed

As a result, this is a full **Edge solution** that does not rely on "always available" connectivity!



# Enhanced with Azure IoT Edge




# Use transfer learning to adapt the models locally