Anomaly Detection and Prediction for Computer Health

Chenyu You, Qiwen Wang, Chao Sun
Mentor: Rok Sosic, Hongwei Wang

Problem Overview

- High service availability is crucial for computer systems
- Anomalous events occur relatively infrequently
- **However**, their consequences can be often in a negative sense
- Also, system error can be seen as a form of **gray failure**, which are fairly subtle failures that are hard to be detected, even when applications are afflicted by them.
- Goal: **apply the ML-based approach to improve computer system availability in network domain**

Key Challenges

- Defining a **representative** normal region
- The boundary between normal and outlying behavior is often not precise
- The exact notion of an outlier is different for different domains
- Availability of labeled data for training/validation/test - (**Major Issue**)
- Data might contain noise
- Abnormal Behavior keeps evolving
- Malicious adversaries

Problem Formulation

- **Task Overview**
  - **Detecting** network anomaly: Determine the anomaly of the current timestamp using the features from the history data.
  - **Predicting** network anomaly: Predicting the anomaly of the future status using the feature from the history data.
- **Dataset Generation**
  - Anomaly network traffic generation:
    - High Bandwidth usage (Iperf)
  - Normal network traffic generation:
    - Send and receive data from a server (Curl)

DataSet - Real-life Server Log

- Real Server Log Dataset:
  - Record all events and activity occurred in the server (**~7 days**)
  - Highly correlated features: network send & received bytes/packets
  - Weakly correlated features: CPU usages, tcp stats, etc.

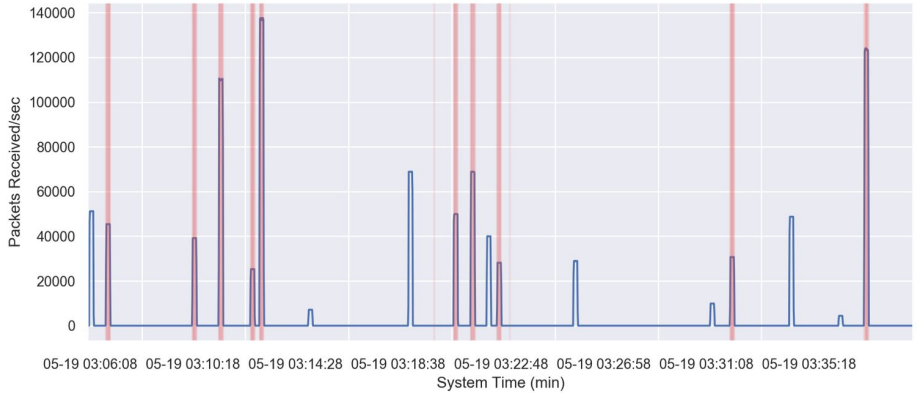

Figure 1: Examples of network traffic dataset in certain time interval

Experiments

- Deal with unbalanced data: Perform oversampling and downsampling
- ML-based Model:
  - LSTM: capturing long-term dependencies in the sequences but have **information loss**
  - Bidirectional LSTM: efficiently make use of past features (via forward states) and future features (via backward states) for a specific time frame
  - Attention mechanism: assign a high attention weight to different parts of source sequence
- Evaluation metrics: precision, recall, F1 score, and AUC

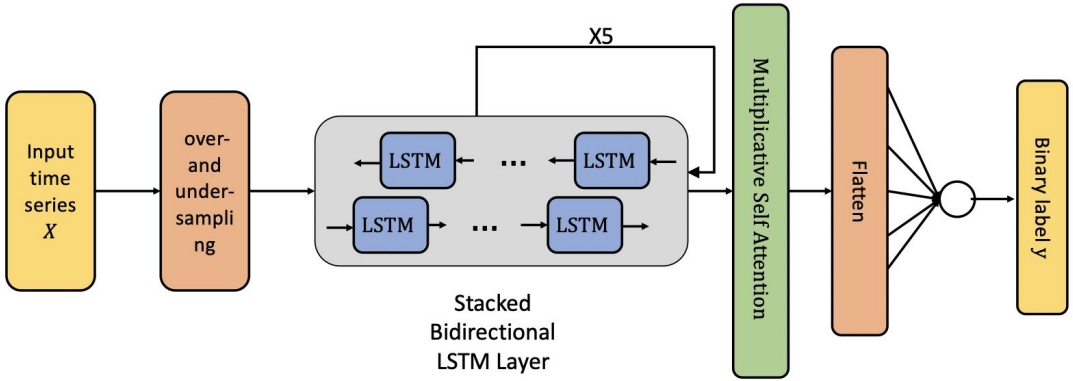

Figure 2: Stacked Bidirectional Self-attention LSTM network architecture.

Anomaly Detection Results

- LSTM-based methods achieved > 40% **better** F1 score than conventional approaches like decision tree
- Incorporate more history data and weakly correlated features improves the model performance
- Best results are achieved with bi-directional LSTM+self attention mechanism (sBiLSAN) with 96% F1 score.

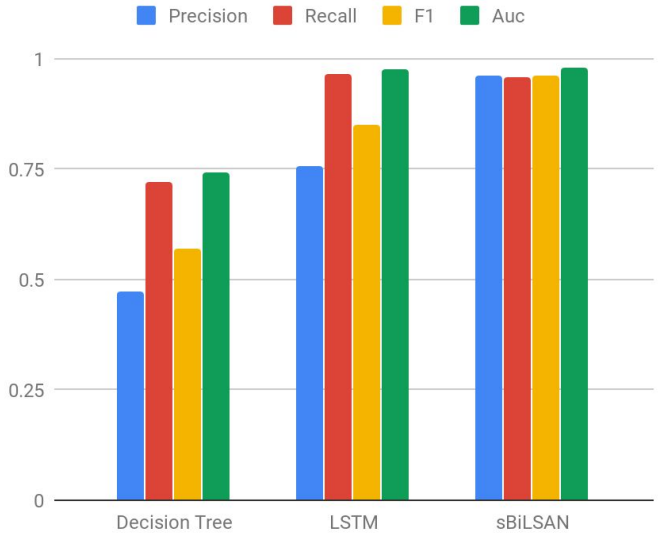

Figure 3: Comparison of detection accuracy between conventional and ML-based methods.

Anomaly Detection Results

- Using the previous 60 seconds data and applying multiplicative attention achieve the best result.

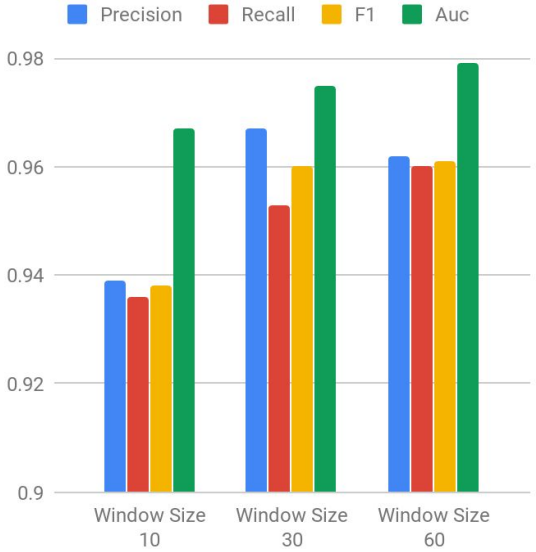

Figure 4: The effects of different history window size on detection accuracy with sBiLSAN.

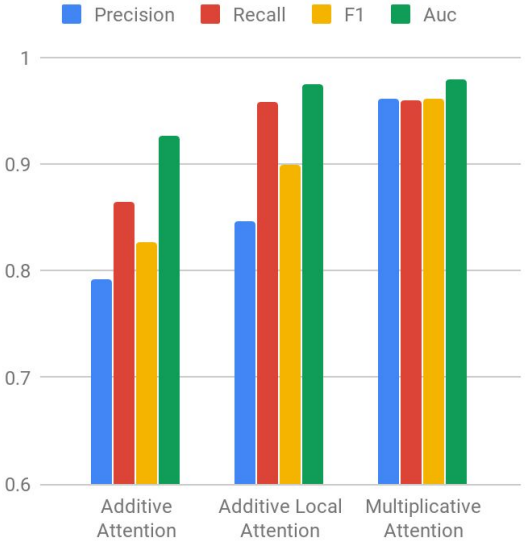

Figure 5: The effect of different self-attention mechanisms on detection accuracy.

Anomaly Prediction Results

- The LSTM model surpasses the baseline models by 80%
- Bidirectional LSTM outperforms the self attention model. The result from 60 seconds ago may not providing much information. The sequence of the data matters more.
- Bidirectional LSTM performs roughly as well as that in the detection task

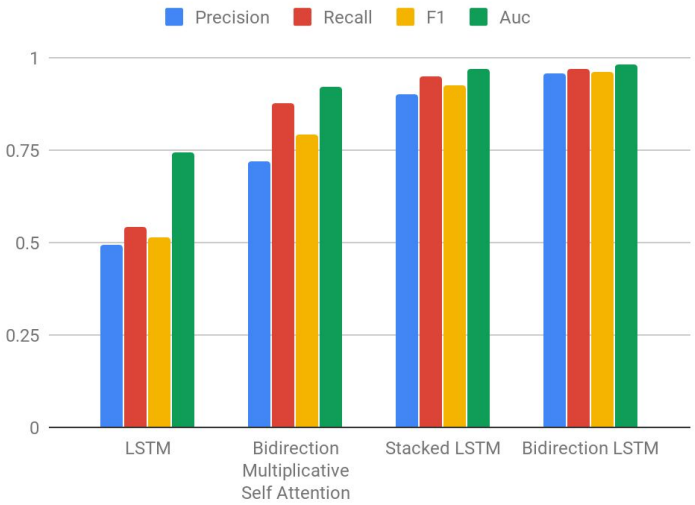

Figure 6. LSTM Prediction Result. Using the features of the previous 60 seconds to predict the state of the future 10 seconds

Prediction Visualization Results

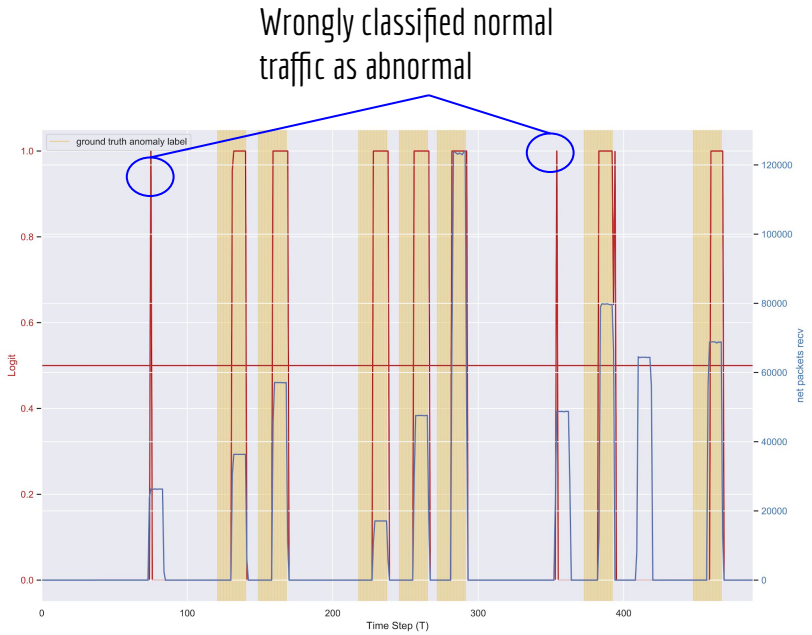

Figure 7: Prediction result on test data with history window size 10.

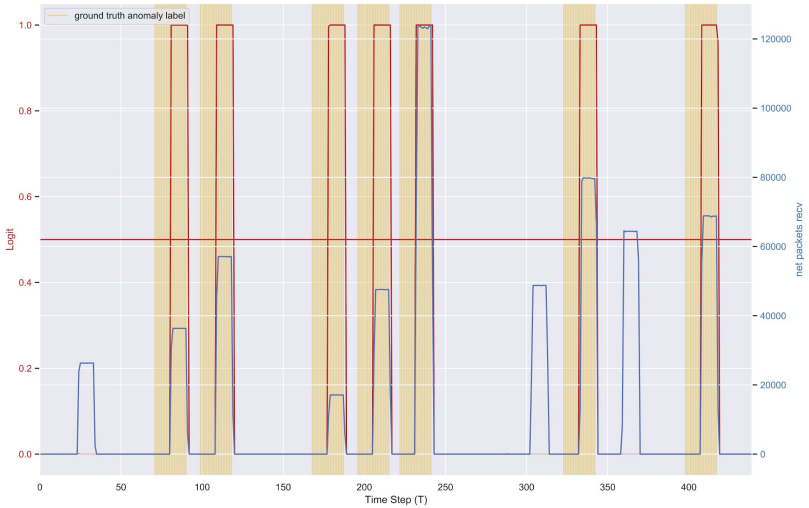

Figure 8: Prediction result on test data with history window size 60.

Conclusion

- Created real-world network traffic dataset with anomaly
- Compared several conventional ML models with LSTM models for anomaly detection and prediction
- The stacked Bidirectional LSTM model with self-attention network achieved the best result of 96% F1 score, produce around 40% performance gain over

Thank you, Rok and Hongwei!