Predicting Roles from Computer Logs using Recurrent Neural Networks

Aaron Tuor, Sam Kaplan, Brian Hutchinson — Western Washington University

Nicole Nichols and Sean Robinson — Pacific Northwest National Laboratory

Overview

Motivation

- Knowledge of user attributes can benefit applications.
  - Customization
  - Managing Resources
  - Security

Approach

- Infer unknown user attributes from information contained in network logs.
- Train Long Short Term Memory Network (LSTM) on sequences of user actions.
- Predict user attributes online.

Background

Cert Insider Threat Dataset

- Synthetic data generated with user models.
- 4000 users, 516 days, 135 million events total.
- Email, web, logon, file and device usage events.
- Accompanying user meta data:
  - Job Title
  - Project
  - Team
  - Supervisor
  - Department
  - Personality Score

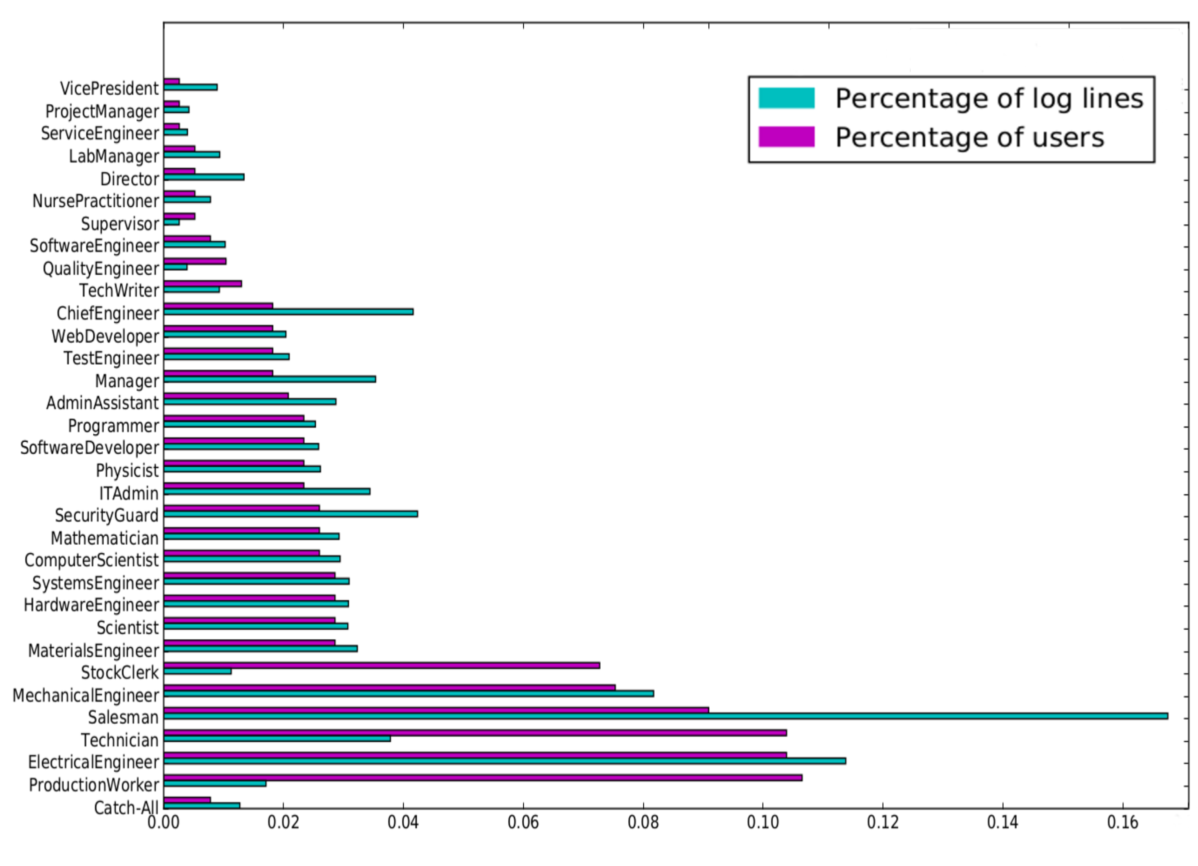
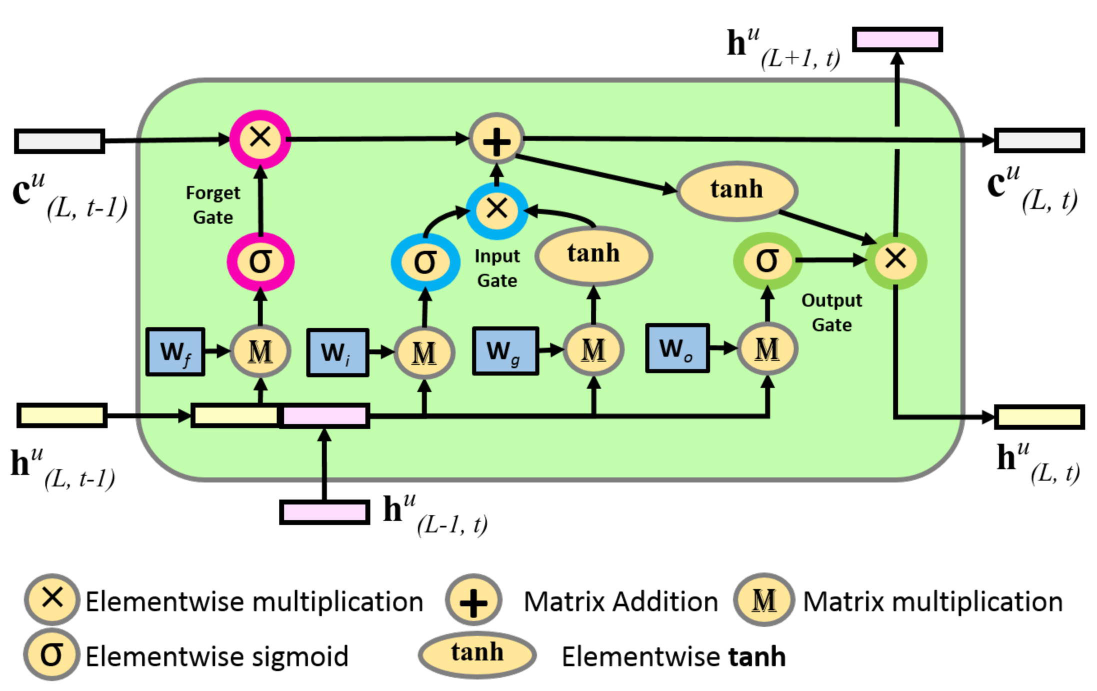


Figure: Distribution of roles by user and line.

Inside an LSTM Layer[a]



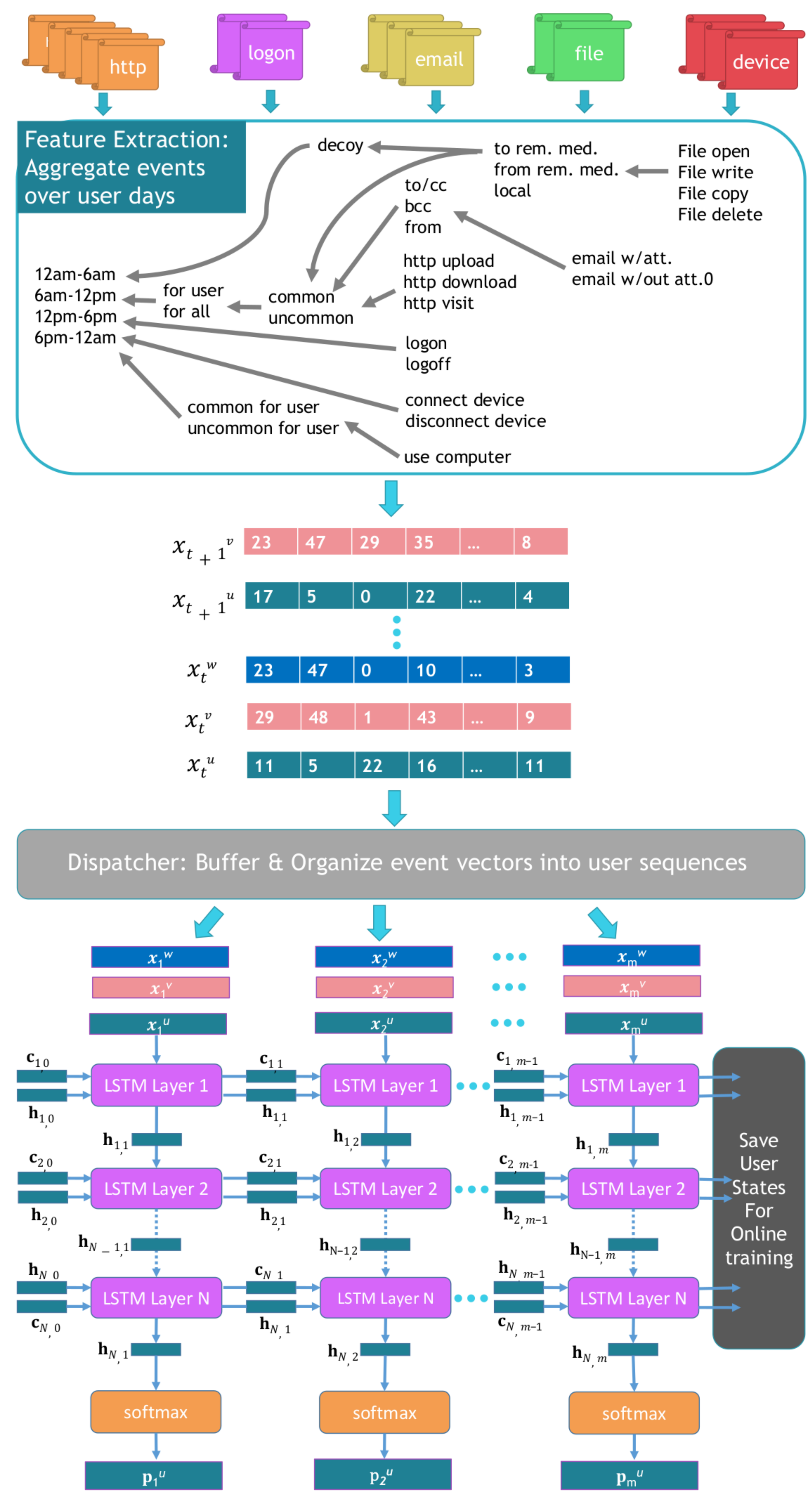
[a] To avoid clutter, bias is not depicted.

Model

Online Training and Prediction

1. Aggregate user features from logging sources.
2. Organize features into sequences of user-day vectors.
3. Train LSTM classifier concurrently on user sequences.

- $\mathbf{x}_t^u$: User u's feature vector for day t.
- $\mathbf{p}_t^u$: Distribution of roles for user u for day t.



LSTM Classifier Equations

Model parameters

$$p_{t,k}^u = \frac{\exp(h_t^u \mathbf{W}_p + \mathbf{b}_p)_k}{\sum_j \exp(h_t^u \mathbf{W}_p + \mathbf{b}_p)_j}, \text{ where}$$

$$\mathbf{h}_t^u = \mathbf{o}_t^u \odot \tanh(\mathbf{c}_t^u)$$

$$\mathbf{c}_t^u = \mathbf{f}_t^u \odot \mathbf{c}_{t-1}^u + \mathbf{i}_t^u \odot \mathbf{g}_t^u, \text{ and}$$

$$\mathbf{f}_t^u = \sigma\left(\mathbf{W}_{f,x}\mathbf{x}_t^u + \mathbf{W}_{f,h}\mathbf{h}_{t-1}^u + \mathbf{b}_f\right)$$

$$\mathbf{i}_t^u = \sigma\left(\mathbf{W}_{i,x}\mathbf{x}_t^u + \mathbf{W}_{i,h}\mathbf{h}_{t-1}^u + \mathbf{b}_i\right)$$

$$\mathbf{o}_t^u = \sigma\left(\mathbf{W}_{o,x}\mathbf{x}_t^u + \mathbf{W}_{o,h}\mathbf{h}_{t-1}^u + \mathbf{b}_o\right)$$

$$\mathbf{g}_t^u = \tanh\left(\mathbf{W}_{g,x}\mathbf{x}_t^u + \mathbf{W}_{g,h}\mathbf{h}_{t-1}^u + \mathbf{b}_g\right)$$

Experimental Setup

- Simulate online scenario for 90 days data.
- 80/10/10 train/dev/test split over users.
- Cross-entropy objective.
- Random hyperparameter search.

Results and Analysis

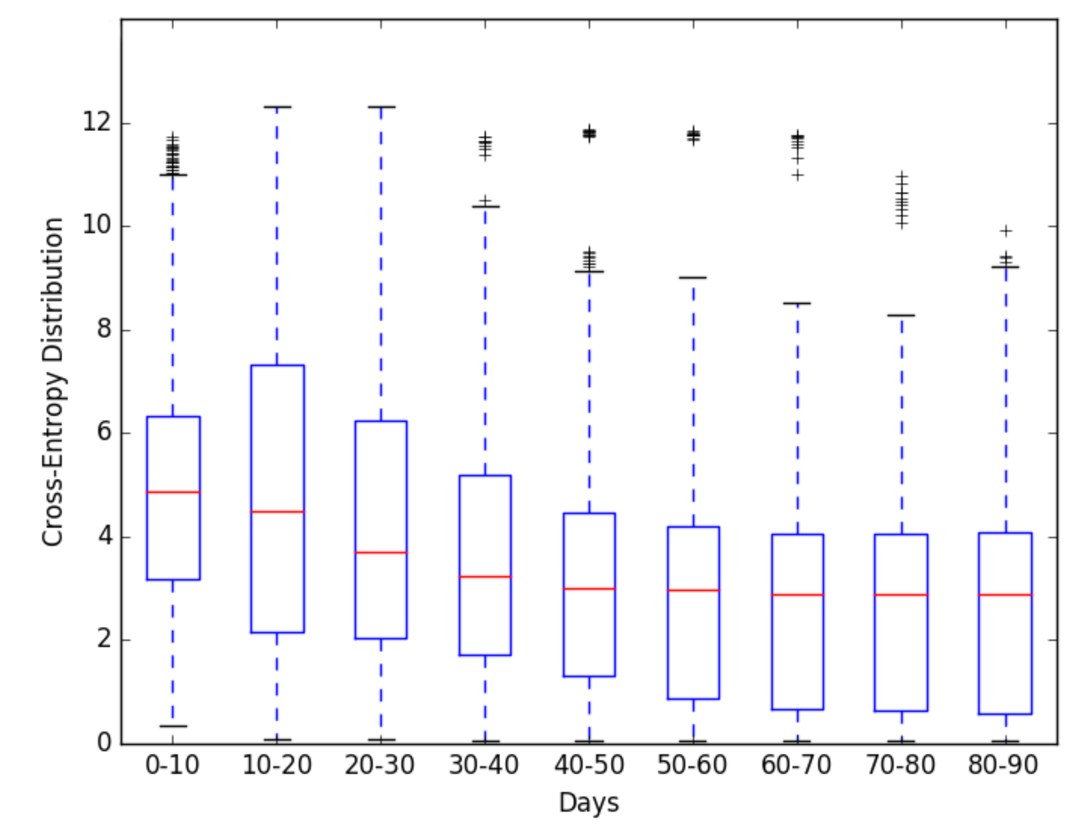


Figure: Cross-entropy as a function of time.

- Performance starts out poor but steadily improves.
- Predictions improve until day 40-50.
- Achieve 38% accuracy after 90 days training.
- 11% baseline to predict majority class role.

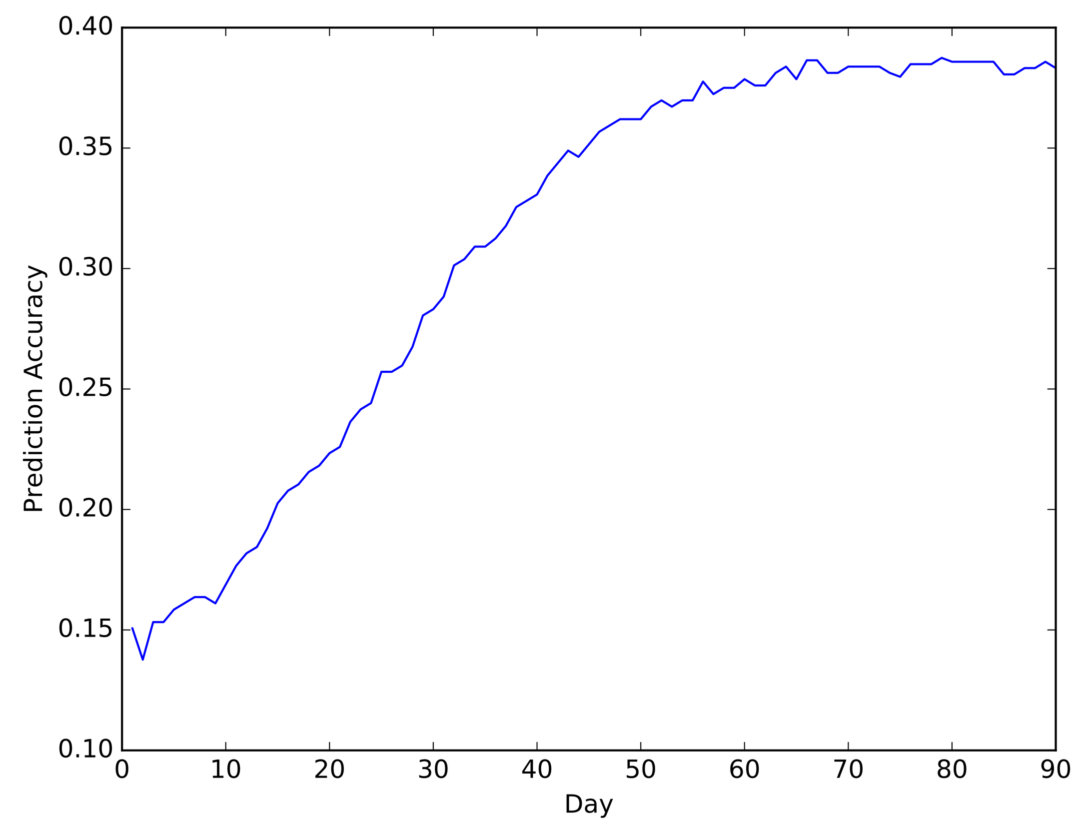


Figure: Accuracy as a function of time.

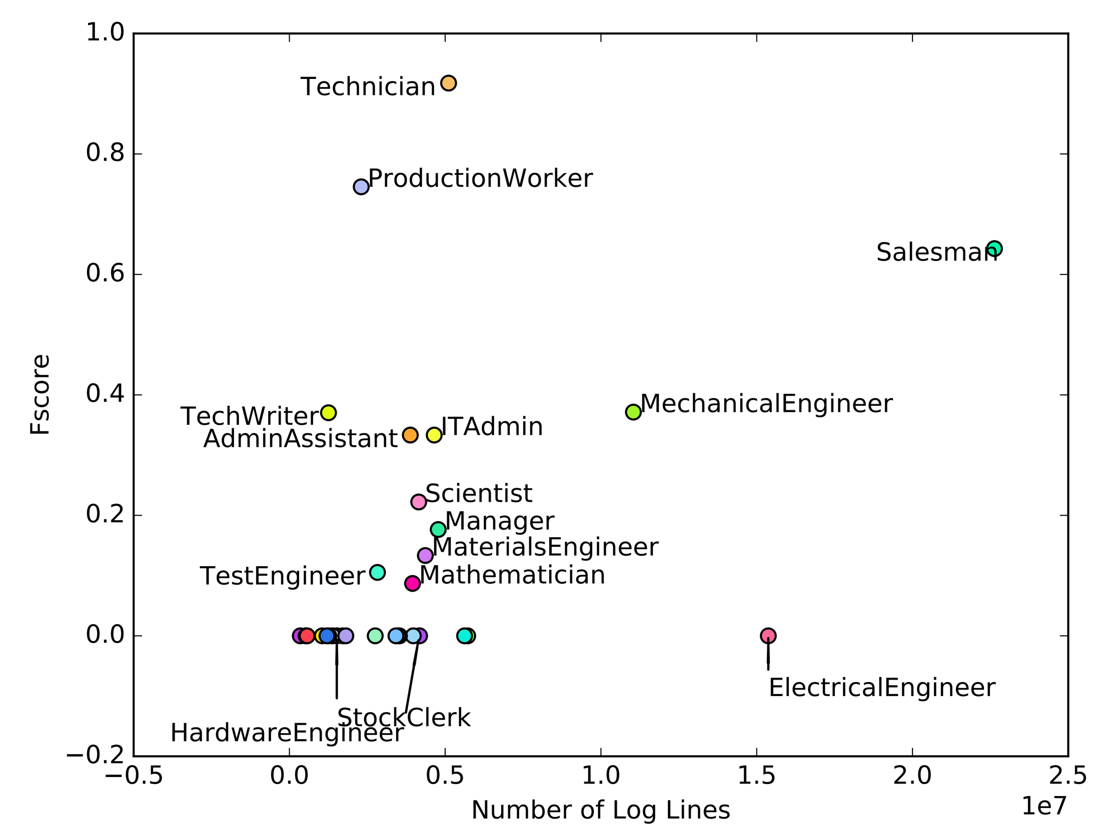


Figure: F-score as a function of log lines.

- Classifiers tend to do better with many examples.
- Expect linear correlation between f-score and log lines.
- Overperforming classes may show distinctive behavior.

Conclusions and Future Work

- 38% accuracy on 33-way classification.
- Method trivially generalizes to other attributes.
- Address class imbalance by random re-sampling.
- Evaluate on real world data sets.