2018

PUNCH

# LESSONS LEARNED FROM DATA SCIENCE APPLICATION TO CYBER SECURITY NETWORK LOGS

JOHN LANKAU, KYLE SMITH, LAUREN DEASON, MICHAEL GEIDE, JACOB BAXTER
PUNCH CYBER ANALYTICS GROUP

This research was developed with funding from the Defense Advanced Research Agency (DARPA).
Disclaimer: The views, opinions and/or findings expressed are those of the authors and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

## Table of Contents

# 1 Executive Overview

As operational cyber security analysts, we have invested time and resources into data science and mathematical approaches for identifying cyber-relevant results in network log data. We will describe the various successes and pitfalls from our experience. The majority of modern operational detection capabilities rely on detection signatures. To help move beyond signatures, data science techniques are actively being explored to detect threats that signatures miss. We will discuss the differences and trade-offs between traditional signatures and data science analytics as applied to network logs. Figure 1 outlines the relative merits of signature and anomaly based detections. This paper’s authors are individuals with backgrounds in cyber security operations and digital forensics incident response (DFIR), and we will touch on specific data science approaches and their utility in operational environments. The intended audience of this paper is cyber security analysts looking to incorporate data science techniques into their playbooks, data scientists looking to apply detection techniques to cyber security use cases, and anyone interested in the intersection of these disciplines. Specifically, there are several categories of cyber-relevant logs where data science techniques can be applied—network, host-logs, static files, emails, and social-networking information to name a few. Our focus is on the applications and lessons learned specifically applying data science concepts to network logs, but the concepts would have applications across a variety of data types.

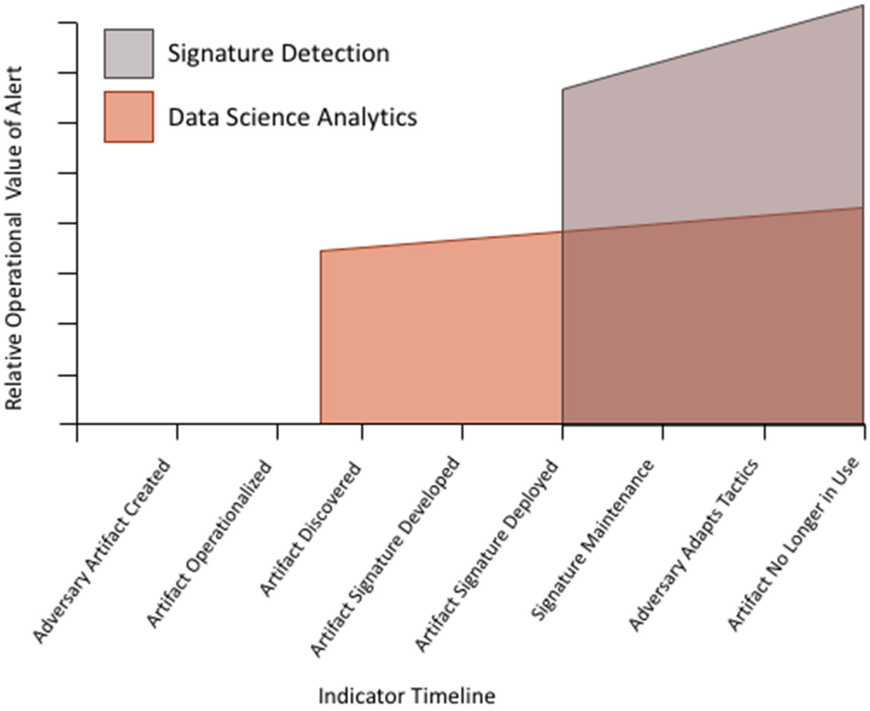


*Figure 1 - The relative value of artifacts detection (domains, IP addresses, hashes, etc.) over time for data science analytics compared to traditional signature-based detection techniques. Adapted from Figure 19 of Ten Strategies of a World-Class Cybersecurity Operations Center [1].*

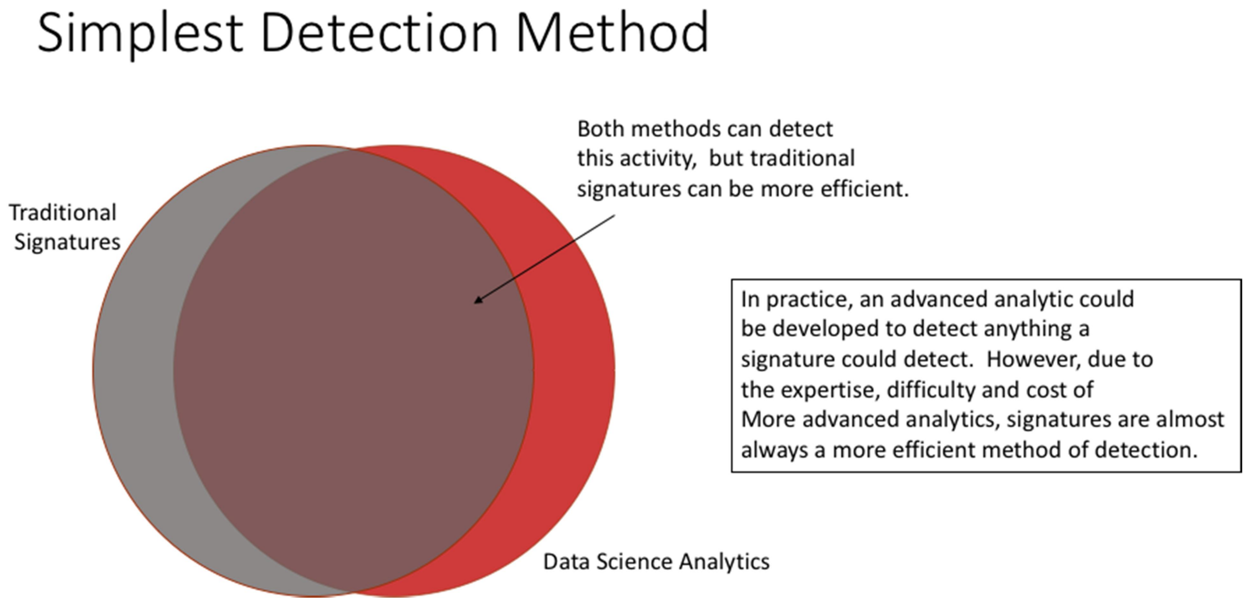
# 2 Measures for Success

Data science techniques can be applied to find new and interesting insights about your data, however these solutions require significant knowledge of the network to which they are applied in order to yield relevant results. Marketers would want you to believe that by quickly and easily applying advanced analytics to your dataset, the needle in the haystack will simply reveal itself. However, this promise drives an unrealistic expectation of data science approaches and can distract from ways to measure success. This leads to a results-driven *"did it detect malicious activity*" methodology that leads us to ignore less glamorous but very useful successes in other ways, such as did we derive new data features or insights that can be leveraged by more traditional methods.

From a research perspective, *"did it detect malicious activity"* is clearly defined as a "win". But this approach can be viewed like scoring a baseball game and only considering homeruns. One of the aspects that makes these analytics so powerful is that they can be applied to new datasets and provide novel results. This works because they are able to highlight statistical anomalies—not necessarily the act of malicious activity, but the aberrations that make the activity different from the baseline. Unfortunately, in practice, the reality of network traffic is that it is extremely volatile and unpredictable, so much so that the vast majority of anomalies tend to be benign and not tied to malicious activity. Instead, these anomalies often trace back to esoteric benign administrator behavior, non-standard architecture, or symptoms of a break-fix done years ago just to keep the network operational while not cleaning up the underlying issue itself. Finding these benign anomalies is not a bug—it is a feature of this approach. However, for the analytics to be successful in a new environment, these types of anomalies need to be investigated, whitelisted, and incorporated into the analytics to prevent them from appearing in future results.

Even when data science techniques find a malicious result, this is not enough information to surmise whether or not the analytic was truly successful or not. One of the lessons discussed in more detail in 4.1 "The Analytic Spectrum" is whether or not the results were discovered with the simplest method available. Data science analytics are often computationally expensive and can be more difficult to investigate compared to traditional signatures and indicator-of-compromise (IOC) matches. They can also be "black boxes" that are difficult to tune and require expertise to curate and interpret. A true measure of success is whether the analytic could detect malicious activity and *would be able to detect malicious activity that no simpler method would also be able to detect.* Operationally this means the analytic is able to identify malicious behavior that was not detected by signatures and IOCs. In other words, if the data science was able to identify malicious traffic that we also detected from known intelligence and Snort signatures, it is not as valuable of a detection as a result that was uniquely found by only the analytic. Figure 2 visualizes the overlap between traditional method and new analytics.

In practice, we found that data science analytics identify suspicious behavior that leads to the creation of signatures for static indicators like strings and patterns for that specific instance of the malicious activity. The shortcoming of such signatures, however, is that they will only be useful until the attack evolves into a new implementation. Analytics based on detecting anomalies and outliers relative to normal behavior are often more resilient to change and may be able to detect both previously seen attacks (that can be detected using signatures) as well as never before seen attacks for which signatures have not been developed. In our research, data science analytics have been applied to networks where traditional signature-based defense is in place, and generated detections that traditional cyber operators have missed.



Both methods can detect this activity, but traditional signatures can be more efficient.

In practice, an advanced analytic could be developed to detect anything a signature could detect. However, due to the expertise, difficulty and cost of More advanced analytics, signatures are almost always a more efficient method of detection.

*Figure 2* - Can an analytic detect malicious activity and would it be able to detect malicious activity that no simpler method would also be able to detect.

Defense-in-depth is always a goal of a successful security posture but it is important to understand the difference between corroborating evidence of existing results and a truly new detection. Only by understanding this distinction can you truly perform a value analysis on new analytic detection methods. Many times, the advanced analytics will deliver anomalous results that are not found to be malicious after investigation. This does not mean that these analytics were not successful. In many cases, the investigation was able to tell us new information about the data or lead to additional building blocks that can be incorporated back into the analytics for better detection next time. Additionally, in our experience we are often running the analytics on real and unlabeled network data—therefore we do not know whether there are any true positives present in the test data set. Due to a lack of training data [2] (discussed in more detail in Section 3.5), running the analytic on several datasets can help us determine the strengths and weaknesses of the analytics.

In one example, we experimented with techniques to identify watering hole attacks in HTTP traffic. A watering hole attack is when a legitimate website is compromised and serves malicious content to the visitors of the site—the compromised site is typically chosen because of a high likelihood that the true intended victim user will visit the site. For example, websites for a local news site for the regional area of the intended victim are prime targets for watering hole compromises. We performed analysis on the HTTP referrer strings to create connections between domains based on understanding which domains refer to other domains within the dataset—sudden changes to patterns and anomalous patterns would be investigated as potential watering hole attacks. In practice, this approach turned out to be too noisy to be reliable for watering hole detection, but by looking at the results differently - based on the domains with the *most* HTTP unique referring domains we derived a technique that can identify likely advertising domains. Advertising domains often appear and behave in ways very similar to malicious domains—they are both cryptic and try to avoid being blocked. Because of this, differentiating between malicious and advertising can prove difficult when looking at individual results. However, by creating a list of likely advertisers observed over an entire dataset, we can apply this label to the results of other analytics to aid analysis in determining whether a result is likely a benign advertisement or a malicious domain.

**Takeaway**: Do not only consider whether a technique identified malicious activity when evaluating data science approaches.  Be sure also take into account if there are simpler methods that yield similar results, or if new insight was gained about the dataset.

# 3 The Data

Referring back to the source data will help us to answer a great deal of questions that arise from analytic results by providing contextual clues. For example, if alerted to a man wielding a knife, the context is key in determining the threat. If the man is located in a dark alley and wearing a mask, the risk is high; but if he is dressed as a chef preparing a meal in a kitchen, the risk is low. From a cyber perspective, data analysis will identify anomalous patterns, groups and outliers but in order to assess threat severity, we need the underlying input and relevant contextual data to facilitate both analysis and investigation. This depends on the Extract, Transform, Load (ETL) process to ensure the available data has maximum utility—in terms of quality, features, detail, and enrichments.[1] Aside from the architecture, the perspective from which we view the data is also extremely important to the analysis. Sensor placement directly dictates traffic visibility and can be the cause of blind spots, aggregation points, and data duplication that can have severe downstream impacts on data analytics.

Figure 3 shows the basics of investigating analytic results. It begins with querying the input data to see what we can learn about the event—whether it is the number of unique hosts that have the same pattern, the traffic immediately before and after the event, or if we have observed

[1] For example, if we are performing analysis on HTTP traffic flow, the network architecture is critical for determining what is “normal” traffic and what is “abnormal” for the given network. Some networks are configured to have workstations connect directly out to the internet, while others send all traffic through a web gateway. These two setups will have fundamentally different traffic patterns and understanding the expected traffic flow gives valuable context that should be incorporated into the analytics to tune them and minimize low-quality results. Another example is understanding the context of the DNS infrastructure within a given environment. Bro DNS logs, for example, give a great view of the protocol as it is observed on the wire within the environment. However, the analysis needs to be aware of whether it is reviewing DNS traffic from hosts to the DNS server, or from the organization DNS server(s) out to the internet.

similar activity before. Information like this provides us with some context about the anomaly, and since it originates from our dataset it is possible to incorporate it back into the analytic. However, not all context will be contained within the input dataset we are working with. For example, network-data can give us publicly accessible domains and IP addresses—where enrichments can help allow us to incorporate these aspects into the dataset and then be leveraged by the analytics. However, if the analytics are focused on more “black box” identifiers such as suspicious user/account behavior, getting to ground truth typically requires reviewing additional information such as HR interviews, employment status, or work schedules. An external investigative team will rarely have the ability to access these resources, and even if they can, they will be difficult to incorporate into an automated analytic so they tend to remain in the investigative domain rather than the data input domain.

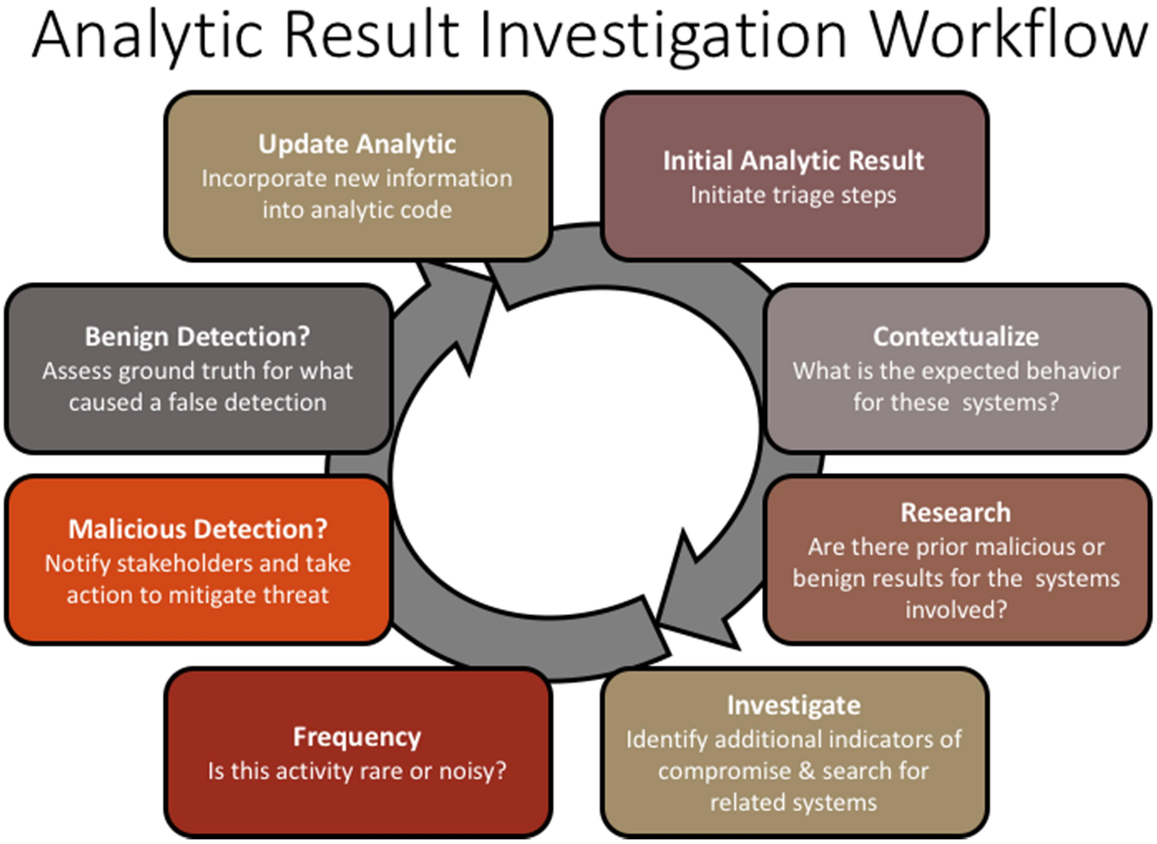

Analytic Result Investigation Workflow
Initial Analytic Result
Initiate triage steps
Contextualize
What is the expected behavior for these systems?
Research
Are there prior malicious or benign results for the systems involved?
Investigate
Identify additional indicators of compromise & search for related systems
Frequency
Is this activity rare or noisy?
Malicious Detection?
Notify stakeholders and take action to mitigate threat
Benign Detection?
Assess ground truth for what caused a false detection
Update Analytic
Incorporate new information into analytic code

*Figure 3 – An Outline of the traditional evaluation and investigation framework an analyst uses to assess new analytics as well as the manner in which it is tuned to eliminate false positives.*

Few network datasets currently available were designed with data science techniques in mind—the data will not always be structured for efficient queries, and there may be significant ETL efforts required to ensure the data structure facilitates advanced analytics. Some log types are better suited for this than others—Bro IDS records complete summaries of events that are observed, which readily lends itself to data science techniques. Logging systems that perform sweeps to create “snapshots” of the environment are effectively sampling the dataset at points in time—which can lead to problems when your goal is to find events that rarely occur and only exist for short periods of time. However, other data types, such as Windows Event Logs, can be cumbersome, cryptic and capture data features differently across different Event IDs. A data source like this can require a significant amount of configuration, research and ETL effort to get the logs into a format that facilitates data science applications.

## 3.1 Data Quality

The data itself will direct what can be learned from the analytics and logging and curating specific data points can be crucial. In some cases, data that was not recorded can be estimated or sampled—but this uncertainty will be magnified downstream, often to the point where the results are low-confidence at best.  The importance of data structure and quality is often overlooked and only realized much later in the process when it becomes significantly more difficult to address. In the *Huntpedia* Chapter 13 "*Leverage Machine Learning for Cyber Threat Hunting"*, Tim Crothers states "On a typical ML project, I spend about 90% to 95% of the overall time curating the data. The age-old analogy of "garbage in, garbage out" certainly applies in spades with ML. The better curated your data, the better your results will be. [3]"   This underlines the oft hidden importance of a quality ETL process. ETL is often far upstream of analysis and quality issues can easily be overlooked or ignored while performing analysis leading to inaccurate or inconsistent results.

Furthest upstream from ETL is data collection—including sensor placement and decisions about the type of data that is being recorded. Due to cost, logistics, existing architecture, or policies, the data collection strategy cannot always be optimized, but the ETL process can help to minimize many of the pitfalls that can arise from collection shortcomings. From the sensor, the data can have several issues—inconsistency, gaps, duplication, and parsing issues are the most common in our experience. Figure 4 identifies the layers of abstraction that occur between data collection and analysis.

**Data inconsistency** can involve information being recorded differently from several sensors—the most common example is something such as the time zone. Many data science analytics depend on timestamps as a critical component of analysis, so the ETL process is when disparate sensors can be normalized to all record on consistent time.[2]

[2] A more local example is the differentiation between "internal" and "external" IP addresses. Many organizations use RFC1918 space for internal devices (such as 192.168.x.x or 10.x.x.x IP addresses), but some organizations use externally routable IP addresses for internal devices or have a mix of both. This will be unique to the organization, and many of the analytics will need to understand what devices are considered "internal" and which are considered "external" to perform properly. This is a great example of a feature that can be approximated but getting a clear and certain understanding from the data provider will provide significant advantages and confidence to analytic results. Another common inconsistency we have observed is the level of aggregation employed by a sensor versus what is required by an analytic. An example of this is the ability to separate domain names into their component - top level domain (TLD) such as .com or .gov, the domain itself, and any associated subdomains. The domain "subdomain.google.com" may be recorded as a single entry, but the ETL process may optimally create additional data points of "subdomain" "google.com" and "com" to facilitate analysis.

Addressing data normalization immediately after the initial data ingestion will prevent several problems during analytic development. A best practice is to perform data normalization as an integral part of the ETL process. This eliminates the issues associated with data scientists each performing their own data transformations—namely future result replication, inability to combine multiple analytics that have different data inputs, and increase in runtime at each analytic execution. This has the added benefit of allowing your most expensive resource—data scientist and analyst time—to focus on solving new problems, not repeatedly rehashing old known data inconsistency issues.

**Data gaps** will occur, and they can be the result of sensor issues or processing issues. Sensor issues, such as packet loss or data corruption can't be fixed with ETL, but it is at this stage where it is most advantageous to detect and validate the data so the gaps can be identified and accounted for during analysis. Many of these gaps are the result of improperly sized sensors that either can't handle the flow of data, are misconfigured to record the wrong data, or don't have enough storage to maintain a large enough buffer before transferring the data to a repository.

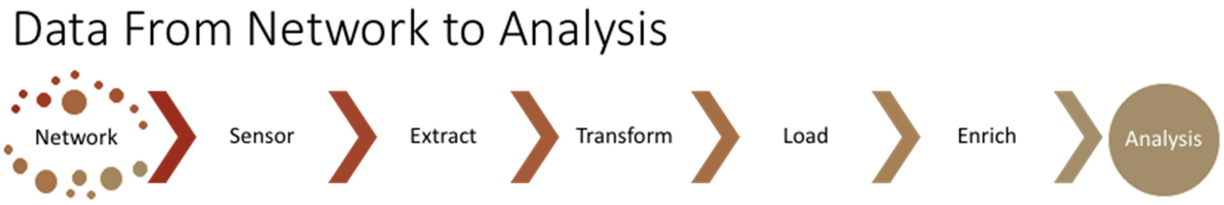


*Figure 4 - The layers of abstraction introduced through observation and analysis of a network*

Data gaps can also occur as a result of the ETL process. Parsers or processes that take the raw data and transform it into a format that allows for massively scalable analysis need to be validated to ensure data integrity. Performing basic "Sanity checks"[3] during initial ETL ingest can quickly identify issues. Unlike sensor-based gaps, data lost during the ETL process can be identified and corrected—as long as data validation processes are followed. Identifying these gaps early can prevent downstream problems with analytic results based on inaccurate assumptions and minimize the times data needs to be re-ingested because of ETL processing flaws.

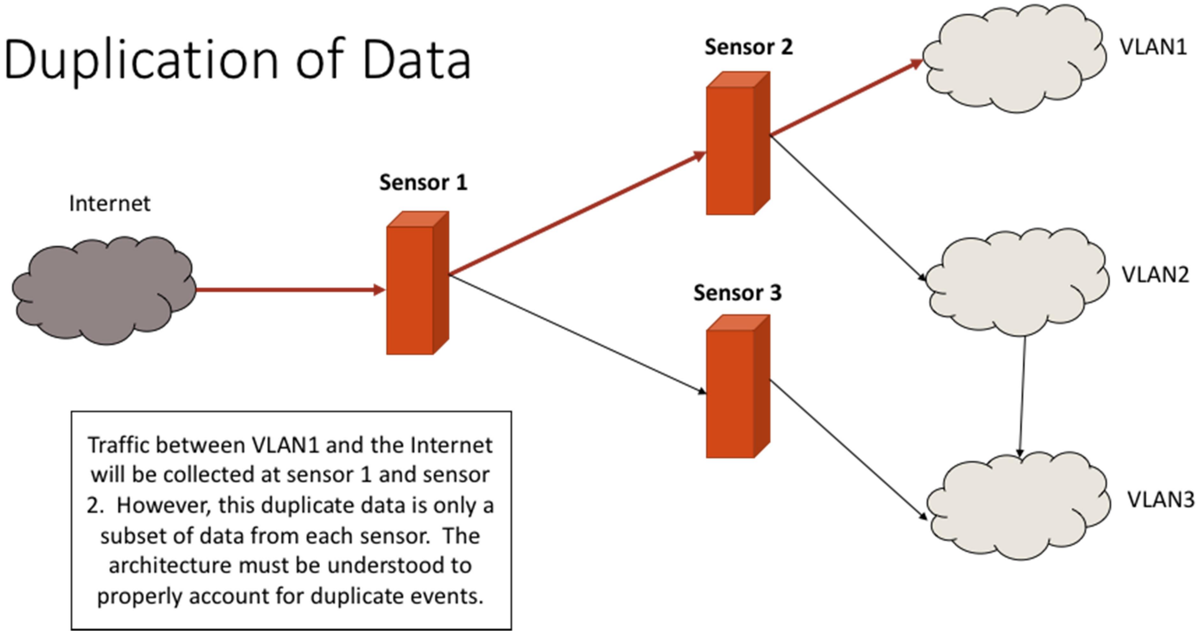
**Data duplication** can lead to its own analytic problems. This problem is often identified by investigating unexpected analytic results—usually requiring significant troubleshooting efforts and wasted analysis and processing time. To minimize this, ensure a data validation process reviews the data and performs checks to identify duplicate records. Duplication can be caused by something simple, such as records being provided multiple times in different formats or as the result of unsynchronized processes or multiple, overlapping sensors.[4] Figure 5 outlines a

[3] For example, column and row counts, manual review of rare data values, special character handling and column name collisions.

[4] One data set provided logs that were archived every 12 hours *and* when the archive size reached a certain threshold. These competing processes resulted in overlapping but not identical records. Neither

possible sensor layout that would cause known data duplication.

Data duplication can become more complicated when there are several sensors in the environment with overlapping visibility. For example, if multiple Bro IDS sensors are deployed within the environment to achieve 100% coverage, there will likely be traffic that passes by multiple sensors. Due to network architecture, the same traffic may appear differently across the sensors—as the true source IP in one log, and a NAT proxy IP address in another log. As the traffic is observed at different points, the timestamps will likely be similar but not exact matches. This type of data duplication is significantly more difficult to identify and remove from the dataset due to its inexact matching and may not be able to be completely eliminated without significant insight into the network architecture, and sensor setup. Duplication records in the database can affect analytics that utilize counts and ratios, artificially making certain data points appear more or less prevalent than they really are. In many cases, the simplest workaround to this is to perform analysis on a per-sensor basis or calculate results in record-sensor dyads.



Traffic between VLAN1 and the Internet will be collected at sensor 1 and sensor 2. However, this duplicate data is only a subset of data from each sensor. The architecture must be understood to properly account for duplicate events.

*Figure 5 - Understanding how your sensors correspond to your network diagram is essential in preventing data duplication that can skew statistical analysis.*

Inherent but often overlooked in all of this is the fact that each data set will be unique and require some level of ETL customization. In our environment, we have worked with several data partners to perform analysis on the datasets we were provided. Many of these involved the same data type—the most common being Bro IDS records. Different versions of Bro IDS provide a different set of Bro IDS logs. For instance, the SMB analyzer and associated logs were introduced in Bro 2.5 as well as custom Bro scripts to include—or exclude—specific fields within each Bro IDS log. Due diligence and care must be performed when ingesting a new dataset—even if the data type has been processed before.

set would provide full coverage, but by combining and de-duplicating the sets full coverage of the dataset was achieved.

**Takeaway:** Data quality is paramount for data science and machine learning applications. Invest heavily in ETL and data validation resources and ensure processes are meticulous, repeatable, and complete. The bottom line is if there is no trust in the data, then there will be no trust in the analytic results either.

## 3.2 Enrichment of Data

Once the data has been recorded, we need to consider how to facilitate our data science use-cases and enrich the data. Whitelists and blacklists have a bad reputation in data science discussions—after all, we are trying to move beyond these manual techniques. In practice, however, the art of anomaly detection uncovers the outliers within your dataset—but the number of anomalies that are actually malicious is still very small [3]. For any given environment, the outliers will need to be investigated and conclusions made. We need to divide the data into “things we know about” and “things that are new to us” so the same results are not re-investigated over and over again. By that very nature—the category of “things we know about” are just the “whitelist” and “blacklist” of our results—so we do not need to reinvestigate them when the analytic is run again. If we think of “whitelists” and “blacklists” as data labels for benign and malicious—it turns out our advanced analytics were using this concept the whole time.

It is rare that an analytic will be so elegant that its output is entirely high confidence, malicious results determined solely from behavior and feature analysis without analyst confirmation. There will almost always be a server that “acts” malicious but is just the network’s unique method of performing an essential business function. Not all lists need to address the question of “benign” or “malicious” as whitelists and blacklists do. As mentioned earlier, one analytic that was developed became a method to identify likely advertiser domains based on their prevalence across the dataset—or even among several datasets. These “advertiser” labels can be factored into the analytics themselves, but they may be more useful to use while investigating the results to see which results are suspected of being advertising domains allowing analysts to avoid spending time researching a benign rabbit hole.

Advanced analytics like this can be useful to generate labels and lists to enrich data. These labels can then be used as features for other analytics to build upon what we have learned. For example, we discovered that many analytics would easily be influenced by internet network scanners such as Nessus. This traffic appears “malicious” in its behavior, but when we add the context of the authorized scanning it was no longer an interesting result. An analytic was developed to detect scanners, both internally and externally with the explicit purpose of labeling the devices as scanners. With these labels applied, known scanners could be excluded from future analytics and resulting in a much cleaner dataset—and more interesting results. Generating analytics with the goals of labeling records is frequently overlooked in the pursuit to find “the bad guy”—but these building blocks can prove instrumental in tuning analytics and in performing incident investigation.

**Takeaway:** Using any available internal or external data enrichments only helps to further build out what you can conclude about your network. Internal network source enhancements including information from places like internal network maps (such as subnet breakdowns and their purpose) and Active Directory user details are a quick and free way to enrich your data as well as being unique to your network. Incorporating external information such as WHOIS and geolocation data to enhance data with information such as the ASN, associated company, country, and city can facilitate analysis by looking for interesting patterns at different aggregation points if something like IP addresses prove to be too granular to derive meaningful results.

## 3.3 Derived Data Features

Other features can be derived from the data and can allow an analyst to decipher the outputs of “black box” analytics. For example, some analytics use scoring methodologies to calculate specific attributes of data entities. Often, several scoring methodologies are combined to create an analytic ensemble—which can incorporate several aspects of the data into a single score. However, these ensemble metrics can sometimes be so complex that they are difficult to decipher in aggregate—and providing a means for the cyber analyst to trace each component of the overall score individually allows the analyst to get a more holistic understanding of the results, especially during the research and development phases of an analytic.

As an example, many of the analytics are based on the timing of events—and as a result, time-based enrichments can greatly enhance analysis. Specifically, one feature we frequently derive is the “first-known” time a specific attribute (IP address, domain, etc.) was observed within the dataset—typically this can be quickly accomplished by taking the “minimum (starttime)” of the associated record. Having a data point like this available, an analyst can use it to perform a more in-depth analysis on domains that are new in the environment as they appear.

**Takeaway:** Don’t limit analysis to the out-of-the-box data that is provided. In many cases, enhancing the data with labels, calculations, and incorporating contextual information into your dataset opens up new dimensions of analysis. Remember to think beyond the data at hand and consider what data enhancements might facilitate your use-case and formulate a path to derive, calculate or import it into the dataset. As you develop enrichments, ensure that as they are incorporated into potential scoring ensembles, that the initial inputs are not abstracted away from the analyst. This has the added benefit of providing additional context to your analysts which can save time while triaging analytic outputs.

## 3.4 How Detailed Is Your Data?

There is often an expectation gap between the questions being asked and the amount of certainty that can be derived from the data available. Not all data provides the same level of detail—and analysis can only go as far as the data allows. Consider the levels of data fidelity associated with PCAP, Bro IDS, and a more generic netflow.

Data can be extremely feature rich, such as PCAP data—a full capture of all activity observed on the wire. The answers are likely in there—but they can be difficult to uncover due to the sheer size of the dataset and lack of data structure. Feature rich data has excellent forensic value if we know where to look for suspicious activity. Netflow records are quite small and can cover a significant range of time to facilitate statistical analysis and can allow us to find anomalous behavior on a very long timescale. However, if we only have netflow records available, it can be difficult to get to the ground truth of the cause of the anomalous behavior. Logs like netflow and DNS are feature poor—they are best reviewed in aggregate and can find promising results in in larger sets—they can tell us where to investigate, but they are not conducive to providing ground truth. Cyber analysts can research and provide a best guess, but the level of detail and certainty in these results will usually be quite low. The graphic in Figure 6 provides some examples.

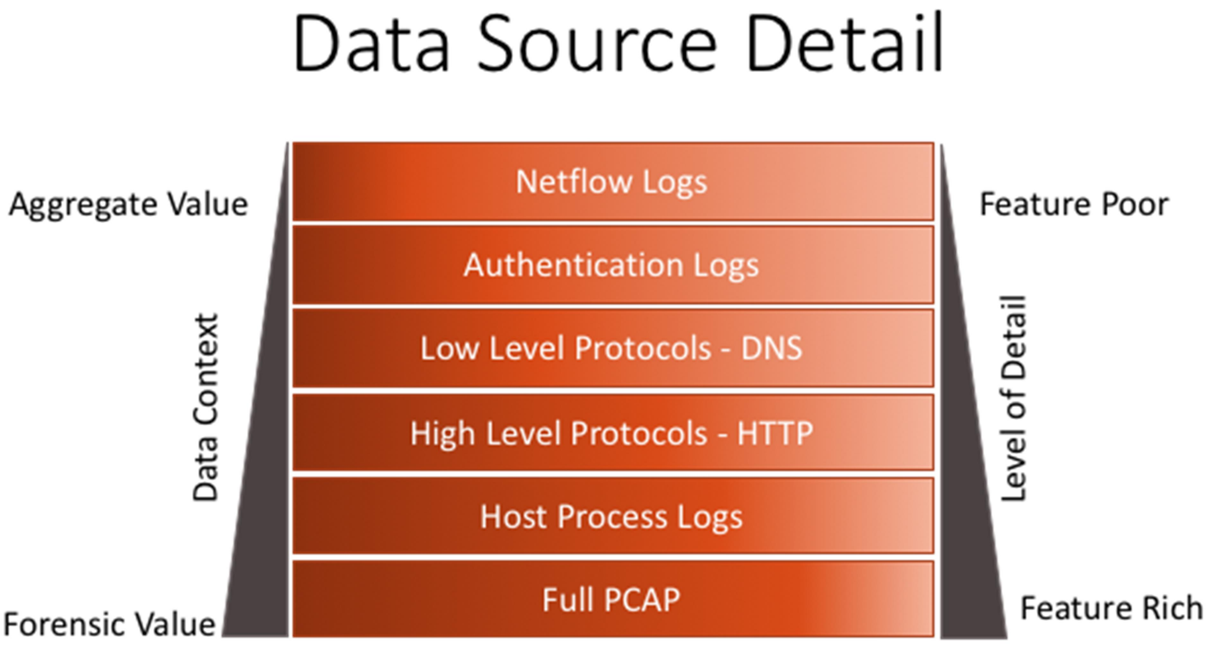


*Figure 6 - As detail increases, the forensic value of the data increases according. Conversely, data with limited detail can be useful in drawing conclusions about a network in aggregate. Adapted from Figure 5 of Ten Strategies of a World-Class Cybersecurity Operations Center [1].*

Bro IDS can help bridge the gap between netflow and PCAP because it provides structure and features across several layers of the OSI model [4]– but it has significantly less space requirements than PCAP. Netflow data typically does not provide more detail than the TCP/UDP, IP and port information (OSI level 4); application and user level data is not available, so there is a significant amount of information simply not available for analysis and investigation. For the most common protocols, Bro IDS is more powerful because it automatically derives many of the most valuable features and uses the connection ID to link the records across the associated logs. It provides structure and correlation that can help get us closer to ground truth.

**Takeaway:** All of our analytics are attempting to maximize the insight we can get from our data—but we must be cognizant of the limits of what insight can be obtained from a particular data type.[5] Sometimes we need to work within the practical constraints of our data where a low certainty conclusion might be better than nothing, but too many inconclusive results can hinder an analyst from effectively being able to compare what techniques worked to which did not. Even if an analytic is focused on a specific protocol or log type, the availability of corresponding data for investigation and ground truth is very important.

## 3.5 Training Data

To begin experimenting with supervised machine learning algorithms (including deep-learning), we need to have labeled training datasets. Ideally, we would be able to apply a labeled dataset to our use-cases to take advantage of supervised learning approaches, however training data can be especially difficult to construct from network logs for several reasons. First, network logs do not necessarily have a one-to-one relationship between malicious activity and recorded events. For example, in the case of a DNS exfiltration "event," hundreds or even thousands of DNS records would need to be observed with a suspicious pattern in order to trigger a legitimate alert—no single log event would be considered "malicious" by itself. This concept can make applying "malicious" or "benign" labels to the event level difficult and likely misleading taken out of context or removed from the aggregate use-case.[6] We are often looking for malicious *behavior*, not simply malicious *events*. Thus, the sum of events can tell us more than the individual isolated events themselves.

The second challenge is ensuring that the labels are relevant to the question being asked. The most common question is "is it malicious?", but as we know even this can be difficult to attribute to a single log entry. Even in cases of malicious traffic, absolute certainty is not always possible resulting in the analyst's "best guess" as to whether something should be considered malicious or not. But beyond that, the labels that are applied to the dataset need to be relative to the question that is being asked. If we managed to get a whitelist and blacklist of domains to apply labels to our dataset, that can give us information to answer the "Is it malicious?" question. But this is just one of the ways we can look to label the data. One of our exercises was to attempt to label devices within the internal environment based on their network behavior to see if their primary purpose could be determined (Domain Controller, DNS server, scanner, web-server,

[5] Netflow does not contain domain names, so that layer of analysis is going to be approximate at best. DNS provides domains but does not provide the specific ports or protocols that communicate with those domains once they have been resolved. HTTP records can provide details such as user agent strings, MIME types, response codes and methods to get a better sense of what likely happened during the transaction—and PCAP can provide us the contents that were exchanged and likely ground truth. In practice if we have DNS analytics that identify rare and suspicious domains, the analytic may not incorporate netflow records, but the investigation into that result would gain great insight in determining the ports used and bytes transferred to IP address associated with the suspicious domain

[6] The unit of analysis is not always known (nor easy to define), labels are not always defined at the "event" or row-level, just as a pixel of a cat image cannot necessarily be labeled as a cat without the context of the rest of the pixels.

etc.). For this use-case, the benign/malicious label does not apply and, instead requiring us to address the question of "*What is this device's role?"*

Acquiring the right data can be one of the most difficult parts of the training data problem. For an ideal, controlled environment, we would want to obtain training data that is fully labeled to appropriately address the questions we are trying to answer. Training sets exist in this space, but they rarely have the breadth, depth, and steady-state noise to reliably model our use-cases in real-world data. Training sets become out-of-date very quickly, which means they have traffic patterns, devices, and threats that quickly lose relevance in an operational environment. One of the most frequently referenced training datasets was created by DARPA [5]—however the most recent iteration is from 2000.  It is no longer a realistic representation of real networks at this point in time.

### 3.5.1 Considerations for Obtaining Data for Supervised Learning

There are a number of important considerations when obtaining a training data set for supervised learning [2] —it is our opinion that these generally concern the method of generation of the data. Training datasets are usually obtained in one of three ways—simulated, anonymized, or third-party provided with a sharing agreement. Each have their own benefits and drawbacks—and using the best match for your specific situation is up to you.

***Simulated Data***

Simulated datasets will replay traffic modelled on observed behavior and are typically not recorded directly from actual human activity. This can lead to unnatural patterns and artifacts that would not typically be observed in real world data. These artifacts can have adverse effects on analytics, as they become tuned for patterns that do not exist in real world scenarios. Analytics can be tuned very well in a simulated environment, but the results will likely be very unpredictable and inaccurate when they are introduced in real-world scenarios. Existing data sets in this category are both small in total size as well as limited in their applicability to modern, more massive networks.

***Anonymized Data***

Anonymized data provides real-world datasets that have scrubbed sensitive and personally identifiable data to make the original user and organization information anonymized. This data will be closer to real-world scenarios, but the scrubbing process to get the anonymization can be very difficult to perform completely. Sensitive information can show up in places you would never expect—URLs can have plaintext of users' identities for instance. Information can be input in the wrong field, such as passwords accidentally being entered into the "username" field. Mistakes like these are easily missed by the scrubbing process. The anonymization process itself will also introduce artifacts that can alter how the analytics process the data, which can lead to unexpected results in real-world scenarios. Further, anonymized data is much less likely to be labeled than a simulated dataset because it was generated from large, real-world enterprises.

***Third Party or Self-Obtained Data***

The last data type is real-world data—provided by a third party or collected internally. This data is the best option for testing analytics in operational environments. If obtained from a third party, it will likely require strict data handling agreements to include a fortified, secured infrastructure to severely limit access to the data. This can put constraints on the personnel that are authorized to access the data, while introducing bureaucratic hurdles. While this is a necessary requirement for this type of data, these restrictions tend to have less of a negative impact on analytic results relative to the drawbacks of simulated or anonymized data due to their inherent lack of highly sensitive information. Third party datasets are often provided with the understanding that results will be provided back to the data provider in exchange for researcher's access to the data. The obvious drawback to this scenario is that the research team tends to focus more on finding actionable results and have less adherence to generating scientific, repeatable analytics due to external pressures to maintain the mutual value of the sharing agreement. Data providers can shape the direction of the analytics based on their expectations and requirements to "find bad," which may not always align with performing repeatable and sound data science analytics.

**Takeaway:** Labeled network data is difficult to acquire—and the resources are going to be either simulated, anonymized, or real-world network datasets. Each have their own benefits and drawbacks that will apply to your use-cases. In many instances, the scarcity of available datasets may be the largest factor to shape your analytic development process, as is the adage "beggars can't be choosers." To attempt to account for the lack of labeled training data, the practical approach we recommend is to *work with live, updated network data and specifically integrate into a capable, functioning SOC environment*. This setup allows a better chance of getting to ground truth of the analytic results and provides the opportunity to use the SOC ticketing system to apply labels to the datasets and improve your detection and analytics over time. This integration also provides insight into how security operations are performed and gives an opportunity for data science personnel and cyber security experts to more closely collaborate on the formation and development of their use-cases.

# 4 The Structure of the Analytics

This section will discuss the approach and techniques of the analytics themselves. We will not go into the mathematical details of specific algorithms, but rather provide aspects to be considered in the analytic development process that are often overlooked or not considered at the onset. By keeping these concepts under consideration, we hope to better inform the goals, approaches, and results of future analytics.

To start, ensure all parties are using consistent terminology of the term "analytic." People may say they want research into "analytics" but the end state of what that looks like can be nebulous. The specific definition of an analytic is *"separating something into component parts or constituent elements"* [6] which implies that an analytic is as simple as representing the data in a new and insightful way. Many analytic concepts can be described in plain language, such as "We look for web traffic with less than 3 source IP addresses, a single destination domain, one unique URI, and only uses the POST method."  It can also be a new approach of looking at the data, such as "To detect potential Denial of Service (DOS) attacks, we look for periods of activity with a higher than average number of IP addresses that have not been previously observed in the data, that is, spikes of newly observed IP addresses."  Both of these examples can be expressed as query strings or simple code, but it is also easy to convey the steps or concepts in plain English and to rapidly implement on new systems. These concepts are simply a new *way of thinking* about the data, and the transition of the idea can be as simple as a conversation—they are basic concepts than can applied to several datasets to find new insight into the data. We would consider techniques like these to be data concepts and not analytics, but they can be referred to as analytics—and this distinction should be made clear up front.

Others consider an analytic to be complex executable code. For example, K-means clustering is an analytic that can be used to measure the distance and grouping between certain entities, but by itself doesn't generate understandable analytic results. However, an implementation of K-means clustering that has chosen a suite of features and has interpretable, repeatable outputs would require executable code to be run. Successfully applying this algorithm to a cyber use case requires significant domain knowledge of both the intricacies of the clustering approach and the data being analyzed. Complex concepts like this are sometimes referred to as "analytics," and often what is meant by the term "analytic" is a specific, production-tested implementation of a data science concept applied to a use-case. Ideally it would be modular and be able to run in another environment with minimal effort—"push button analytics."  Even better, it would be able to produce a small set of high-confidence results and could be run by an entry-level analyst that does not need to understand what the analytic is doing under the hood but has enough knowledge and understanding of how to interpret and investigate the results.

Clearly, there is a vast difference in the amount of effort, expertise and understanding between the examples above—especially over how to transition an analytic from the research realm to the operational realm. Both interpretations are valid, as this is just a symptom of the popularity of the term "analytic" being used (and over-used) in the current landscape. Whichever definition is used, it's important to ensure that all stakeholders in the process are in agreement with the expectation and understanding of how the term "analytic" will be used during the effort.

## 4.1 The Analytic Spectrum

The analytic spectrum is a general concept that shows the relationship between the cycles that will be spent developing an analytic compared with the cycles that will be spent investigating the results.  It is directly related to the specificity of the use-case the analytic is trying to address. There are exceptions, but normally a more generic use-case will have more generic results, which will require additional time to investigate. A more specific use-case will typically require significant analytic development time, but the results will likely be highly confidence and require less investigation time. The benefits of general use-case analytics include the ability to be rapidly deployed to new environments with minimal changes—so they are excellent candidates for understanding the intricacies of a new network, but they will likely produce relatively uninformed results that need to be investigated in the context of the dataset. The relationship is shown in Figure 7.

When exploring a new dataset, our team was often given the initial guidance of “see what you can find.”  With no previous domain knowledge of the dataset, we were unaware of the specific problem areas, network architecture, and gaps of the data provider. This lead to more generic use cases, looking at traffic patterns for abnormal traffic patterns based on general concepts like port usage, bytes transferred, and unusual node-edge relationships. These analytics often resulted in identifying things such as authorized vulnerability scanners, legitimate administrator activity, and customized internal processes. The analytics were relatively generic, but the time needed to investigate and analyze the results was significant because of a high number of false positives.

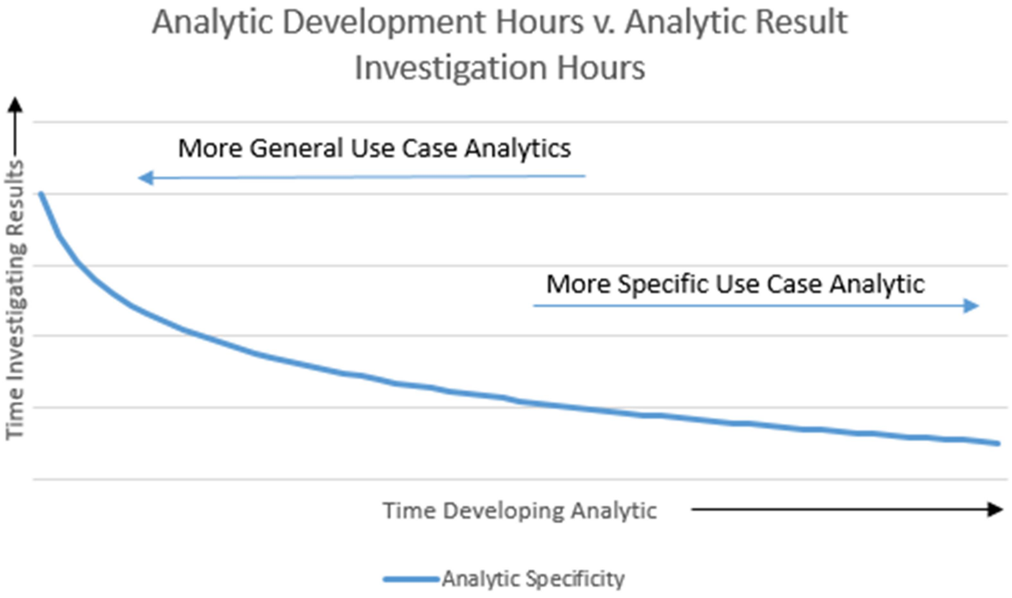


*Figure 7 - The amount of time required for an analyst to investigate results from an analytic is correlated with how specific of a use case the analytic is seeking to detect.*

More specific use-cases can take significant time to develop, and they typically can incorporate unique context and data points that may not be available in all datasets. This could will include narrowing your focus to business critical risks first rather than employing a scattershot approach across the entire enterprise. These use cases will likely only apply to a narrow set of conditions, but the results can be very high confidence. These can lend themselves to push-button analytic solutions because of the amount of information and context that goes into producing the results. However, because of the dependence on dataset context and esoteric details that will differ from one dataset to another, they are typically poor candidates for analytics that can quickly be deployed to new environments. It can be done, but often requires extensive tuning and configuration before the analytic can match its performance within the original dataset. Crafting a defined and specific use case allows a data scientist to more intelligently select features and construct better models. This also includes developing unique pre and post processing steps to better structure the data. It has been our experience that following these basic guidelines has a significant impact on the quality of the results. For example, using different clustering algorithms often matters very little relative to the impact that feature selection and model engineering have in creating successful analytics [7].

With all analytic development, it is important to ask three questions:

- Can it detect something new or reinforce a different, lower-confidence detection?
- Is this analytic the best approach for this use-case?
- Is the goal to find something new or to reduce existing workload?

***Can it detect something new?***

It is always exciting when your analytic has a legitimate result—it has found what it is looking for! However, a check to see if this is a new result or one that was found previously will help us determine the functionality of this result. For example, if a complex, data science technique detects a malicious activity which can be detected faster and with equal or greater confidence, such as using a simple count and descending sort, the utility of the computationally expensive data science approach is significantly reduced. Care must also be taken to understand the *reason* for the detection (or lack thereof)—not all datasets will contain the attack, so a lack of results does not mean the analytic did not necessarily work.

***Is this analytic the best approach for this use-case?***

In general, an analyst's goal is to detect malicious activity with the simplest approach possible—this does not outright mean that a data science analytic with the same result as a simpler query is not successful. Often, malicious activities have multiple behaviors that differentiate it from the benign traffic within a dataset—so even if the specific result of malicious behavior was already found, if the data science approach leveraged a different feature set to identify the result it could have significant value and boost confidence of specific signatures amongst the noise. The ability to identify unique, suspicious aspects within the data is what should be evaluated—not only the specific results themselves.

***Is the goal to find something new or to reduce existing workload?***

The advanced analytics may not find new results, but they can still be successful if they provide insight or context that significantly reduces the investigative workload. For example, in one use case, the cyber analysts needed to spend 40+ hours manually stitching together two disparate but related datasets—there was no common feature to join on other than the timestamp. However, an analytic pipeline or data engineering process was developed to conduct probabilistic record linkage across the two logs. Once this was generated, the resulting output allowed the cyber analysts to perform the same analysis in a matter of minutes. The analytic wasn't intended to find a result, but rather its goal was to create new data points that significantly reduced the existing workload. These types of analytics won't necessarily have direct malicious results to point to, but their development significantly improved the analyst workflow and created new features which enabled the generation of additional analytic results and investigations.

**Takeaway:** Establish consistent definitions between all stakeholders on the terms and expectations of the effort. Ensure stakeholders understand the analytic spectrum, and the tradeoffs between general and specific use-cases to maintain realistic expectations.

## 4.2 The Semantic Gaps

One of the primary goals of applying data science analytics to cyber relevant data is to provide high-confidence results to entry level cyber analysts to investigate and triage. The goal is to leverage the advanced logic driven results of data science but lower the level of expertise needed to interpret the results. However, for this to be successful, it is important that the analytic results are expressed in terms that have meaning to the cyber analyst [2].

Many of the data science driven algorithms take the dataset and perform feature engineering. These features, such as outlier scores, distance metrics, clustering similarities, and relational graph edges, drive results—these features often allow for the identification of anomalous enterprise activity, helping anomalies to statistically stand out from the rest of the data. This type of analysis is the heart of what drives the data science– but it can be an impediment to investigating the results for the cyber analyst.

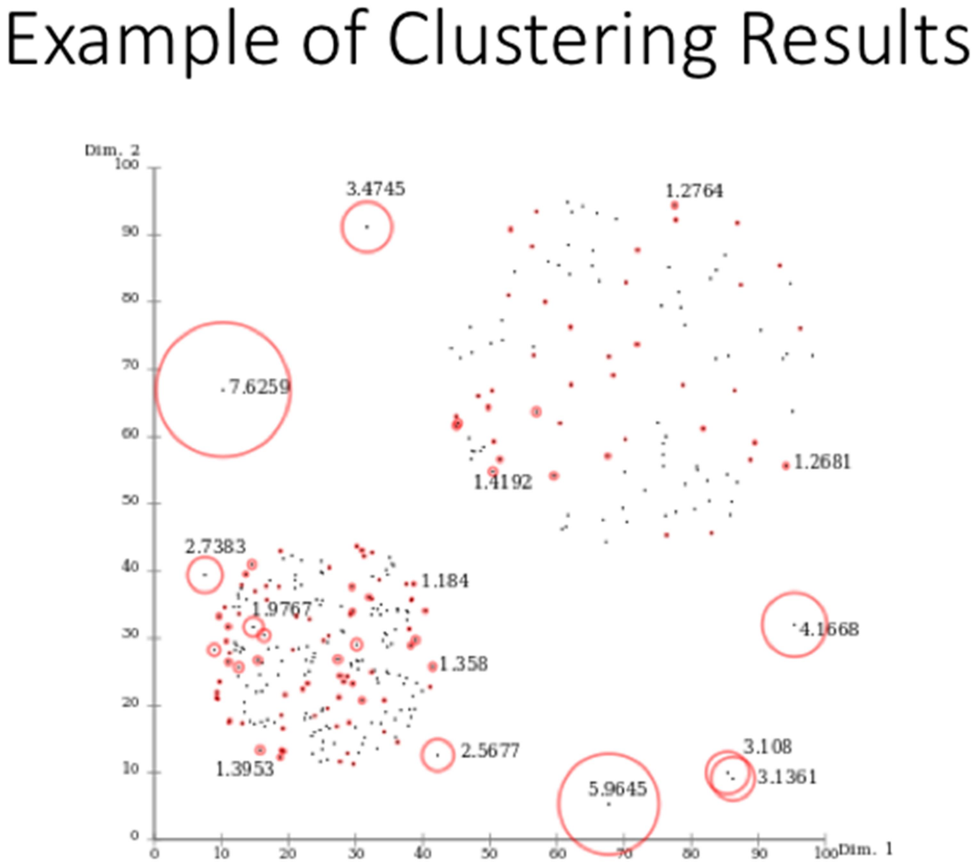


*Figure 8 - It is non-intuitive to identify suspicious cyber security behavior from this graph, so translating this output into something familiar to a cyber analyst is critical.*

Cyber analysis is rooted in Digital Forensics and Incident Response (DFIR)—evidence driven analysis. In order to investigate the data science analytic results, we need to assess “what happened” to identify the ground truth explanation of the results.  This normally includes investigating traffic patterns, IP addresses, ports, bytes transferred, domains, user-agents, and other tactile data points that can be attributed to a timeline of “what happened.”

When cyber analysts receive analytic results that explain anomalies in terms of mathematical terms, they can be difficult to interpret and bring back to the tactile, sequence-based logs to find an explanation as seen in Figure 8. For example, if a cyber analyst is provided a list of results from a run of K-means clustering looking for anomalous IP addresses and the top result is given an outlier score of -7.6259, the cyber analyst needs to investigate several aspects of that IP address to determine what makes it unlike the rest of the data. This is not a specific lead, and the analyst would need to investigate several aspects of the result to begin to make sense of possible reasons for this outcome. A cyber analyst is likely to ask, “When was this IP acting suspicious?” and that depends on the window of data that the analytic analyzed, and in fact cannot point to any specific row in a log. Going through several results like this, cyber analysts may become fatigued and are more likely to misunderstand the results—this could result in improper investigation or overly hasty triaging as too complicated for analysis in favor of more immediate, clear results. Even if the accuracy of the data science analytics is high, if their accuracy cannot be successfully explained to the target audience (cyber analysts) they do not provide much operational utility.

This problem exists at various levels—not all data science driven analytic approaches have this issue to the same degree. Analytics aimed at identifying time sequences, for example, may perform analysis that can be presented in terms of the data itself, and so results that show “the following IP address had periodic activity every 6 minutes to the suspicious domain” is readily understood and interpreted by cyber analysts. However, more abstract data science concepts, such as graph analysis and clustering methods, tend to rely heavily on engineered features producing results that are far abstracted from the original data and harder to interpret.

**Takeaway:** To do their jobs, cyber analysts have to investigate the technological first-principles that create anomalous artifact trails identified by data science. The goal should be to provide results in the terms of the governing first-principle data, not in terms of the mathematical analysis or engineered features that were used to identify the anomalies. Even using plain English results such as “The analysis found that this host was 98% different than the normal profile” does not give the cyber analyst a place to start their investigation, or a specific event or time range to focus on [2]. To be successful, the results need to be provided using the terms of the original dataset as much as possible—and minimize the level of abstraction that is provided back to the target audience.

## 4.3 The Base Rate Fallacy

One of the most overlooked aspects of data science driven analytics is the base rate fallacy [8]. The base rate fallacy is a logical flaw that leads people to assign greater emphasis to specific information (such as the results of an analytic run) over generic information (such as how likely an event truly is within the dataset). In other words, malicious cyber events are very rare in most data sets—often on the order of one in a million or one in a billion records. Thus, the actual chance that we have detected a malicious event is extremely rare—yet we tend to think that our analytic results are very likely to be true malicious detections.

For a real world example, we can look to medicine. Doctors frequently administer tests on patients to see if they have a specific trait, such as left-handedness. Let us say that we know the **base rate** of occurrence of this trait is 25% of the population.  On a particular test we know the percent chance of getting a true positive (**recall**) is 99%. We also know that the false positive rate is 10%. With these factors, people tend to believe that the true detection rate is close to the recall of 99%, however if we were to run 1000 tests we would have 247.5 true positive detections and 75 false positive detections, for a positive prediction value (**precision**) of only 77% as outlined in Figure 9.

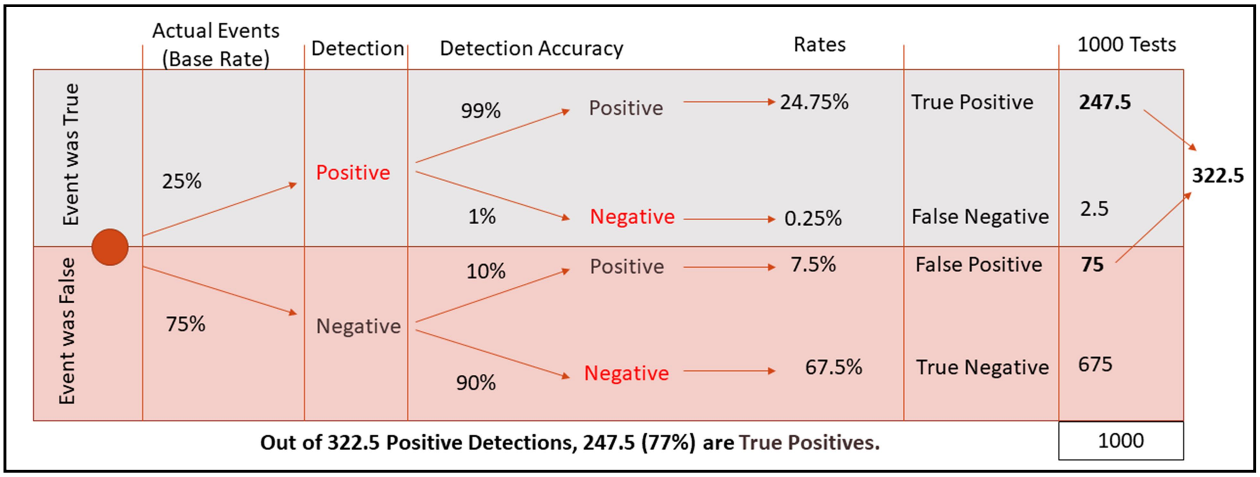
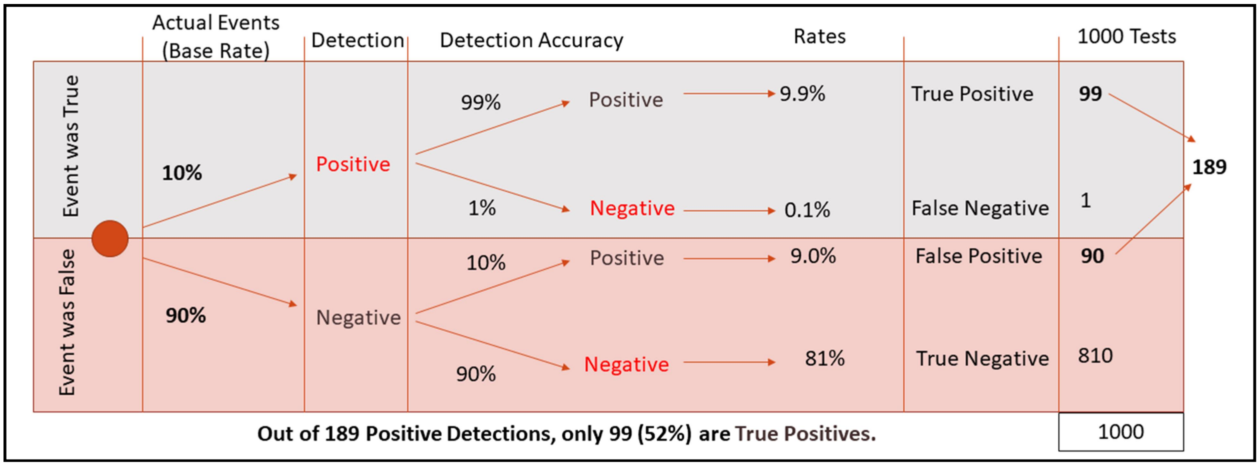


*Figure 9 - Both recall—the odds of a true positive—and the false positive rate dramatically impact the precision.*

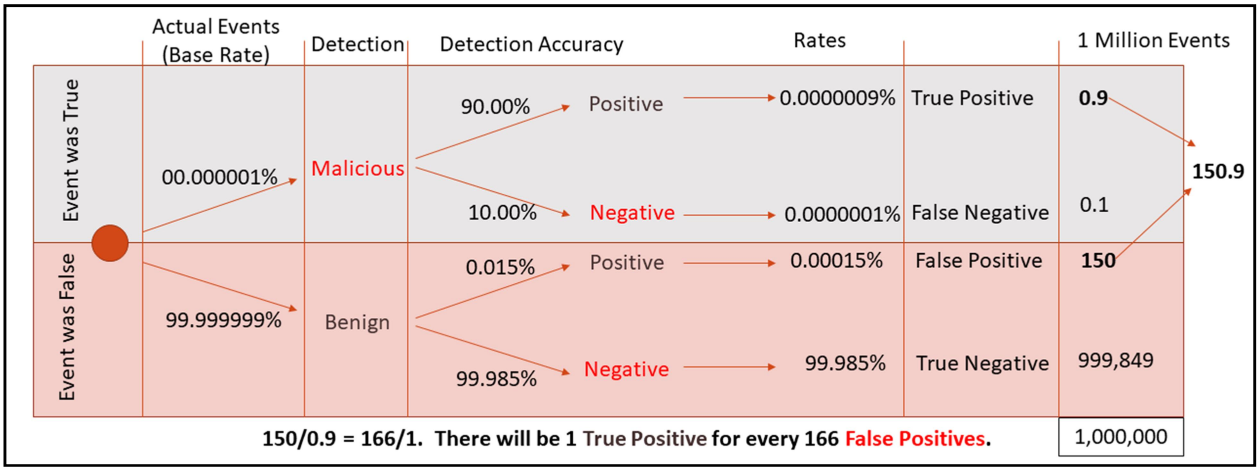
Using the same recall and false positive rates but applying them to a scenario with a base rate of 10% shows how the base rate can impact the results. In this scenario, 1000 tests will result in 99 true positives and 90 false positives—resulting in 52% precision. Effectively what is happening is the relatively small false positive rate is being magnified by 90% of the data while the 99% detection rate only applies to 10% of the data as can be seen in Figure 10. The number of instances in the dataset has a significant impact on the resulting precision [9].



Out of 189 Positive Detections, only 99 (52%) are True Positives.

*Figure 10 - The lower the Base Rate, the more dramatic of an impact your false positive has on your results.*

Applying this to cyber-security use cases, the problem is magnified significantly. Cyber use cases are looking for that needle in a haystack—malicious or anomalous events that occur very infrequently in a data set. If we take a signature or technique that can detect malicious activity with 99% accuracy and a tiny 0.015% false positive rate and apply it to a scenario with a base rate of 1 event per 1 million logs we start get a sense of the scale of this problem. In 1 million tests, this scenario will have 0.9 true positive detections and 150 false positive detections—effectively 1 true positive for every 166 false positives as outlined in Figure 11. Precision at these levels can result in significant analyst hours investigating and following-up on events with no consequence.



150/0.9 = 166/1. There will be 1 True Positive for every 166 False Positives.

*Figure 11 - Here our minuscule base rate penalizes even the smallest of false positive rates.*

We can use the same scenario but let's say we successfully tuned our detection analytic to have 100% accuracy—in other words, we know it can detection every single instance of the malicious event in the dataset. However, the false positive rate remains unchanged at 0.015%. In this improved scenario, we still have 150 false positives for every 1 true positive detection—which still leads to a significant investment in analyst cycles spent investigating benign activity.

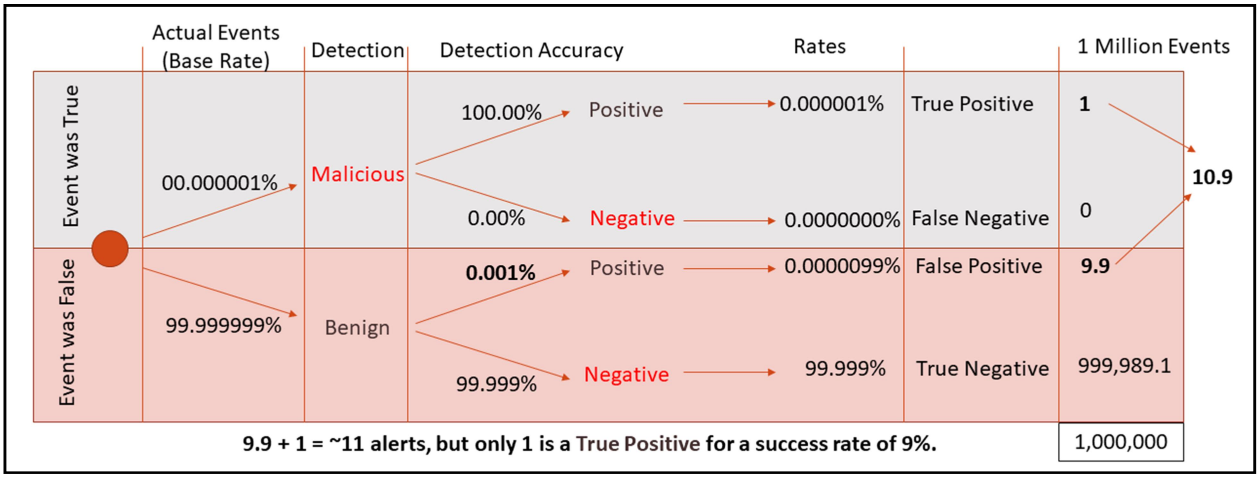


*Figure 12 - At larger scales, minimizing false positive rates pays substantial dividends for your cyber analysts.*

We can start to see a significant improvement if we are able to improve the false positive rate. If we are able to reduce this from 0.015% to 0.001%, in our 1 million test scenarios we now detect 9.9 false positives for every 1 true positive for 9% precision as is outlined in Figure 12. While this means the cyber analysts will need to investigate 10 times as many false positives as truly malicious events, it is still a marked improvement from the 150 results discussed earlier.

The examples above are theoretical—in practice, with an unlabeled dataset it is extremely difficult to know the true event base rate of your data as well as the recall and false positive rate of your analytics. Values can be sampled and estimated, but without a complete labeled dataset to test you will likely not be able to know these specific values in practice. The values are also volatile and change over time as specific attacks techniques evolve. Despite these challenges, the key takeaway is to get an understanding of the problem set—even “pretty good” detection of very rare events in extremely large datasets can be dwarfed by the detection of benign events.

**Takeaway:** As discussed earlier, analytics are often judged and touted based on the number of significant, real-world detections they have discovered. However, the number of false positives that were generated and investigated en route to those impressive finds is often overlooked. We recommend placing an emphasis not only *on increasing the detection accuracy* of the analytics but also expending significant resources into understanding and *reducing the false-positive detections* as much as possible. This can be something as simple as post-processing results with known whitelists to remove the bulk of the false positives, but one of the primary goals of data science analytics is to reduce the workload on cyber-expertise personnel—their time is a limited resource and needs to be treated as such. Each analytic is different—there may be cases where the impact of the threat the analytic is detecting is so great that the acceptable number of false positives is significant. But there are others where a cost-benefit of the analytic accuracy compared to analyst time needed to follow-up should be explored and evaluated. This is another reason why specific use-cases and analytics can improve the downstream process—they have a narrow goal and can typically lead to reduced false-positives as part of the analytic.

# 5 Conclusion

Data science and cyber security are both relatively new fields of study.  As such, we must carefully consider our approach in the research and implementation of new analytics for network defense to avoid spending unnecessary cycles on implementation and support problems. Crafting specific use cases and carefully selecting features plays a significant role in the success of an analytic; therefore, ensuring that we perform this initial due diligence is critical. The measurements, the data, and the analytics themselves need to be carefully and methodically understood early in the building process so they can be leveraged effectively. The choice of one algorithm over another is frequently dwarfed in effect by the importance of solid feature selection and data engineering. We also need to think critically about the shortcomings and pitfalls of advanced cyber security analytics that we develop and observe, to ensure that we do not misplace our confidence in results before they are fully proven or understood. We hope that the lessons learned and considerations discussed in this whitepaper will help the community move closer to that goal.

# References

[1] C. Zimmerman, Ten Strategies of a World-Class Cybersecurity Operations Center, Bedford, MA: The MITRE Corporation, 2014. Available: https://www.mitre.org/sites/default/files/publications/pr-13-1028-mitre-10-strategies-cyber-ops-center.pdf.

[2] R. Sommer and V. Paxson, "Outside the Closed World: On Using Machine Learning for Network Intrusion Detection," in *Proceedings of the IEEE Symposium on Security and Privacy 2010*, Oakland, CA, 2010. Available: http://www.icsi.berkeley.edu/pubs/networking/outsidethe10.pdf.

[3] T. Crothers, "Leveraging Machine Learning for Cyber Threat Hunting," 2017. [Online]. Available: https://sqrrl.com/media/huntpedia-web-2.pdf.

[4] V. Beal, "The 7 Layers of the OSI Model," 03 April 2018. [Online]. Available: https://www.webopedia.com/quick_ref/OSI_Layers.asp.

[5] MIT Lincoln Laboratory, "DARPA Intrusion Detection Data Sets," 2000. [Online]. Available: https://www.ll.mit.edu/ideval/data/.

[6] Merriam-Webster, "Analytic," 5 June 2018. [Online]. Available: https://www.merriam-webster.com/dictionary/analytic.

[7] A. Emmott, D. Shubhomoy, T. Dietterich, A. Fern and W.-K. Wong, "A Meta Analysis of the Anomaly Detection Problem," arXiv:1503.01158v2, 2016. Available: https://arxiv.org/abs/1503.01158.

[8] J. Bollinger, B. Enright and M. Valites, Crafting the Infosec Playbook: Security Monitoring and Incident Response Master Plan, Sebastopol, CA: O'Reilly Media, 2015.

[9] N. F. e. a. Rusland, "Analysis of Naïve Bayes Algorithm for Email Spam Filtering across Multiple Datasets," in *IOP Conference Series: Materials Science and Engineering, Volume 226*, Melaka, Malaysia, 2017. Available: http://iopscience.iop.org/article/10.1088/1757-899X/226/1/012091