Ivan Miller - Problem Set 2 for Adversarial AI. Tuesday, Sept. 20, 2022

All three of these problems will concern the following scenario. Suppose you have a network that includes d machines that can all transmit information outside the network. For each minute of the day when the network is active, you have a data point where each is a non-negative integer expressing how many bytes of information were transmitted out of the network by machine i in that minute. Suppose you collect this data for N minutes of network operation. (So you have N data points, and each data point consists of d numbers. These are ordered by time.) We're going to think through some steps for searching this data for suspicious activity.

1. **Imagine that your d machines behave roughly similarly when network operations are normal (with some natural variance), but halfway through your data collection period, one of them became infected with malware and started transmitting a fair amount of extra data out of the network, for many of the subsequent minutes. Describe how you could detect this and identify the infected machine.**
   I would use mean as an estimate to detect the extra data being transmitted, since it will be sensitive to an increase in bytes sent even if it's coming from just one machine. I would calculate the total network data transmitted from all **d** machines by minute and use that metric to calculate mean and median number of bytes transmitted. Comparing the means with the median value and flagging discrepancies above a certain threshold could help identify the moment in time when the distribution became skewed and mean and the median stopped being approximately equal, as we would expect from a normal distribution.

   Additionally, to alleviate potential short-term fluctuations that may be occurring in the data naturally, I would experiment with rolling averages of total bytes sent across all machines in a given minute (I'd start with a 5-, 7-, or 10-min rolling average to identify the best suited option).

   Once the approximate period of time when the network started transmitting extra data is identified (with additional considerations mentioned below in mind) we could turn to a similar report but with rolling average that is being calculated for individual machines (not network total) to identify the infected machine.

   Extra considerations: plotting a time series chart could also help us visualize changes in the amount of data transmitted by the network and spot trends such as different levels of network usage by hour of the day (likely to be less busy at night) and day of the week (weekdays vs weekends).

   Steps in a nutshell:
   1. Total network data transmitted from all **d** machines by minute.
   2. Keep track of mean and the median to see if they are approximately the same.
   3. Calculate 5-, 7-, or 10-min rolling average of network's bytes sent.
   4. Calculate rolling average of each machine's bytes sent.
   5. Visualize the data as a time series to spot changes over time or as histogram/box plot of the distribution in a given moment to spot potential outliers.

2. **Next imagine that the malware may have infected multiple machines. Can your detection method from the first problem be used to identify all of the infected machines without mis-identifying many uninfected ones? Discuss and explain your reasoning. If not, try to describe a modified detection strategy that will work to identify all of the infected machines without too many false positives.**
   Assuming in its normal state the data produced by the network is approximately normal we could keep track of how the mean and the standard deviation of the bytes sent changes over time. Then we could flag machines that are more than two standard deviations away from the mean in a given timeframe to identify outliers.

   Additionally, learning what the baseline metrics historically looked like for a given hour of a given day of the week may be helpful in better understanding whether a certain value that we are seeing is out of ordinary or whether it's within range that we've seen before

3. **Now imagine you are the hacker designing the malware, and it will get to infect about 1/3 of the total machines. How might you design your strategy for transmitting data out of the network without it being detected? Can you defeat the detection method designed above or not? Explain your reasoning.**
   In order to "fool" the detection methods outlined above I would take the following steps:
   1. Limit the incremental amount of data (let's call it **M)** that each infected machine is allowed to transmit to avoid early detection.
   2. Gradually increase **M** at set intervals to increase the mean number of bytes transmitted by the network in a way that would look more natural, as opposed to having a suspicious spike.
   3. Limit the number of machines per set interval of time (day) that each infected machine is allowed to infect, again, to avoid a sudden increase in the number of bytes that infected machines transmit out outside.

   As the number of infected machines increases while being undetected, an adversary gains more power in significantly changing the mean, however, since they are limited to controlling only up to ⅓ of the machines, the second set of flags that we based on comparing means with medians should theoretically help detect this unusual activity.