Use Neural Structured Learning for Beaconing Detection

Hanjun Li (hl3339), Boyu Liu (bl2788), Yitao Liu (yl4343), Yiyang Sun (ys3284), Banruo Xie (bx2168)

** Names ordered alphabetically by team members' last and first name; every member has an equal contribution to the project.*

1 Introduction

In cybersecurity, malicious software designed to harm or extract information from programmable devices or networks are considered "malware". Due to the modern-day firewalls, "beaconing", a malware variant that manages to bypass firewalls by reconnecting to an intermediate "command & control" (C&C) server, has emerged.

We implement a Neural Structured Learning (NSL) model [1, 2] to effectively detect the beaconing data, and it has better performance in accuracy and F1 score comparing to other models.

2 Data

2.1 Data Contents

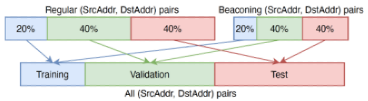
The dataset includes differentes attributes of netflow data, including categorical attributes like source and destination addresses (SrcAddr and DstAddr), start time (StartTime), protocol type, direction, etc.; and numeric attributes like duration, total package, total bytes, etc.

Besides the features in the given dataset, we also include two features indicating whether source and destination addresses are bogon.

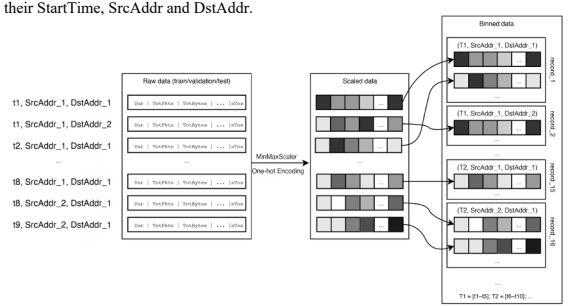
Labels of regular/ beaconing are determined by (SrcAddr, DstAddr) pairs. StartTime, SrcAddr and DstAddr are used together to bin the data (refer to the *Data Binning Procedure* figure).

2.2 Data Processing

We split original dataset into training, validation and test sets based on source and destination addresses pair. Proportions of training, validation and test pairs are 20:40:40 for both regular and beaconing pairs.



Then, we bin the data by a time window of 1 second, data are put into bins based on their StartTime, SrcAddr and DstAddr.



Data Binning Procedure

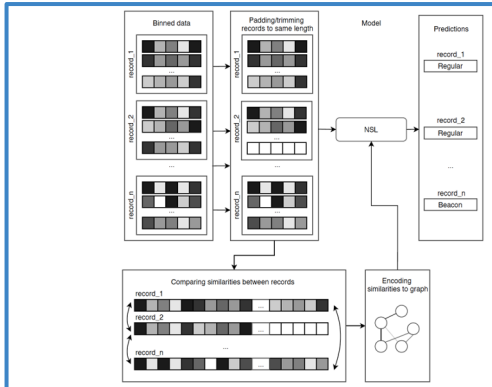
3 Methodology

For the binned data, one approach is calculating the aggregate statistics (summation, averaging, counting, etc.) [3], the other approach is using the sequential data in each bins directly.

The NSL model uses the sequential data as input. In the model training stage, NSL applies the graph regularization to the base model (which can be various types of neural networks), to improve the model performance [1].

The base model we chose is the Long Short-Term Memory (LSTM) model, which can handle the sequential data as input. The graph is generated by the training data using the following procedure:

- Flatten each binned sequential record to 1D record.
- Calculate similarities between the flattened records.
- Generate graph, in which an edge is added when the similarity score between two records is above a threshold; set edge weight to the similarity score.



Our NSL Model

4 Result

We include other models for performance comparison: Logistic Regression (LR) and Artificial Neural Network (NN), which use aggregated data; and LSTM model, which uses sequential data.

	Recall	Precision	Accuracy	F1
LR	87.540%	**99.986%**	90.583%	0.93350
NN	87.558%	**99.986%**	90.596%	0.93360
LSTM	89.530%	99.893%	92.022%	0.94428
NSL	**95.440%**	99.285%	**96.038%**	**0.97325**

5 Conclusion

- NSL model performs best in recall, accuracy, and F1 score.
- The aggregated models (LR and NN) have a lower F1 score than the non-aggregated models (LSTM and NSL); due to the information loss when calculating the aggregation data.
- NSL has a better F1 score than the LSTM; because of the introduction of graph structure during the training stage.

Reference

[1] T. D. Bui, S. Ravi, and V. Ramavajjala, "Neural Graph Learning: Training Neural Networks Using Graphs," in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, Feb. 2018, pp. 64–71, doi: 10.1145/3159652.3159731.

[2] "Graph regularization for sentiment classification using synthesized graphs," TensorFlow. https://www.tensorflow.org/neural_structured_learning/tutorials/graph_keras_lstm_imdb

[3] R. Kozik and M. Choraś, "Pattern Extraction Algorithm for NetFlow-Based Botnet Activities Detection," Security and Communication Networks, Oct. 17, 2017.

COLUMBIA UNIVERSITY
Data Science Institute

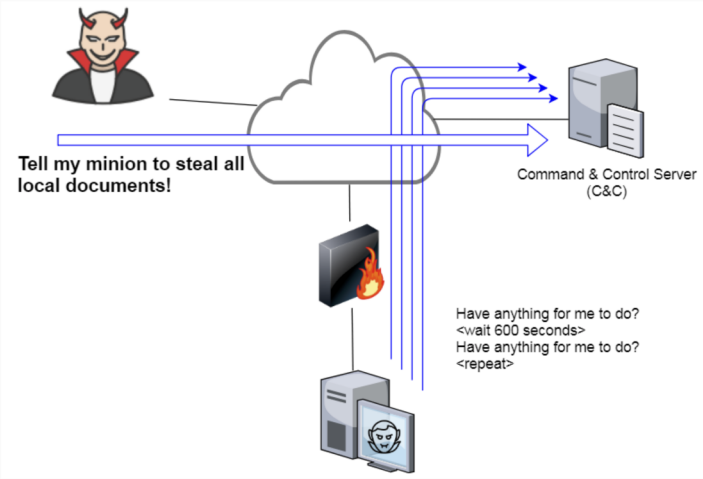
Use Neural Structured Learning for Beaconing Detection

Agenda

- Introduction
- Data Contents
- Data Processing
- Model
- Summary

Introduction

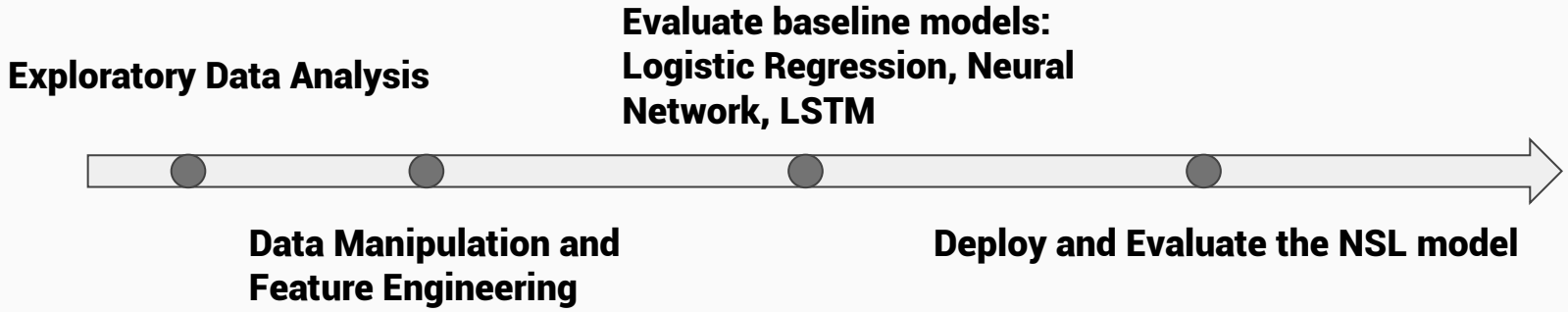
In cybersecurity, malicious software designed to harm or extract information from programmable devices or networks are considered “malware”. Due to the modern-day firewalls, “beaconing”, a malware variant that manages to bypass firewalls by reconnecting to an intermediate “command & control" (C&C) server, has emerged.



Demonstration of the basic setup for beaconing [1]
Figure source: www.activecountermeasures.com/blog-beacon-analysis-the-key-to-cyber-threat-hunting/.

Goal and Outcome

In the capstone project, we implement a Neural Structured Learning (NSL) model [2, 3] to effectively detect the beaconing data, and it has better performance in accuracy and F1 score comparing to other models.



Data Contents

The summary of dataset is shown in the figure on the right. The dataset includes different attributes of netflow data, including categorical attributes like source and destination addresses (SrcAddr and DstAddr), start time (StartTime), protocol type, direction, etc.; and numeric attributes like duration, total package, total bytes, etc.

```
Data columns (total 13 columns):
 #   Column      Non-Null Count   Dtype
---  ------      --------------   -----
 0   StartTime   105469 non-null  object
 1   Dur         105469 non-null  float64
 2   Proto       105469 non-null  object
 3   SrcAddr     105469 non-null  object
 4   Sport       102676 non-null  object
 5   Dir         105469 non-null  object
 6   DstAddr     105469 non-null  object
 7   Dport       102676 non-null  object
 8   State       105469 non-null  object
 9   sTos        78421 non-null   float64
 10  dTos        24255 non-null   float64
 11  TotPkts     105469 non-null  int64
 12  TotBytes    105469 non-null  int64
```

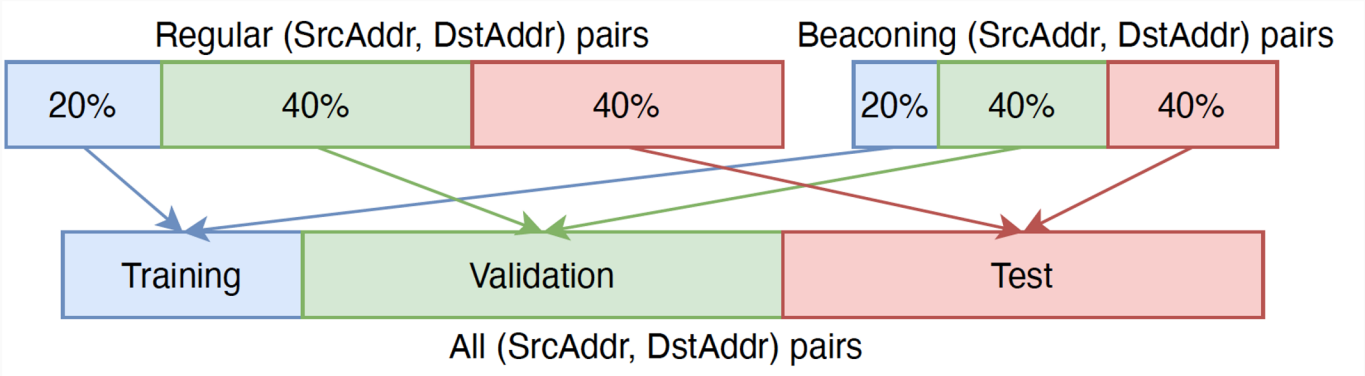
Data Contents

Besides the features in the given dataset, we also include two features indicating whether source and destination addresses are bogon.

Labels of regular/ beaconing are determined by (SrcAddr, DstAddr) pairs. StartTime, SrcAddr and DstAddr are used together to bin the data (which will be explained in the following slides).

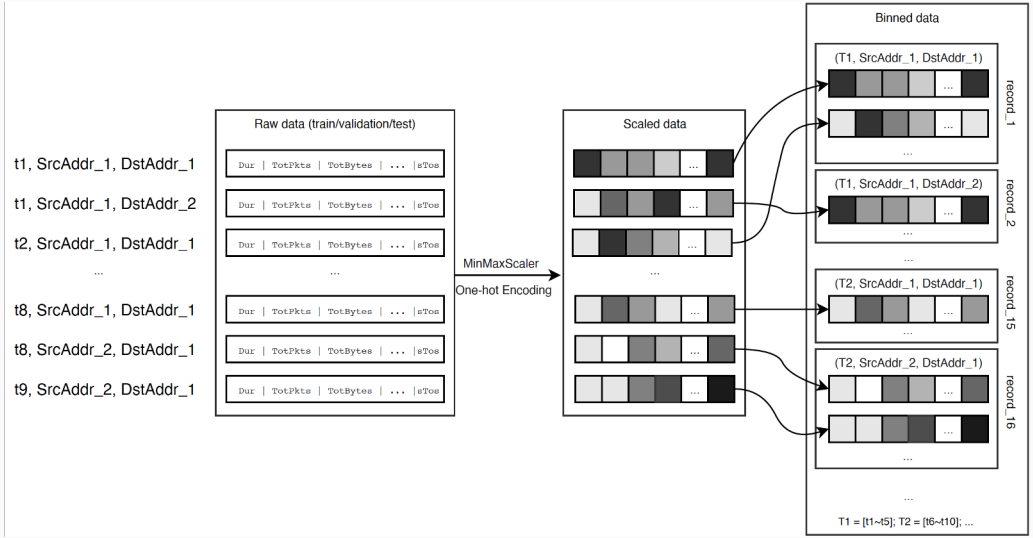
Data Processing

We split original dataset into training, validation and test sets based on source and destination address pairs. Proportions of training, validation and test pairs are 20:40:40 for both regular and beaconing pairs. (As shown in the figure below).



Data Processing

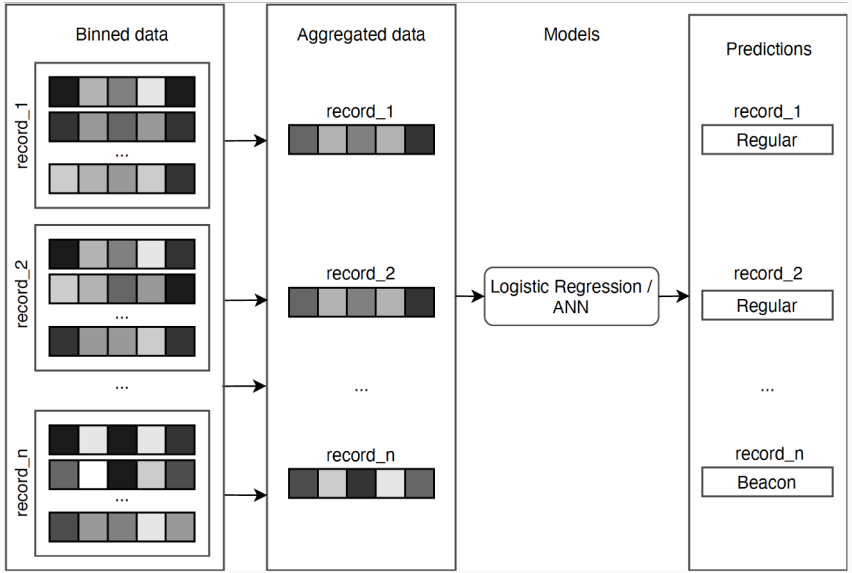
Then, we binned data by a time window of 1 second, data are put into bins based on their StartTime, SrcAddr and DstAddr. (Data binning procedure is shown in the figure on the right).



Model

For the binned data, one approach is calculating the aggregate statistics (summation, averaging, counting, etc.) [4], the other method is to use the sequential data in each bin directly.

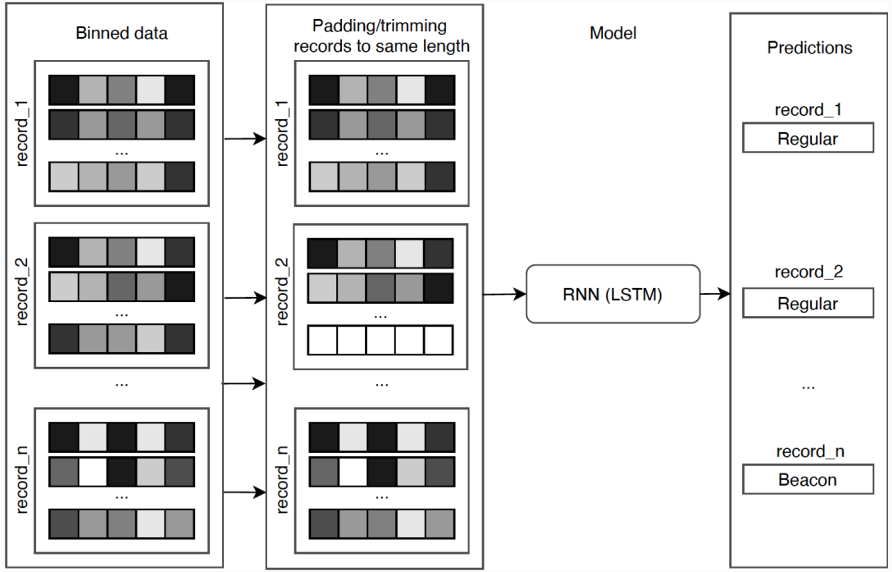
Pipeline of using *aggregated* data as input (as shown on the right).



Model

For the binned data, one approach is calculating the aggregate statistics (summation, averaging, counting, etc.) [4], the other method is to use the sequential data in each bin directly.

Pipeline of using *sequential* data as input (as shown on the right).



Model

We tried 4 models：

- Using aggregated data:
  - Logistic Regression (LR)
  - Artificial Neural Network (NN)
- Using sequential data:
  - Long Short-Term Memory (LSTM) model
  - **Neural Structured Learning (NSL) model**

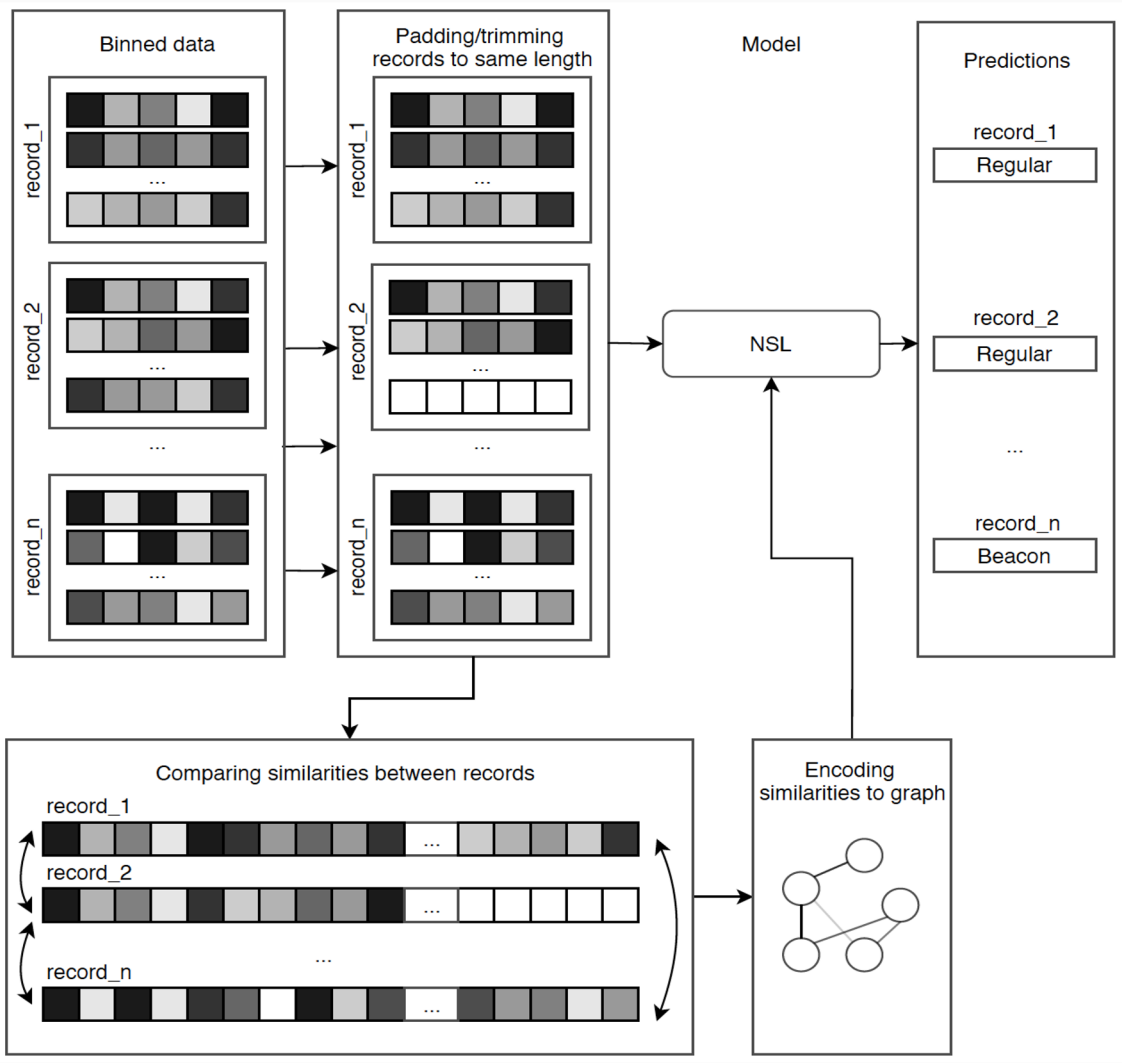
Model - NSL

In the model training stage, NSL applies the graph regularization to the base models (which can be various types of neural networks), to improve the model performance [2].

The base model: Long Short-Term Memory (LSTM) model, which can handle the sequential data as input.

The graph is generated by the training data using the following procedure:

- Flatten each binned sequential record to 1D record.
- Calculate similarities between the flattened records.
- Generate graph, in which an edge is added when the similarity score between two records is above a threshold; set edge weight to the similarity score.


Binned data
record_1
record_2
...
record_n
Padding/trimming records to same length
Model
NSL
Predictions
record_1
Regular
record_2
Regular
...
record_n
Beacon
Comparing similarities between records
Encoding similarities to graph

Result

	Recall	Precision	Accuracy	F1
LR	87.540%	**99.986%**	90.583%	0.93350
NN	87.558%	**99.986%**	90.596%	0.93360
LSTM	89.530%	99.893%	92.022%	0.94428
NSL	**95.440%**	99.285%	**96.038%**	**0.97325**

Conclusion

- NSL model performs best in recall, accuracy, and F1 score.
- The aggregated models (LR and NN) have a lower F1 score than the non-aggregated models (LSTM and NSL); due to the information loss when calculating the aggregation data.
- NSL has a better F1 score than the LSTM; because of the introduction of graph structure during the training stage.

Future works

- Use more data from the CTU-13 dataset, and try a better way to solve the class imbalance problem.
- Find a better way to build the graph.
- Optimize the model training time without hurting the performance.
- Reorganize our code so that it can automate the train and detect process.

Reference

[1] Brenton, Chris. “Beacon Analysis - The Key to Cyber Threat Hunting.” *Active Countermeasures*, 21 Feb. 2018, www.activecountermeasures.com/blog-beacon-analysis-the-key-to-cyber-threat-hunting/.

[2] T. D. Bui, S. Ravi, and V. Ramavajjala, “Neural Graph Learning: Training Neural Networks Using Graphs,” in Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, Marina Del Rey, CA, USA, Feb. 2018, pp. 64–71, doi: 10.1145/3159652.3159731.

[3] “Graph regularization for sentiment classification using synthesized graphs,” TensorFlow. https://www.tensorflow.org/neural_structured_learning/tutorials/graph_keras_lstm_imdb

[4] R. Kozik and M. Choraś, “Pattern Extraction Algorithm for NetFlow-Based Botnet Activities Detection,” Security and Communication Networks, Oct. 17, 2017.

Capstone Team

Mentors and instructor: Yiwen Zhang, Emma Pan, Professor Suman Jana

Team member: Hanjun Li (hl3339), Boyu Liu (bl2788), Yitao Liu (yl4343), Yiyang Sun (ys3284), Banruo Xie (bx2168)

** Names ordered alphabetically by team members' last and first name; every member has an equal contribution to the project.*

Thank you