

# Dynamic Ensembles for Anomaly Detection: An Application in Malicious Domain Investigations for US Army NETCOM

Lydia Barit, Kathryn Dula, Blake Jacobs, Harrison Leinweber, John McCormick, Roberts Nelson

## 1 Executive Summary

### 1.1 Project Overview and Discussion

The purpose of our project was to help NETCOM understand the effectiveness of anomaly detection models in use cases pertinent to the Command's mission and effectively incorporate these models into its current workflow. We accomplished this through an extension to NETCOM's existing Apache Airflow malicious URL detection pipeline. We developed an additional Directed Acyclic Graph to engineer features, run four anomaly detection algorithms (xStream, isolation forests, local outlier factor, and one-class spanning vector machine), and generate Shapley Additive Explanations for the data. We then present these results and explanations in a web-interface of our own design. Finally, we encapsulate our work within a sample risk framework to show how it can contribute to cyber risk management practices both on the front line and across the Command.

### 1.2 Project Deliverables

The following comprise our final set of deliverables:

1. **Project Report:** A document which details the datasets we analyzed, model selection rationale, our design process, and a sample risk framework.
2. **Final Project Presentation:** Audiovisual presentation delivered on 3 May 2022 which summarized this project report and provided an opportunity for stakeholders to seek clarification and further information.
3. **Project Video:** A condensed version of the project presentation is available here: <https://youtu.be/zBE4twWFRD8>
4. **User Interface Code:** Implemented in RShiny, our front end allows the analyst to interact with the data and models and provides the ability to see detailed anomaly score explanations, mute features, and mark specific URLs for further investigation. Directions for installation are provided with the user interface code.
5. **Amazon MWAA Cloud Pipeline:** A stand-alone Directed Acyclic Graph (DAG) and corresponding source code is provided to clean, score, and generate explanations for the example NETCOM NULLFOX data. The example data, intermediate results, and final data files are also provided for clarity in interpreting the source code.

## 1.3 Documentation

### 1.3.1 User Interface

Source code for the R-Shiny web application consists of a single R script “app.R” and an adjacent directory ‘/www’, which contains the .jpg files used in the application’s display. These have been included in a compressed zip file labeled “NETCOM\_App\_Prototype.zip” and are available on the projects github page.

The app script file has been documented throughout with in-line comments and headers. These headers have been structured in a outline format, which when loaded into R-Studio provides a clear organizational structure for reviewing, understanding and modifying the app. The code is split into four primary sections: Loading the Data from AWS (which includes some additional data processing), the UI, and the Server. Each of these sections additionally have helper functions which are essential to procedurally generating the app based on the features included in the data. These functions will be essential to maintaining the app moving forward, given the high likelihood that features may change as the pipeline is improved and tweaked. However, even with the procedural generated code, the app has strong coupling with the decision to include the four anomaly detection algorithms. Adding more algorithms (or removing some of the existing ones) will require significant refactoring.

Additionally, the app is currently connected to the AWS bucket created by the DAG scripts and the Amazon MWAA Cloud Pipeline. This can be modified by editing which aws s3 bucket is being referenced and changing the AWS authorization id and key. Though there is weak coupling with the specific data structure generated by the pipeline as designed, this can be changed if desired by refactoring the data processing portion of the app.

Finally, regarding the deployment of the app, a number of options are available for deploying an R-Shiny app. The easiest method is deploying the app to shinyio, a free service provided by RStudio. This service can be upgraded to a premium tier which provides additional resources and bandwidth dedicated to the app. Additionally, a third party or enterprise hosted R Shiny server could be established. This capability is provided by the AI2C’s development platform Coeus.

### 1.3.2 Amazon MWAA Cloud Pipeline

Source code for a stand-alone DAG in Amazon Managed Workflows for Apache Airflow (MWAA) is provided along with example data that was produced by the DAG. This DAG was tested on Amazon MWAA to ensure that the requirements file and directory structure are correct and that the DAG can run without error. The code and data is provided as a compressed zip file with the name “NETCOM\_cloud\_backend.zip”. Inside this directory are two subdirectories: “dags” and “data”.

The “dags” directory contains all the source code for running the DAG. In Amazon MWAA you should set the root of the DAG to this directory. The directory has a “requirements.txt” file that AWS MWAA reads to manage the DAG dependencies. The DAG itself is provided in “anomaly\_detection\_dag.py”. The source code that the DAG calls is located in the “src/” directory. The source code includes “CleanEnrichedData.py”, “CreateAnomalyScores.py”, and “CreateSHAP.py” which handle the data cleaning, scoring, and SHAP explanations respectively. Additionally, the directory includes the source code for the python implementation of xStream which

you can also find on Github.<sup>1</sup>

The “data” directory contains the example NULLFOX output data (enriched.altIP.csv), the intermediate results in the pipeline (CleanEnrichedData.csv), as well as the final outputs of the pipeline. The pipeline produces five final outputs. The first is “enriched.altIP\_with\_scores.csv” which is a copy of the original inputs to the pipeline with the four anomaly scores added to the end of the data. The other four outputs are SHAP values for each of the four anomaly detection methods. These files are intended to be used as concrete examples of the inputs and outputs of the DAG as it progresses to clarify the comments and documentation provided in the source code.

In order to get integrate the DAG into NETCOM’s NULLFOX workflow, the only changes that should be needed are some resource strings for Amazon S3 buckets and object keys. All locations in the code where these changes need to happen are marked with a “TODO” comment to aid in the transition to NETCOM’s production environment.

## 1.4 Suggestions for Future Work

We suggest the following five work streams for future work:

- Searching for “malicious URLs” in an unlabeled data set leads to difficulties in evaluating model performance. Websites can present numerous threats to the enterprise (ex. spam, phishing, drive-by downloads, data exfiltration, or general espionage). We suggest determining the specific threat that NETCOM is searching for in order to allow for better model evaluation and comparison.
- Not all anomalous URLs are malicious. Further research is needed to determine whether anomaly detection models are the best paradigm for investigating malicious URLs. We suggest investigating additional families of algorithms to see if there exists an alternative which performs better on this data.
- We suggest increasing the window of observation from one week. Evaluating performance and noise on a well-labeled dataset over periods ranging from bi-weekly, monthly, bi-monthly, and semi-annually would be ideal.
- Further work is needed in order to enable our pipeline to work in-stream and on a dynamic dataset.
- We suggest utilizing our sample risk framework to create and maintain an organization-specific risk register which provides analysts and managers the details needed to map *anomaly* scores to *risk* scores. This register would allow for effective and efficient prioritization and response to findings.

---

<sup>1</sup><https://github.com/cmuxstream/cmuxstream-core/tree/master/python>

## 2 About our Client

The Network Enterprise Technology Command (NETCOM), formally established in October 2002, leads global operations for the Army’s segment of Department of Defense Information Networks (DODIN). Its mission is to ensure freedom of action in cyberspace while denying the same to the United States’ adversaries. To accomplish this, NETCOM has set up a team-of-teams structure, one of those teams being the Data Science Directorate(DSD), established to provide integrated, advanced analytic capabilities to enable objective decision-making in support of DoD Information Network Operations. Our client, NETCOM DSC-PIT, is a Data Science Center (DSC) based in Pittsburgh, PA that directly supports the DSD. Academic partnerships are a key strategic capability of DSC-PIT, and they approached us in order to leverage Carnegie Mellon University’s machine learning knowledge to further NETCOM’s mission.

## 3 Project Objectives

The NETCOM data science directorate has an enduring interest in developing algorithmic methods for detecting malicious internet traffic on the US Army’s network. Of particular concern is developing a system for detecting emerging threats from newly created domains. Despite this interest, the current anomaly detection pipeline for analyzing new internet domains is largely non-functioning. The current system is slow, tedious and predominantly manual, wasting a significant amount of man hours. Moreover, NETCOM does not have a formalized set of internal processes or procedures to evaluate new anomaly detection capabilities before putting them to use. This lack of a model evaluation process is coupled with under-developed risk response framework making the output of the any algorithmic process unreliable and often unactionable.

Our solution to these nested problems was to (1) develop a model evaluation methodology and apply it to specific anomaly detection algorithms of interest to NETCOM, (2) improve the efficiency and dependability of their anomaly detection pipeline by integrating ensemble models, post-hoc explanations (XAI), and a front-end interactive user interface, and (3) create a risk-response framework that will inform how NETCOM analyst turn the anomaly scores into actionable risk scores.

We see these separate project goals and the specific set of solutions we developed as part of a holistic methodology for evaluating the effectiveness of algorithms and putting them into production. In order to display the efficacy of this methodology, we aim to create an anomaly detection report system that generates post-hoc explanations for why a specific event was anomalous, leading to easy understanding and rapid response by a human network analyst. This will leverage an existing algorithm, created by a team of Carnegie Mellon researchers led by Dr. Leman Akoglu, in the DSC-PIT research pipeline as a proof of concept that will enable DSC-PIT to more efficiently evaluate future machine learning (ML) models introduced by research partners.

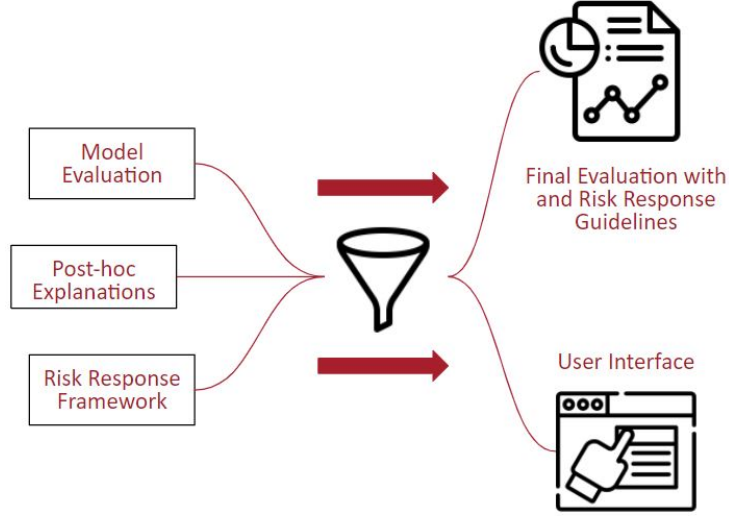


Figure 1: Project lines of effort and key deliverable products

In addition to the above objectives, the testing and evaluation of the aforementioned algorithm will help to shed more light on its specific capabilities and utility to DSC-PIT. Further, the development of a human readable support system off of the basis of our methodology will form a front-line foundation to manage risk.

## 4 Value and Business Impact

The primary business impact our solution is the improved efficacy and efficiency of the anomaly detection process, which will lead to increased precision of post-hoc investigations while maintaining high recall of malicious traffic. In order to get a baseline to compare the value of our solution, it's useful to consider the existing anomaly detection process. Currently, DSC-PIT has an AWS-deployed pipeline which identifies which newly registered internet domains that have been visited by web users on the US Army enterprise network. Publicly available data on these web domains is then aggregated. This enrichment includes virus total score, which is a commonly used cyber security metric displaying how many mainstream security softwares block the website. Noticeably, the current pipeline does not have any system for

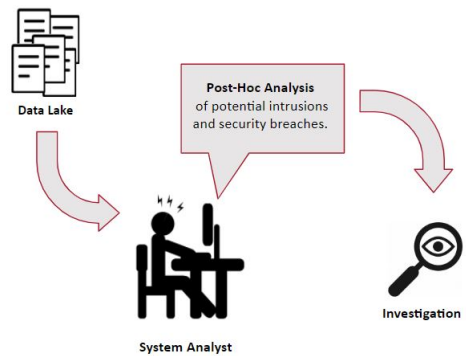


Figure 2: The existing anomaly detection work flow employed by NETCOM DSC-PIT.

ranking which of these websites are most anomalous or prioritizing further investigation. As such, the data scientists at DSC-PIT are forced to manually compare these observations each week.

Outlier detection algorithms are a natural means to address this issue, since they can provide a consistent way to identify observations that are at least statistically anomalous. However, when employing an algorithm to replace a human being, there is a natural loss of interpretability and clear understanding. This imposes an inherent challenge in the work flow, since the investigators will be less likely to trust a model than another human analyst. Moreover, even if they do trust the algorithm, a simple anomaly score can provide an explanation making it unclear what specifically to investigate about a new domain.

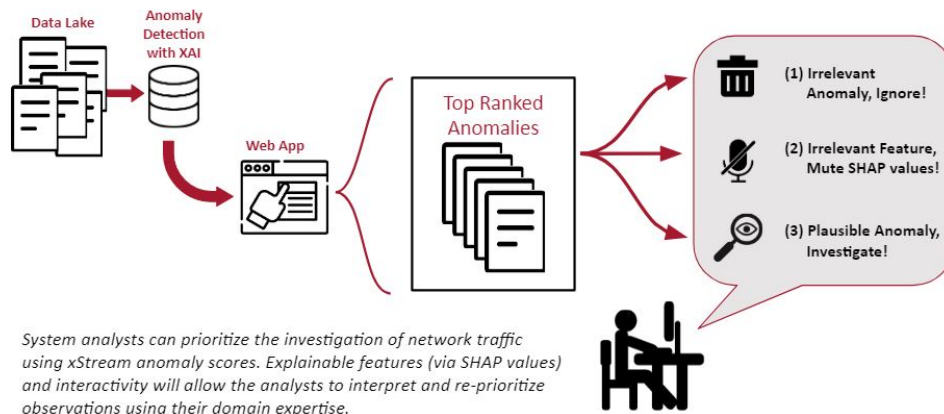


Figure 3: The Proposed workflow based on human-computer interaction. Though domains are ranked by a dynamic anomaly detection ensemble, the decision to prioritize for investigation is left to a human user.

Thus our proposed workflow relies not only improving the speed and accuracy of anomaly detection through automation, but also human-computer interaction based on XAI and an interactive user interface. Specifically, we believe our proposed solution offers two measurable improvements in the pipeline and three capabilities that were not previously available. The improved workflow is expected to decrease time spent on analysis and increase precision of anomalous traffic investigations. (In layman’s terms, we expect that NETCOM will be able to identify more true threats faster). These two improvements will be measurable and verifiable once the model is integrated into the existing workflow.

Additionally, the new workflow has additional features that were simply not present in the prior pipeline. As such these new features cannot be compared directly against the old baseline, but will contribute new forms of value. First, and most obviously, the explainability and interpretability offered by the XAI implementation are new to our implementation and will offer value in model trustability and fidelity throughout the pipeline. The dynamic ensembling of the models (discussed in detail in the user interface section) is an additional feature that does not have a clear baseline. However, the ability to rapidly experiment with new model configurations and features is a powerful tool for DSC-PIT data scientists. Finally, the risk management which provides guidelines and informed decision-making throughout the process is another contribution that NETCOM currently has no means to employ.

## 5 Project Methodology

### 5.1 Data Sources

Currently, Army NETCOM DSD collects a massive amount of network data and stores them in a cloud-based data lake. The data is streaming continuously into the lake, and in most cases the data is unlabeled. Through discussions with the NETCOM DSC-PIT team, we highlighted two key areas of interest: (1) network intrusion and (2) malicious domains. In order to prototype and verify our approach in these areas, we used a publicly available data set for each category. Additionally, we received an anonymized data set from NETCOM to demonstrate our methodology on identifying potentially malicious domains visited on the Army network. These data sets are described in detail below.

#### 5.1.1 Network Intrusion Data

##### Data Description

This dataset we used in our first case study is called CICIDS2017 [24]. It is a simulated, computer-network, intrusion detection system (IDS) dataset. This dataset was simulated as follows:

*To create a comprehensive testbed, we designed and implemented two networks, namely Attack-Network and Victim-Network. The Victim-Network is a high secure infrastructure with Firewall, Router, switches and most of the common operating systems along with an agent that provide the benign behaviors on each PC. The Attack-Network is a completely separated infrastructure designed by a router and switch and a set of PCs with public IPs and different necessary operating systems for executing the attack scenarios. The following sections discuss the infrastructure, benign profile agent and attack scenarios[24].*

In this data, each observation is a packet of information sent across the simulated network. The observation is given the label 'BENIGN' if the packet was sent by a Victim process and another label corresponding to the attack type if sent by an Attacker process. The features describe aspects of the packet such as its size, and average byte flow rate.

We chose to use this labeled dataset because it contained the most common operating systems (Windows, Linux and Macintosh), and simulated 12 current intrusion attack types across a typical "work week" of network traffic. By virtue of the labels, our team is able to use supervised machine learning techniques to tune the hyper parameters of the models and analyze their respective anomaly detection performances. The dataset is highly imbalanced, and thus representative of real world web intrusion attack conditions and appropriate for anomaly detection. Malicious traffic (i.e. an intrusion attack) represented 20 percent of the total network traffic. The dataset is also very large, with just shy of 3 million observations. Due to computing constraints, we took a representative, random sample of 10,000 observations to conduct our anomaly detection analysis on. This random sample maintained the original class imbalance at about 20 percent, and included observations from each of the 12 attack types. Most importantly this is a relevant problem set for NETCOM, part of whose mission is to defend the U.S. Army's cyber networks. This dataset presents an analysis on the efficacy of using unsupervised anomaly detection to prioritize investigative manpower on malicious network traffic in NETCOM.

## Data Processing

This dataset was relatively simple to prepare for use with the anomaly detection algorithms. We removed the small proportion of rows with incomplete observations before taking the representative sample of the remaining, as discussed above. We made this decision because there was sufficient data remaining to complete our analysis without needing to introduce imputation errors into the data. All but one of the 79 original features were numeric and simply had to be normalized. We dropped eight features that were effectively empty (columns only containing zeros). Finally, we had to transform the single categorical feature representing a packet’s port number of entry on the destination process. We used a truncated one-hot-encoding process to break-up the original port number feature into domain-relevant binary features. For the well-known ports (0-1023), we one-hot-encoded each unique port. Then we engineered two additional binary features, one for the registered ports (1024-49151) and one more for the dynamic ports (49152-65535) where any destination port value that falls within one of those ranges would result in a 1 in its respective engineered binary feature [8].

Our rationale behind this encoding scheme was to extract the maximum amount of information while reducing noise. When a computer connects on a well-known port, it will virtually never switch ports for the same service, so a one-to-one encoding is appropriate. Additionally, unlike the other port ranges, a system generally requires administrator privileges to bind to a port in this range (a further certification of the port’s intended use).<sup>2</sup> The registered ports are more loosely defined and do not offer the same one-to-one guarantee as the well-known ports, which is why we decided to group them all into one class.<sup>3</sup> Ports in the ephemeral range are usually dynamically generated by the system. We decided to group connections in this range into one class because these connections generally substantially differ from those in the other two ranges and port numbers generally do not repeat themselves.

### 5.1.2 Publicly Available Malicious Domain Data

#### Data Description

This dataset is hosted publicly on Kaggle and was created by a team of undergraduate researchers to provide a labeled dataset of malicious and benign URLs. The URLs were collected in a "low interactive client honeypot to isolate network traffic," with additional resources used to generate more features (ie. Whois) [27]. The malicious URLs were collected from the following blacklist websites:

- [machinelearning.inginf.units.it/data-andtools/hidden-fraudulent-urls-dataset](https://machinelearning.inginf.units.it/data-andtools/hidden-fraudulent-urls-dataset)
- [malwaredomainlist.com](https://malwaredomainlist.com)
- [zeuztacker.abuse.ch](https://zeuztacker.abuse.ch)

---

<sup>2</sup>Malware is capable of abusing this port range; however, most use cases involve masquerading with the traffic type that is native to the port. In the unlikely case malware uses an abnormal traffic type over a well-known port, it is even more unlikely that the malware would dynamically change which well-known port it is operating on, as that would cause significant, noticeable side-effects on the victim machine.

<sup>3</sup>It may have been a good idea to identify common registered ports in our dataset or common ports for abuse (1337, 4444, 8888, 12345, etc.) and one-hot-encode them; however, we decided against this due to the large number of additional features it would introduce.



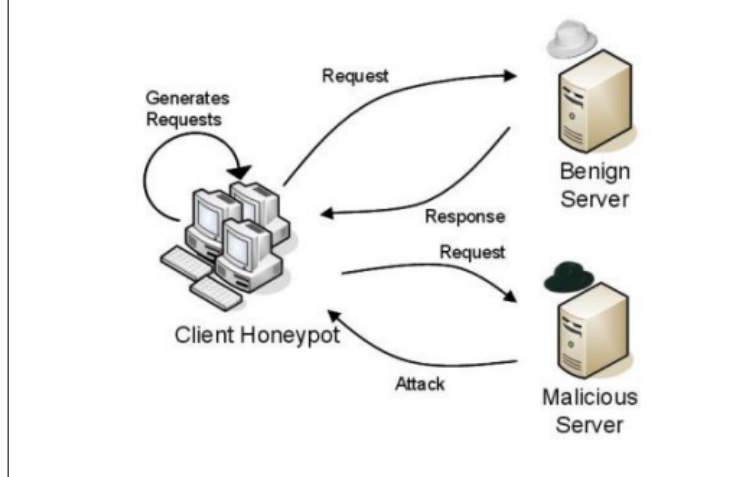


Figure 4: The data was collected with a honeypot.[21]

There are 1781 observations in the dataset each with 20 features. Each observation is a unique URL with features consisting of categorical and numeric information like country of registration and content length of the HTTP headers for the URL.

This dataset was selected to use for a case study because it is the closest public, labeled dataset to the one NETCOM is looking to apply anomaly detection to. It is important to note that this dataset has a much less information rich feature space than NETCOM’s real data. We completed the analysis to give NETCOM a rough inclination of how these unsupervised models will perform on their real unlabeled malicious URL dataset. In other words, the analysis of this dataset offers a baseline expectation for how the models will perform.

## Data Processing

A nontrivial amount of data engineering went into preparing this dataset. First, a few features were removed that couldn’t offer intelligible numeric information such as the URL string and various dates. We chose to steer clear of doing a Bag of Words or n-gram analysis on the URLs because the anomaly detection models we are using are not intended to handle that sort of data. Without using something like a long short-term memory (LSTM) neural net model that can take into account time series, it also doesn’t make sense to include dates.

Next we dealt with missing values. Most features with null values were categorical so without augmenting with another source of data, the imputation of missing values is highly prone to error. Instead, we filled null values with the string *'missing'* as a work around to be able to One Hot Encode the feature later. For the numeric features with null values, we chose to replace them with 0.0 based on domain guidance.

Lastly, there was a data integrity issue with the *'WHOIS\_STATEPRO'* feature. Aside from missing values, there were also values that were misspelled or formatted in multiple ways. For example California was represented as : *'CA'*, *'USA'* and *'CAL'*. So we hand-created dictionaries to correct for the most common state formats and country codes before one hot encoding.

This process resulted in the preservation of all data points and a final count of 455 features.

### 5.1.3 NETCOM Malicious Domain Dataset

#### Data Description

Currently, NETCOM collects data on newly visited domains on the Army network and enriches them through a cloud pipeline called NULLFOX shown in Figure 5. They have an interest in detecting malicious domains in this dataset using anomaly detection methods.

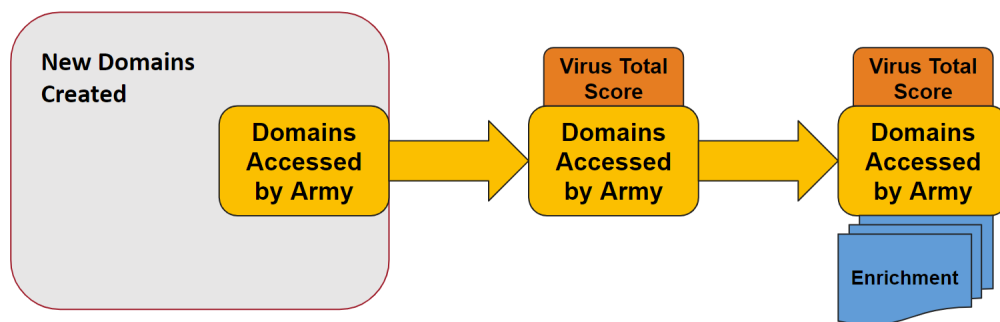


Figure 5: This figure is a simplified version of NETCOM’s data collection pipeline.

The dataset did not have labels distinguishing malicious from non-malicious websites. The data contained 696 observations and 25 features of things like the VirusTotal Score, the age of the domain and the number of active certificates the domain has. Though this dataset still revolves around malicious URLs the content of the information is much different than that of the Kaggle dataset which makes it difficult to compare the two.

#### Data Processing

This data set had the same registration location data integrity issues as the last and two hand-made dictionaries were created to handle the formatting discrepancies as a result.

Other than dropping 10 features that provided repetitive data, the rest of the features were generically one-hot-encoded or normalized as is. The only exception was a group of 3 features that needed to be exploded before being one-hot-encoded because the features included lists of categorical variables for singular observations. Additionally we imputed missing values for the domain age variable by using the earliest certificate date in the *Cert* feature. The final cleaned dataset had 696 observations with 606 features.

Note: generally it is best that the number of observations *far* exceeds the number of features in a dataset, otherwise model results will not be robust. This stems from the fact that there will not be sufficient data to sort through the noise in the feature space. It is recommended that more data is collected before conducting anomaly detection to account for this problem. Otherwise the models may train to random noise rather than true patterns in the dataset.

## 5.2 Anomaly Detection

### 5.2.1 Background

Anomaly detection, also referred to as outlier detection, is an unsupervised data mining technique that identifies abnormal data within a set of data. These techniques have been demonstrated to work well on detecting network intrusions as summarized by M. Ahmed et al. [2]. In their work, they divide anomaly detection techniques into three primary categories: clustering models, statistical models, and classification models. Though each of these model types are used to find anomalies in data, they do so with different techniques and assumptions. Therefore, we expect that models across these categories will perform differently depending on the exact data set being considered. In our work, we use four specific algorithms split across these multiple categories to achieve robust outlier detection on the data. Specifically we used xStream and local outlier factor as cluster-based detectors, isolation forest as a tree-based detector, and one-class support vector machines as a classification detector.

### 5.2.2 Models Considered

#### **xStream**

In 2018 Manzoor et al. proposed a state-of-the-art clustering-based anomaly detector specifically designed for streaming data — a method they called xStream (Outlier Detection in Feature-Evolving Data Streams) [16]. Since its publication, NETCOM DSD has been working towards applying the method for anomaly detection. The algorithm is unique in its ability to efficiently detect outliers in contexts where data points and feature space may be evolving over time. The xStream model works by projecting data points to a lower dimensional space and performing density estimation to detect outliers at multiple scales. For example, consider Figure 6 which shows xStream scores produced on a toy data set. The model is able to identify different types of anomalies within the data. It correctly assigns high anomaly scores to the sparse and dense clusters of points to the top left and right of the image while assigning low scores to the sparse cluster in the middle of the data (the main benign cluster). It uses a smaller dimensional scale to score the cluster in the bottom left of the chart. The dense center of the cluster is assigned a low to medium score while the sparse points around the outside of the cluster are assigned a high anomaly score.

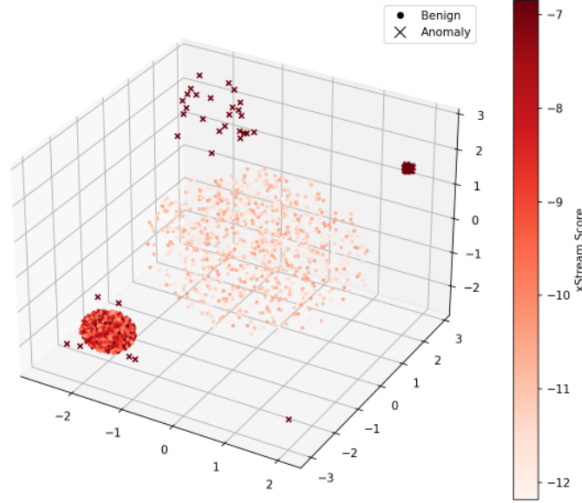


Figure 6: xStream is a density estimation based anomaly detector proposed by Manzoor et al. that can detect outliers at multiple scales and in contexts with evolving data point values and feature space [16].

### Local Outlier Factor

A more common cluster-based approach to outlier detection is local outlier factor (LOF) which was first introduced in 2000 by Breunig et al [4]. In this approach, they use the density of neighbors to score a point by its degree of isolation — a metric they define as the local outlier factor. Local outlier factor has long been used to detect anomalies in network intrusions, so we deemed it appropriate to use it as part of our approach [12].

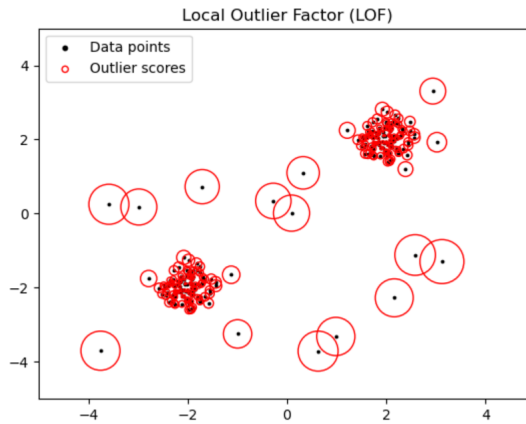


Figure 7: Local Outlier Factor is a cluster-based approach to outlier detection that uses local densities of points to score the isolation factor of each observation [19].

### Isolation Forest

Isolation forest is a tree-based anomaly detector that uses isolation rather than profiling normal observations in the data to find outliers [13]. This method uses an ensemble of isolation trees on the data set and provides an anomaly score inversely proportional to the average depth of an observation in these trees. These trees work by randomly partitioning the data. Normal points in the data take more partitions to be isolated while anomalous points take fewer. This means that anomaly risk is inversely proportional to the depth of the tree at which a point is isolated. The original authors report that isolation forest typically outperforms models like local outlier factor while being much more computationally efficient [13].

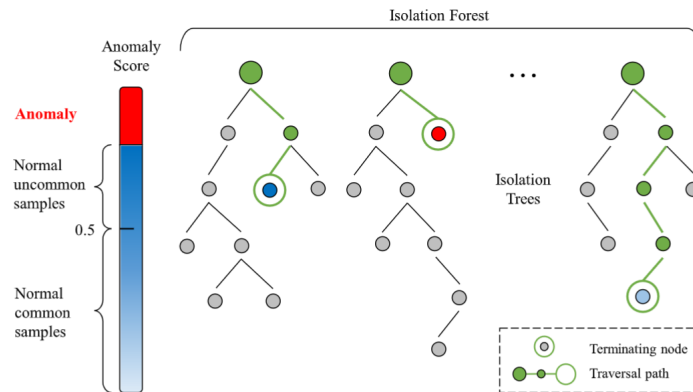


Figure 8: Isolation forest is a tree-based anomaly detector that finds anomalies by randomly partitioning the data an ensemble of isolation trees. Figure taken from [29].

### One-Class Support Vector Machine

One-Class Support Vector Machines (OCSVM) are a popular choice among the classification-based anomaly detection algorithms. The method was first proposed by Schölkopf et al. [23] in 2000 to extend the support vector algorithm to unsupervised learning. In this approach, the algorithm finds a hyper sphere that fits most of the data in a minimum volume. The samples that do not fall within the hyper sphere are considered outliers [9]. The distance from the point to the hyperplane define the anomaly score for that point (points further away will be assigned a higher score).

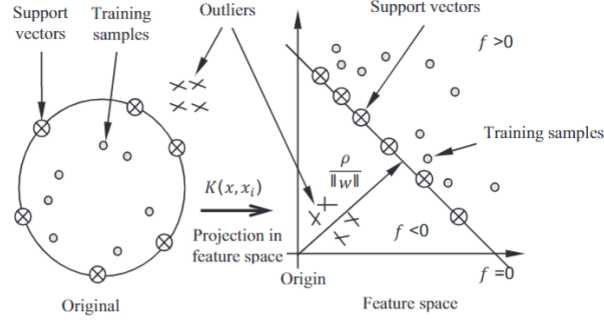


Figure 9: One-Class Support Vector Machines extend the support vector algorithm to unsupervised learning where they learn a hyper sphere that fits most of the data in a minimum volume [9].

### 5.2.3 Rank Aggregation

In order to generate a more robust anomaly ranking of the data, we propose combining the four separate models described above (xStream, LOF, Isolation Forest, and OCSVM) in an ensemble. Because each of these models produces scores across different domains and at different scales, we propose to standardize each model's score and then aggregate the results by either taking the mean or median of the standardized scores.

### 5.2.4 Anomaly Detection for Malicious Actors

Anomaly detection algorithms like the ones we propose using have demonstrated success in finding anomalous data in a wide field of disciplines to include network intrusion and fraud detection. However, anomalous data do not always correspond to bad actors like a denial of service attack or malicious internet domains. This is demonstrated visually in Figure 10 where on the left-hand side of the figure, anomalous data has a high coincidence rate with malicious data so you would expect anomaly detectors to be effective in separating out most of the malicious data. Conversely, the right-hand side of the figure demonstrates a low coincidence rate, so anomaly detection would be less effective in finding truly malicious traffic. For NETCOM's mission set, we expect that bad-faith actors would work to look as benign and non-anomalous as possible, so an essential task in this project is to evaluate the effectiveness of anomaly detection in specific NETCOM use-cases, namely network intrusion and malicious domains.

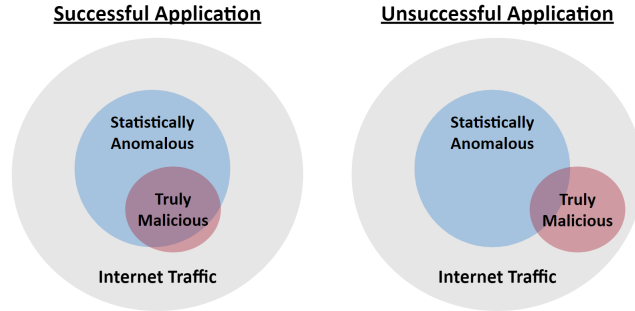


Figure 10: Anomalies do not always effectively capture malicious data, especially if you are looking for bad-faith actors. One goal of this research was to quantify the effectiveness of anomaly detection algorithms for NETCOM’s mission set[5].

### 5.3 Post-Hoc Explanation

Machine learning models have become increasingly complex and are much harder for experts to interpret and explain. Understanding how the model makes classifications, and being able to trust that they are accurate and will perform equally on real-world data is important if it will be used for decision making [22]. Artificial Intelligence that is easily understood and explained by a human domain expert is known as Explainable AI (XAI), and when this explanation is applied after the model has already made its prediction it is known as Post-Hoc Explanation.

#### 5.3.1 General Approach

Current approaches to XAI can be broken down into two categories: local and global explanations. Many XAI methods explain the model’s classification through the use of a simplified binary feature mapping in order to see which features contributed to the output [14]. Global explanation describes how much each feature contributes to the model’s overall decision making and local explanations describe how each feature contributes to the classification of any individual point [1].

Lindberg et al. introduced SHAP (SHapley Additive exPlanation) in 2017 as an entirely new class of additive feature importance methods. SHAP unifies six different explanatory methods and is proven to provide a unique to this class of methods while adhering to these three properties[14]:

1. **Local Accuracy** - The generated model’s output has to match the output of the original model for the corresponding input values.
2. **Missingness** - Any missing feature will have a value of zero.
3. **Consistency** - If a model changes such that the feature’s contribution changes, then the feature’s value will change accordingly.

SHAP values are difficult to calculate exactly, but they’re easy to approximate; several model specific algorithms have been created in order to decrease computation time [14] [15].

### 5.3.2 Our Approach

We use SHAP to generate explanations in this problem because it provides a proven, unique solution to the class of problem we're trying to solve. In addition to that, there is a polynomial-time algorithm for producing SHAP explanations from tree-based models [15] and SHAP allows for a global comparison of the results instead of a purely local interpretation. The fast computation of the explanations is essential for NETCOM's applications where the number of observations might increase over time. We are able to take advantage of the special properties listed above to provide NETCOM with a dynamic and interactive web interface for anomaly investigation. And the major advantage of SHAP is that it benchmarks to the average score across the data, allowing the SHAP feature values to be interpreted relative to the rest of the data rather than just in a small neighborhood.

We generate the SHAP values for each model - observation pair through a four step process shown in Figure 11. First, we generate anomaly scores with a model like xStream. Second, we fit an XGBoost regression model to the input feature values  $X$  and the anomaly scores  $y$ . Third, we run the tree-based SHAP algorithm on the XGBoost model to create the SHAP explanation model. And fourth, we take each observation and pass it through the SHAP explanation model to generate the feature scores for each observation.

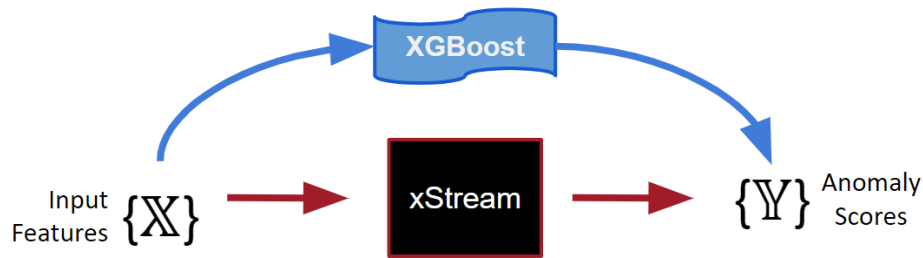


Figure 11: In order to generate explanations for the anomaly scores, we fit a XGBoost regressor to find correlations between the input features and anomaly scores in the data set. We then produce a SHAP explainer for the model and run each observation through the explainer to generate the individual feature contributions of each observation.



## 5.4 AWS Pipeline

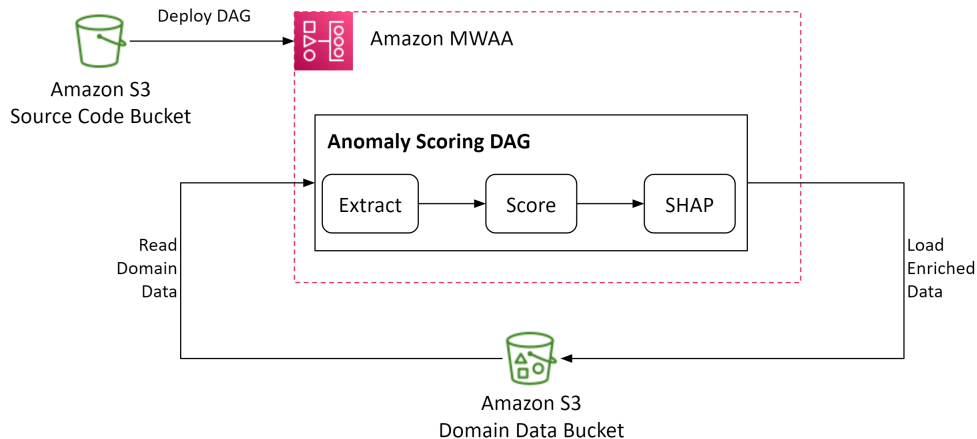


Figure 12: The anomaly scoring and XAI generation occurs in a cloud pipeline hosted on Amazon Managed Workflows for Apache Airflow.

In order to seamlessly incorporate our product with NETCOM DSC-PIT, our back-end pipeline is deployed to Amazon Managed Workflows for Apache Airflow (MWAA) as shown in Figure 12. This pipeline is intended to run inline with the current NETCOM DSC-PIT MWAA workflow shown in Figure 5. The source code for the process is stored in an AWS S3 bucket and scheduled with a directed acyclic graph (DAG) that directs the execution of the pipeline for MWAA. The domain data generated by NETCOM’s current workflow is stored in another AWS S3 bucket. When the DAG is executed by MWAA, the first script extracts relevant features from the data and pre-processes for model fitting. Next, the data is scored by four different anomaly detection models (xStream, LOF, Isolation Forest, and OCSVM). Finally, SHAP values are calculated for each observation-model pair. These results are saved back to the original data S3 bucket to be fed into downstream applications like our proposed user interface later.

## 5.5 User Interface

In order to demonstrate the combination of XAI and anomaly detection algorithms we developed a fully functioning web application prototype. As discussed in the business value section, integrating this prototype into their workflow can both expedite their analytic processes and offer new features not previously available. Ready to ‘plug-and-play’, once connected to DSD’s AWS backend on Gov-Cloud the prototype will be fully operational for a beta release.

After considering different options, ultimately we choose to develop the app in R-Shiny, a package and tool suite designed by RStudio to make data-focused web apps using the R programming language. R-Shiny fit several key criteria. First, the language is designed to be easily extensible providing a simple framework for rapid development without limiting complex reactivity between user inputs and the display. Second, not only does R have a significant following among Army data scientists (which will make maintaining the app more feasible), there are a number of options available for deploying an R-Shiny on Army internal platforms. Specifically, NETCOM DSC-PIT

has access to COEUS, a data science platform under development at the US Army Artificial Intelligence Integration Center, and Gov Cloud, which simplifies the process of deploying applications to the cloud. The major drawback of R-Shiny is apparent difficulties with scaling to multiple users and rapid changes to data. It is unlikely that NETCOM will deal with these issues anytime soon, but if scaling is desired the app will likely have to be refactored into javascript.



Figure 13: A screenshot from the user interface prototype. Currently deployed at [https://jtmccorm.shinyapps.io/NETCOM\\_AD\\_Prototype/](https://jtmccorm.shinyapps.io/NETCOM_AD_Prototype/)

The prototype provides several specific features that will improve the existing workflow and contribute new capabilities as well:

1. **Rank anomalies by Ensemble Scoring.** As discussed in the anomaly detection review, we propose ensembling several algorithms together by taking a standardized mean. The new domains visited by soldiers are displayed according to this aggregated score, immediately prioritizing action.
2. **Explain individual model outputs using SHAP values.** Users can click on the entry for a specific domain and produce the SHAP plots associated with that observation. Essentially, this will show the features most important in leading the model to score the observation a certain way.
3. **Identify observations to investigate or ignore.** Though the ensemble anomaly detection scores will prioritize how domains are displayed, the decision to investigate or ignore an observation is left to the analyst using the application. These annotations will be added to the dataset and made available for download.

4. **Explore relationships between variables and scoring algorithms.** In addition to the ranking specific anomalies and displaying their SHAP scores, the web application supports an interactive exploratory data analysis plot. The user can choose what variables to plot against one another. This allows the analyst to visually identify potentially malicious traffic and validate the models by comparing them against one another.
5. **Dynamically mute features preventing skew from irrelevant features.** Taking advantage of the additive property of SHAP values, we can subtract the effect of a specific feature from the model outputs. This may be desirable if certain features have been identified as irrelevant by a subject matter expert.
6. **Download the dataset with annotations and updated scores.** After modifying models and annotating specific results, the data can be downloaded recording these new features and notes. This allows end users to perform specialized analyses after the fact prior to beginning investigations.
7. **Provide tiered access based on credentials.** Due to the flexibility and control being offered by the app, we felt it may be necessary to restrict certain features to only experienced analyst or data scientists in order to protect against confirmation bias.
8. **Control which anomaly scoring methods are included within the ensemble.** If the analyst identifies negative trends in the performance of one model, it may be advantageous to remove that model from the ensemble. However, given that we are dealing with unlabeled data we felt this could be too drastic of an action to take without careful deliberation by a trained data scientist. As such we restricted this tool to “developer mode”.
9. **Fine tune feature weights of individual models.** If certain features have been identified as distorting results or not being weighted high enough by certain models, the “feature weight” of covariates can be set anywhere between 0 and 1.5. This allows deliberate tweaking of models in order to tune them to specific types of attacks or malicious websites, a capability not currently in the hands of NETCOM DSD.

Many of the additional features of the app rely on the additive property of SHAP values. Specifically, because all the SHAP values for an observation add up to its final score, we can dynamically remove features from a model by subtracting the features SHAP value from the overall score. This creates powerful interactivity since it effectively allows the user to simulate re-running the model without spending computational resources or time to have this done. In addition, to simply removing a SHAP value it “reweighted” by multiplying it by certain factors, allowing further control over the model outputs. This rapid modification of individual models and how they fit together is what we call dynamic, interactive ensembling.

## 6 Data Analysis / Model Performance Evaluation

In this section we will discuss the results of our work with each of the data sets. We will present tentative conclusions and discuss obstacles in the analysis that should be addressed in future work.

## 6.1 Network Intrusion Data

In this dataset we focused on the following model's performance: xStream, Isolation Forest and Local Outlier Factor. This figure simply bins observations by anomaly score, then for further granularity each observation is given a color according to it's true label in the data.

The tall, isolated blue bar on the left-hand side of the graph without any red bars mixed in indicates that xStream was able to distinguish between many of the benign instances from malicious ones. This is why you can see the model reaches 100% recall before looking at every observation in the bottom recall curve on the right-hand side of the figure below.

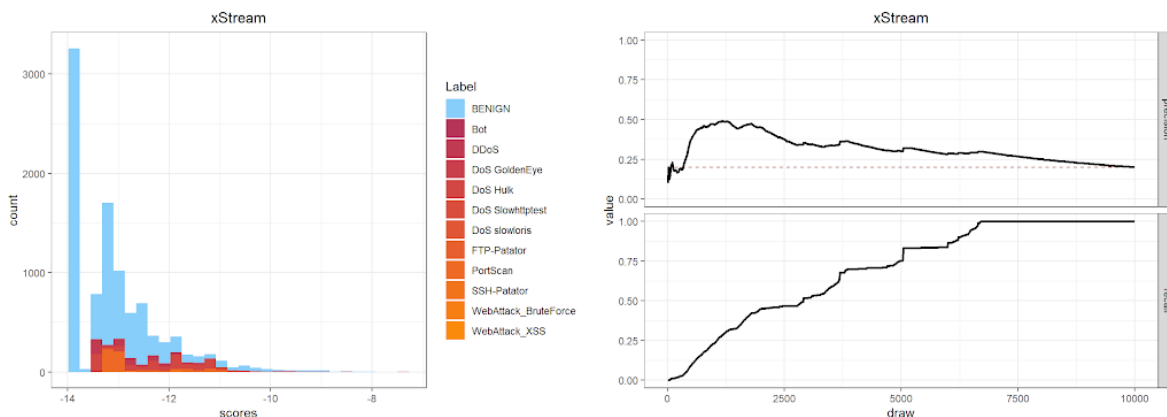


Figure 14: xStream performs well on this dataset. It separates out a large chunk of the benign data, and achieves 100% recall earlier than any of the models as a result.

In contrast, Local Outlier Factor did not isolate a large proportion of benign observations from malicious ones, which results in a worse recall performance than xStream. Now shift focus to the precision curve in the upper-right of the figure below. The early and high spike in precision, allows Local Outlier Factor to initially outperform xStream's ranking, meaning it ranks more truly malicious observations as most anomalous in the top 500 ranked observations than xStream did.

Practically, this means that using LOF the systems analyst would see more malicious observations initially, but using xStream the analyst would see ALL of the malicious observations sooner. This demonstrates that each anomaly detection model has it's own strengths and by intelligently combining ranks from multiple models you may be able to improve both precision and recall performance over all individual models. We will discuss test this rank aggregation later in the paper.

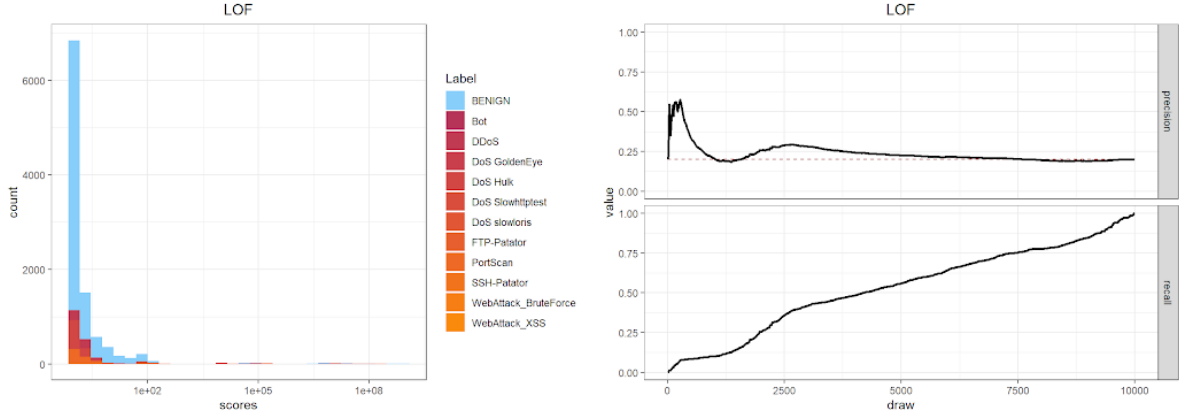


Figure 15: Local Outlier Factor has high precision early on, then it's performance in both precision and recall are similar to randomly guessing.

Unlike the previous two models, Isolation Forests performance on this dataset was not especially impressive. It did not perform much better than random guessing at any point in the top-k analysis, which is displayed by the lack-luster precision and recall graphs in the figure below.

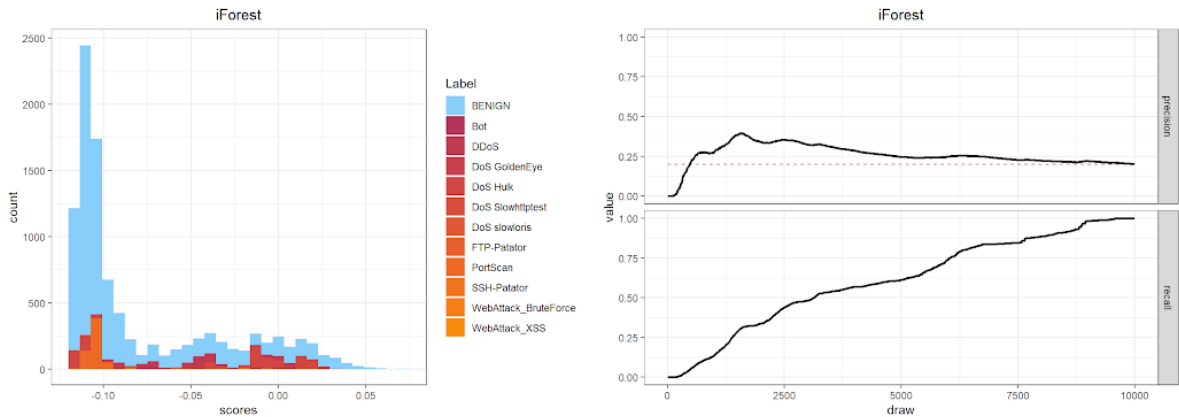


Figure 16: Isolation Forest did not outperform either xStream or Local Outlier Factor in precision or recall.

Another way to analyze performance is to look at each model by attack-type rather than looking at overall performance on ALL malicious data grouped together. In the figure below you will see three tables that represent the first rank where each model ranked a malicious observation for each type of attack. This further supports our previous analysis that xStream outperforms the other models, because you will see most attacks appear for the first time at lower ranks (sooner) using xStream than for other models. However, we can see in certain cases other models perform better on some attacks types. For example, LOF does much better at identifying Botnet attacks than either xStream or Isolation forest does. Whereas, Isolation forest actually does better at identifying

Cross-site scripting attacks than the other two models.

xStream		iForest		LOF	
label	first_rank	label	first_rank	label	first_rank
DoS GoldenEye	1	BENIGN	1	BENIGN	1
BENIGN	2	DoS Hulk	138	PortScan	8
DoS Slowhttptest	12	DDoS	293	DoS Hulk	10
DoS Hulk	33	DoS slowloris	305	DoS Slowhttptest	424
PortScan	136	DoS Slowhttptest	412	Bot	501
SSH-Patator	224	WebAttack_BruteForce	574	DDoS	631
DDoS	243	DoS GoldenEye	754	DoS GoldenEye	833
DoS slowloris	327	WebAttack_XSS	1224	SSH-Patator	940
FTP-Patator	341	SSH-Patator	1604	FTP-Patator	1167
WebAttack_XSS	3071	PortScan	2133	DoS slowloris	1567
WebAttack_BruteForce	3121	FTP-Patator	2730	WebAttack_XSS	3333
Bot	4212	Bot	3068	WebAttack_BruteForce	4727

Figure 17: This figure displays the first rank at which each model found each attack type. Again we see that xStream generally outperformed the other two models. However, in select attack-types the other models performed better.

Now, we will look at how robust each model’s performance is to changes in hyper-parameter settings. This is important because real-world data will not be labeled, thus it is difficult to tune hyper-parameters well. Part of a model’s attractiveness is thus determined by how well it can perform regardless of its hyper-parameter settings.

Below we see 12 scatter plots. The top, left-most plot shows the three models performances on all the attack types, the remaining plots show performance by attack. The red-dashed line denotes the baseline occurrence rate of each attack, thus models falling below the line are essentially guessing and not performing well. Each model is displayed using a different color in the plots, with each point being one instance of the model trained with a different set of hyper-parameters. The x-axis denotes Average Precision (across various subsets of top-ranked observances), and the y-axis is the Average Area Under the Receiver Operator Characteristic curve. The best performing models will have high values for each performance metric and thus be close to the upper-right hand corner of the plot.

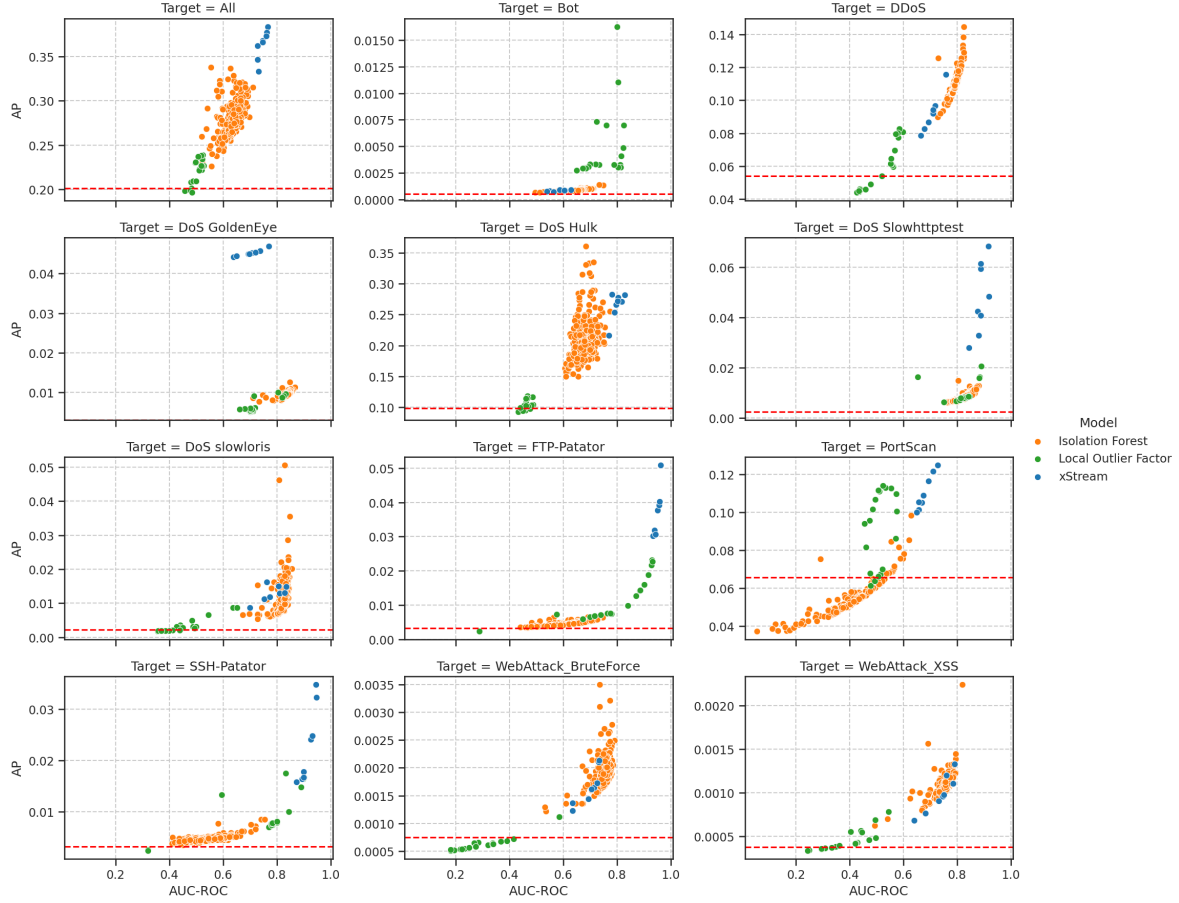


Figure 18: xStream generally outperforms the other models, however in Botnet, Distributed Denial of Service and Cross-Site Scripting attacks the other two models outperform xStream.

From the figure above, we can see that xStream generally outperforms the other models on most attacks, and is more robust to hyper-parameter settings as well. Thus we conclude that on this dataset xStream is the better overall instrument from detecting Malicious Web-Intrusion Attacks.

## 6.2 Publicly Available Malicious Domain Data

Remembering that this dataset did not have very rich features, we can see the impact of that in the results. The anomaly detection models were not as robust to changes in the hyper-parameter space in this use case. We can see from the figure below, that over half of the model performed no better than randomly guessing. The exception is Local Outlier Factor which performs above the baseline in every instance. These less stellar results may indicate that the feature space did not provide enough information to the models to make them robust, or potentially that anomaly detection may not be the best suited approach to detecting malicious URLs. Further investigation is needed to come to a definite conclusion.

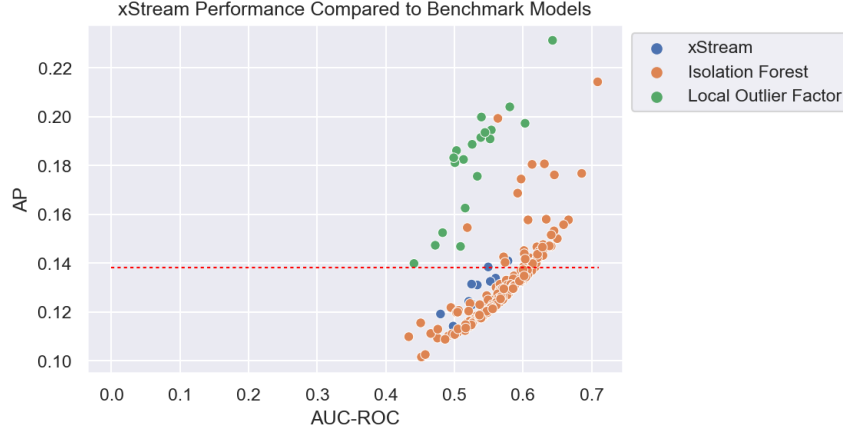


Figure 19: On this dataset only Local Outlier Factor seemed to perform well. xStream and Isolation Forest were very sensitive to hyper-parameter tuning, and did not outperform the baseline in most instances.

### 6.3 NETCOM Malicious Domain Data

This enriched dataset from NETCOM was unlabeled. This means that the type of analysis we conducted on the previous two datasets isn’t directly possible. However, we did not want to simply hand an untested product to NETCOM so we came up with a proxy label.

The proxy label used VirusTotal score as an estimate to the maliciousness of the website. VirusTotal is a website that “analyze[s] suspicious files, domains, IPs and URLs to detect malware and other breaches, [and] automatically share them with the security community” [28]. If NETCOM’s mission is that detect similar malicious URLs on the Army’s network then our proxy score will reflect the unlabeled anomaly detection results relatively well. However, if NETCOM is seeking to detect other forms of malicious URLs this proxy label will NOT provide much insight into the models’ performances. One last point to keep in mind is that in order for us to use the VirusTotal feature as the label, we had to remove it from the dataset. This is taking away a feature with a lot of information, thus when it’s added back into the feature space the models’ performances will likely improve. All of this to say, that the results to follow are not robust conclusions, but simply indications of how the models may perform in NETCOM’s real pipeline.

With all the considerations listed above in mind, we can dig into the results displayed in the figure below. Both Local Outlier Factor, and xStream seem relatively robust to changes in hyper-parameter settings and outperform the baseline. Isolation forest in some instances outperforms the two other models, however it is also extremely sensitive to hyper-parameter tuning. When looking at these results, it’s difficult to determine which model outperforms the others. Instead our team decided to ensemble the scores and add an additional model to try and improve the anomaly detection results.



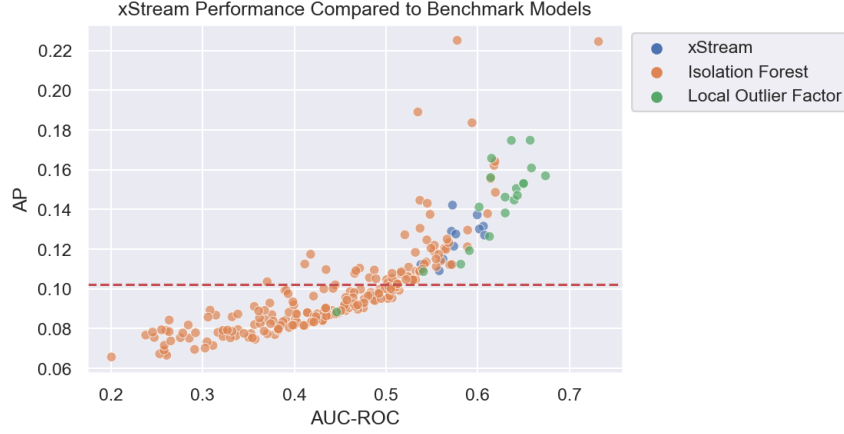


Figure 20: Model Performance on the NETCOM data with VT Score as a proxy label. We see significant amount of variance based on hyper-parameters, suggesting the model's may be more sensitive to tuning with this data set and task.



Figure 21: This figure displays results for the NETCOM enriched dataset, using Virus Total Score as a proxy data label for maliciousness. In the chart on the left neither OCSVM's nor the Mean Rank Aggregation's performance dominates the other, but collectively they have the two best performing rankings in this limited use case. By interacting with the contribution of each model a data scientist can improve the results of the aggregated ranking (shown on the right). In this case we muted the contributions of xStream and Isolation Forest to improve the Mean Rank Aggregation's performance.

As you can see from the figure above, when all models are included in the rank aggregation scores both median and mean aggregation scores perform on par with the best performing model (One Class Support Vector Machine). However, when the worst performing models are muted, the mean aggregation scores dominate the rest of the individual model rankings. Seeing these improvements we included mean rank aggregation in our final product the NETCOM, with the ability to mute models to improve the performance. This functionality was discussed in more detail

earlier in the user interface section of the report.

## 7 Risk Framework

As stated by the Department of Defense’s (DoD) 2018 Cyber Strategy, “American prosperity, liberty, and security depend upon open and reliable access to information” [17]. Facilitating this requires “secure and resilient networks and systems” [17] that are continuously assessed and monitored against strategic objectives, and those under NETCOM’s purview are no exception. As the DoD continues to “leverage automation and data analysis to improve effectiveness,” NETCOM must obtain and maintain high levels of security and resiliency across all of its critical services and assets in order to “[ensure] freedom of action in cyberspace while denying the same to our adversaries” [6, 17]. To do this, NETCOM must manage cyber risk in a systematic way across the Command. The following risk guidelines will assist NETCOM in doing just that.

There are many cyber risk management frameworks in existence, a few of which are currently being used in the Army and DoD. We chose to leverage the OCTAVE FORTE (Operationally Critical Threat, Asset, and Vulnerability Evaluation For the Enterprise) process, a process model developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (CMU) “that helps executives and other decision makers understand and prioritize the complex risks affecting their organization” [25]. OCTAVE FORTE is highly informative yet flexible and as a result, facilitates the simultaneous use of other risk management methodologies. Many of its tenets are also based on standards already used within the Army and DoD, such as National Institute of Standards and Technology (NIST) standards including NIST SP 800-39 (Managing Information Security Risk: Organization, Mission, and Information System View) and NIST SP 800-37 (Risk Management Framework for Information Systems and Organizations: A System Life Cycle Approach for Security and Privacy), otherwise known as NIST RMF [25].

The following guidelines are laid out in order of the steps of OCTAVE FORTE and reference cyber risk management strategies and projects already underway in the Army and DoD, namely Project Sentinel and the DoD’s 2018 Cyber Strategy. In particular, this section highlights NETCOM DSC-PIT’s role in managing risk within NETCOM, referencing the Nullfox workflow and our team’s added anomaly detection capabilities as a use case throughout.

Through a systematic, enterprise-wide approach to risk management, NETCOM can effectively achieve its strategic objectives while managing uncertainty, and NETCOM DSC-PIT can understand exactly how their front line efforts translate into Command-wide value.

### 7.1 Establish Risk Governance

The first step toward a synchronized risk management effort is establishing a risk governance structure. A risk governance structure spells out how individuals within an organization communicate with each other to effectively manage risk. These structures are typically split into three tiers as exemplified in Figure 22 below from NIST SP 800-39 [18]. Having effective communication between and within each of these governance tiers facilitates strategic alignment and accountability. It also facilitates a feedback loop that gives each organizational level a voice, allowing for risk to be addressed in a multifaceted way [25]. The following tier descriptions explain a recommended tiered

governance structure for NETCOM, highlighting where NETCOM DSC-PIT fits in.

### **Tier 1**

Creating and promoting a risk governance structure must come from the top. Leaders and executives at the organizational level are most in tune with strategic objectives that drive the Command's mission at large and thus, understand how these objectives can be seized or threatened. As such, they are best positioned to handle *strategic risks*. To do so, this tier must establish strategic direction and approve cyber risk policies created at a more operational level (typically among Tier 2 cyber risk managers) [25]. Further, this tier must establish a Command-level risk appetite statement that acknowledges the Army-wide risk threshold defined through Project Sentinel initiatives[11]. Anyone currently closest to overseeing, developing, and iterating on NETCOM's mission statement and defining its objectives is the best poised to take on a risk leadership role in this tier. Likely, these will be a set of high-ranking leaders that are assigned an additional role as risk leaders. These leaders should form a "risk board" that collectively makes risk-based decisions to achieve strategic objectives, and it is this risk board that will assign the risk managers that operate in the second tier.

### **Tier 2**

Risk managers within this tier represent key communicators in the risk governance structure. Not only are they responsible for communicating feedback from Tier 3 risk owners up to the risk board, but they are also responsible for communicating higher-level, policy-driven changes from the risk board downward. Risk managers in this tier should have an understanding of the critical assets and services that drive NETCOM's mission, and the higher level security requirements of these assets/services, such as their confidentiality, integrity, and availability requirements (explained further in Section 7.4). Further, they must create and/or disseminate cyber risk policies and procedures through the Command and facilitate a feedback loop so they can be monitored and improved upon. Those best positioned to take on an additional risk management responsibility in this tier are high-ranking leaders within each of the "teams-of-teams" created to support NETCOM's mission [6]. Related to our application, this means that at least one cyber risk manager should represent NETCOM DSD on the risk committee that sits in this tier.

### **Tier 3**

Tier 3 is typically composed of subcommittees of high-performing managers that are closest to an organization's daily operations [25]. As such, this is the tier in which all data science centers, including NETCOM DSC-PIT, should have risk owner representation. As the gatekeepers to academic and industry partnerships, the DSCs are best poised to understand the *tactical risks* - risks associated with the every day deployment and operation of services and assets - that emanate from acquiring, using, and maintaining research products from third parties [20]. As a designated risk owner in this tier, a NETCOM-DSC PIT investigator is responsible for translating cyber risk policies and procedures into practical protection mechanisms at the asset/service level. This would include processes such as implementing applicable controls and monitoring their effectiveness over time. Further, as new risks present themselves and current risks change, pertinent information will likely need to be reported up the governance structure to assess any possible effects on strategic objectives [25]. As a front-line manager of these cyber risks, an investigator must also communicate the effectiveness of any metrics used to prioritize cyber risks and assess controls [25].

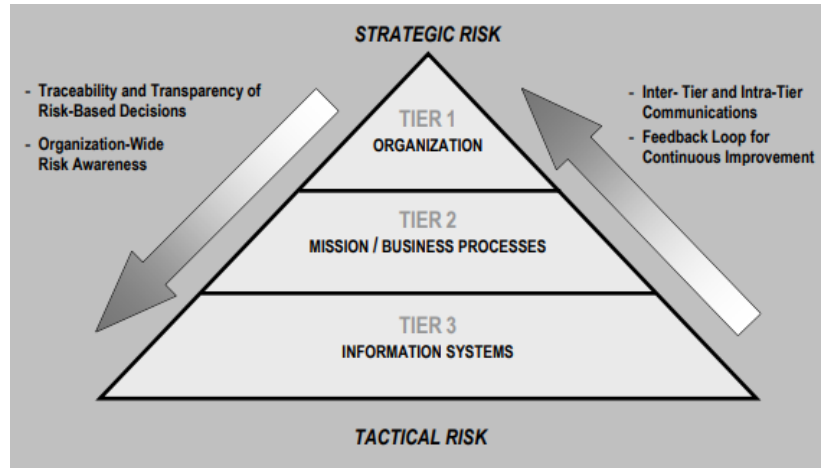
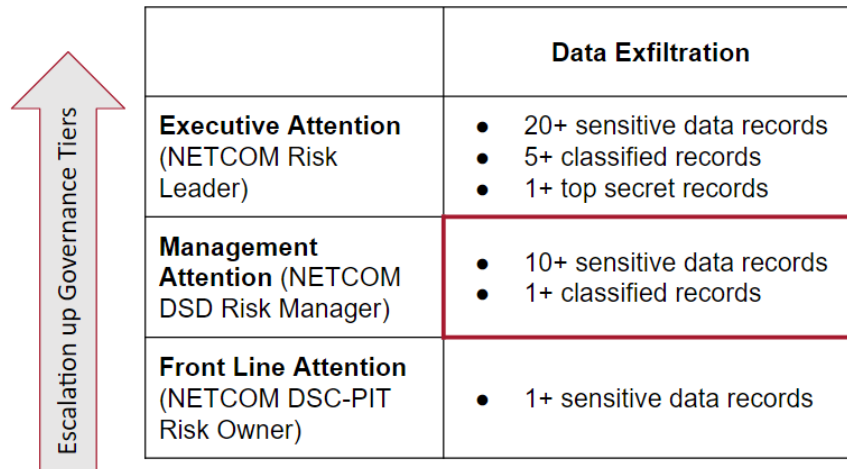


Figure 22: Multitiered Organization-Wide Risk Management Governance Structure [18]

## 7.2 Establish Risk Appetite

One of the goals of Project Sentinel is to “institute a risk threshold” that is consistently applied across the Army: “with the adoption of a risk threshold, decisions for where to spend resources become apparent, necessary, and more precise” [11]. Having this threshold in place will not only help with efforts to prioritize risk, but also help dictate how to respond to them, leading to effective allocations of time and other resources. This kind of threshold forms the foundation of a risk appetite statement.

As defined by NIST, a risk appetite statement reflects “the types and amount of risk, on a broad level, an organization is willing to accept in its pursuit of value” [3]. Typically, risk appetite statements are broken down into escalation levels and impact areas, general categorizations of pain points that can impact mission attainment. Each row within an impact area dictates the conditions required to escalate particular risks up the governance structure. These quantitative values represent *risk tolerances* for each tier, the amount of risk that can be tolerated and addressed at that level based on that tier’s capabilities. Figure 7.2 below gives a simplified example of a risk appetite statement, focusing on Data Exfiltration as an impact area.



	Data Exfiltration
<b>Executive Attention</b> (NETCOM Risk Leader)	<ul style="list-style-type: none"> <li>• 20+ sensitive data records</li> <li>• 5+ classified records</li> <li>• 1+ top secret records</li> </ul>
<b>Management Attention</b> (NETCOM DSD Risk Manager)	<ul style="list-style-type: none"> <li>• 10+ sensitive data records</li> <li>• 1+ classified records</li> </ul>
<b>Front Line Attention</b> (NETCOM DSC-PIT Risk Owner)	<ul style="list-style-type: none"> <li>• 1+ sensitive data records</li> </ul>

Figure 23: A simplified Risk Appetite Statement with one impact area, Data Exfiltration, the three Risk Governance escalation tiers, and escalation requirements

### 7.2.1 Risk Appetite Example Use Case

Consider a case where the anomaly detection tool identifies anomalous activity whereby 14 different DODIN IP addresses have accessed a particular URL to make purchases. Combining VirusTotal score information and additional threat intelligence, a NETCOM DSC-PIT investigator determines that this URL is malicious, likely hosted by a Nation State Actor (NSA), and is extracting the payment information of multiple soldiers. What the risk appetite statement above dictates is that this risk needs to be communicated to the risk manager level (i.e. to a NETCOM DSD risk manager) to be addressed most effectively since more than 10, but less than 20, sensitive data records have been exfiltrated.

## 7.3 Scope Critical Assets and Services

Understanding which assets and services are most critical to achieving NETCOM’s mission is crucial for prioritizing cyber risks. Importantly, understanding the web of assets that support services helps to identify where the “crown jewels” lie [7]. These represent any asset that, if gravely impacted, can disrupt critical services, even if they do not always seem as individually important on a daily basis. Having a systematic way to identify, catalog, and document assets gives everyone in the Command a heightened understanding of asset types, owners, dependencies, and security requirements.

Specific to the anomaly detection tool, keep in mind that in addition to being a control, it is also an information and technology asset to the Command. As such, it should be subject to the aforementioned cataloging, monitoring, and review to ensure it is utilized for appropriate purposes and remains effective in its application.

## 7.4 Identify Resilience Requirements for Assets

Critical assets and services are deemed as such for a reason, often meaning that they must be heavily guarded, unable to or difficult to be changed, constantly available, or a combination of these. These characteristics represent aspects of confidentiality, integrity, and availability, “the fundamental resilience requirements for information security” [25]. Defined more specifically by ISC<sup>2</sup>:

- **Confidentiality** involves preserving authorized restrictions on information access and disclosure, including means for protecting personal privacy and proprietary information.
- **Integrity** involves guarding against improper information modification or destruction and includes ensuring information non-repudiation and authenticity
- **Availability** involves ensuring timely and reliable access to and use of information by authorized users [10]

Determining which of these requirements must be held and to what degree helps to define how resilient a particular asset/service is. In turn, this helps to inform an investigator’s risk analysis by informing risk impact ratings. Typically, assets/services that must be highly resilient will result in a larger impact if compromised by a realized risk.

Additionally, since the anomaly detection tool is an asset in and of itself, it must have its own resilience requirements determined. From a confidentiality perspective, ensuring the data fed into the tool is not leaked is crucial. Relatedly, although the tool involves the use of black-box algorithms, any other information about how the tool works should also be kept confidential so that adversaries are less likely to evade its capabilities. From an integrity perspective, changing even the smallest elements of the tool, such as flipping a switch to exclude a model from analysis, can greatly impact which anomalies are chosen for investigation. Finally, if the tool is unavailable, NETCOM DSC-PIT must discern what may transpire as a result and consider whether any back up or alternative capabilities are available.

In addition to drawing these resilience requirements against the CIA triad, in tools leveraging machine learning algorithms, resilience must also be assessed from a trust perspective. Without clear verifiability, it must be recognized that investigators may lose trust in the system or that bias may be unknowingly introduced [26].

## 7.5 Measure Current Capabilities

In order to respond to risks most effectively, NETCOM DSC-PIT must have a working knowledge of the controls available to them. As defined in OCTAVE FORTE, controls are “the methods, policies, and procedures that the organization uses to respond to risk and meet its strategic objectives” [25]. These controls tend to be “technological, physical, or administrative” in nature [25]. The anomaly detection tool, for instance, is a form of technological control, and is detective rather than preventative. As such, it is an ideal tool for creating a *defense in depth* strategy, a strategy where multiple layers of controls are used to address risks [25]. For example, in our use case of analysing URL data, an investigator would use the anomaly detection tool to find anomalous, potentially malicious activity, and use a preventative control, such as an intrusion prevention system (IPS) to block access to that particular URL.

Creating a prioritized list of controls is an essential first step to take in this effort. Fortunately, this aligns with current projects, since a primary goal of Project Sentinel is to devise a subset of NIST RMF controls that is most applicable to the kinds of assets and services that NETCOM uses and provides [11]. However, NETCOM DSC-PIT should still remain vigilant in the assessment of this control set, understanding clearly the protections they provide for critical assets and services, and the protections they don't (i.e., assessing where gaps in protection lie).

## 7.6 Identify Risks, Threats, and Vulnerabilities to Assets

OCTAVE FORTE breaks down risk considerations into three steps: Identifying Areas of Concern, Identifying Threat Scenarios, and forming Impact Statements [25]. Identifying Areas of Concern involves understanding what could negatively affect the operation of the Command's critical assets and services. Typically, this understanding is initially provided by asset/service owners and enhanced by risk owners when constructing threat scenarios. Thus, when relating this to the governance structure, this primarily would involve cyber risk managers within DSD, roping in risk owners within the DSCs to give further insight. Threat scenarios dive more deeply into these areas of concern, identifying threats or families of threats and the vulnerabilities they can exploit in critical assets/services. This hedges on current threat intelligence and knowledge of the ways different asset types can be affected. When moving to assess the impact of realized threat scenarios, the previously defined resilience requirements are crucial to have on hand. Understanding how each of the CIA Triad (confidentiality, integrity, and availability) components is affected by a threat scenario and reconciling this with resilience requirements forms impact statements. For example, if a threat scenario were unlikely to result in a confidentiality breach for an asset/service that has high confidentiality requirements, the impact would be relatively low, and as such, that particular risk may not be of high priority for that asset/service.

## 7.7 Analyze Risks Against Capabilities

Once controls are prioritized, asset resilience requirements are determined, and potential risks to those assets are identified, NETCOM DSC-PIT must decide which and how many of these controls are appropriate to address each risk and analyze how well they would do so [25].

In this step, the unique cyber risk management characteristics of the anomaly detection tool come to light. First and foremost, as discussed in Section 7.5 the tool acts as a technological monitoring control. For example, in the malicious URLs use case, the tool monitors by summarizing key attributes around suspicious outward DNS requests on a weekly basis. Where this tool arguably brings more value, though not as clearly in the form of a control, is in the way it surfaces risk indicators (these are discussed further in Section 7.8).

When interacting with the tool, an investigator is able to use domain expertise to determine which of these indicators are important to consider, subsequently refer to each model's anomaly score and/or the combined ensemble score, and iterate on this process to determine which anomalies should be investigated. However, this information alone does not provide enough information to form a risk. These data points and features indicate that there *may* be a malicious threat actor (internal or external to the Command) at large who *may* be attempting to exploit a vulnerability in a particular asset, but they do not provide any certainty in these regards. As such,

more than anything, the tool acts as a prioritization mechanism for anomalies that require further investigation.

For this slate of anomalies, investigators can then leverage threat, vulnerability and previously determined exposure information (including the CIA requirements and other related resilience requirements for assets and services of concern) to rate the impact and likelihood of the risk if it were to be realized. At present, the DoD Cyber Strategy leverages a Low, Moderate, High scale for qualifying impact and likelihood [17]. With both of these ratings, an inherent risk score - or, a risk score that rates a risk before any protections are applied to respond to it - can be formed. Figure 24 visually summarizes this risk scoring process.

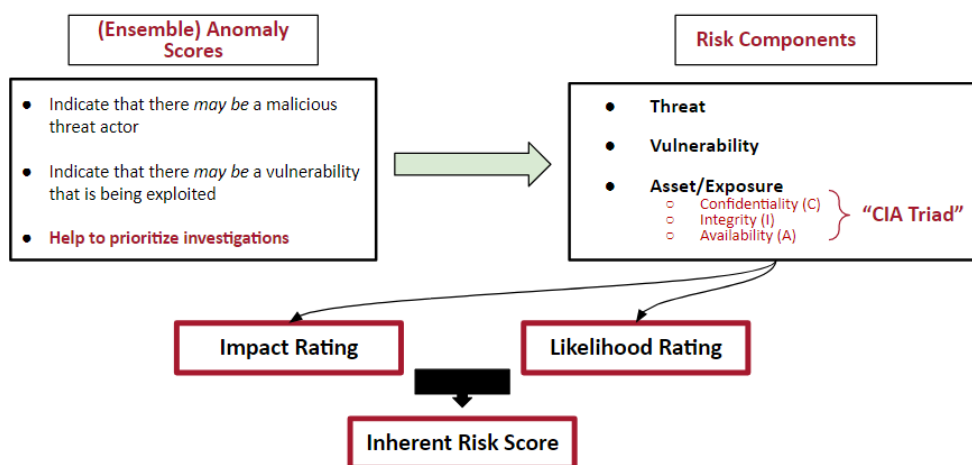


Figure 24: Moving from an anomaly score to a risk score

Once an inherent risk score is formulated, the prioritized set of controls and their capabilities formed in Section 7.5 should be consulted to figure out if there is any way to reduce the level of residual risk, the risk that remains once protections are applied. Utilizing a heat map to display these rankings helps to indicate senses of urgency with addressing risk; however, if a risk is in a green area, this does not mean it can be ignored, and if a risk is in a red area, it does not automatically warrant directing an excessive amount of resources (beyond what NETCOM’s risk appetite dictates) to address it.

The example in Figure 25 below shows how this process is carried out. Consider a risk whose inherent risk score is Moderate-High given its Moderate impact rating and High likelihood rating. Given the characteristics of the risk, a set of controls is picked to address it. An assessment of the applicability of those controls their current health reveals that they would be moderately effective at reducing this risk. As such, the residual risk score is Moderate, a reduction from the inherent risk score of Moderate-High.

Keeping track of all of this tactical risk information in a DSC-level risk register will equip NETCOM DSC-PIT with a standardized way to assess and prioritize risk. NETCOM should also consider utilizing a risk register to manage strategic risks. Typically, risk registers managing front line risk are more detailed and contain metrics that can “roll up” into “higher-level” risk registers.



This helps with communication up the governance structure, detailing how risks identified at the front line ultimately end up impacting wider scale strategic objectives.

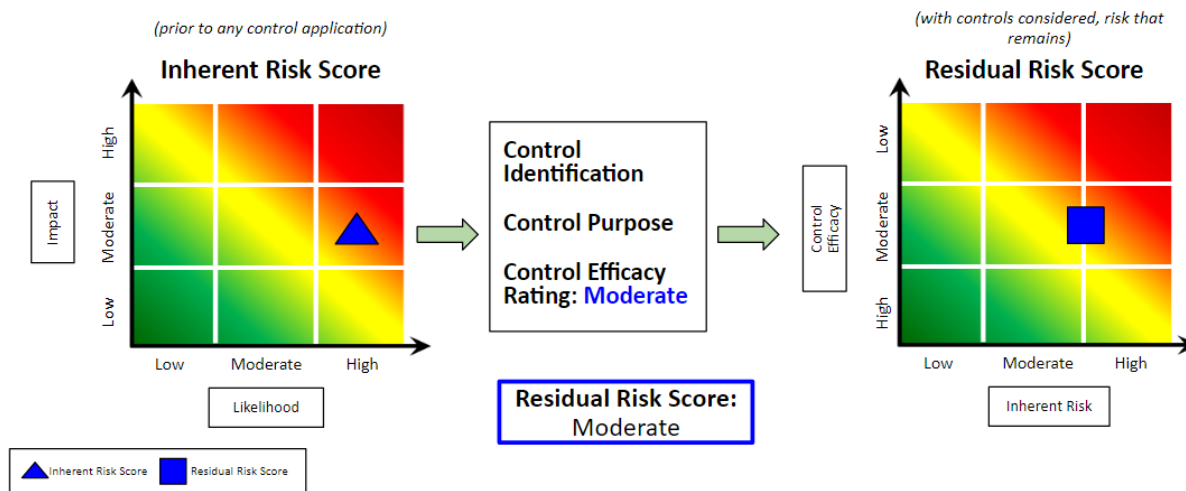


Figure 25: Formulating a residual risk score from an inherent risk score

## 7.8 Plan for Response and Implement Response Plans

Equipped with a detailed understanding of the cyber risks NETCOM DSC-PIT is likely to face on the front line and a prioritized set of available controls, response plans should be developed to address these risks in the most efficient and effective way possible [25]. Creating these response plans first involves developing sets of key risk indicators (KRIs), a list of possible signs that a particular risk is coming to fruition [25]. Reconciling KRIs with the controls available, an investigator can determine:

- **How to respond to a risk.** Typically this will be in the form of either mitigating, transferring, avoiding, accepting, enhancing, exploiting, or sharing the risk [25]. All risk response plans should adhere to risk tolerances specified in the risk appetite statement, meaning that a choice of response and the controls needed to enact it result in an impact within the Command's risk appetite. Enhancing and exploiting a risk should only occur when a clear opportunity is present rather than a threat. **NOTE:** accepting a risk means that an investigator acknowledges that this risk could come to fruition, but does not require any controls to address it. This does not mean that the risk should be ignored. From the perspective of the anomaly detection tool, any anomalous activity that exhibits characteristics of a typically accepted risk must still be slated for investigation to allow for adequate monitoring. Accepted risks should always be monitored over time should they evolve into more impactful risks.
- **Whether multiple risks can be responded to in the same way.** Being closest to the tactical cyber risks the Command faces, DSC investigators are in the best position to identify

any risks that are interdependent. These are such that, when realized, can cause “ripple effects” across the Command (i.e., force other risks to also be realized) [25]. An in-depth assessment of risk characteristics and KRIs will help to identify these risks. While discovering these inter-dependencies can be worrying, they can also indicate ways to make response plans more efficient. Response plans that, due to risk similarities, can address multiple threats at once are key for NETCOM DSC-PIT to identify in order to ensure efficient resource allocation.

Once appropriate response plans are created, implementing them additionally involves establishing ways to measure success and set realistic milestones. DSC investigators are best positioned to oversee this implementation. Measuring success should involve feedback loops between investigators and other risk responders to ascertain where improvements can be made, both in the response plans themselves and for tuning analytics. Responding to risks inherently heightens domain knowledge around those risks/families of risks, and that knowledge should be harnessed by any investigator interacting with the anomaly detection tool to inform future investigations.

## 7.9 Monitor and Measure Effectiveness

In this step, NETCOM shifts from an application perspective to an evaluation one [25]. After applying guidance from the previous steps, measures must be put in place to monitor the effectiveness of all cyber risk management program elements. Typically on the enterprise level, this is done by creating metrics that are applicable to assessing each of the impact areas (defined in the risk appetite statement in Section 7.2) and framed around specific critical assets and services. An example of an enterprise-level metric surrounding human capital would be soldier retention year-on-year, or quarter-on-quarter, depending on how frequently risk thresholds deem this should be reassessed.

From NETCOM DSC-PIT’s perspective, similar metrics can be developed to monitor and measure the anomaly detection tool’s effectiveness. Although these metrics are more granular than that of the soldier retention example above, insights taken from their measurement will have widespread impact. Metrics to consider include:

- The percentage of anomalies slated for investigation that were actually malicious
- The percentage of anomalies ignored that were actually malicious (this is much more difficult to measure and requires additional investigation)
- How often the tool is used. This can be measured by determining the frequency with which anomalies are chosen to be investigated or ignored on a weekly or monthly basis, for example.

When contemplating changes or improvements to the tool, the following metrics may be of interest:

- If new models are added into the tool OR previous models are deprecated, the percentage of anomalies investigated, before and after this change, that were actually malicious
- The percentage of instances in which a particular feature is muted among anomalies slated for investigation. For example, say that an investigator slates 50 anomalies for investigation that he knows are all phishing attacks. Of these anomalies, it is determined that 95% of them were slated for investigation after muting Feature A. This may tell an investigator that Feature A is unimportant when working with a detection tool for phishing attacks.

In addition to utilizing these more quantitative, concrete metrics, NETCOM DSC-PIT investigators and risk owners should constantly assess their governance structure communications. Namely, NETCOM DSC-PIT should make note of how well risk information is being communicated down to the team and how well and often the DSC is reporting risk information up the governance structure.

## 7.10 Review, Update, and Repeat

Cyber risk management is an iterative process. Operating amidst a quickly evolving threat landscape requires NETCOM to be nimble in its cyber risk management practices, especially as Army and DoD-wide projects define new goals to reach and standards to follow. Both annual and more frequent policy reviews are required to keep up with these kinds of changes and any others that have material effects on the Command's operation (often referred to as "tripwires"). These review considerations also apply to controls.

Specifically with the anomaly detection tool, a few of OCTAVE FORTE's improvement areas serve as good indicators of where reviews should be focused and from where unexpected changes could emanate [25]:

- **Investment and procurement:** Over time, as use cases change and the knowledge to address them advances, NETCOM DSC-PIT may wish to assess the algorithmic makeup of the tool. Will any current models used become less popular over time? As such, should any models be offloaded? Will any new anomaly detection models become of interest and require further research? Should more advanced capabilities, beyond VirusTotal scores, be considered to capture more accurate indicators of malicious activity? These are the types of questions NETCOM DSC-PIT investigators can ask when periodically reviewing the tool, and answers to these questions can reveal whether further investment in research is required.
- **Training:** Project Sentinel was, in part, inspired by four key challenge areas identified when the Army first adopted NIST RMF in 2015, one of which was training [11]. In addition to any Command-mandated cyber risk training, it may be helpful for NETCOM DSC-PIT to leverage training opportunities that can help soldiers carry out their daily operations. So as to be applicable to multiple work streams, an example of this kind of training could revolve around building basic cyber security skills that, among other benefits, would help investigators to provide enhanced domain knowledge to feature selection when working with the anomaly detection tool.
- **Policy Changes:** As NETCOM matures in its risk management practices, there will be room to explore categorizing, prioritizing, and ranking risks at a more granular level of detail, either by expanding qualitative descriptions or moving to quantitative measures. As this happens, NETCOM DSC-PIT must evolve to ensure that translations from anomaly scores to risk scores capture any additional nuances.

## 8 Conclusion and Recommendations

We demonstrate that by combining anomaly detection and SHAP explanations in a user-focused application and pairing it with a robust risk-management framework, NETCOM DSC-PIT can better monitor and investigate potentially malicious traffic on the Army network. Furthermore, we provide NETCOM products to put this approach into production immediately. We provide a modular AWS-based pipeline to extend their current NULLFOX process with anomaly scoring and SHAP explanations. We also provide a prototype R-Shiny user interface to aggregate the data and enable human experts to better investigate anomalies within the data. We recommend that NETCOM integrate these tools into their production data science pipeline immediately and begin working with the end-users of the products to refine and improve on them.

## 9 Future Work

We recommend the following five work streams for future work on this project.

1. Searching for “malicious URLs” in an unlabeled data set leads to difficulties in evaluating model performance. Websites can present numerous threats to the enterprise (ex. spam, phishing, drive-by downloads, data exfiltration, or general espionage). We suggest determining the specific threat that NETCOM is searching for in order to allow for better model evaluation and comparison.
2. Not all anomalous URLs are malicious. Further research is needed to determine whether anomaly detection models are the best paradigm for investigating malicious URLs. We suggest investigating additional families of algorithms to see if there exists an alternative which performs better on this data.
3. We suggest increasing the window of observation from one week. Evaluating performance and noise on a well-labeled dataset over periods ranging from bi-weekly, monthly, bi-monthly, and semi-annually would be ideal.
4. Further work is needed in order to enable our pipeline to work in-stream and on a dynamic dataset.
5. We suggest utilizing our sample risk framework to create and maintain an organization-specific risk register which provides analysts and managers the details needed to map *anomaly* scores to *risk* scores. This register would allow for effective and efficient prioritization and response to findings.

## 10 Lessons Learned

While working on this project, we learned many lessons related to successful project management as well as technical details on implementing anomaly detection on NETCOM relevant data.

From the standpoint of successful project management, we learned the following key lessons.

- **Scoping the project early and often with the client is essential.**

It took us a few weeks to fully scope the project and identify how we could best help NETCOM leverage our work. We could have streamlined this process if we had better dialogue on the desired *impact* of the work in the early stages rather than focusing on the implementation details.

- **Adopting a team-of-teams approach helps divide work and promote ownership within the project.**

With six people working on the capstone and multiple lines of effort, successfully managing individual work and contributions was important. We found that using a small team-of-teams approach was effective for us. We split up the group into three teams: model evaluation, post-hoc explanations, and risk management. Each team owned their respective line-of-effort but cross-talk and collaboration was encouraged between the teams.

We also learned some key lessons concerning the technical aspect of the project.

- **xStream requires standardized or normalized feature data.**

While working with Dr. Akoglu on applying xStream in this project, we learned that the feature data needs to be either standardized or normalized in order for the algorithm to be effective. This makes sense considering it uses density estimation to evaluate the data, but it was not explicitly discussed in the xStream paper.

- **Shapley additive explanations enable a dynamic approach to anomaly scoring.**

We originally intended for SHAP values to be used only for post-hoc explanations, specifically for human experts to make sense of the anomaly scores associated with individual observations in the data. However, we found that SHAP values can also be used to dynamically update anomaly scores by adjusting individual feature weights. This enables the user interface to approximate the effect of removing features from the data without having to re-run the anomaly detectors on every permutation of features.

## References

- [1] Kjersti Aas, Martin Jullum, and Anders Løland. Explaining individual predictions when features are dependent: More accurate approximations to Shapley values. *arXiv e-prints*, page arXiv:1903.10464, March 2019.
- [2] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016.
- [3] Matt Barrett, Matt Barrett, Jeff Marron, Victoria Yan Pillitteri, Jon Boyens, Stephen Quinn, Greg Witte, and Larry Feldman. *Approaches for Federal Agencies to Use the Cybersecurity Framework*. US Department of Commerce, National Institute of Standards and Technology, 2020.
- [4] Markus M Breunig, Hans-Peter Kriegel, Raymond T Ng, and Jörg Sander. Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD international conference on Management of data*, pages 93–104, 2000.
- [5] William R. Claycomb, Philip A. Legg, and Dieter Gollmann. Guest editorial: Emerging trends in research for insider threat detection. *J. Wirel. Mob. Networks Ubiquitous Comput. Dependable Appl.*, 5:1–6, 2014.
- [6] U.S. Army Network Enterprise Technology Command. Mission and vision. <https://netcom.army.mil/>, 2022.
- [7] The MITRE Corporation. Crown jewels analysis. <https://www.mitre.org/publications/systems-engineering-guide/enterprise-engineering/systems-engineering-for-mission-assurance/crown-jewels-analysis>.
- [8] Walter Goralski. Chapter 11 - user datagram protocol. In Walter Goralski, editor, *The Illustrated Network (Second Edition)*, pages 289–306. Morgan Kaufmann, Boston, second edition edition, 2017.
- [9] Yasmine Guerbai, Youcef Chibani, and Bilal Hadjadji. The effective use of the one-class svm classifier for handwritten signature verification based on writer-independent parameters. *Pattern Recognition*, 48(1):103–113, 2015.
- [10] (ISC)<sup>2</sup> Inc. Cissp glossary - student guide. <https://www.isc2.org/Certifications/CISSP/CISSP-Student-Glossary>.
- [11] Nancy Kreidler. Project sentinel - the army announces cybersecurity risk management framework reform. [https://www.army.mil/article/230900/project\\_sentinel\\_the\\_army\\_announces\\_cybersecurity\\_risk\\_management\\_framework\\_reform](https://www.army.mil/article/230900/project_sentinel_the_army_announces_cybersecurity_risk_management_framework_reform), Dec 2019.
- [12] Aleksandar Lazarevic, Levent Ertoz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 25–36. SIAM, 2003.
- [13] Fei Tony Liu, Kai Ming Ting, and Zhi-Hua Zhou. Isolation forest. In *2008 eighth ieee international conference on data mining*, pages 413–422. IEEE, 2008.

- [14] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. 12 2017.
- [15] Scott M Lundberg, Gabriel Erion, Hugh Chen, Alex DeGrave, Jordan M Prutkin, Bala Nair, Ronit Katz, Jonathan Himmelfarb, Nisha Bansal, and Su-In Lee. From local explanations to global understanding with explainable ai for trees. *Nature machine intelligence*, 2(1):56–67, 2020.
- [16] Emaad Manzoor, Hemank Lamba, and Leman Akoglu. xstream: Outlier detection in feature-evolving data streams. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1963–1972, 2018.
- [17] United States Department of Defense. 2018 department of defense cyber strategy summary. Technical report, United States Department of Defense, Arlington, V.A., 2011.
- [18] National Institute of Standards and Technology. Managing information security risk: Organization, mission, and information system view. Technical Report National Institute of Standards and Technology Special Publication (NIST SP) 800-39, U.S. Department of Commerce, Washington, D.C., 2011.
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [20] U.S. Army NETCOM Data Science Center Pittsburgh. Spring 2022 capstone initiation brief, 2022.
- [21] Mohammad Puttaroo, Peter Komisarczuk, and Renato Amorim. Challenges in developing capture-hpc exclusion lists. 10 2014.
- [22] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery.
- [23] Bernhard Schölkopf, John C Platt, John Shawe-Taylor, Alex J Smola, and Robert C Williamson. Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471, 2001.
- [24] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *ICISSP*, 2018.
- [25] BA Tucker. Advancing risk management capability using the octave forte process. Technical report, Carnegie Mellon University, Pittsburgh, P.A., 2020.
- [26] BA Tucker. A risk management perspective for ai engineering. Technical report, Carnegie Mellon University, Pittsburgh, P.A., 2020.
- [27] Christian Urcuqui, Andrés Navarro, José Osorio, and Melisa García. Machine learning classifiers to detect malicious websites. In Javier Bustos-Jiménez and Sandra Céspedes, editors,

*Proceedings of the 3rd Spring School on Networks co-located with (ChileCON 2017), Pucon, Chile, October 19-20, 2017*, volume 1950 of *CEUR Workshop Proceedings*, pages 14–17. CEUR-WS.org, 2017.

- [28] VirusTotal. <https://www.virustotal.com/gui/home/upload>.
- [29] Yupeng Xu, Hao Dong, Mingzhu Zhou, Jun Xing, Xiaohui Li, and Jian Yu. Improved isolation forest algorithm for anomaly test data detection. *Journal of Computer and Communications*, 9(8):48–60, 2021.