BotNet Detection Using ML

RuntimeTerrors • 12.08.2020

Outline

Steps Involved

- Preprocessing
  - Each instance of data(row) is a pair of IPs which represent a particular flow
  - Packets are grouped by network flow.
  - Aggregate metrics are calculated for each unique flow.
- Feature Engineering and Prediction
  - For each flow additional features are created based on the extracted features to allow better prediction.
  - Then the data is passed into a rule based classifier.

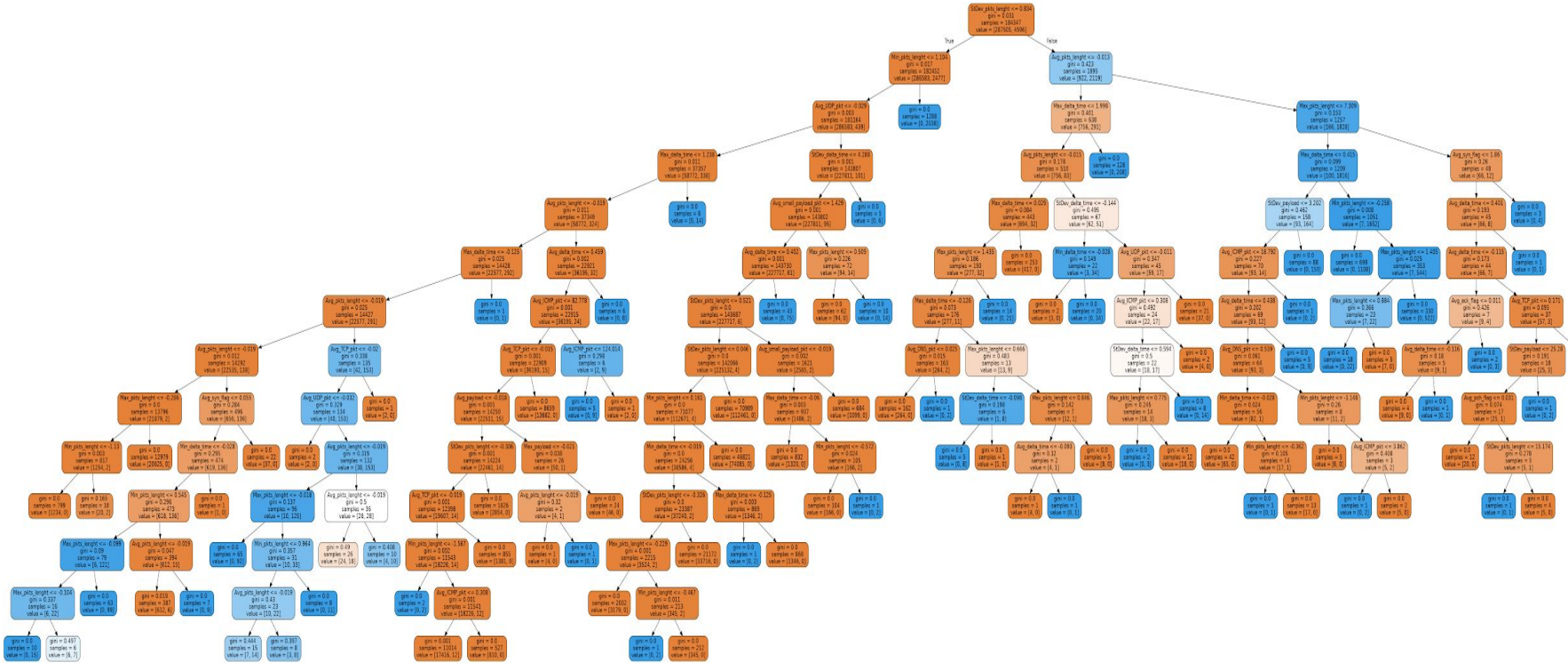
Preprocessing

- For each network capture file, the **Packets are grouped by flow** and the following **Aggregate Features** are extracted.
- Avg_syn_flag Avg_urg_flag Avg_fin_flag Avg_ack_flag Avg_psh_flag Avg_rst_flag Avg_DNS_pkt Avg_TCP_pkt Avg_UDP_pkt Avg_ICMP_pkt **Duration_window_flow** **Avg_delta_time** Min_delta_time Max_delta_time StDev_delta_time **Avg_pkts_length** Min_pkts_length Max_pkts_length StDev_pkts_length Avg_small_payload_pkt Avg_payload Min_payload Max_payload StDev_payload Avg_DNS_over_TCP
- Some additional features were created through **Feature engineering** also, Available in the above list.

Training and Prediction

- Since most of the features extracted are aggregates, we decided that a **Rule Based** approach would we better.
- Scaled the data using a **Standard Scaler**.
- Fit the data on a **Random Forest Classifier**
- **500 estimators** are run on the data in parallel.
- The Prediction is averaged over the output of all these estimators.
- Got very high accuracy, F1 score, recall and precision.

Plot for a single Estimator



Improvements

- Due to the the availability of limited computing resource, the aggregate features were engineered manually and Extracted. The problem with this is some **Important Patterns** in data is ignored/not used.
- To Make the prediction a lot more general it is required to use some sort of **Time Series Based Encoder Neural Network**, Then the output of this neural network can be used for prediction purposes using any suitable model.
- The main advantage about our approach is that it doesn't require **large computing resources** and **can be execute 'on-line'** directly on a stream of packets.
- The performance can be improved further by freezing the model or **implementing these rules in C**.

Business Perspective

- Analysis can be done either on-line directly or offline
- Extension to already existing softwares
- Intrusion Detection Systems require detection to be done on live stream of packets
- Model is very light-weight.
- Royalty deals with already existing IDS software providers.

- In summary, We would like to improve prediction by using a **Time Series Encoder**, Improve performance by **implementing the rules in C** and generating binary, Sell/Market it to **Intrusion Detection Software Providers**.