**Module 7: Portfolio Milestone: Final Research Paper**

John Paul Harris

Colorado State University Global

Course Code: 23FB-MIS581-1

Instructor: Osama A. Morad, Ph.D.

October 1, 2023

Abstract

The Capstone Project developed Machine Learning algorithms to analyze the BETH cybersecurity dataset, composed of modern network traffic with integrated cloud services activity and actual Internet malicious activity. The null hypothesis stated that malicious events will be indistinguishable from network background noise. The alternate hypothesis stated that ML algorithms could identify the malicious activity as anomalous events. The research supported the alternate hypothesis, thus rejecting the null hypothesis. The literature review identified supervised, semisupervised, unsupervised, and self-supervised methods for developing anomaly detection models with out-of-distribution data. The ML techniques included neural networks (supervised learning), one-class SVM (semisupervised learning), and isolation forests (unsupervised or self-supervised learning) for classification-based anomaly detection. One-class SVM was selected as the semisupervised method to analyze the BETH cybersecurity dataset because it was the most used novelty detection technique in the literature review. Isolation forest was selected as the unsupervised method because of its appropriateness for classification-based anomaly detection with out-of-distribution data. Supervised methods, such as neural networks, were not chosen for study because of the complexity of training with out-of-distribution data. Both ML algorithm models detected malicious network activity as anomalous events. The one-class SVM outperformed the isolation forest algorithm with 92% overall accuracy, 100% sensitivity, and 92% specificity. The isolation forest parameters were optimized. However, the optimized model still suffered from low specificity and sensitivity to malicious events.

*Keywords:* anomaly detection, cybersecurity, network intrusion detection, machine learning, out-of-distribution analysis, BETH dataset, one-class SVM, isolation forest

**Module 7: Portfolio Milestone: Final Research Paper**

**Introduction**

Cybersecurity is a dynamic field where threat actors' tactics, techniques, and procedures (TTPs) continually advance. Information security methods to identify malicious TTPs include signature-based or anomaly-based detection (Center for Internet Security [CISA], n.d.). Signature-based detection identifies the indicators of compromise (IOCs) of known threats (CISA, n.d.). However, new attacks do not have known IOCs. Thus, signature-based identification fails to detect unknown attacks, commonly called zero-day attacks. In contrast, Shiravi et al. (2012) explain that anomaly and novelty detection have great potential for detecting zero-day attacks. Hence, the Capstone Project will investigate anomaly- and novelty-based detection methods to identify malicious network activity.

Generally, identifying malicious network activity requires packet inspection at a firewall or analysis of aggregated network flow statistics (Buczak & Guven, 2016). Another option is out-of-band network traffic inspection, where network activity is captured and inspected on another computer distinct from the firewall. The advantage of out-of-band inspection is the availability of more computing power to perform analysis, which allows for a better understanding of network dynamics with robust machine-learning algorithms (Dillard, 2020).

Machine learning algorithms can learn a network activity baseline, a snapshot of normal network behavior. Then, ML algorithms can identify anomalous behavior and report it to an IT administrator for further analysis. For example, candidate ML algorithms should be able to decipher remote user login activity, Secure Shell (SSH) activity, file transfer (FTP or SFTP) activity, and Internet usage by authorized accounts from abnormal activity due to a botnet, crypto mining, or adversarial lateral movement. Also, excessive traffic generated from a single networked machine to other internal machines via uncommon Dynamic/Private ports can indicate a worm or viral replication (Easttom, 2018). ML algorithms should be able to recognize excessive alterations to the network baseline and unusual port activity as an anomaly and notify a network administrator.

ML algorithms must train with datasets derived from modern network traffic flows to provide a realistic benchmark for ML algorithms. Shiravi et al. (2012) explain, “As network behaviors and patterns change and intrusions evolve, it has very much become necessary to move away from static and one-time datasets toward more dynamically generated datasets” (p. 357). Training datasets must include modern network protocols as they also continually advance. For example, the Chrome web browser uses the new QUIC protocol to communicate with Google services (Google Peering, n.d.). Thus, network activity with the Chrome web browser shows QUIC traffic instead of conventional HTTP communications. As new protocols are built to replace older ones, ML algorithms must be exposed to and trained with these protocols.

Highnam et al. (2021) state that currently available cybersecurity datasets are over a decade old, such as the KDD Cup 1999, may include simulated user activity, and do not address modern network architectures like cloud services. In addition, the authors state that currently available datasets do not incorporate kernel processes or network logs, which can provide valuable data for detecting anomalous or malicious behavior. A modern, realistic cybersecurity dataset should include real network traffic with integrated cloud services activity and the names of executed processes.

**Problem Statement**

ML algorithm development must utilize training datasets from current network traffic flows, including modern cloud service activity, protocols, threat tactics, techniques, and procedures (TTPs) to identify today’s novel, zero-day network attacks.

**Objectives**

The research aim is to develop ML algorithms to analyze out-of-band network traffic data to identify anomalous or novel behavior. First, ML algorithms will be trained with a modern-day cybersecurity dataset. The training will provide the ML algorithms with a baseline of standard network activity. The training data set will include vast records of network activity, providing ample data to build a normal profile. Next, it is hypothesized that the ML algorithms will detect malicious network activity in a test data set as anomalous or novel. Well-performing algorithms should identify malicious activity as anomalous with high true positive rates and low false positive rates or specificity.

**Overview of Study**

Recent corporate network security breaches have led to the expansive loss of customers’ personally identifiable information and sensitive health records (Medibank, 2023; Western Digital, 2023). These cases have shown that abnormal network activity is detected when the attackers exfiltrate large amounts of data from the corporate network. However, the early stages of network intrusion are also characterized by anomalous network activity (Highnam et al., 2021). The early stages of network intrusion may include anomalous user activity, abnormally executed process names, and unusual user accounts. Thus, detecting the early stages of network intrusion would allow an organization to protect information assets before they are exfiltrated.

Machine learning algorithms can be trained to detect network anomalies and novelties. Anomaly detection is an unsupervised ML technique, while novelty detection is a semisupervised technique. Thus, the study will assess the performance of unsupervised and semisupervised ML techniques with a cybersecurity dataset that contains malicious network activity. Well-performing ML techniques are candidates for network intrusion detection systems (NIDS) implementation. In addition, novelty detection, the ability to recognize never-seen-before events as different, has the potential for zero-day attack detection.

Training ML algorithms presents challenges because abnormal events are rare and hard to label. Chandola et al. (2009) explain that “anomalous behavior is often dynamic in nature, e.g., new types of anomalies might arise, for which there is no labeled training data” (p. 10). Thus, developing ML algorithms that detect never-seen-before (out-of-distribution) activity is more effective. This study will evaluate the ability of anomaly and novelty detection ML algorithms to identify out-of-distribution malicious activity.

**Research Questions and Hypotheses**

**Research Questions**

O’Leary (2017) recommends crafting research questions by first defining five domain areas: (1) topic, (2) context, (3) goals, (4) nature of questions, and (5) relationships. The Capstone research topic is anomaly and novelty detection in the context of information system communication networks. The goal is to identify malicious network events. The nature of anomaly and novelty detection is to find abnormal events. Finally, relationships between network event attributes and malicious activity will be sought. Accordingly, the research questions are:

1. Can anomaly detection methods identify malicious events in network data?
2. Are there attributes of information system network events that correlate to malicious activity?

**Hypotheses**

Chandola et al. (2009) explain that anomaly detection “is to define a region representing normal behavior and declare any observation in the data which does not belong to this normal region as an anomaly” (p. 15:3). However, Chandola et al. describe challenges to this approach as (1) difficulty defining the normal region, (2) malicious actors masking their behavior to appear normal, (3) normal behavior evolving over time, and (4) data noise appearing as anomalies. Data networks have all four of these challenges. For example, network activity includes data sent from many hosts to many destinations via numerous protocols. Thus, detecting anomalies by a protocol or destination IP address would lead to a high false positive rate. In addition, normal network activity evolves as new protocols are added, and older, inefficient protocols are obsoleted.

Sanchez (2004) discusses the need to periodically re-tune ML algorithms to the background network activity to account for changes over time. Sanchez (2004) states, “Network traffic is not static; thus, the normal and attack traffic evolves over time” (p. 12). The algorithms must be re-trained to the evolving network behavior. If not, then the false positive rate will increase. With the importance of adjusting the algorithms to the background network behavior, it is crucial to define the hypothesis statements. The null hypothesis for anomaly detection in network intrusion detection systems is that all network activity is normal traffic or background noise (Sanchez, 2004). Conversely, the alternate hypothesis is that not all network traffic is normal or background noise. The formally stated hypotheses statements are:

All network events are part of the set of normal traffic events

A network event is not part of the set of normal traffic events

The hypothesis test will be conducted with semi-supervised and unsupervised anomaly and novelty detection machine learning algorithms.

**Literature Review**

Chandola et al. (2009) survey the research and machine learning techniques for anomaly detection. Chandola et al. explain, “Techniques that operate in a semisupervised mode, assume that the training data has labeled instances for only the normal class” (p. 10). Thus, the BETH dataset is an ideal candidate for semisupervised model development since the training data does not contain anomalous instances. In addition, Chandola et al. recommend unsupervised learning where models are developed without training data. However, Chandola et al. explain that unsupervised learning methods depend on the “assumption that normal instances are far more frequent than anomalies in the test data” (p. 11). Finally, Yang et al. (2022) find that self-supervised methods, such as isolation forests, are appropriate for developing a classification-based model with out-of-distribution data.

Patcha and Park (2007) provide an overview of anomaly detection techniques and report that principal component analysis (PCA) is widely used in intrusion detection research. Patch and Park state that the advantage of PCA is the reduction of high-dimensional network traffic data to uncorrelated core components. Shyu et al. (2003) report that PCA outperformed other anomaly detection algorithms, like the Local Outlier Factor, when analyzing the KDD CUP99 network intrusion evaluation dataset. Thus, current research should assess the performance of PCA with the BETH cybersecurity dataset.

Patcha and Park (2003) describe classification-based intrusion detection methods to include neural networks like the Elman recurrent network and support vector machines (SVM). Chandola et al. (2009) report that one-class SVM is appropriate for classification-based anomaly detection techniques. Yang et al. (2022) find that one-class SVM and isolation forest apply to anomaly detection. Finally, Nassif et al. (2021) performed a systematic review of anomaly detection with machine learning techniques. They found that “SVM is the most used technique as either standalone or in hybrid models” (Nassif et al., 2021, p. 78669).

Chandola et al. (2009) recommend semi-supervised techniques like neural network classification-based anomaly detection. In addition, Nassif et al. (2021) found that neural networks “have emerged as a practical technology” (p. 78669) for building ML-based NIDS. Finally, Patcha and Park (2003) found that neural networks have been prominent in their literature overview for classification-based intrusion detection.

The Scikit-learn (2023) library recommends the unsupervised techniques of isolation forest and One-class SVM for anomaly and novelty detection. Both algorithms are ideal for high-dimensional data (Scikit-learn, 2023; Chandola et al., 2009). Another Scikit-learn (2023) recommended unsupervised algorithm for anomaly detection with nonparametric data is the Local Outlier Factor (LOF), which may be a good candidate for the Capstone Project.

**Research Design**

**Methodology**

The research will quantitatively analyze data collected from cloud-hosted, Internet-facing honeypot Linux computers. Highnam et al. (2021) developed the BPF-extended tracking honeypot (BETH) cybersecurity dataset to capture actual Internet activity, including malicious Internet events. The extended Berkeley Packet Filter (eBPF) interacts with the Linux kernel, amassing statistics on network-initiated events (Anand, 2017). Highnam et al. (2021) collected the eBPF data to construct the BETH dataset. The dataset contains event IDs, process IDs, thread IDs, process names, user IDs, and similar network-initiated features from the Linux kernel.

The research aims to develop ML algorithms to detect malicious network activity not seen in training datasets (out-of-distribution). Shiravi et al. (2012) explain that algorithm evaluation works best when abnormal behavior is only included in the testing dataset. In addition, out-of-distribution evaluation best represents the real-world environment in which intrusion detection systems are expected to detect malicious activity and zero-day attacks. Finally, Highnam et al. (2021) state that testing algorithms against datasets with modern-day attack elements are essential to evaluate anomaly-based detection algorithms. Thus, the BETH test dataset incorporates legitimate ransomware, malware, or other cyber attack elements.

Semisupervised and unsupervised ML algorithms will be used for out-of-distribution anomaly and novelty detection. Semi-supervised techniques are well-suited for out-of-distribution analysis because training requires data sets with non-contaminated, labeled normal activity (Chandola et al., 2009). The BETH cybersecurity dataset contains only labeled normal training data. Conversely, unsupervised techniques require training data where “normal instances are far more frequent than anomalies in the test data” (Chandola et al., 2009, p. 11). Thus, the BETH training and testing datasets will be combined to develop an unsupervised anomaly detection algorithm.

**Methods**

The BETH dataset comprises over 8 million network events from 23 Linux honeypots hosted on a cloud service provider. The events represent real network activity with ransomware, malware, and other cyber threats. Highnam et al. (2021) selected clearly defined events that could be appropriately labeled, resulting in a dataset with 1,141,078 records and 16 features.

The BETH dataset contains pre-split data of 60% training, 20% validation, and 20% testing data. The researchers performed the split to ensure the testing dataset contained suspicious and ‘evil’ (malicious) events while the training and validation datasets contained only suspicious events. The pre-split dataset configuration is designed to evaluate the anomaly detection algorithm's ability to identify out-of-distribution abnormal behavior. The ‘evil’ category represents the modern-day attack elements captured from the 23 honeypots hosted on the cloud service provider.

Data preparation will include variable encoding and data transformations. Two categorical variables with high cardinality (*processName* and *hostName*) will be encoded using a multivariate md5 hash algorithm. The hash will be set to 8 bits, resulting in eight new columns to store the hash, one column per bit. Data transformations will include five numeric variables (*mountNamespace*, *returnValue*, *processId*, *parentProcessId*, and *userId*) transformed into binary or trinary variables.

The BETH training dataset will be used to train a semisupervised one-class SVM algorithm. Alternatively, the training and the contaminated testing dataset will be combined to train the unsupervised isolation forest algorithm. The training data will provide the ML algorithms with a baseline of standard network activity. In addition, the training data set will include vast records of network activity, providing ample data to build a normal profile.

The second development phase will be tuning the algorithms with the validation data set. The validation dataset does not contain malicious activity, preserving the out-of-distribution analysis for the testing phase. Thus, validation aims to identify the hyperparameters for each model that optimize inlier specificity (true zeroes / (true zeros + false zeroes)).

The final step will test the ability of each algorithm to accurately detect malicious network activity as anomalous or novel in the testing dataset. Well-performing algorithms should have decent true positive rates with low false positive rates or specificity. Specificity measures the ability of the algorithm to detect abnormal activity without labeling most activity as abnormal.

The research will be conducted using the Python data analytics tool. Specifically, the Anaconda Distribution (<https://www.anaconda.com/download>) will be installed on a single computer to run the Python language. The advantage of Python is the availability of machine learning algorithms from the Scikit-learn library (<https://scikit-learn.org/stable/index.html>), like Isolation Forest and One-class SVM. The Scikit-learn Isolation Forest and One-class SVM algorithms will test the hypothesis.

**Limitations**

The BETH dataset is unique because it is a cybersecurity dataset but does not contain network traffic features like IP addresses, TCP, or UDP port numbers. Instead, it collects network-initiated Linux kernel events from the Internet-facing honeypots. The advantage of the dataset is that it consists of real network activity within a modern cloud architecture with actual attack methods. The limitation of the data is that it does not explain the source of the network-initiated event, like the source IP address. Instead, the dataset represents a network of external sensors monitoring the events of 23 honeypots. In a real-world scenario, such data would fulfill the requirements of an intrusion detection system, alerting administrators to malicious activity. However, it is the job of forensic investigators to locate and isolate the source of malicious activity through other tools or techniques.

**Ethical Considerations**

The Capstone Project will analyze the BETH cybersecurity dataset that Highnam et al. (2021) developed. The data was derived from honeypots (servers opened to the Internet) set up by the researchers on a public cloud service provider to capture network activity with actual attacks from ransomware, malware, or other cyber threats. None of the information systems captured proprietary network activity or the actions of real individuals. Instead, the researchers simulated normal network activities associated with user activity to create background noise. However, the captured malicious activities were actual attacks from the Internet. Thus, the data is owned by the researchers who have made it available for public use. In addition, the data does not contain personally identifiable or proprietary information. Finally, collecting and distributing the data aims to improve network security and protect individuals and organizations from criminal activities.

**Findings**

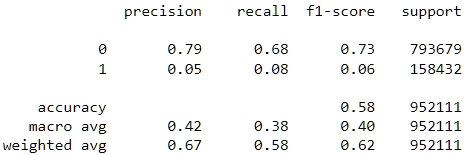
**Isolation Forest**

Isolation forest is an unsupervised ML algorithm that learns a function to separate normal from anomalous events. Isolation forests must learn from a contaminated dataset containing anomalous events. Thus, the BETH training and testing datasets were combined to create a large, contaminated training dataset comprising 793,679 normal and 158,432 anomalous events.

The initial isolation forest model with default parameters from the Scikit-learn library showed an overall accuracy of 58%. However, the ability to recognize malicious events as anomalies (sensitivity) was low at 8%. In addition, the ability to accurately classify events as anomalies (specificity) was low at 5%. The model labeled 355,495 events as anomalous when only 158,432 malicious events existed. Figure 1 shows the classification report for the initial isolation forest model with precision (specificity) and recall (sensitivity) values. Figure 2 shows the confusion matrix for the initial isolation forest model.

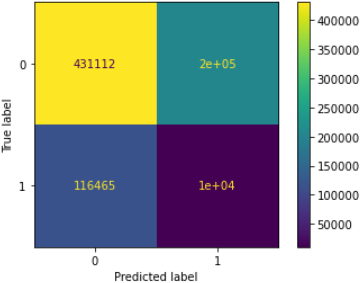
**Figure 1**

*Initial Isolation Forest Classification Report*



**Figure 2**

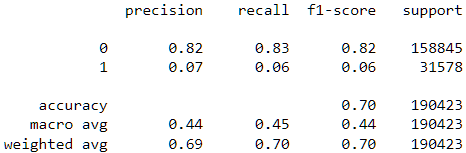
*Initial Isolation Forest Confusion Matrix*



The isolation forest algorithm was tuned by searching for the best-performing *contamination*, *n\_estimators*, and *max\_samples* parameters. The best-performing parameters were *contamination = 0.15*, *n\_estimators = 100*, and *max\_samples = .1*. After tuning, the isolation forest algorithm performed better on the test dataset, with 70% overall accuracy. However, the tuned isolation forest algorithm still suffered the same low sensitivity (6%) and specificity (7%) scores. The tuned isolation forest algorithm better identified normal events but not malicious ones. Figure 3 shows the tuned isolation forest classification report, and Figure 4 shows the confusion matrix.

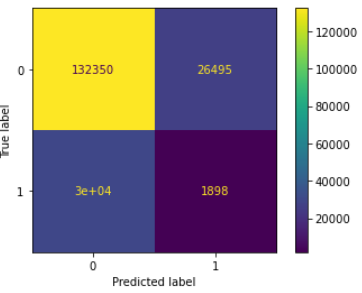
**Figure 3**

*Tuned Isolation Forest Classification Report*



**Figure 4**

*Tuned Isolation Forest Confusion Matrix*



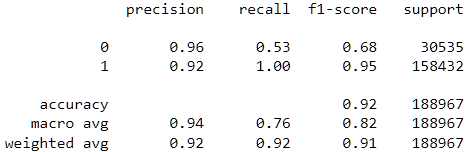
**One-class SVM**

One-class SVM is a semi-supervised ML algorithm that finds a hyperplane to separate normal from anomalous events. The semi-supervised method requires the training dataset to contain only normal events (no contamination), while the test dataset contains a mixture of normal and malicious events. The BETH dataset is already configured for the semi-supervised method with a noncontaminated training dataset. However, the complexity and time drastically increase with the number of training samples for one-class SVM. Thus, the one-class SVM was trained with a random subsample of 200,000 records from the larger BETH training dataset of 763,144 records. Training the one-class SVM on a modern computer with 200,000 training samples took about two hours.

The one-class SVM performed well on the test dataset, with 92% overall accuracy, 100% sensitivity to malicious events, and a low 8% false positive rate (1 - specificity). The model predicted anomalous events with 92% accuracy, where events were genuinely malicious. Figure 5 shows the one-class SVM classification report, and Figure 6 shows the confusion matrix.

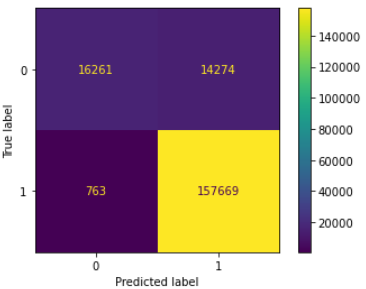
**Figure 5**

One-class SVM Classification Report



**Figure 6**

*One-class SVM Confusion Matrix*



**Discussion**

Both ML algorithms detected malicious network activity as anomalous events. Hence, the answer to the research questions posed by this Capstone project is that ML anomaly detection methods can identify malicious events in network data. In addition, ML algorithms can detect features in information system networks that correlate to malicious activity.

The one-class SVM outperformed the isolation forest ML algorithm with 92% overall accuracy. The high accuracy, specificity, and sensitivity to malicious events of the one-class SVM make it an excellent candidate for use in network intrusion detection systems. The isolation forest algorithm had 70% overall accuracy. However, the accuracy score was due to classifying most events as normal with an unbalanced dataset. In addition, the fine-tuned parameters could not improve specificity and sensitivity to malicious events.

Both ML algorithms rejected the null hypothesis that all network traffic is normal or background noise. Both ML algorithms could distinguish normal from anomalous events in the BETH cybersecurity dataset. Thus, there are features of malicious network activity that can be detected by ML anomaly or novelty detection algorithms.

**Conclusion**

The Capstone Project developed Machine Learning algorithms to analyze the BETH cybersecurity dataset, composed of modern network traffic with integrated cloud services activity and actual Internet malicious activity. The null hypothesis stated that malicious events will be indistinguishable from network background noise. The alternate hypothesis stated that ML algorithms could identify the malicious activity as anomalous events. The research supported the alternate hypothesis, thus rejecting the null hypothesis.

The literature review identified supervised, semisupervised, unsupervised, and self-supervised methods for developing anomaly detection models with out-of-distribution data. The ML techniques included neural networks (supervised learning), one-class SVM (semisupervised learning), and isolation forests (unsupervised or self-supervised learning) for classification-based anomaly detection. In addition, Scikit-learn recommended the unsupervised Local Outlier Factor (LOF) for anomaly detection.

One-class SVM was selected as the semisupervised method to analyze the BETH cybersecurity dataset because it was the most used novelty detection technique in the literature review. Isolation forest was selected as the unsupervised method because it was identified in the literature review as appropriate for classification-based anomaly detection with out-of-distribution data. Supervised methods, such as neural networks, were not chosen for study because such methods are not well-suited for analysis with out-of-distribution data.

Both ML algorithm models detected malicious network activity as anomalous events. The one-class SVM vastly outperformed the isolation forest algorithm with 92% overall accuracy, 100% sensitivity, and 92% specificity. The isolation forest model was optimized by identifying the best-performing hyperparameters. However, the optimized model still suffered from low specificity and sensitivity to malicious event detection.

**Recommendations**

The paper shows that the one-class SVM algorithm has great potential for detecting malicious network activity. The one-class SVM model had an overall 92% accuracy, 100% sensitivity to malicious events, and an 8% false positive rate (1 - specificity). The one-class SVM model could be further tuned by identifying the best-performing *nu* and *gamma* parameters for a specific network environment. It is recommended that one-class SVM be further studied within different network environments or with different cybersecurity datasets to assess its versatility and capabilities to identify malicious activity.

Research to further optimize the isolation forest algorithm should be conducted to assess its full potential for malicious network event detection. The Capstone Project performed a simple preliminary hyperparameter search for the best performers. Other hyperparameter search techniques should be employed to optimize the model further.

**References**

Anand, P. (2017, September 17). *An intro to using eBPF to filter packets in the Linux kernel*. Opensource.com. <https://opensource.com/article/17/9/intro-ebpf>

Buczak, A. L., & Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys & Tutorials*, *18*(2), 1153–1176. <https://doi.org/10.1109/COMST.2015.2494502>

Center for Internet Security. (n.d.). *Election security spotlight – signature-based vs. anomaly-based detection*. <https://www.cisecurity.org/insights/spotlight/cybersecurity-spotlight-signature-based-vs-anomaly-based-detection>

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection : A survey. *ACM Computing Surveys (CSUR)*, *41*(3), 1–58. <https://doi.org/10.1145/1541880.1541882>

Dillard, J. (2020, July 23). *The 101 series: Out-of-band vs inline network security*. Garland Technology. <https://www.garlandtechnology.com/blog/the-101-series-out-of-band-vs-inline-network-security>

Easttom, C. (2018). *Network defense and countermeasures: Principles and practices* (3rd ed.). Pearson Education, Inc.

Google Peering. (n.d.). *QUIC*. <https://peering.google.com/#/learn-more/quic>

Highnam, K., Arulkumaran, K., Hanif, Z., & Jennings, N. R. (2021). BETH dataset: Real cybersecurity data for anomaly detection research. *International Conference on Machine Learning 2021 Workshop on Uncertainty and Robustness in Deep Learning*. <http://www.gatsby.ucl.ac.uk/~balaji/udl2021/accepted-papers/UDL2021-paper-033.pdf>

Medibank. (2023, February 23). *Cyber event timeline*. Retrieved June 11, 2023, from <https://www.medibank.com.au/health-insurance/info/cyber-security/timeline/>

Nassif, A. B., Talib, M. A., Nasir, Q., & Dakalbab, F. M. (2021). Machine learning for anomaly detection: A systematic review. *IEEE Access*, *9*, 78658-78700. <https://doi.org/10.1109/ACCESS.2021.3083060>

O’Leary, Z. (2017). *The essential guide to doing your research project* (3rd ed.). Sage Publishing.

Patcha, A., & Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer Networks*, *51*(12), 3448–3470. <https://doi.org/10.1016/j.comnet.2007.02.001>

Sanchez, M. D. (2004, February 23). *Anomaly detection: How to apply hypothesis testing to IDS events for quality control*. GIAC Certifications. <https://www.giac.org/paper/gsec/3741/anomaly-detection-apply-hypothesis-testing-ids-events-quality-control/105998>

Scikit-learn. (2023). *2.7. Novelty and outlier detection*. <https://scikit-learn.org/stable/modules/outlier_detection.html#outlier-detection>

Shiravi, A., Shiravi, H., Tavallaee, M., & Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, *31*(3), 357–374. <https://doi.org/10.1016/j.cose.2011.12.012>

Shyu, M. L., Chen, S. C., Sarinnapakorn, K., & Chang, L. (2003). *A novel anomaly detection scheme based on principal component classifier* [Workshop presentation]. Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, Melbourne, FL, United States. <https://apps.dtic.mil/sti/pdfs/ADA465712.pdf>

Western Digital. (2023, May 5). *Western Digital provides update on network security incident*. <https://www.westerndigital.com/company/newsroom/press-releases/2023/2023-05-05-western-digital-provides-update-on-network-security-incident>

Yang, J., Zhou, K., Li, Y., & Liu, Z. (2022). Generalized out-of-distribution detection: A Survey. *arXiv*. <https://doi.org/10.48550/arXiv.2110.11334>