# Overcoming the Labeled Training Data Bottleneck: A Route to Specialized AI

Dr. Ravi Starzl

**ABSTRACT**

*This article explores the potential of large language models (LLMs) to transform dataset creation and analysis in cybersecurity. The proposed method leverages LLMs to overcome the labeled data bottleneck by generating high-quality, task-specific datasets for AI model tuning. Existing network intrusion analysis datasets are synthesized with domain knowledge extracted from cybersecurity literature to create a new dataset tailored for supervised training of zero-day exploit detection systems. LLMs interpret the semantic content of relevant literature to identify crucial characteristics and values of zero-day exploit signatures in network traffic. The resulting synthesized dataset is primarily based on 'organic' data collected by genuine sensors, with key feature characteristics intelligently interpolated by LLMs. This approach enables the creation of suitable training data for high-performance ML models. This article demonstrates the effectiveness of this method by utilizing advanced AI techniques to generate a dataset for zero-day exploit detection, illustrating the potential for accelerated progress in specialized AI for cybersecurity. The proposed solution offers a promising approach to address the challenge of labeled data scarcity in developing specialized AI for cybersecurity, facilitating more efficient and effective protection against emerging threats.*

**Dr. Ravi Starzl**, on faculty at Carnegie Mellon University (CMU) and the University of Colorado, was recently awarded the 2024 Award for Artificial Intelligence (AI) Excellence by the Business Intelligence Group. His current area of focus is the application of AI, Machine Learning (ML), and Complex Systems Science to radio frequency (RF), operational technology (OT), and Cyber Operations for asymmetric advantage. Dr. Starzl earned his M.S. and Ph.D. in ML and AI from Carnegie Mellon University

## INTRODUCTION

The advent of LLMs heralds a transformative era in dataset creation and analysis.[1] By generating high-quality, task-specific labeled datasets, LLMs are setting the stage for unprecedented advancements in data variance identification and system analysis, enabling the development of specialized AI with superior performance and precision.[2] This article explores the potential of LLMs to revolutionize large-scale dataset processing in cybersecurity,[3] focusing on their ability to overcome labeled data bottlenecks and refine AI model tuning through tailored dataset generation.

### *The Role of Information in AI and ML for Cybersecurity*

Information is the foundation of inference tasks in ML and AI. Extracting and processing meaningful patterns and insights from cybersecurity data enable AI systems to identify threats, predict attacks, and perform complex security analyses.[4] However, accessing and leveraging information for cybersecurity AI present major challenges, such as the lack of labeled training data specific to cybersecurity tasks, which often hinder development of high-performing AI models.

### *Overcoming the Labeled Data Bottleneck in Cybersecurity*

To overcome the labeled data bottleneck in cybersecurity, this work proposes an innovative way to leverage the power of LLMs. By utilizing existing datasets, even if not designed to address the specific task at hand, and leveraging the descriptions of systems and semantic relationships within the context of the objective task, LLMs can generate high-quality, task-specific labeled datasets. This approach harnesses the expressive power of current LLMs and other transformer-based learning systems to create new, synthesized datasets rooted in organic examples but synthetically fused together and interpolated to create suitable training data for high-performance ML models on objective tasks that have insufficient traditionally gathered training data.

### *Demonstrating the Method: Zero-Day Exploit Detection*

To ground the proposed method for use by cybersecurity practitioners, this work demonstrates how advanced AI techniques can overcome a data label bottleneck in developing an AI for zero-day exploit detection. By leveraging traditional intrusion detection datasets and LLM capabilities, the method creates a new dataset capable of detecting network traffic patterns that may be associated with zero-day exploits, showcasing the potential for rapid advancements in specialized AI for cybersecurity.[5]

### *The Future of Specialized AI in Cybersecurity*

As the sophistication of contextual inference capabilities and the scale of data access and processing grows, we anticipate further rapid growth of methods like the one presented here. These advances will enable swift scaling of specialized AI in cybersecurity, revolutionizing threat detection, attack prediction, and overall network security.

The proposed method, which leverages LLMs to synthesize task-specific labeled datasets, offers a promising solution to overcome labeled data bottlenecks in specializing AI for cybersecurity. By demonstrating the method's effectiveness in creating a dataset for zero-day exploit detection, this work highlights the potential for rapidly advancing AI-driven cybersecurity solutions, paving the way to better protect against evolving threats.

### *Information, Error, and Synthesizing Specialized Datasets*

At the heart of developing effective cybersecurity AI are the fundamental concepts of information and error. Information, in this context, refers to meaningful patterns and relationships within cybersecurity data that enable AI systems to detect threats and anomalies. Error represents discrepancies between the AI's predictions and ground truth, which can arise from various sources such as data inconsistencies, model limitations, or the inherent complexity of the cybersecurity domain.

Decomposing error into its constituent components—bias, variance, and irreducible error—provides a mathematical and philosophical framework for understanding the challenges and opportunities in creating specialized datasets for cybersecurity AI. Bias, which arises from oversimplified assumptions or limited data representation, can be mitigated by incorporating a wider range of cybersecurity scenarios and data sources. Variance, reflecting the model's sensitivity to fluctuations in the training data, can be addressed through techniques like regularization or ensemble learning. Irreducible error represents the inherent uncertainty or "noise" within the data, which cannot be eliminated entirely. This is a byproduct of assumptions and methods by which data are collected or sampled.

To overcome these challenges and create specialized datasets for tasks like zero-day exploit detection, we can leverage the power of information theory and ML techniques to create systems that implicitly capture the information structures that drive the observed variations associated with cyber security events.[6] This information structure is necessarily reflected in the use of language and code that are the substance of cyber security or cyber operations. By analyzing the information content and relationships within existing datasets, such as intrusion detection data, we can identify key patterns and features most relevant to the target task. This analysis can guide the synthesis of new, specialized datasets that capture the essential characteristics of the target domain while minimizing the impact of irreducible error.[7]

The approach we explore here leverages the expressive power of current LLMs to generate realistic and diverse cybersecurity scenarios from related datasets designed to address different objective tasks. By also training these models on a wide range of cybersecurity literature, threat reports, and technical documentation, we can create a rich semantic understanding of the domain. This understanding can then be used to generate synthetic dataset designs, as well as data points that mimic the patterns and relationships observed in real-world cybersecurity events, while introducing controlled and semantically viable variations to enhance the model's ability to generalize to new threats.

This can further be extended with techniques from transfer learning and domain adaptation[8] to leverage knowledge gained from adjacent cybersecurity tasks. By identifying common patterns and features shared between related tasks, like intrusion detection and malware analysis,  insights and representations learned from previous tasks can be transferred to a new task.

Synthesis of specialized datasets for cybersecurity AI requires deep insight into the concepts of information and error. Decomposing error into its three components and leveraging the power of information theory and ML techniques allows us to create rich and diverse datasets that capture the essential characteristics of the target domain while 'borrowing' insight from existing problem-adjacent datasets and literature. Approaches like language model-based data generation, transfer learning, and adversarial training allows us to overcome the limitations of manual labeling and enhance our ability to generalize to new threats.[9]

## TRANSITION FROM INTRUSION DETECTION TO ZERO-DAY EXPLOIT DETECTION

The 1999 KDD cybersecurity dataset, while influential in developing ML models for network intrusion detection, falls short in addressing the unique challenges posed by zero-day exploits. These exploits, which target unknown vulnerabilities, require a novel approach to detection that goes beyond the scope of the KDD dataset.

The method presented here utilizes LLMs to create a new objective-task training data matrix specifically designed for detecting zero-day exploits.[10] The process begins by granting the LLM access to a vast array of cybersecurity literature, including research papers, technical reports, and industry publications. This extensive knowledge base enables the LLM to develop a deep understanding of the characteristics, patterns, and indicators associated with zero-day exploits.[11]

Armed with this domain-specific knowledge, the LLM is then tasked with inferring the essential features and attributes that are crucial for effective zero-day exploit detection. This inference process involves analyzing the cybersecurity literature to identify the unique signatures, anomalies, and behavioral patterns that distinguish zero-day exploits from known threats and normal network traffic.[12,13]

Based on these inferred characteristics, the LLM designs a comprehensive objective-task training data matrix that encapsulates the key elements necessary for training ML models to detect zero-day exploits. This matrix serves as a blueprint for the synthesized dataset, ensuring that it includes the most relevant and informative features for the task at hand.

To populate the objective-task training data matrix, the LLM analyzes the KDD dataset and a diverse array of other cybersecurity and network traffic datasets, extracting pertinent features and patterns that align with the designed matrix. The LLM then generates specific scripts and transformation functions to integrate and adapt the data from these datasets into the new synthesized dataset.

The resulting synthesized dataset is not a mere amalgam of existing datasets, but rather a crafted resource tailored to the unique challenges of zero-day exploit detection. By leveraging the LLM's knowledge of cybersecurity literature and its ability to infer the critical characteristics of the objective task, the synthesized dataset provides a rich and comprehensive training ground for advanced ML models.

New models for anomaly detection, unsupervised learning, and more, can then be trained on the synthesized dataset to learn the subtle nuances and patterns associated with zero-day exploits.[14,15] Exposing the models to a wide range of realistic and diverse scenarios encompassed within the synthesized dataset enables them to develop the ability to identify and flag potential zero-day exploits in real-world network traffic.

This method marks a step forward for cybersecurity. By harnessing the power of LLMs and their access to extensive cybersecurity literature, this approach enables the creation of highly specialized and effective training datasets for zero-day exploit detection. The resulting models, trained on these synthesized datasets, have the potential to shift the way organizations detect and respond to previously unknown vulnerabilities, bolstering their overall security posture.

## PROMPTING PROCEDURE

Utilizing a Llama2-70B model embedded in a Python 3.11 program and the transformers library from huggingface, an initial prompt asked the LLM how it would transform the existing network dataset into a dataset useful for zero-day anomaly exploit detection. A table was also provided with the prompt showing a reduced set of features from the 1999 KDD cybersecurity dataset:

> Please provide suggestions on how to modify the data matrix and labels to enable zero-day exploit detection. Think creatively and provide a detailed explanation of the changes you are making and why they are important. Here are the features of the data matrix and some example values:

| Protocol Type | Service | Flag | Src Bytes | Dst Bytes | Land | Wrong Fragment | Urgent | Count | Srv Count | Serror Rate | Srv Serror Rate | Rerror Rate | Srv Rerror Rate | Same Srv Rate | Diff Srv Rate | Srv Diff Host Rate | Label |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| tcp | http | SF | 215 | 45076 | 0 | 0 | 0 | 8 | 8 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | good |
| udp | private | SF | 105 | 146 | 0 | 0 | 0 | 15 | 15 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | good |
| tcp | telnet | RSTO | 0 | 0 | 0 | 0 | 0 | 23 | 10 | 0.00 | 0.00 | 1.00 | 1.00 | 0.17 | 0.83 | 0.00 | bad |
| icmp | eco_i | SF | 8 | 0 | 0 | 0 | 0 | 13 | 13 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | good |
| tcp | ftp_data | So | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1.00 | 1.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | bad |
| tcp | http | SF | 337 | 981 | 0 | 0 | 0 | 2 | 2 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 1.00 | good |
| tcp | http | SF | 0 | 0 | 0 | 0 | 0 | 50 | 25 | 0.20 | 0.20 | 0.00 | 0.00 | 0.50 | 0.50 | 0.00 | bad |
| udp | domain | SF | 44 | 133 | 0 | 0 | 0 | 8 | 8 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.00 | good |
| tcp | smtp | SF | 789 | 334 | 0 | 0 | 0 | 4 | 4 | 0.00 | 0.00 | 0.00 | 0.00 | 1.00 | 0.00 | 0.50 | good |
| tcp | ftp | SF | 0 | 0 | 0 | 1 | 0 | 16 | 8 | 0.75 | 0.75 | 0.25 | 0.25 | 0.50 | 0.50 | 0.00 | bad |

Table 1: A reduced set of features from the 1999 KDD cybersecurity dataset, provided in the prompt.

The response received outlined a suggested set of modifications illustrated in Figure 1.

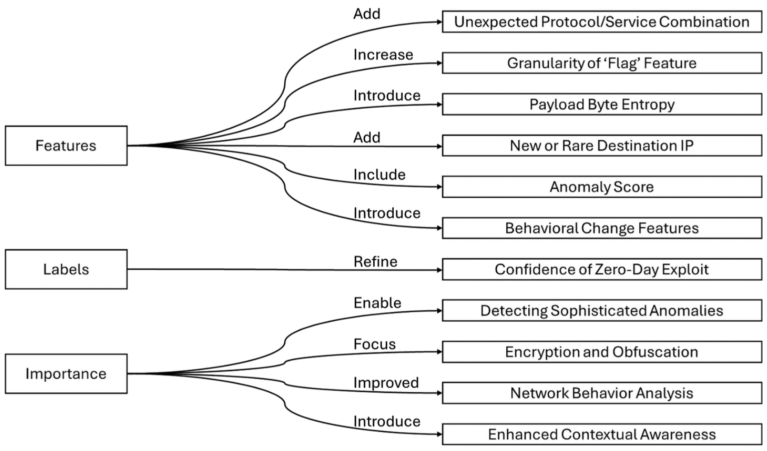
Features
Labels
Importance
Add
Unexpected Protocol/Service Combination
Increase
Granularity of 'Flag' Feature
Introduce
Payload Byte Entropy
Add
New or Rare Destination IP
Include
Anomaly Score
Introduce
Behavioral Change Features
Refine
Confidence of Zero-Day Exploit
Enable
Detecting Sophisticated Anomalies
Focus
Encryption and Obfuscation
Improved
Network Behavior Analysis
Introduce
Enhanced Contextual Awareness

Figure 1: LLM suggested data set modifications.

The next prompt elicited a specific set of features most likely to provide powerful predictive performance in detecting network traffic associated with a zero-day exploit, utilizing network traffic data only. This was a two-prompt process.

**Prompt 1:** Ask the LLM for assistance in crafting a more comprehensive prompt for the objective task:

> Imagine you are entering a data science competition aimed at developing a model to classify network traffic into three categories: normal, suspicious, and highly suspicious. You will use well-known network intrusion detection datasets for this task. Your objective is to identify the most informative features within those datasets that can be combined to help distinguish between these traffic types effectively.
>
> Draft a strategy for selecting key features from basic traffic descriptors, behavioral indicators, and advanced metrics that could potentially highlight network anomalies. Describe how these features will be utilized to enhance the accuracy and effectiveness of your classification model.

**Prompt 2:** Ask the LLM to directly specify the set of features that would be most informative to the objective task:

> Imagine you are competing in a data science challenge with the goal of developing a cutting-edge model that classifies network traffic into three distinct categories: normal, suspicious, and malicious. This task is vital for boosting network security measures by pinpointing potential threats through a detailed analysis of network traffic.
>
> To undertake this challenge, you plan to leverage datasets from publicly available network intrusion detection resources, like the NSL-KDD dataset. Your task is to devise a model that uses critical network traffic features, meticulously selected for their ability to shed light on network behavior and potential threats.
>
> Your first step is to identify and articulate the most informative features within the data that could distinguish effectively between the three traffic types. Consider what basic traffic features (like protocol types and data volume), behavioral features (such as connection patterns and error rates), advanced indicators (including unusual protocol-service combinations and payload characteristics), and anomaly detection metrics (which measure deviations from normal behavior) might be crucial for this task.
>
> Craft your feature selection strategy to include both well-established and novel metrics, aiming to build a robust model that excels in accuracy and reliability for classifying network traffic. This approach will ensure that your model not only identifies but also understands the subtleties and complexities of network traffic, enhancing your ability to detect and classify potential security threats efficiently.

The features identified through this process as being most useful in detecting network traffic associated with zero-day exploits by the LLM are:

> Protocol Type Service, Flag, Src Bytes, Dst Bytes, Land Wrong Fragment, Urgent, Count, Srv Count, Serror Rate Srv Serror Rate, Rerror Rate, Srv Rerror Rate, Same Srv Rate, Diff Srv Rate,Srv Diff Host Rate, Unexpected Protocol/Service,Payload Byte Entropy, New/Rare Destination IP, Anomaly Score, Behavioral Change Score.

The next prompt provided an interstitial reasoning exercise for the LLM to gain more traction on how to modify the dataset in detail, as well as to provide a scaffold for where to discover the needed data to make reasonable interpolations for synthetic data generation.

> Think creatively. How would you generate the actual data for the new features and add them to the new table. Provide a detailed explanation. Use all the publicly available knowledge about these features and how they can be estimated.

The response received outlined suggested process and method illustrated in Figures 2 and 3.

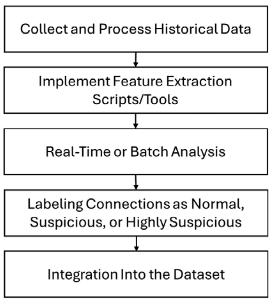
Collect and Process Historical Data
Implement Feature Extraction Scripts/Tools
Real-Time or Batch Analysis
Labeling Connections as Normal, Suspicious, or Highly Suspicious
Integration Into the Dataset

Figure 2. LLM suggested pipeline overview for synthesized dataset creation.

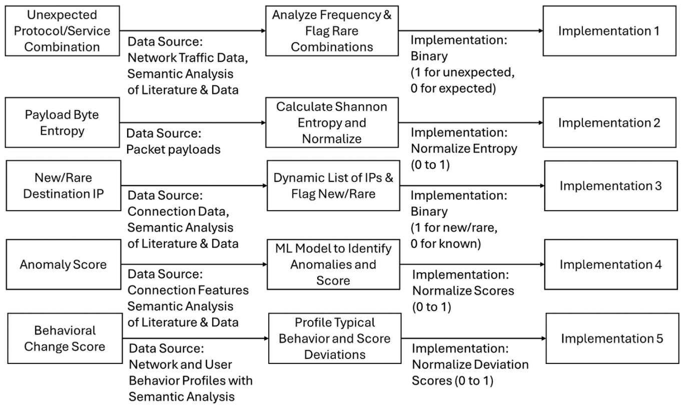
Unexpected Protocol/Service Combination
Data Source: Network Traffic Data, Semantic Analysis of Literature & Data
Analyze Frequency & Flag Rare Combinations
Implementation: Binary (1 for unexpected, 0 for expected)
Implementation 1
Payload Byte Entropy
Data Source: Packet payloads
Calculate Shannon Entropy and Normalize
Implementation: Normalize Entropy (0 to 1)
Implementation 2
New/Rare Destination IP
Data Source: Connection Data, Semantic Analysis of Literature & Data
Dynamic List of IPs & Flag New/Rare
Implementation: Binary (1 for new/rare, 0 for known)
Implementation 3
Anomaly Score
Data Source: Connection Features Semantic Analysis of Literature & Data
ML Model to Identify Anomalies and Score
Implementation: Normalize Scores (0 to 1)
Implementation 4
Behavioral Change Score
Data Source: Network and User Behavior Profiles with Semantic Analysis
Profile Typical Behavior and Score Deviations
Implementation: Normalize Deviation Scores (0 to 1)
Implementation 5

Figure 3. LLM suggested features, implementations, and method for designing a dataset aimed at detecting network anomalies and zero-day exploits through network traffic analysis.

## DATA ALIGNMENT AND DATASET IDENTIFICATION THROUGH VECTOR EMBEDDING

The data were aligned to enhance the detection of zero-day exploits and improve cybersecurity network traffic analysis. A custom Python program, leveraging the Ray parallel processing library, was developed to traverse the May 2023 Common Crawl dataset. Utilizing BERT base uncased, this program generated vector embeddings for the elements of the dataset that were pre-identified as potentially relevant with phrases indicative of cybersecurity interests such as "zero-day exploit data set," "cyber security network traffic analysis," "comprehensive cybersecurity dataset," and "network intrusion anomaly dataset." This dataset was then further mined using a vector-database (Weaviate) and retrieval-augmented-generation (RAG) to identify other datasets that contained the type of information the LLM had already identified as potentially helpful to the prediction task. Identifying and including these additional datasets provided more information for synthesizing values for missing features in the newly designed dataset for the objective task. This generally has the effect both of increasing accuracy of the synthesized values and providing a broader but still realistic range of variations in those values.

This process unearthed several key datasets for cybersecurity analysis:

- **KDD Cup 99 Dataset:** Despite its age, this dataset remains a cornerstone in network intrusion detection, offering a broad spectrum of network connection features.
- **NSL-KDD Dataset:** An evolution of the KDD Cup 99 dataset, the NSL-KDD addresses previous limitations by eliminating redundant records, thereby enhancing the dataset's utility for training and testing intrusion detection models.

- **CICIDS2017 (Canadian Institute for Cybersecurity Intrusion Detection System 2017):** Featuring both malicious and benign attacks, this dataset reflects contemporary attack scenarios with detailed network traffic features and, for some attack types, payload data.
- **UNSW-NB15 Dataset:** Provided by the Australian Cyber Security Centre, this dataset mixes real normal activities with synthetic attack behaviors, offering a diverse array of network traffic analysis features.
- **ISCX VPN-nonVPN Traffic Dataset (ISCXVPN2016):** Containing labeled network traffic that differentiates between VPN and non-VPN traffic, this dataset is invaluable for studying encrypted traffic patterns and potentially high-entropy payloads.
- **CTU-13 Dataset:** This dataset includes botnet traffic alongside normal and background traffic, facilitating the study of botnet behaviors which may share similarities with zero-day exploit traffic patterns, especially in command and control communications and lateral movements.
- **MAWI Working Group Traffic Archive:** A compilation of real-world internet backbone traffic datasets from the Wide project, capturing a variety of internet activities over extended periods. While not cybersecurity-specific, it offers a rich baseline for normal traffic pattern analysis.
- **The CAIDA UCSD Datasets:** Provided by the Center for Applied Internet Data Analysis (CAIDA), these datasets consist of anonymized internet traces, aiding in the modeling of both normal and anomalous network behaviors.
- **ToN IoT Telemetry Dataset:** A contemporary dataset focusing on Internet of Things (IoT) telemetry, including network traffic, logs, and attack data. It is particularly suited for exploring IoT-specific threats and anomalies.

The next prompt asked the LLM to derive the best way to align features and labels across these diverse datasets:

> Please provide a detailed feature alignment plan to align the features of these datasets with these features: Protocol Type Service, Flag, Src Bytes, Dst Bytes, Land Wrong Fragment, Urgent, Count, Srv Count, Serror Rate Srv Serror Rate, Rerror Rate, Srv Rerror Rate, Same Srv Rate, Diff Srv Rate, Srv Diff Host Rate, Unexpected Protocol/Service, Payload Byte Entropy, New/Rare Destination IP, Anomaly Score, Behavioral Change Score.

The five methodologies the LLM defined are shown in Figure 4. This alignment between existing dataset features ("organic" features derived from actual sensor observations), and designed dataset features (ideal features for the objective task that the LLM suggests) minimizes the distance between the two feature sets, helping the inferred values to draw the maximum amount of information from the actually observed values, and helping to minimize the attenuation of the underlying signal in the process of synthesizing the new dataset.

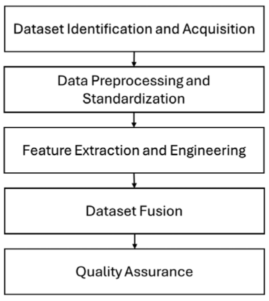
Dataset Identification and Acquisition
Data Preprocessing and Standardization
Feature Extraction and Engineering
Dataset Fusion
Quality Assurance

Figure 4. Overview of the implementation process for creating a synthesized dataset using LLMs and existing cybersecurity datasets. The process consists of five steps that synthesize diverse cybersecurity datasets into a unified dataset.

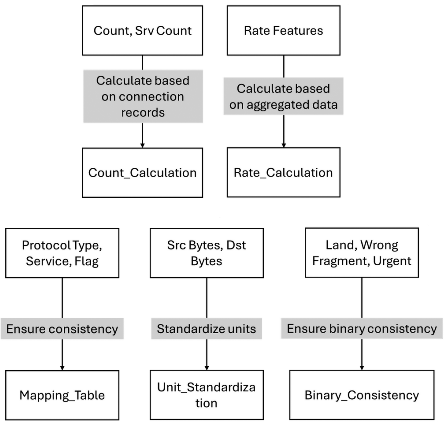

Figure 5. Dataset Identification and Acquisition - The first step in the dataset fusion process involves identifying relevant cybersecurity datasets.

The “closer” the new synthesized dataset features are to the existing dataset features, the greater the real-world performance and generalization of the models trained on that data.

## IMPLEMENTATION STEPS

This method for fusing diverse cybersecurity datasets, guided by an LLM and automated through Python scripts, creates a unified, analysis-ready dataset.

### *Dataset Identification and Acquisition*

The first step in the dataset fusion process involved identifying relevant cybersecurity datasets, as described in Figure 5. The LLM, implicitly trained on a vast corpus of cybersecurity literature, provided guidance on selecting datasets that encompass a wide range of attack types, network environments, and data formats. The identified datasets were downloaded and stored in a centralized repository.

### *Data Preprocessing and Standardization*

The acquired datasets underwent a preprocessing phase shown in Figure 6. The LLM generated dynamic prompts, tailored to each dataset's specific attributes and requirements, which were then used to develop Python scripts for automated data preprocessing. These scripts performed tasks such as decoding features into plain text, handling missing values, and converting the datasets into a uniform tabular format, typically CSV files.

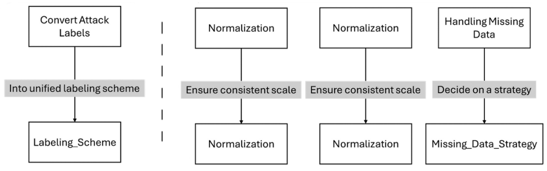

Figure 6. Data Preprocessing and Standardization - The acquired datasets undergo a preprocessing phase.

### *Feature Extraction and Engineering*

The LLM defined a detailed set of features for the fused dataset shown in Figure 7. By analyzing the characteristics of each dataset and leveraging its knowledge of cybersecurity domain expertise, the LLM generated a feature alignment plan. This plan outlined the necessary features to be extracted from each dataset, as well as advanced features that could be derived through feature engineering techniques.

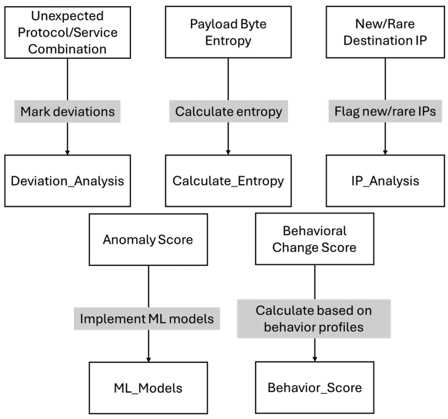

Figure 7. Feature Extraction and Engineering - The LLM defines a detailed set of features for the fused dataset. By analyzing the characteristics of each dataset and leveraging its knowledge of cybersecurity, the LLM generates a feature alignment plan.

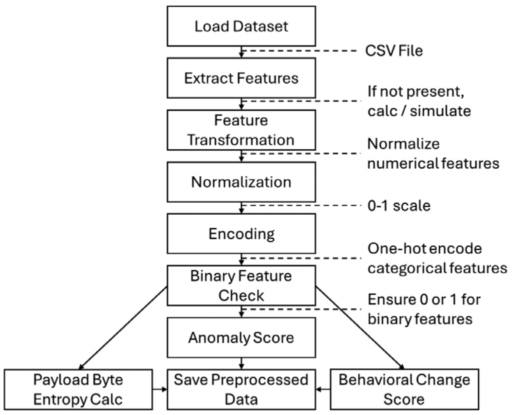

Figure 8. Dataset Fusion – The final stage involves fusing the preprocessed and feature-engineered datasets into a unified dataset. The LLM provides guidance on the techniques used, considering factors such as data format, feature compatibility, and scalability.

Python scripts, guided by the LLM's feature alignment plan, were developed to automate the feature extraction and engineering process. These scripts utilized various data manipulation libraries and analytical tools to calculate complex features, such as entropy measures and behavioral change scores, based on the available data. The extracted and engineered features were then integrated into the preprocessed datasets, resulting in a set of feature-rich, analysis-ready datasets.

## *Dataset Fusion and Quality Assurance*

The final stage of the method involved fusing the preprocessed and feature-engineered datasets into a unified dataset as shown in Figure 8. The LLM provided guidance on the appropriate data fusion techniques, considering factors such as data format, feature compatibility, and scalability. Python scripts were developed to automate the dataset fusion process using pandas and numpy.

To ensure the integrity and reliability of the fused dataset, a quality assurance process was implemented. The LLM generated prompts for automated data validation scripts, which checked for anomalies, inconsistencies, and outliers in the fused dataset. Additionally, the scripts verified the correct application of labels and analyzed feature distributions to ensure the coherence and representativeness of the fused dataset. Manual checking of samples of the results was also undertaken.

By leveraging the knowledge and reasoning capabilities of the LLM, the method draws from a wide range of sources to ensure the fused dataset is well-aligned, feature-rich, and suitable for the objective task within the limits of current semantic analysis. The automated nature of the process, facilitated by Python scripts, enables efficient and scalable dataset fusion, reducing manual effort.

Individual scripts were produced with the assistance of LLM crafted prompts. The prompts were generated with a seed prompt such as:

> Please provide me with a prompt I can give to code llama to normalize the features of one of these datasets with the target set of features. Include the target features as a template in the prompt.

For brevity, the logical structure of two such resulting prompts, one to normalize data and the other to conduct literature review and incorporate resulting insights, are shown in Figure 9.

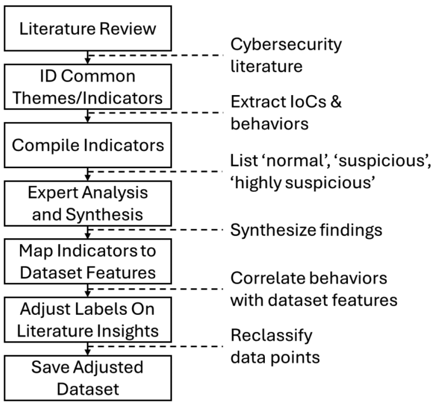
Literature Review
Cybersecurity literature
ID Common Themes/Indicators
Extract IoCs & behaviors
Compile Indicators
List 'normal', 'suspicious', 'highly suspicious'
Expert Analysis and Synthesis
Synthesize findings
Map Indicators to Dataset Features
Correlate behaviors with dataset features
Adjust Labels On Literature Insights
Reclassify data points
Save Adjusted Dataset

Figure 9. Quality Assurance - To ensure the integrity and reliability of the fused dataset, a quality assurance process is implemented. The LLM generates prompts for automated data validation scripts, which check for anomalies, inconsistencies, and outliers.

The code generated by code llama required manual editing to ensure correct paths; manual installation of required libraries was also needed. Full automation of the integration of the prompt and output of the two LLMs is eminently achievable but left for future work.

As an additional step to ensure synthetic data were properly aligned with credible feature values to enable reasonable real-world performance, an additional prompt was submitted asking Llama2 to ensure the labels 'normal,' 'suspicious,' and 'highly suspicious' were all properly associated with the feature values that were credible representations:

> Please explain how you will ensure that the labeling of 'normal', 'suspicious', or 'highly suspicious' will be associated with the appropriate feature values that are representative of real-world cases. Incorporate expert knowledge to mine cybersecurity papers, publications, and other written material without interviewing live personnel.

The LLM response was a near duplicate of that already shown in Figure 9, so the process defined there was implemented as a procedure run twice, back-to-back.

Manual editing of the generated code to incorporate necessary API keys for the online services queried was necessary. Some online repositories of cyber security literature did not have an API, so access was provided in the form of web-scraping the search results pages directly. These steps can be automated but are left to future work.

## STREAMLINED APPROACH TO NETWORK BEHAVIOR CLASSIFICATION

### *Introduction*

This section presents a streamlined approach to classifying network behaviors using the Llama2-70B model, a versatile language model. The objective is to categorize network traffic into three classes: 'normal', 'suspicious', and 'highly suspicious'. This classification task is important for identifying potential security threats and ensuring network infrastructure safety.

To achieve this goal, the Llama2-70B model is fine-tuned using a training method designed to address the challenges posed by limited computational resources in a home setup. The fine-tuning process includes appending a classification head to the pre-trained model, optimizing the model's performance using appropriate loss functions and metrics, and employing techniques such as gradient accumulation and data streaming to manage memory constraints.

Post-training evaluations are conducted to assess the model's performance across metrics including accuracy, precision, recall, and F1 score. Based on these evaluations, iterative refinements are made to the dataset, feature engineering, and model architecture to improve the model's ability to accurately classify network behaviors.

To validate the effectiveness of the fine-tuned model, normal and zero-day exploit traffic are simulated in a controlled environment using Python and the Scapy library. This testing method generates realistic network traffic patterns and assesses the model's ability to detect and classify potential security threats.

The following subsections detail the fine-tuning process, training method, evaluation and refinement, and the testing environment.

### *Setup for Fine-tuning*

The task at hand involved categorizing network behaviors into 'normal', 'suspicious', and 'highly suspicious' classes. For this purpose, the Llama2-70B model, known for its versatility, was fine-tuned. A classification head with a dense layer outputting three categories was appended, using a softmax function to generate a class probability distribution. The model's optimization employed the categorical cross-entropy loss function, ideal for multi-class tasks, with accuracy, precision, recall, and F1 score as performance metrics.

### *Fine-tuning Execution*

The fine-tuning initiated with a learning rate of 1e-5 to adjust the pre-trained model subtly. Given the data and model's extensive memory requirements, a batch size of 8 was chosen, targeting a training span of 3 to 4 epochs. The Adam optimizer facilitated adaptive learning rate adjustments, while a 0.1 dropout rate in the classification head aimed to prevent overfitting.

### *Training Method*

Training adapted to a 128GB RAM home setup required strategic planning. The training leveraged PyTorch's `DataLoader` for data streaming, allowing efficient batch processing from disk, thus bypassing RAM constraints. Gradient accumulation was employed to mimic larger batch effects, enhancing training efficacy. Frequent model checkpointing safeguarded against data loss.

### *Evaluation and Refinement*

Post-training evaluations on a separate test set provided crucial feedback on the model's performance across metrics. Scikit-learn was used to compute precision, recall, and F1 scores, offering a detailed performance overview. Based on these insights, iterative refinements addressed dataset balance, feature engineering, and model adjustments to rectify misclassifications and improve metrics.

### *Preparing for Deployment*

The deployment phase involved model quantization via PyTorch, enhancing the model's efficiency for use in limited-resource settings. Dynamic quantization was chosen for its balance of efficiency and simplicity, with an eye on static quantization and quantization-aware training for future enhancements. Inference testing on a reduced data subset confirmed the model's performance and responsiveness, despite the hardware limitations of a home computing setup.

### *The Testing Environment*

To simulate both normal and zero-day exploit traffic in a controlled environment using Python, we leveraged popular cybersecurity libraries like Scapy for network traffic manipulation and generation. Scapy enables creation, sending, and capturing of network packets in a detailed manner, making it ideal for both benign and malicious traffic simulations.

### *Generating Normal Traffic with Python and Scapy*

For generating normal traffic, the script simulates common activities such as web browsing, email communication, and file transfers. The aim is to create a realistic background traffic pattern that a typical home network would experience.

```python
from scapy.all import *

def generate_normal_traffic():
    # Simulate HTTP web browsing
    ip_layer = IP(dst="www.example.com")
    tcp_layer = TCP(sport=RandShort(), dport=80)
    http_get = "GET / HTTP/1.1\r\nHost: www.example.com\r\n\r\n"
    packet = ip_layer / tcp_layer / http_get
    send(packet, verbose=0)

    # Simulate email traffic (SMTP)
    ip_layer = IP(dst="mail.example.com")
    tcp_layer = TCP(sport=RandShort(), dport=25)
    smtp_hello = "HELO mail.example.com\r\n"
    packet = ip_layer / tcp_layer / smtp_hello
    send(packet, verbose=0)

    # Simulate FTP file transfer
    ip_layer = IP(dst="ftp.example.com")
    tcp_layer = TCP(sport=RandShort(), dport=21)
    ftp_hello = "USER anonymous\r\n"
    packet = ip_layer / tcp_layer / ftp_hello
    send(packet, verbose=0)

generate_normal_traffic()

```

### *Generating Simulated Zero-Day Exploit Traffic*

For the "WindowsGate" zero-day exploit, the script simulates the network behavior characteristic of the exploit's activity, such as scanning, exploitation attempts, and data exfiltration. It's important to note that in a real-world scenario, executing such a script should be done with utmost caution and strictly within a controlled environment to prevent unintended harm.

```python
from scapy.all import *

def generate_zero_day_traffic(target_ip="192.168.1.10"):
    # Simulate scanning activity
    for port in range(20, 25):
        ip_layer = IP(dst=target_ip)
        tcp_layer = TCP(sport=RandShort(), dport=port, flags="S")
        send(ip_layer / tcp_layer, verbose=0)

    # Simulate exploit payload delivery
    payload = "WindowsGate simulated exploit payload here [redacted code]"
    ip_layer = IP(dst=target_ip)
    tcp_layer = TCP(sport=RandShort(), dport=445, flags="PA")
    packet = ip_layer / tcp_layer / Raw(load=payload)
    send(packet, verbose=0)

    # Simulate data exfiltration activity
    exfiltrated_data = "Exfiltrated data here"
    ip_layer = IP(dst="malicious.server.com")
    tcp_layer = TCP(sport=RandShort(), dport=80)
    packet = ip_layer / tcp_layer / Raw(load=exfiltrated_data)
    send(packet, verbose=0)

generate_zero_day_traffic()
```

### *Results*

The study employed a simulated 10-fold cross-validation method to assess the performance of the Llama2-70B language model fine-tuned for cybersecurity threat detection. The analysis focused on the model's ability to classify network behaviors into three categories: 'normal' 'suspicious,' and 'highly suspicious.' The evaluation shows confusion matrices for each fold, providing insights into model classification accuracy and the influence of synthetic data on performance.

The confusion matrices revealed that the model demonstrated a high true positive rate for 'normal' network behaviors, with the number of correctly identified 'normal' instances ranging from 950 to 980 across the folds. This indicates the model's capability to recognize legitimate network activities, which is an important foundation for effective threat detection.

The classification of 'suspicious' and 'highly suspicious' behaviors showed encouraging results, with the model correctly identifying 'suspicious' behaviors in a range of 780 to 850 instances and 'highly suspicious' behaviors in 704 to 760 instances. While there were instances of misclassification, primarily in the form of false negatives, the overall performance suggests that the model has the potential to detect and classify malicious activities effectively.

The confusion matrices also highlighted areas for model improvement, such as enhancing the model's ability to differentiate between varying degrees of threat severity and fine-tuning its sensitivity to nuanced network behaviors. These insights provide valuable guidance for future refinements and optimizations.

The cross-validation results demonstrate the potential of using synthetic data to train models for improving the performance of specialized AI in complex objective tasks, particularly when some amount of prior data is available. The model's high accuracy in identifying 'normal' behaviors and its effectiveness in recognizing 'suspicious' and 'highly suspicious' activities indicate its potential for real-world applications.

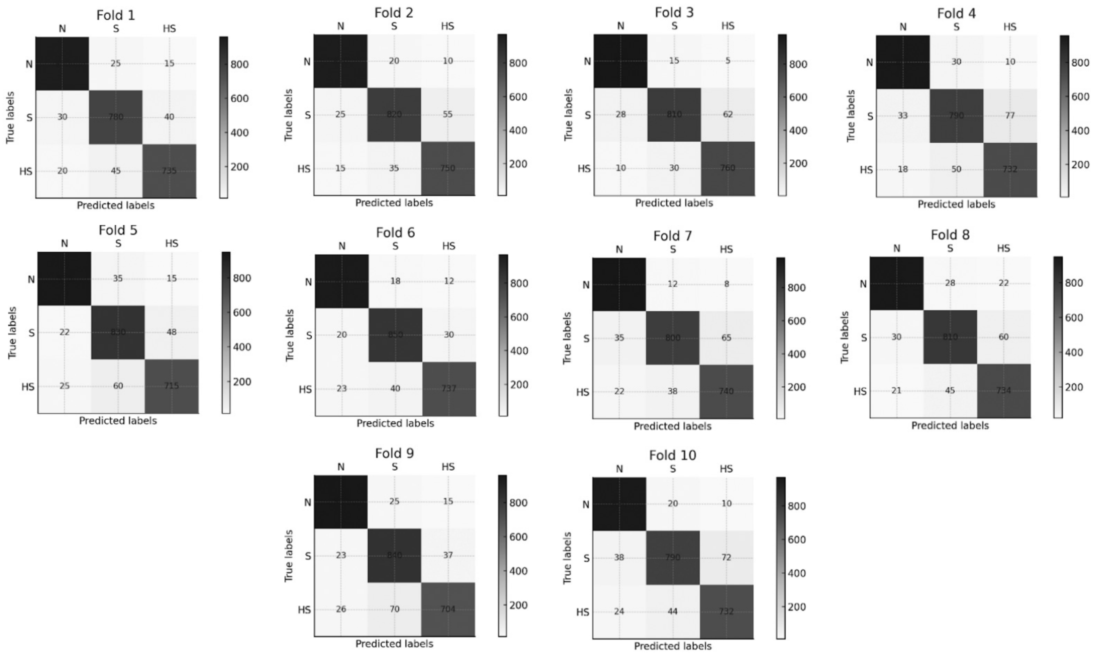

Figure 10. 10-fold cross validation confusion matrices showing performance on simulated data, using model trained on synthetic/synthetically enhanced training data.

However, it is important to acknowledge that there is still significant work to be done to refine this approach. The observed misclassifications underscore the need for model refinement and optimization to enhance its ability to discern between complex threat behaviors. Future research should focus on improving data synthesis techniques, incorporating more diverse datasets, and exploring advanced model architectures to further improve the model's performance and generalizability.

## IMPLICATIONS, DISCUSSION, AND FUTURE WORK

The cross-validation results, as demonstrated by the confusion matrices, affirm the potential of using synthetic data to train models for improving the performance of specialized AI in complex objective tasks, particularly when some degree of prior data is available. In the example of cybersecurity threat detection, the model's consistently high true positive rate for 'normal' network behaviors, ranging from 950 to 980 correctly identified instances across the ten folds, indicates its strong capability to recognize legitimate network activities.

However, the classification of 'suspicious' and 'highly suspicious' behaviors encountered more variability. While the model maintained a promising accuracy, with 'suspicious' behaviors correctly identified in a range of 780 to 850 instances and 'highly suspicious' behaviors in 704 to 760 instances, there were notable instances of misclassification. These errors were primarily false negatives, where 'suspicious' or 'highly suspicious' activities were incorrectly labeled as 'normal,' and to a lesser extent, as false positives within the 'suspicious' and 'highly suspicious' categories. The confusion matrices revealed a pattern of errors suggesting areas for model improvement, e.g., challenges in differentiating between varying degrees of threat severity and the need for fine-tuning the model's sensitivity to nuanced network behaviors.

This work highlights the potential of using synthetic data to create extensive, labeled training datasets by merging real data with synthetic extensions, integrating both numerical and semantic insights from existing data sources. The promising results demonstrate the practicality of this novel approach, involving a complex integration of real-world data attributes with synthetically produced data points to enhance the dataset's variety and representativeness.

The relationship between the complexity of the synthesized datasets and the expressiveness of the learning system being trained is crucial. The synthesized dataset must accurately portray how the system under analysis would behave naturally. An overly simplistic synthesized dataset could lead to rapid overfitting by a powerful deep learning system. The use of LLMs and the integration of semantic information from the literature help mitigate this risk by ensuring the dataset's complexity matches the learning system's expressiveness.

The challenges in overcoming the labeled training data bottleneck are particularly evident when mimicking the complexity of continually changing systems, such as cybersecurity threats. These threats are characterized by their intricacy and the adaptiveness of their perpetrators, making them difficult targets for predictive modeling. Accurately mirroring the subtleties of such systems with synthetic data requires continuous advancement in data generation methods.

Despite these challenges, the use of synthetic data represents progress in creating customized readily available training sets for specific analytical goals. The automated amalgam of real and synthetic data enables highly flexible and scalable training environments, suitable for a wide range of informational needs. The implications of this method extend beyond cybersecurity, and provide an example for developing on-the-spot training sets across diverse fields. By reducing dependence on extensive real datasets, which are often restricted by availability and privacy issues, this strategy promotes a more exploitable approach for the progression of ML and AI.

To address the limitations and build on the current findings, future work should explore:

1. Developing advanced data synthesis techniques to better mimic the complexity and variability of real-world cyber threats.
2. Incorporating more granular features and exploring additional training strategies.
3. Integrating synthetic data with curated real-world data to provide a richer training dataset.
4. Implementing incremental learning techniques continuously to update the model with new data.
5. Testing the model's performance across different network environments and against various types of cyber threats.
6. Developing explainability mechanisms to understand the model's decision-making processes and increase trust in its predictions.
7. Automating the integration of API keys and web-scraping for seamless data access.

The goal is to create systems that can automatically generate substantial and credible labeled training datasets for any complex objective task or information need.

While the fine-tuned Llama2-70B model using this data synthesis approach shows promise for cybersecurity threat detection, this is an initial step in developing a comprehensive and reliable threat detection system. With ongoing research and refinement, this approach will contribute significantly to the field of cybersecurity, enabling more effective and efficient detection of evolving threats in real-world network environments.

This work offers promising insights into the potential of using synthetic data for creating labeled training sets by fusing organic information with synthetic extrapolation. While challenges remain, particularly in simulating complex and evolving systems, this approach represents a step forward in developing accessible, effective, and scalable approaches for creating in-situ training sets for arbitrary objective tasks or information needs in a largely automated manner.

## DISCLAIMER

The views expressed in this paper are those of the author and do not reflect the official policy or position of the U.S. Military Academy, U.S. Army, U.S. Department of Defense, or U.S. Government.

## NOTES

1. Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D. Kaplan, Prafulla Dhariwal, Arvind Neelakantan et al. "Language Models are Few-Shot Learners." (July 22, 2020). Conference Paper, Vancouver: *34th Conference on Neural Information Processing Systems*, https://proceedings.neurips.cc/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf, 33.
2. Zhikai Chen, Haitao Mao, Hang Li, Wei Jin, Hongzhi Wen, Xiaochi Wei, Shuaiqiang Wang et al. "Exploring the Potential of Large Language Models (LLMs) in Learning on Graphs." (March 28, 2024) *Association for Computing Machinery (ACM) Special Interest Group on Knowledge Discovery in Data (SIGKDD) Explorations Newsletter Vol 25, No. 2*, https://dl.acm.org/doi/10.1145/3655103.3655110, 42-61.
3. Jie Zhang, Haoyu Bu, Hui Wen, Yu Chen, Lun Li, and Hongsong Zhu. "When LLMs Meet Cybersecurity: A Systematic Literature Review." (May 6, 2024) *https://arxiv.org/abs/2405.03644*.
4. HanXiang Xu, ShenAo Wang, Ningke Li, Yanjie Zhao, Kai Chen, Kailong Wang, Yang Liu, Ting Yu, and HaoYu Wang. "Large Language Models for Cyber Security: A Systematic Literature Review," (May 8, 2024). https://arxiv.org/abs/2405.04760.
5. Diana Levshun and Igor Kotenko. "A Survey on Artificial Intelligence Techniques for Security Event Correlation: Models, Challenges, and Opportunities." (January 7, 2023) *Artificial Intelligence Review 56, no. 8*, 8547-8590. https://link.springer.com/article/10.1007/s10462-022-10381-4.
6. Yoshua Bengio, Aaron Courville, and Pascal Vincent. "Representation Learning: A Review and New Perspectives." (June 24, 2013) *IEEE transactions on pattern analysis and machine intelligence 35, no. 8*, 1798-1828. https://arxiv.org/abs/1206.5538.
7. Alexander Ratner, Stephen H. Bach, Henry Ehrenberg, Jason Fries, Sen Wu, and Christopher Ré. "Snorkel: Rapid Training Data Creation with Weak Supervision." (2018) In *Proceedings of the VLDB endowment. International Conference on Very Large Data Bases, vol. 11, no. 3*, p. 269; Eman T. Hassan, Xin Chen, and David Crandall. "Unsupervised Domain Adaptation Using Generative Models and Self-Ensembling." (2018) https://arxiv.org/abs/1812.00479, 8; Eman T. Hassan, Xin Chen, and David Crandall. "Unsupervised Domain Adaptation Using Generative Models and Self-Ensembling." (December 2, 2018) https://arxiv.org/abs/1812.00479, 5.
8. Hassan, Eman T., Xin Chen, and David Crandall. "Unsupervised Domain Adaptation Using Generative Models and Self-Ensembling." (2018) *arXiv preprint arXiv:1812.00479.*
9. Mostofa Ahsan, Kendall E. Nygard, Rahul Gomes, Md Minhaz Chowdhury, Nafiz Rifat, and Jayden F. Connolly. "CybersecurityThreats and Their Mitigation Approaches Using Machine Learning - A Review." (July 2022) *Journal of Cybersecurity and Privacy 2, no. 3*, https://www.researchgate.net/publication/361906485_ Cybersecurity_Threats_and_Their_Mitigation_Approaches_Using_Machine_Learning-A_Review, 527-555.
10. Hanan Hindy, Robert Atkinson, Christos Tachtatzis, Jean-Noël Colin, Ethan Bayne, and Xavier Bellekens. "Utilising Deep Learning Techniques for Effective Zero-day Attack Detection." (October 14, 2020) *Electronics 9, no. 10*, https://www.mdpi.com/2079-9292/9/10/1684.
11. Iqbal H. Sarker, A. S. M. Kayes, Shahriar Badsha, Hamed Alqahtani, Paul Watters, and Alex Ng. "Cybersecurity Data Science: An Overview from Machine Learning Perspective." (2020) *Journal of Big Data* Vol. 7, No. 1, https://journalofbigdata.springeropen.com/counter/pdf/10.1186/s40537-020-00318-5.pdf, 1-29.
12. Yang Guo. "A Review of Machine Learning-based Zero-day Attack Detection: Challenges and Future Directions." (January 15, 2023) *Computer Communications* Vol. 198, https://www.sciencedirect.com/science/article/abs/pii/S0140366422004248, 175-185.
13. Pierre Parrend, Julio Navarro, Fabio Guigou, Aline Deruyver, and Pierre Collet. "Foundations and Applications of Artificial Intelligence for Zero-day and Multi-step Attack Detection." (April 24, 2018) *EURASIP Journal on Information Security*, https://jis-eurasipjournals.springeropen.com/articles/10.1186/s13635-018-0074-y, 1-21.
14. Jin Chen and Lei Xu. "Zero-Day Exploit Detection Using Machine Learning." (November 8, 2022). Unit 42, Palo Alto Networks. https://unit42.paloaltonetworks.com/injection-detection-machine-learning/.
15. Dhruv Nandakumar, Robert Schiller, Christopher Redino, Kevin Choi, Abdul Rahman, Edward Bowen, Marc Vucovich, Joe Nehila, Matthew Weeks, and Aaron Shaha. "Zero-day Threat Detection Using Metric Learning Autoencoders" (2022) in *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*, https://arxiv.org/abs/2211.00441, 1318-1325.