Journal of Database Management
Volume 31 • Issue 1 • January-March 2020

A Service Architecture Using Machine Learning to Contextualize Anomaly Detection

Brandon Laughlin, University of Ontario Institute of Technology, Oshawa, Canada
Karthik Sankaranarayanan, University of Ontario Institute of Technology, Oshawa, Canada
Khalil El-Khatib, Ontario Tech University, Oshawa, Canada

ABSTRACT

This article introduces a service that helps provide context and an explanation for the outlier score given to any network flow record selected by the analyst. The authors propose a service architecture for the delivery of contextual information related to network flow records. The service constructs a set of contexts for the record using features including the host addresses, the application in use and the time of the event. For each context the service will find the nearest neighbors of the record, analyze the feature distributions and run the set through an ensemble of unsupervised outlier detection algorithms. By viewing the records in shifting perspectives one can get a better understanding as to which ways the record can be considered an anomaly. To take advantage of the power of visualizations the authors demonstrate an example implementation of the proposed service architecture using a linked visualization dashboard that can be used to compare the outputs.

KEYWORDS

Context, Explanation, Intrusion Detection System, Network Flows, Outlier

INTRODUCTION

Monitoring network flows (NetFlows) is an essential part of securing networks. This has become increasingly difficult as the amount of traffic being generated has outgrown the ability to effectively analyze them (Cisco Systems, 2018). In addition to the increasing scale, network data is coming in at faster rates and there is a larger variety of data sources to deal with (Habeeb et al., 2019). With such a large influx of information, analysts are not able to identify threats in a timely manner leading to exploits persisting on networks and only discovered once damage has already been done (Secureworks, 2018). This places an increasing importance on more automated methods such as network intrusion detection systems (NIDS). Existing work on NIDS can be placed into two main categories: signature based, and anomaly based. Signature detection is based on existing attack knowledge using specific criterion for threat detection (Fernandes, Rodrigues, Carvalho, Al-Muhtadi, & Proença, 2019). Anomaly detection establishes baselines and looks for activity that appear as outliers. With the scale of modern big data, security analysts are facing difficulties in reviewing all of the network flows determined as threats by the NIDS (Cisco Systems, 2018).

DOI: 10.4018/JDM.2020010104

Compared to finding anomalies in other applications, the analysis of network security data adds additional challenges. There is a wide range of contexts that will influence whether something is anomalous and if it is anomalous, whether or not it is due to malicious activity. For example, the specific user, their role in the organization, the device in use, the application being used or even the time of day are important considerations for analyzing an outlier. With the increased accessibility and reduced cost of computing power in recent years, machine learning (ML) has increasingly become a tool used to address these challenges (Buczak & Guven, 2016).

Over time, these ML techniques have become very complex to the point where only experts of the system can understand how the system works. It is important to have a clear explanation as to how a certain anomaly rating was generated with more context than just an outlier score as output. Without a proper understanding of the underlying properties used to produce the output it is difficult for an analyst to translate the resulting outlier scores into information that can be acted upon. This is challenging as advanced ML algorithms such as deep learning act as a black box that provide little to no justification as to the results of the classifier (Wang & Siau, 2019).

Most research in machine learning for cybersecurity has been done using supervised learning in which labels are included in the data that identify attacks within the dataset (Buczak & Guven, 2016). While supervised approaches reduce the number of false positives, the dependence on labels is a large limitation (Sommer & Paxson, 2010). As the threats facing networks change very fast, even new datasets become irrelevant quickly as adversaries adjust strategies to avoid detection. Developing labeled datasets can also be very expensive and time consuming and is not very scalable. Training machine learning models without labels in an unsupervised setting can remove these limitations; however, they bring their own set of challenges. One of the largest challenges is the validation of the system (Sommer & Paxson, 2010). By building ML models without labels there is no direct method to assess the accuracy. Not only does this make comparing ML models and choosing the best one very difficult, without an effective validation method, building an easy to understand model is even more difficult.

Network security specialists have extensive background knowledge in protecting networks and a general sense of normal network conditions. This makes them suitable candidates for managing an Intrusion Detection System (IDS) although they may not have the experience needed to setup the ML aspects. There are many steps involved in the data science pipeline including data preprocessing, feature engineering, model selection and tuning model hyper parameters (Zimek et al., 2014). While experienced data scientists may be comfortable configuring these options, many security experts do not have the prerequisite skills (Sacha et al., 2017). One option to integrate the ML components is to employ data scientists to help develop the IDS. However, if the security experts using the system had limited involvement in its development then they may have trouble interpreting the ML results. One needs to abstract the details of the ML operation without losing the ability to extract actionable intelligence from the results. Pure automation would remove the analyst and produce non-optimal results, while depending on the analyst to develop the system would be very difficult and time consuming. It is crucial to balance between abstracting the ML operations behind the scenes and being able to interpret the results.

Whenever a record has been identified as anomalous by a NIDS there is the challenge of determining whether this activity is malicious and how to translate this into something actionable that a network operator can respond to. To help reduce the complexity and time needed to handle a response, it is helpful to have the system provide extra context with an alert. This context can come from baseline data provided alongside the alert to enable exploratory data analysis. This additional context for each alert can allow for better threat intelligence. Instead of relying completely on the analyst to find anomalies or using a ML classifier without context, the authors propose a service architecture that will combine both approaches to take advantage of the scale of ML with the expertise of a human analyst.

The proposed service helps provide context and an explanation for the outlier score given to any network flow record selected by the analyst. The service constructs a set of contexts for the record using