GRACE HOPPER CELEBRATION

ANITA B.ORG

Network Traffic Packets Classified as Textual Images for Intrusion Detection

Myriam Leggieri | @iammyr

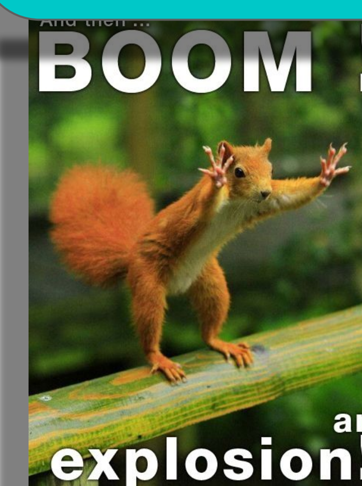
#GHC18

Let’s Connect!

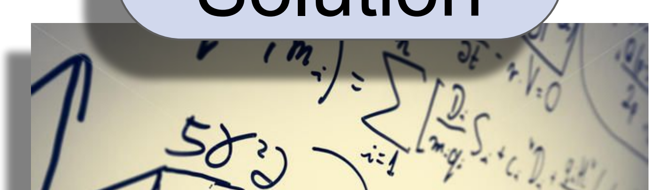
#GHC18

To Decrypt or not to?

Promising Results



The “Never Decrypt” Solution



Myriam Leggieri

workday.

@iammyr

Security Engineer

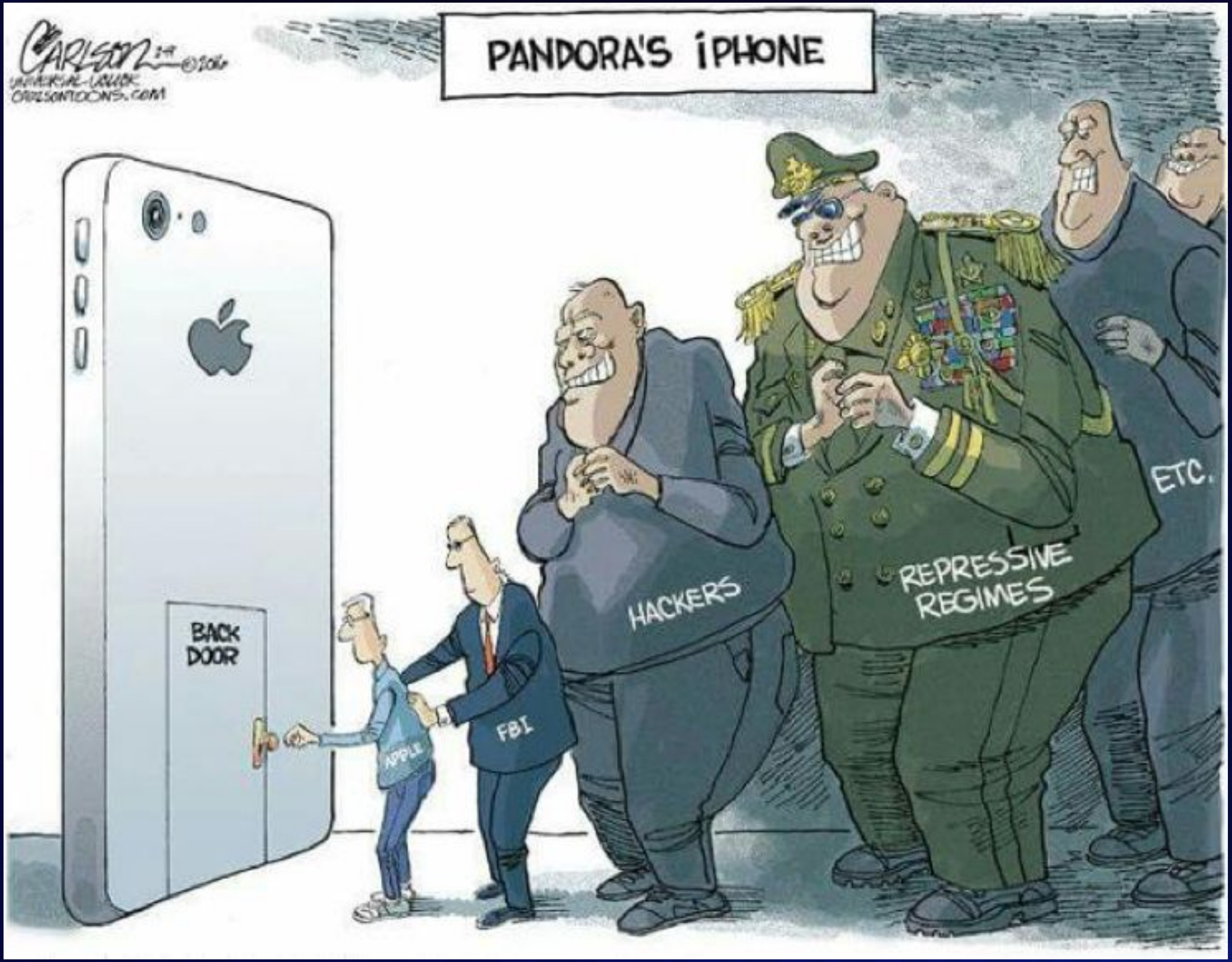
Design and Code Security Review

Development of Security Tools

Security Awareness

Pentesting

Encryption vs. Security



The Problem

Proposed Solution

Monitoring Encrypted Traffic without Decrypting

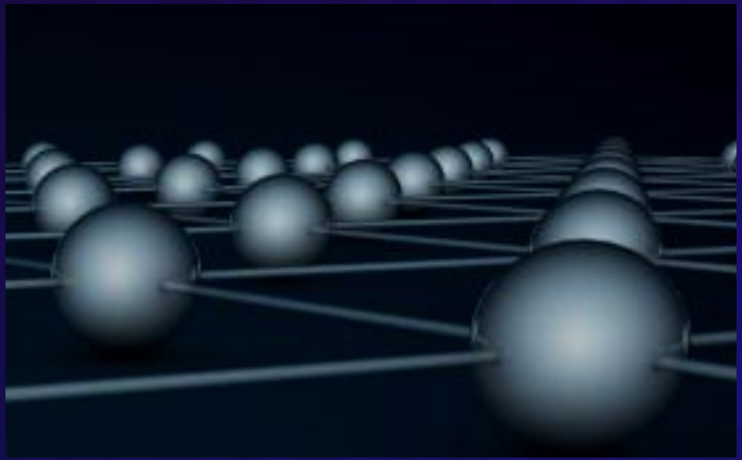
Convolutional Neural Networks (CNN)

Long Short Term Memory Networks (LSTM)

Conditional Random Fields (CRF)


WHY?

CNN → Learn Spatio-Temporal features



LSTM → Learn Long-Term Memories

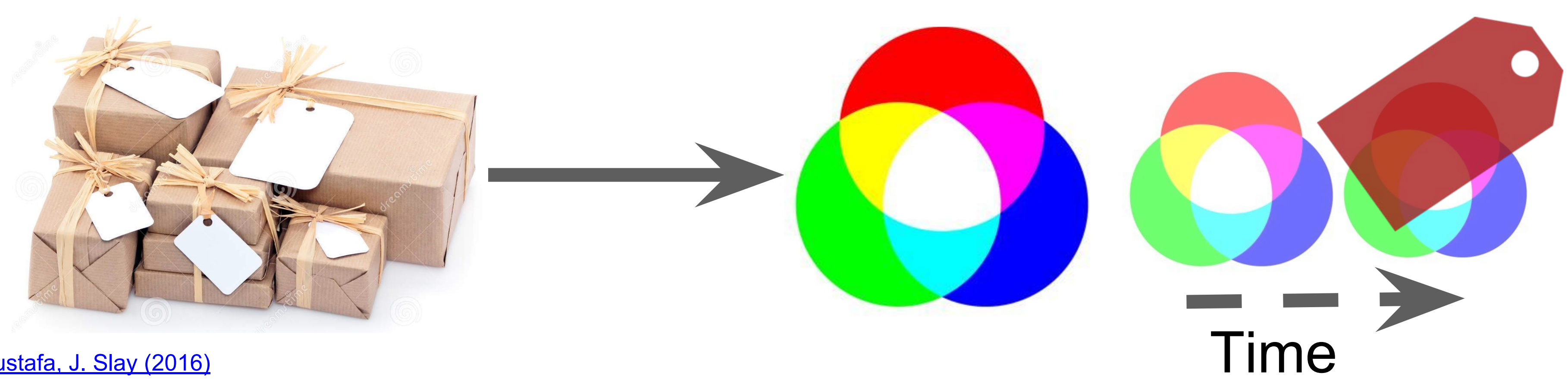
CRF → Learn from Textual Metadata

DataSet

- 100 GB
- 2.540.044 raw pcap traffic
  - 175.341 training set
  - 82.332 test set
- Sets of 4 packets
- 49 features each

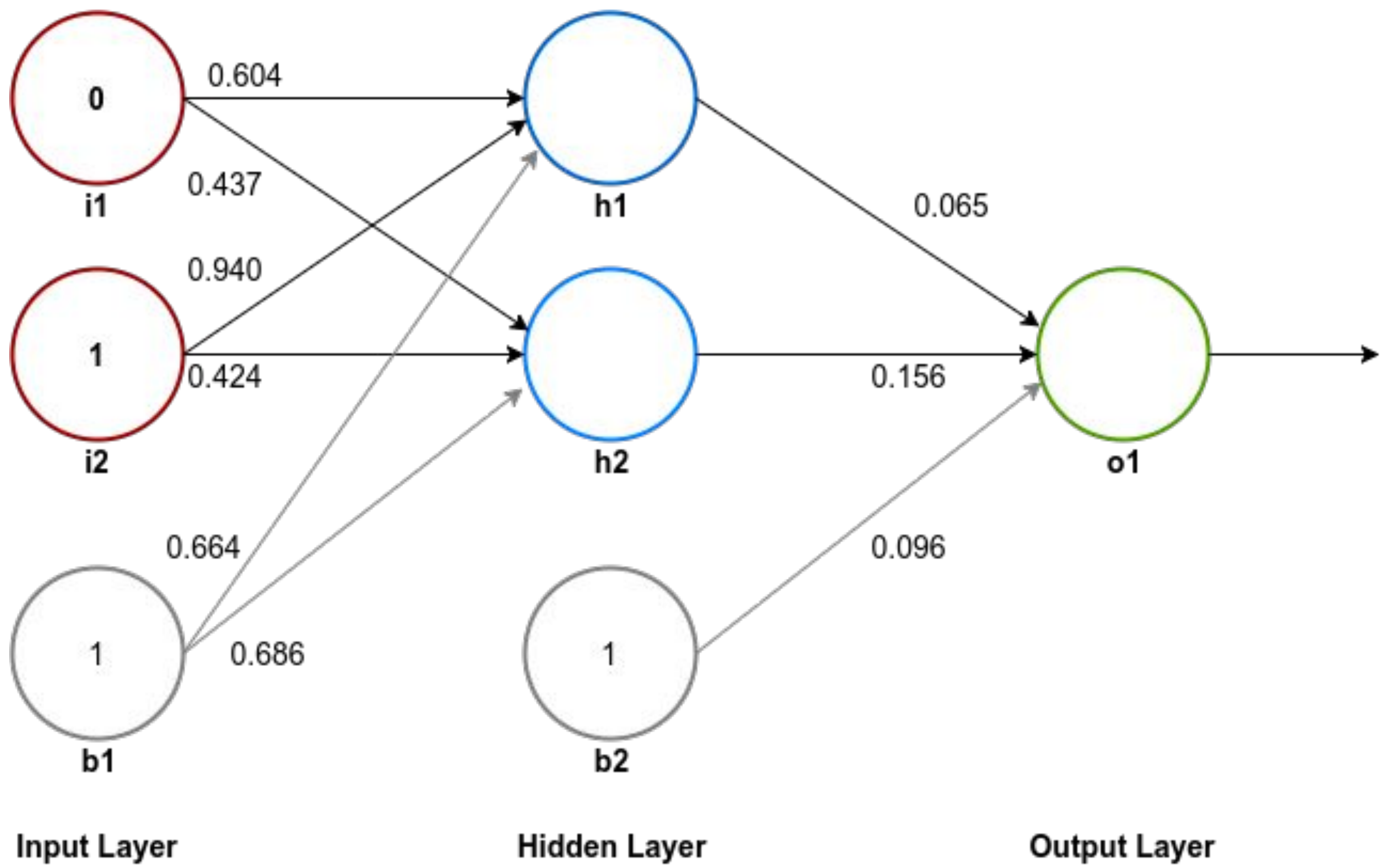
1. Fuzzers
2. Analysis
3. Backdoors
4. DoS
5. Exploits
6. Generic
7. Reconnaissance
8. Shellcode
9. Worms

R - 4 x 1 time/space -related features
G - 3 x 37 src/dest, IP+Port -related features
B - 4 x 11 remaining features
0<px<255 → normalise input



[The UNSW-NB15 data set by N. Moustafa, J. Slay (2016)]

NN



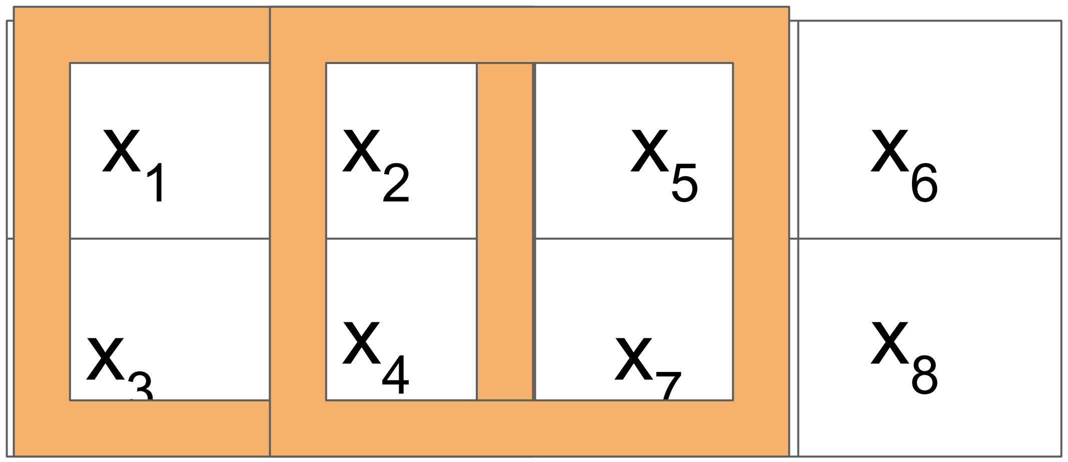
- Activation Function = $1/(1+e^{-x})$
- Weights = strengths of rel. = slope of sigmoid
- Bias = control on when to activate
- Goal: find min of cost(X)

→ **gradient(cost(X)) = 0**

^^Intractable → **Backpropagation in Gradient Descent Algorithm**

https://www.surenderthakran.com/images/articles/tech/implement-back-propagation-neural-network/xor-neural-network-weights.png

CNN



Convolution = moving filter

Overlapping area = stride[1, 1]

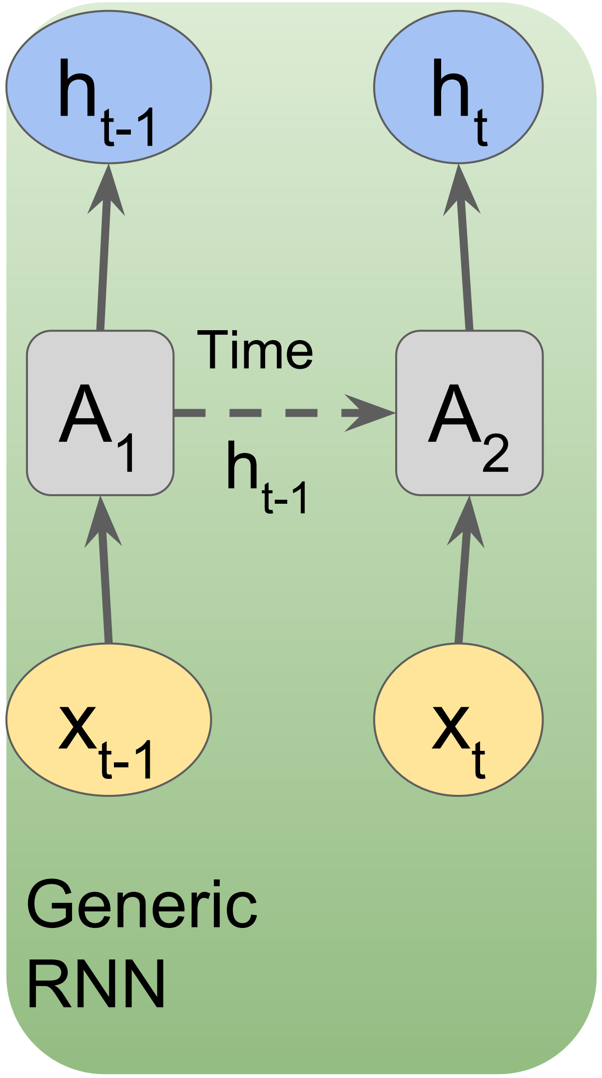
1. Sparse connections
2. Constant Weights in Filter

Less # Params

Filter → Features → Channel

Non-Linear Activation Function = Rectified Linear Unit (ReLU)
Max() Pooling on sliding window, stride [2, 2]
Batch Normalisation
Loss Function: Softmax Cross Entropy
Flatten m chans of X x Y pooling matrices into vector[X x Y x m]

LSTM



Vanishing Gradient Problem

$x * y = z;\; y > 1;\; z \gg 1$

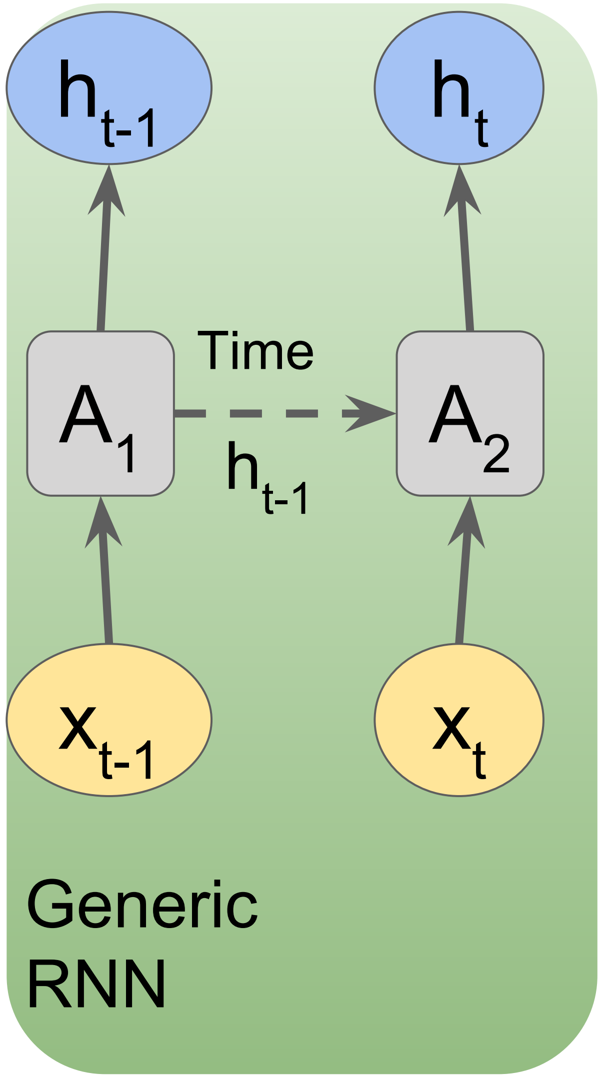
↓

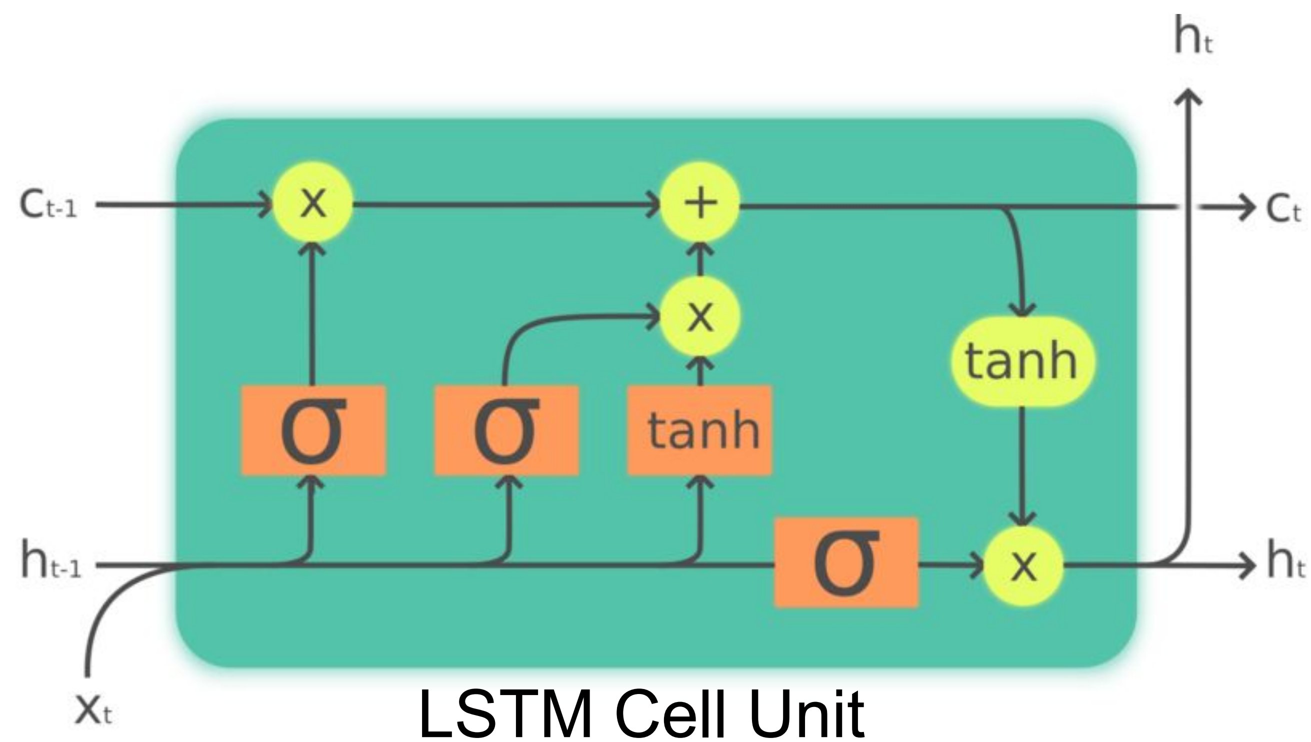
Truncate/Squash

$x * y = z;\; y < 1;\; z \ll 1$ ⟶ ? → LSTM!

Gambler wins 97 cents on every dollar → bankrupt!

LSTM





LSTM Cell Unit

$$
\begin{aligned}
f_t &= \sigma_g(W_f x_t \boxed{+} U_f h_{t-1} \boxed{+} b_f) \\
i_t &= \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= f_t \circ c_{t-1} + i_t \circ \sigma_c(W_c x_t + U_c h_{t-1} + b_c) \\
h_t &= o_t \circ \sigma_h(c_t)
\end{aligned}
$$

1. Forget gate
2. Input gate
3. Output gate

(x_t, h_{t-1})

CRF

Interdependence of attackers' movements

Given X sets of packets, Y labels,

1) Goal: find transition matrix T that minimises the neg log likelihood

$$\sum_{y'}\sum_{i=0}^{n} Log(P(x_i|y'_i)T(y'_i|y'_{i-1})) - \sum_{i=0}^{n} Log(P(x_i|y_i)T(y_i|y_{i-1}))$$

(avg-ed for the whole dataset → HUGE denominator

BUT current label only depends on previous label

aka

Forward-Backward Algorithm

2) Goal: use T to find the most likely seq of labels, given a seq of packets

aka

Viterbi Algorithm

Promising Results

#GHC18

Promising Results

- Learning (car) Traffic as Images (2017)
  - Goal: network speed prediction
  - CNN || RNN || LSTM
    - CNN performs best in long-term predictions
    - CNN outperforms with +42.91% on avg acc
- Network Traffic Classifier for IoT (2017)
  - Goal: infer the application/service used
  - CNN + LSTM
    - 2 CNN layers + 1 LSTM outperforms with 96% accuracy
- Implementation in progress

Future Work

- Bidirectional LSTM (having access to future packets for a given range of time)
- ELU may be better as an activation function than ReLU
- Try changing in CNN:
  - loss function,
  - hyperparameters for convolutional and pooling layers (filter size, pooling size, polling method),
  - depth of the CNN
- Identify which packets' metadata are more relevant for classification
- Include the encrypted payload among the considered features

@iammyr