RAHNEMA
COLLEGE

Introduction

Detecting anomalies within datasets are of great interest in multiple domains. Banks are interested in detecting fraud in applications and transactions; cybersecurity analysts are interested in detecting unexpected incidents in logs and in traffic patterns; systems administrators are interested in unexpected behavior in deployed systems and applications, and so on. Anomaly detection plays a significant role in the field of Web security, and log messages recording detailed system runtime information has become an important data analysis object accordingly. Traditional log anomaly detection relies on programmers to manually inspect by keyword search and regular expression match. Although the programmers can use intrusion detection systems to reduce their workload, yet the log system data is huge, attack types are various, and hacking skills are improving, which make the traditional detection not efficient enough. To improve the traditional detection technology, many anomaly detection mechanisms have been proposed in recent years, especially the machine learning method. Classical approaches to anomaly or outlier detection use many statistical techniques. These include density-based or clustering algorithms such as k-Means and its variations, random forests, Bayesian networks, support vector machines, hidden Markov models and such. With deep learning techniques coming into play, neural-network approaches are becoming more and more popular, among those neural networks capable of generating their inputs like GAN and autoencoders proven to be almost the most efficient ones when facing these sorts of problems. The past few years have seen many impactful methods on applying deep neural networks to this problem. Out of these, the domain of cybersecurity alone has seen hundreds of papers of interest.

Anomaly detection is usually a big data analytics problem, and is often a real-time streaming analytics problem as well. The latter arises from the need to analyze real-time events, and respond within well-defined time periods, such as when handling cyber security threats, or when analyzing online transaction requests for potential fraud. This obviously calls for a big data analytics pipeline
that has streaming capability. In this project you are asked to detect intrusion in the data. Feel free to read more about this anomaly and about the way it can be detected.

Approach

In this project, you are asked to identify intrusion in a system, which relies on the analysis of logs collected in real-time from the log aggregation systems of an enterprise. (if you don't know what a log is, pause here and search) Since you are not given any tags indicating the anomalous behavior, you must present an unsupervised anomaly-based "intrusion" detection solution focused on user footprint analysis; this part will be explained in more detail in the subsequent sections.
Let's dive into some details:
Suppose we are to equip "our customers'', who typically are different (web-based) service providers, with a tool or an algorithm to analyze their systems as well as their users. It turns out such a system would be highly personalized; meaning its design and applicability, must follow (be on the basis of - will be bound by) the specific structure and technologies a specific web service provider is using (it is designed for), and cannot be transferred to another customer, or applied to any other case. For instance, if we ought to design such an analytical tool, based on database performance analysis, it would be restricted to that specific database and not any other one. It might not even be used by another customer using the exact same database as the original customer; It is the direct result of its dependency on the type of usage and the structure of data stored in that database. In spite of this, there is one exception: web-servers!!

Web servers

Web Servers are usually a good starting point both for analysis of user behavior and the behavior of the system itself. These are a few reasons why: - There are only a restricted number of protocols used by various web-servers (usually just one!); Any algorithm capable of analyzing the HTTP based traffic, can be applied to almost all other situations. - System logs in this case have a really simple structure. HTTP protocol data is text based which enables one to extract lots of information, solely by processing web-server logs. Note that a web-server does not have to do anything complicated. It just has to log everything that goes in or out . This feature makes it extremely convenient to apply the existing algorithms in NLP to extract numerical features from these logs. - The main interaction of users with the system passes through this point, which provides the analyst with the digital footprint. These footprints happen to say a lot about a system and its users.

Data structure - 2..0 HTTP:

HTTP is the acronym for Hypertext Transfer Protocol. The structure provides the path between the requests received from the web browser or client and the responses received from the web servers. The data such as HTML documentations, images, videos, query results are delivered by HTTP using TCP/IP (Transmission Control Protocol/Internet Protocol) on the World Wide Web. HTTP is a protocol that provides the communication of the web browsers and servers in a common language. Requests and responses are performed by following language rules specified by HTTP.

Data used for these kinds of tasks (anomaly detection in HTTP log data) is mainly the metadata of HTTP traffic, extracted from HTTP headers in the format of a set of key-value pairs. Some other information extracted from lower layer protocols (TCP/IP) could also be of great importance like IP and related information that can be extracted from an IP(e.g. IpGeoLocation). To get a sense of what we are dealing with, suppose you type "example.org" in your browser. The following image shows what kind of information will be communicated between you and the server. The requests and the responses of "example.org" are in red and blue colors respectively. As we have mentioned before, the image only shows the metadata communicated in this process, not the actual content like documentations, images, videos, etc.

Evaluation

For this part we evaluate the project based on the evaluation metrics of unsupervised algorithms and also you can do some post processing to evaluate how good your model detects anomalies.