MEERKAT

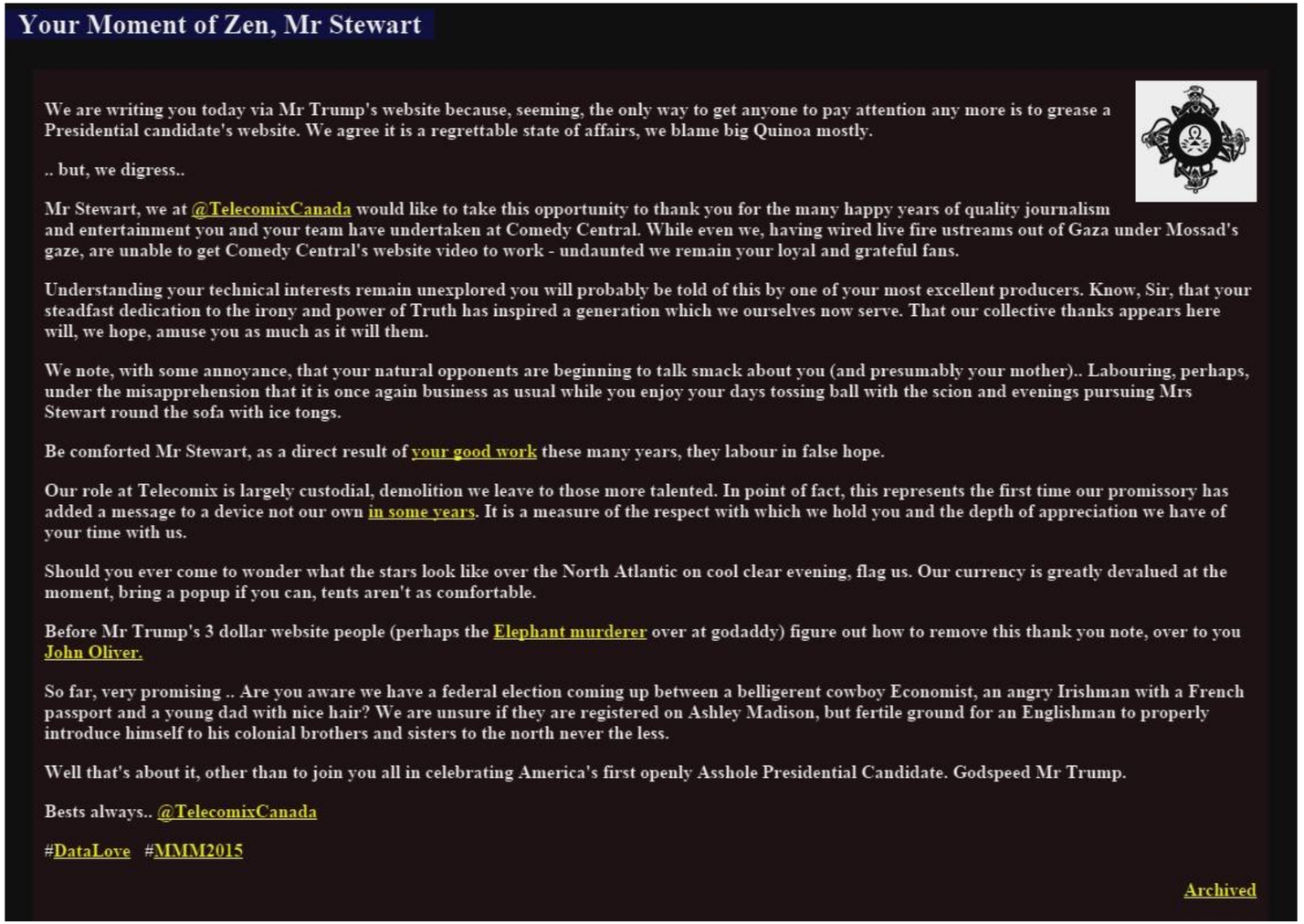
Detecting Website Defacements through Image-based Object Recognition

THE COMPUTER SECURITY GROUP AT UC SANTA BARBARA

Kevin Borgolte kevinbo@cs.ucsb.edu
Christopher Kruegel chris@cs.ucsb.edu
Giovanni Vigna vigna@cs.ucsb.edu

August 13th, 2015
USENIX Security 2015

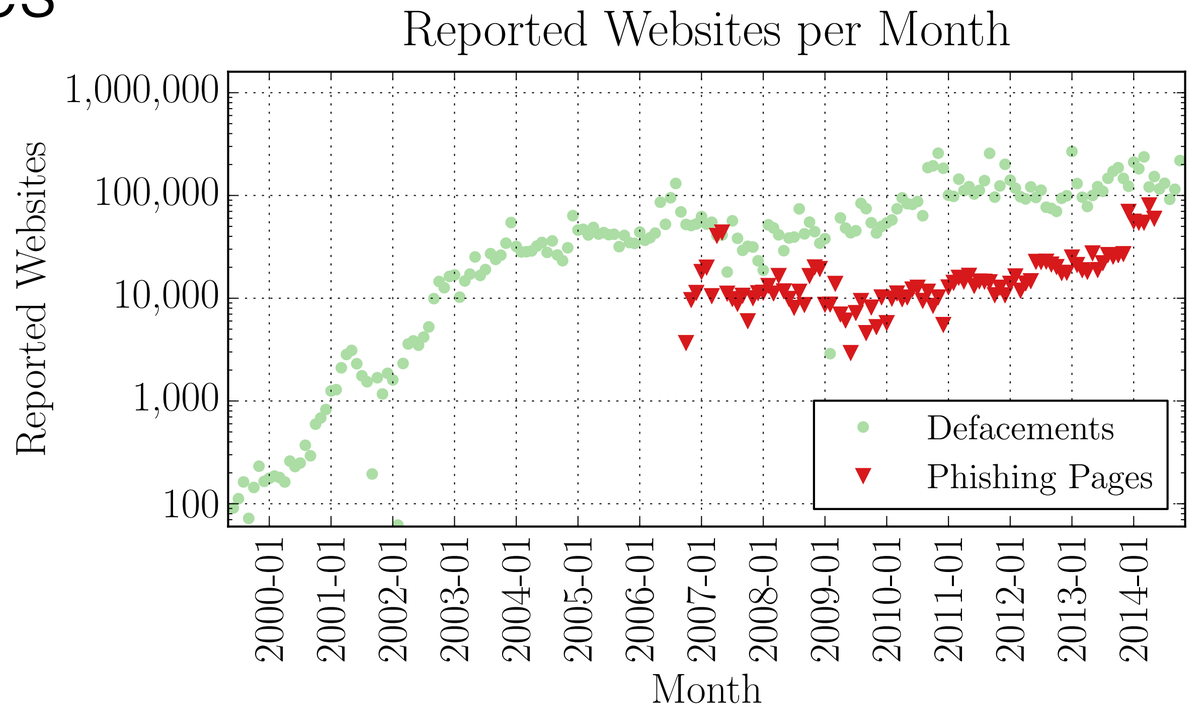
Defacements



Source: The Register, August 3rd 2015, http://www.theregister.co.uk/2015/08/03/trump_website_hacked/

Defacements: Scale

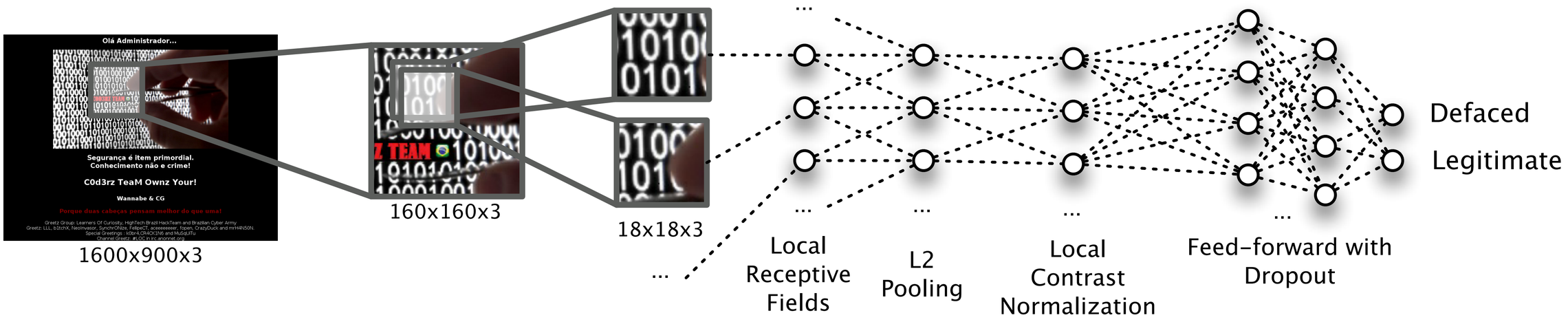
- Prolific defacers:
  Team System Dz, 2,800 websites in 10 months (~10/day)
- Over 4,700 manually-verified defacements each day (Zone-H)
- Defacements to Phishing Pages
  Average: ~7 to 1
  Maximum: ~33 to 1
- Over 53,000 websites from top 1 million lists were defaced in 2014



Approach

- Prior work looks at websites' code, host-based IDS etc.
  - Compares to prior version / known good state
- MEERKAT: Visually, like a human analyst
  - Render website in browser
  - Take screenshot
  - Does the screenshot looks like a defacement?
- No previous version of website needed
- No manual feature engineering

Approach: Deep Neural Network



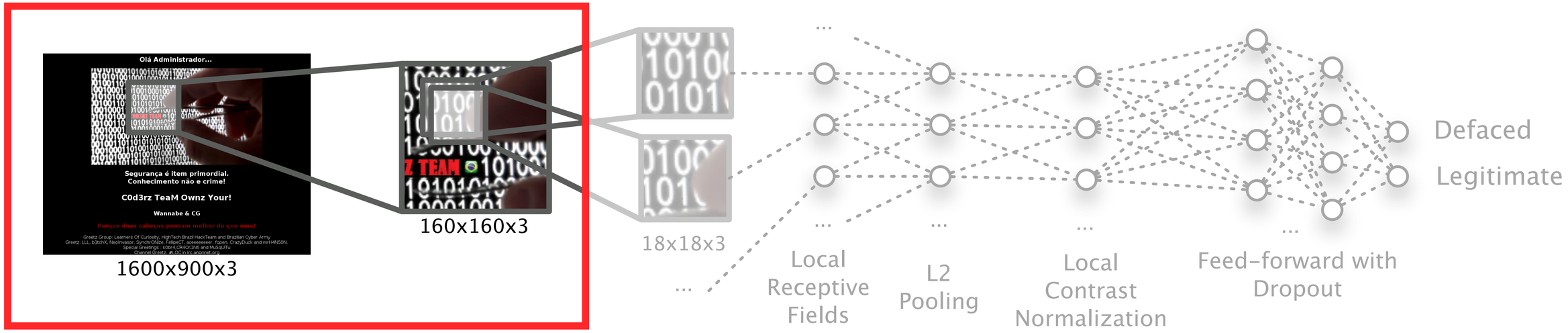
Screenshot Collection

Window Extraction

Feature Learning

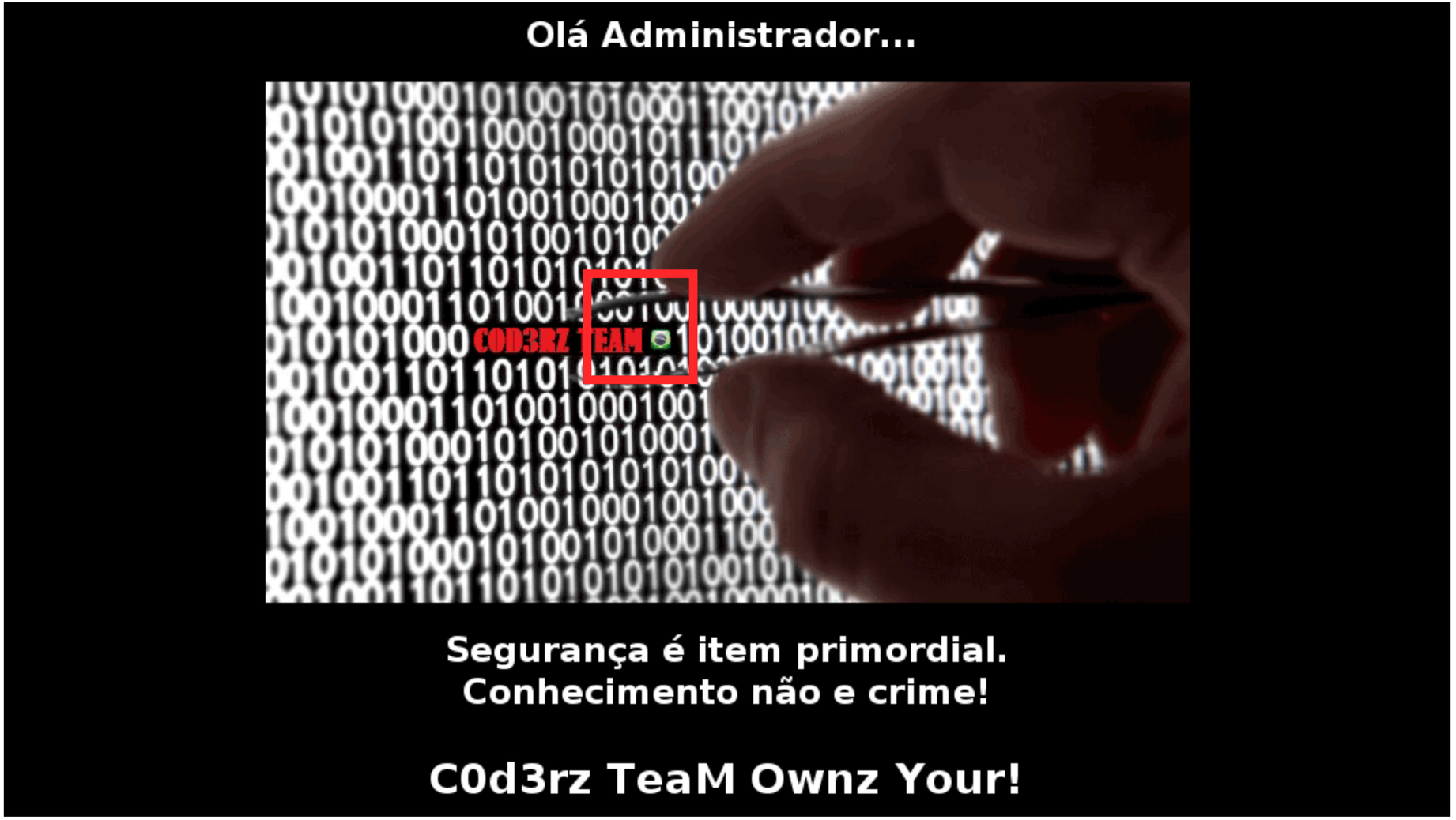
Classification

Approach: A Window “Into” The Defacement



- Full-size screenshots impractical; window “into” defacement instead
- Size of window?
  - Too large ⇒ overfit (high variance)
  - Too small ⇒ underfit (high bias)
- Extract window from which part of the screenshot?

Approach: Representative Window Extraction (1)



Approach: Representative Window Extraction (2)

- Sample windows from 2-dimensional Gaussian distribution
  - Bias heavily toward center of page
  - If outside of screenshot, resample
- Why?
  - Center of page is descriptive!
  - Non-trivial to poison training set

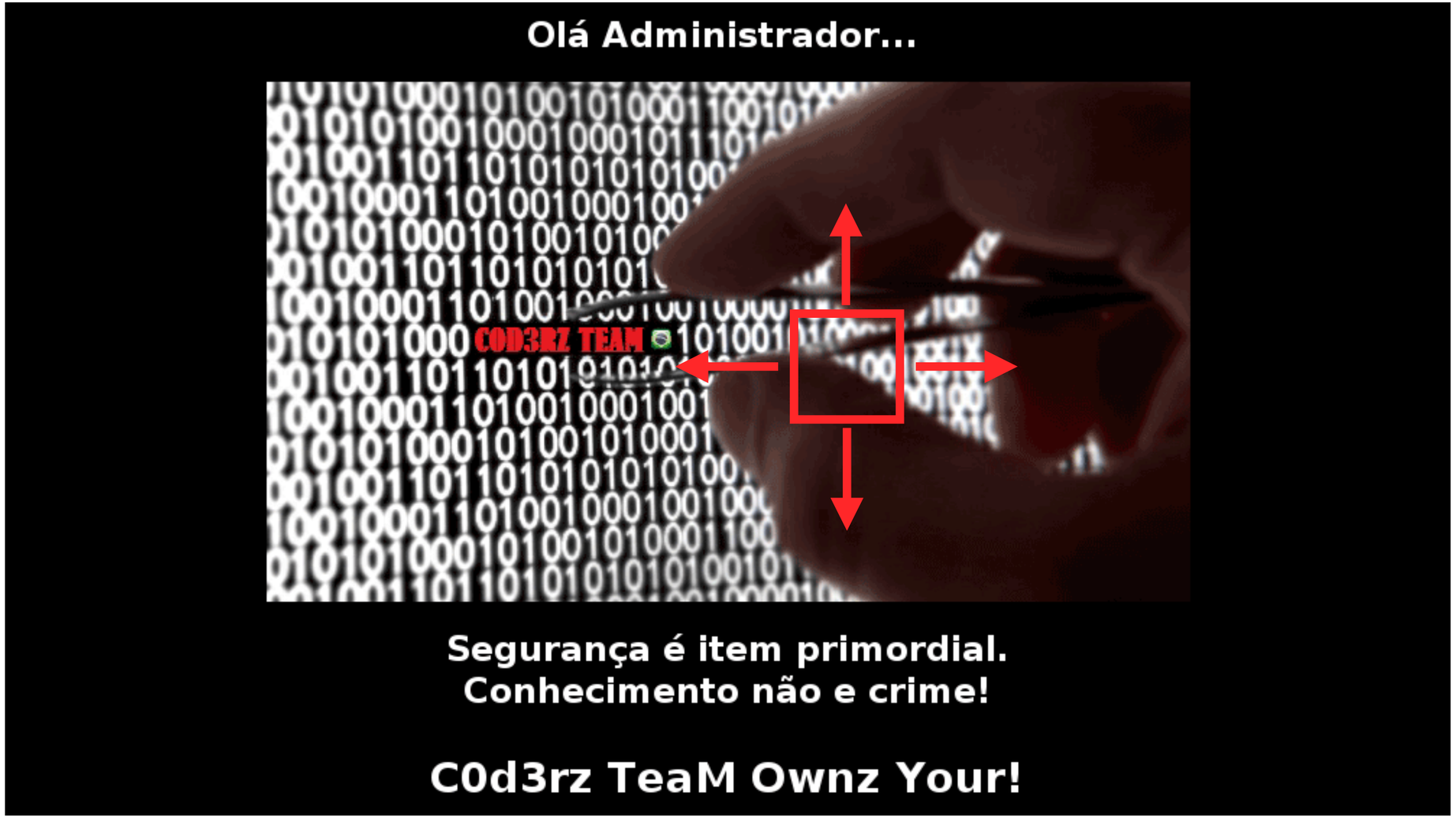
Approach: Deep Neural Network

- Feature Learning:
  Stacked Auto-Encoders
- Classification:
  Feed-Forward Neural Network with Dropout
- Implemented on-top of Caffe
- Trained on GPU, training time in weeks

Approach: Features Learned

- Color combinations
  - Green on black? Black on white/bright gray?
- Letter combinations
  - Broken and mixed encodings
- Leetspeak
  - “pwned” or “h4x0red”
- Typographical and grammatical errors
  - “greats to” or “visit us in our website”
- Defacement group logos

Approach: Detection



Evaluation

- Dataset
  - 10,053,772 defaced websites = positives
    - Defaced websites manually verified by Zone-H
  - 2,554,905 legitimate websites = negatives
    - Legitimate websites not verified, might be defaced
- Dataset skewed toward defacements
  - Report Bayesian detection rate (BDR): P(true positive|positive)
- Unverified legitimate websites ⇒ TPR & BDR are lower-bounds!

Evaluation: Traditional

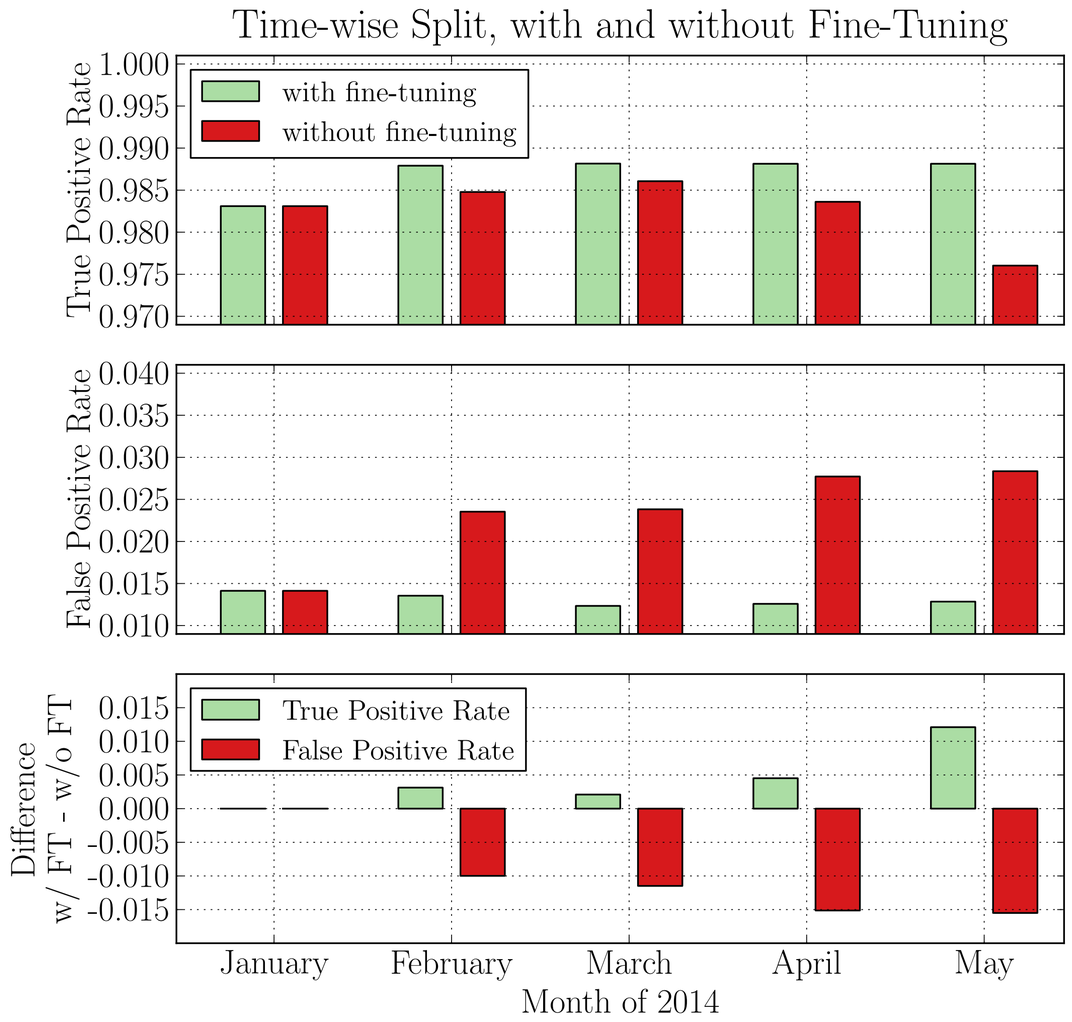
- 10-fold cross-validation
- Results:
  - TPR: avg. 97.878% [97.422%, 98.375%]
  - FPR: avg. 1.012% [0.547%, 1.419%]
  - BDR: avg. 99.716% [99.603%, 99.845%]
- Traditional evaluation has problems:
  - Same defacement possibly in two bins
  - Defacements from 1998 vs. 2014

Limitations

- Fingerprinting and delayed defacements
- Tiny defacements
- Huge advertisements
- Concept drift (natural and adversarial)
  - Major: learn new features from new data (no feature engineering)
  - Minor: adjust weights of deeper classification layer

Limitations: Minor Concept Drift & Fine-Tuning

- Train on Dec 2012 to Dec 2013
  - 1.78 million defacements
  - 1.76 million legitimate pages
- Test on Jan to May 2014
  - 1.54 million samples, 50/50 split
- Fine-tune Jan, Feb, Mar, Apr
- BDR in Jan: 98.583%
  - w/o FT drops to 97.177%
  - w/ FT increases to 98.717%
- Team System Dz started Jan 2014!



Conclusion

- Introduced MEERKAT
- Learns features automatically, match domain knowledge
- Does not require prior version of website
- Outperforms state of the art
- Gracefully tackles minor and major concept drift

Thank you for your attention!

Questions?

kevinbo@cs.ucsb.edu
http://kevin.borgolte.me
twitter: @caovc