AnomaLLMy: Open-Source Anomaly Detection for Small Business Networks on a Budget

pet6r

University of Arizona

CYBV 498: Senior Capstone – Defense and Forensics

Jordan Vanhoy

April 20, 2025

Abstract

Small and medium-sized businesses (SMBs) face significant cybersecurity challenges due to resource constraints and the complexity of existing security solutions, leaving them vulnerable to evolving cyber threats. This paper presents AnomaLLMy, an open-source framework designed to provide accessible network anomaly detection and AI-driven analysis specifically tailored for budget-conscious SMB environments. The framework employs a baseline comparison approach, utilizing Python scripts to establish normal network behavior profiles (including trusted devices/OUIs and allowed protocols) and monitor traffic via SPAN port mirroring for deviations. Detected anomalies are structured into CSV files, leveraging blank rows to delineate distinct events, and subsequently processed by Large Language Models (LLMs) accessed locally via Ollama. Guided by engineered system prompts, the LLMs provides contextual analysis and interpretation of flagged anomalies, aiming to accelerate understanding for users with limited IT expertise. Supplementary Jupyter Notebooks simplify performance metric comparison across different LLMs and estimate token usage to aid in resource planning and cost management. AnomaLLMy demonstrates the viability of using open-source tools and accessible LLMs to enhance SMB security posture, while emphasizing the necessity of modular design, user commitment to baseline maintenance, and required human oversight for validating AI inference.

Keywords: cybersecurity, network anomaly detection, small and medium-sized businesses (SMBs), Large Language Models (LLMs), open-source software, prompt engineering, baseline analysis, pickle files, Jupyter Notebook

Table of Contents

Open-Source Anomaly Detection for Small Business Networks on a Budget

In today's digital world, small businesses increasingly rely on networks for everything from processing payments to managing customer information. This reliance makes them vulnerable to cyberattacks. A single breach to a small local coffee shop could expose customer payment data and disrupt online ordering, potentially leading to financial losses and reputational damage. Small and medium-sized businesses (SMBs) face significant challenges in implementing proper security measures due to their limited resources and the complexity of existing solutions (Sreejith et al., 2022, p. 1). Recognizing these challenges, AnomaLLMy offers a viable open-source solution for small and medium businesses to defend their cybersecurity posture in the face of the exponential rise of threats, even with minimal IT expertise.

With the popularity of artificial intelligence (AI) and the new technologies that have emerged from it throughout recent years, researchers have documented remarkable results in various domains that have improved humanity for the better. However, these same technologies can be weaponized by threat actors to conduct malicious activities. The interconnected nature of our digital world largely means a single vulnerability like the flap of a butterfly's wings can ultimately cause a tornado. Analogous to the "butterfly effect", when one part of a system collapses, the whole system will collapse with large effects. "The impact of potential cyber threats has been extended from malicious uses of AI technologies to enable larger-scale and more powerful attacks" (Kaloudi & Li, 2021, p. 3). This dark side of AI will only grow larger as new automation technologies and frameworks evolve to compete with expensive solutions today. While the good and bad hackers try to one-up each other, SMBs will ultimately find themselves outmatched in this technological arms race, lacking the resources, expertise, and infrastructure to implement the systems necessary to protect against these growing threats.

Literature Review

Small Business Information Security: The Fundamentals (NISTIR 7621 Revision 1)

This document, NISTIR 7621 Revision 1, titled, "Small Business Information Security: The Fundamentals," is a reference guide developed by the National institute of Standards and Technology (NIST) to help small businesses understand and implement basic cybersecurity practices. The document presents these fundamentals in non-technical language. Designed for small businesses (both for-profit and non-profit) that often have limited information security resources, the report details security incident outcomes and emphasizes the need for information security risk management (NIST, 2016, p. 1). The publication is structured around the Framework for Improving Critical Infrastructure Cybersecurity (CSF) but without its abundant and heavy terminology.

It uses the Framework's five core processes: Identify, Protect, Detect, Respond, and Recover to organize its guidance. Background and key concepts are covered which include the impact of business compromise using the CIA triad (Confidentiality, Integrity, Availability) (NIST, 2016, p. 2). It provides steps for safeguarding information that the five functions of the CSF contain like access control, employee training, installing firewalls, data backups, and incident response planning. The last section offers everyday security practices for employees to follow such as being cautious with emails, using strong passwords, and separating personal and business devices. The end of the document contains appendices with a glossary, references, information about the CSF, risk analysis worksheets, and sample security policy statements that all businesses can benefit from.

Version's 2024 Data Breach Investigations Report

This report created by Verizon marked its 17th edition with the aim of informing various threat actor types, their tactics, and chosen targets for their cybercrime. The report analyzes a dataset of 30,458 real-world security incidents, including a record high of 10,626 confirmed data breaches spanning 94 countries, collected with the help of contributors and Verizon's own Threat Research Advisory Center (VTRAC) (Verizon, 2024, p. 5). The DBIR (Data Breach Investigations Report) investigates the pathways to breaches, seeking to identify the most likely action and vector groupings that lead to successful intrusions given the current threat landscape. Key findings in the 2024 DBIR highlight a significant increase in attacks exploiting vulnerabilities as the critical path to initiate a breach, almost tripling from the previous year, largely driven by the impact of zero-day vulnerabilities like those affecting MOVEit.

The report also notes that the human element remains a significant factor, being a component in 68% of breaches (Verizon, 2024, p. 8). Additionally, the influence of third-party connections including software supply chain issues has grown to account for 15% of breaches (Verizon, 2024, p. 8). The DBIR examines threat actors, their motives, and the assets and attributes involved in security incidents, providing a comprehensive overview of the current cyber threat landscape. The report seeks to offer useful insights and metrics to help organizations enhance their cybersecurity posture.

A Context Aware Anomaly Behavior Analysis Methodology for Building Automation Systems

This dissertation paper by Zhiwen Pan informs about the increasing frequency of Internet of Things (IoT) fueled development of Building Automation Systems (BAS) and Smart Home Systems (SHS) designed to improve convenience and efficiency. However, this interconnectedness can expose these systems to serious cybersecurity threats. Traditional IP network security mechanisms are often inadequate for BAS and SHS due to the limited resources of embedded devices and the use of specialized protocols (Pan, 2017, p. 10). Existing intrusion detection techniques include signature-based (misuse detection) and anomaly-based methods; while misuse detection struggles with unique attacks, anomaly detection faces challenges such as high false positive rates (Pan, 2017, p. 23). To address these limitations, the dissertation provides the integration of context awareness with anomaly behavior analysis (ABA). This approach, termed Context Aware Anomaly based Behavior Analysis (CAABA), intends to enhance detection accuracy by incorporating heterogeneous information from various sources to better understand the operational context of BAS and SHS, moving beyond traditional network protocol analysis (Pan, 2017, p. 26).

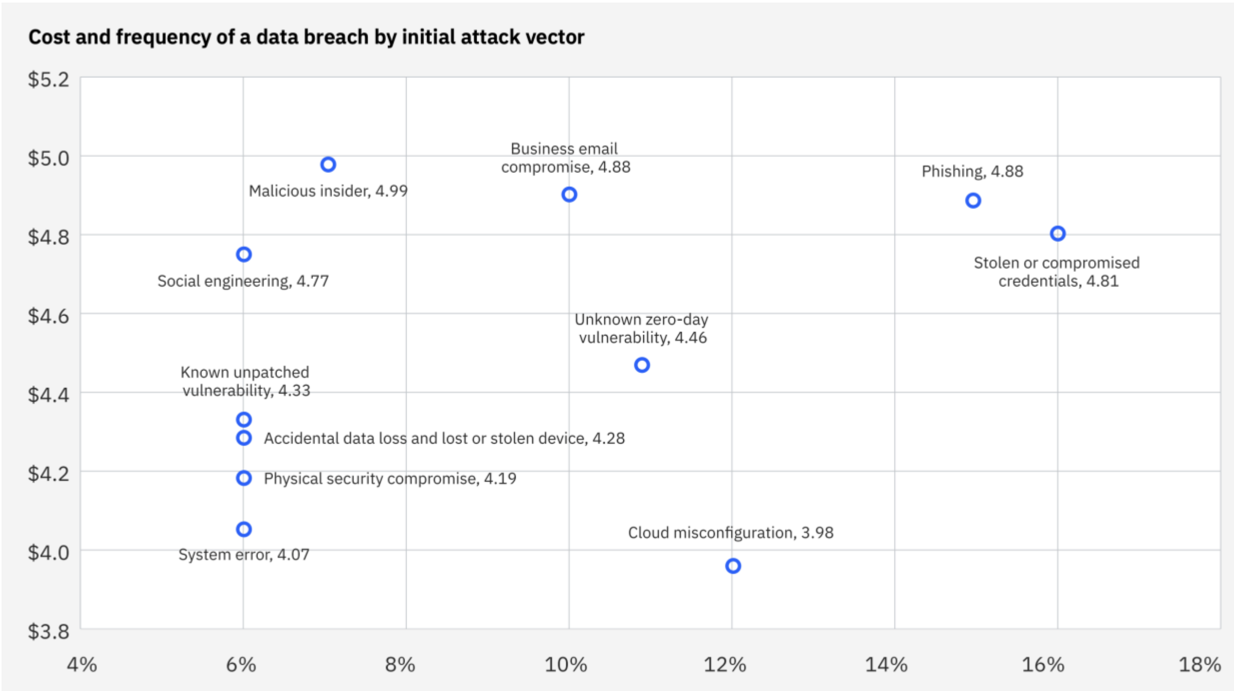
SMB Cybersecurity Challenges

This vulnerability of small and medium-sized businesses stems from their tight resources and unique operational challenges, creating a major disadvantage in the cybersecurity landscape. Unlike large corporations, local businesses cannot afford a dedicated security team. These businesses frequently rely on just one or two overworked IT staff members that handle everything from computer setups to website management. Consequently, cybercriminals target small businesses because they frequently lack strong information security measures (National Institute of Standards and Technology, 2016). While larger organizations also face cybersecurity concerns, the nature of those threats often differs due to their scale and resources. For instance, Verizon's (2024) Data Breach Investigations Report (DBIR) found that, “there is no substantial difference between large organizations (55%) and small organizations (47%) in the “Basic Web Application Attacks" pattern” (p. 43). This highlights that web application vulnerabilities can impact businesses of all sizes equally, though SMBs encounter greater difficulty in addressing these vulnerabilities due to their limited technical capacity.

The persistent threat of unpatched vulnerabilities remains a critical issue for SMBs. Even with good intentions, many SMBs struggle to consistently implement security practices, potentially leading to neglected default configurations, infrequent software updates, inadequate cybersecurity training, or a lack of proactive threat monitoring. The cumulative effect of these gaps creates an environment where attackers can easily exploit vulnerabilities. The significant financial impact of such exploits is highlighted in IBM Security's 2024 Cost of a Data Breach Report, which found that breaches originating from the exploitation of a known, unpatched vulnerability cost businesses an average of $4.33 million (IBM Security, 2024, p. 13), as illustrated in **Figure 1**. Recognizing this reality and understanding the potential for high damages offers an opportunity for SMBs to use anomaly detection to identify and respond to threats before they escalate. By establishing a baseline of normal activity, businesses can be alerted when deviations occur, allowing them to investigate further into suspicious network behavior.

Figure 1

Cost and Frequency of Data Breaches by Initial Attack Vector



Note. Measured in USD millions; percentage of all breaches. Adapted from IBM Security (2024), Cost of a Data Breach Report 2024, p. 13. Copyright 2024 by IBM.

This network traffic analysis can be better understood when compared to a traffic highway system. You would expect a certain high flow of cars during rush hour and fewer cars late at night. A sudden traffic jam or a car speeding in the wrong direction would immediately raise suspicion. Similarly, network traffic is the flow of data between internal devices on a network and the outside world. By establishing a baseline, which acts as a record of normal network activity, it includes the known devices, the type of traffic, and the destinations it's going to—you create a map of your network's typical behavior. This baseline then serves as a reference point for detecting anomalies, which are deviations from the predefined norm. As Pan (2017) explains, "Anomaly Based Intrusion Detection is a security mechanism that uses [a] baseline model to describe the normal behaviors of a system, so that malicious behaviors occurred in a system can be detected by comparing the observed behavior to the baseline model" (p. 9).

Interpreting these deviations from an established baseline requires contextual understanding and pattern recognition that can be difficult for conventional rule-based systems. This creates an opportunity for advancements like Large Language Models (LLMs) in network security analysis. As Kotb et al. (2025) note, "Classical Machine Learning (ML) algorithms play a crucial role in extracting meaningful patterns and insights from data" (p. 8), representing an earlier stage in the evolution of AI techniques that have paved the way for LLMs. LLMs utilize vast datasets of text to develop a broader understanding of language and relationships. By predicting subsequent words or tokens, they can analyze network communications, correlate traffic patterns, and infer potential threat context based on user prompts—ultimately assisting security teams in distinguishing between benign deviations and genuine risks.

The “LLM” In AnomaLLMy

The AnomaLLMy framework was developed to assist in anomaly detection and provide contextual reasoning. The initial development of AnomaLLMy was aimed towards Industrial Control Systems (ICS) and Operational Technology (OT) environments deliberately focusing on local and secure processing by using Ollama’s library of models. Ollama enables users to download and run LLMs on their own systems, minimizing reliance on external cloud services while ensuring data privacy. This architecture is particularly valuable for SMBs, offering greater control over sensitive network data and reducing dependence on costly cloud infrastructure. The availability of resource-efficient LLMs through Ollama allows deployment on readily accessible and low-end systems, making security solutions attainable for SMBs with limited resources.

Beyond leveraging the existing models from industry leaders like IBM and Meta, AnomaLLMy's integration with Ollama allows broader access to specialized models through Hugging Face. This secondary platform enables users to upload and share custom-trained and fine-tuned models, providing opportunities for enhanced contextual reporting tailored to specific industries. For example, cybersecurity models available on Hugging Face demonstrate this potential, such as the *Lily-Cybersecurity-7B-v0.2-GGUF* model provided by QuantFactory (2024). This model is specialized in network analysis, security operations, and monitoring, which can enhance AnomaLLMy's responses. Hugging Face also provides quantized versions of leading models, which essentially means that they have been optimized to reduce computational demands without severely sacrificing accuracy, further providing the accessibility for low resource environments. This combination of flexible model access and optimized performance identifies AnomaLLMy's potential for addressing cybersecurity challenges within SMB settings.

Prompt Engineering

Next, internal prompts direct a model to produce an intended type of analysis or response. Specifically, **Figure 2** provides an example of an engineered prompt.

Figure 2

Internal LLM System Prompt

```
You are a highly skilled cybersecurity analyst specializing in
identifying and reporting anomalous connections within a small
business network environment.

Your task is to analyze the following network connection data
that has been flagged as anomalous by our baseline detection
system and provide detailed insights. Your analysis will be
included in security reports and reviewed by human experts.

Task Overview: Analyze the following connection data from our
network environment. Each group of data represents a connection
conversation that was flagged as anomalous because:
```

```
1. It contains an unknown device (OUI not in our baseline), OR
2. It uses a protocol not in our allowed protocol baseline, OR
3. Both of the above reasons

Port notes: EPH is an ephemeral port (port > 1024) used by
clients. Pay special attention to the protocols, outgoing and
incoming ports, and manufacturers (MFGs).

Response Requirements for each connection group:

1. Device Identification: Identify and describe the devices
involved based on their manufacturer (MFG) names and MAC
addresses.
2. Communication Details: Specify the protocols used, IP
addresses, and ports (both source and destination). Provide
information about the purpose of the ports if known (e.g., 443
for HTTPS).
3. Traffic Volume: Analyze the CNT field, which represents packet
counts for each connection.
4. Risk Assessment: Evaluate the risk level (Low, Medium, High,
Critical) of these anomalous connections. Explain your reasoning
based on the protocols, devices, and communication patterns.
5. Recommendations: Suggest specific actions for security
personnel (block, monitor, investigate, or allow).

Format your response using clear headings and bullet points for
readability. Security personnel will use
your analysis to make decisions about these anomalous
connections.
```

Note. Large Language Model prompt for network connection analysis taken from AnomaLLMy's codebase. Own work.

This unique process provides the LLM explicit roles and characteristics that will be presented in the analysis results. "The strength of this approach lies in its ability to adapt to different tasks through simple modifications to prompt statements, eliminating the need for retraining the entire model" (Xu et al., 2025, p. 3). In the provided prompt example, elements like the "Task Overview" specify the input data (anomalous connection details), while other sections dictate the analysis and output, supporting the prompt's directive role. However, prompt effectiveness can vary considerably across different LLM models. A prompt optimized for one model might not yield the desired results with another. For example, the mentioned model *Lily-Cybersecurity-7B-v0.2-GGUF* from Hugging Face, already fine-tuned for cybersecurity tasks may require less detailed prompting than a general-purpose model needing more explicit context. Therefore, the modular nature of prompt engineering is necessary, pushing users to refine prompts and pair them effectively with suitable models to achieve optimal responses.

Core Workflow

The core operational workflow of AnomaLLMy follows a logical progression designed to transform raw network data into useful insights. It begins with Baseline Creation, where the unique profile of normal network behavior is established for the specific environment. Following this setup, the Anomaly Detection phase involves actively monitoring network traffic via the designated SPAN port and comparing it against the baseline to identify deviations. Once potential anomalies are flagged, they are passed to the LLM Analysis stage, where a chosen LLM interprets the data and provides contextual explanations. The workflow concludes in Report Generation, which structures the AI's findings into a readable document format for review and decision-making.

Baseline Creation

To construct the detailed baseline map described previously with our highway traffic example, AnomaLLMy focuses on two key technical steps: identifying trusted devices and listing approved protocol communication types. For the first step, device identification is required to inventory all network equipment such as computers, servers, printers, employee laptops, and so on. A vital piece of information gathered during this inventory is the

Organizationally Unique Identifier (OUI), which is derived from the first part of each device's unique MAC address. Knowing the OUI is important because it tells us the device's manufacturer (like Dell, Apple, Netgear, etc.). Capturing this manufacturer information via the OUI for all known, trusted assets adds a layer of verification to the baseline, helping AnomaLLMy confirm that the devices connecting are indeed the ones expected. **Figures 3** and **4** provide snippets from their corresponding Python scripts, illustrating how this baseline information (such as trusted OUIs and allowed protocols) is defined in the code.

Figure 3

Known OUIs to Manufacturers

```
KNOWN_OUI_MFG_DATA = {
    "B827EB": "Raspberry Pi Foundation",      # B8:27:EB
    "005056": "VMware, Inc.",                 # 00:50:56
    "001C42": "Cisco Systems, Inc",           # 00:1C:42
    "00155D": "Microsoft Corporation",        # 00:15:5D
    "F875A4": "Dell Inc.",                    # F8:75:A4
    "A0D3C1": "Apple, Inc.",                  # A0:D3:C1
    "001517": "Intel Corporation",            # 00:15:17
}
```

Note. Declaring a dictionary of known OUI manufacturer data in Python, found in the `create_oui_baseline.py` module from AnomaLLMy’s codebase. Own work.

Figure 4

Allowed Protocols

```
ALLOWED_PROTOCOLS = {
    "TCP",
    "UDP",
    "ICMP",                         # Covers ICMP for IPv4
    "IPV6-ICMP",                    # Covers ICMP for IPv6
    "ARP",
}
```

Note. Declaring a dictionary of allowed protocols in Python, found in the `create_protocol_baseline.py` module from AnomaLLMy's codebase. Own work.

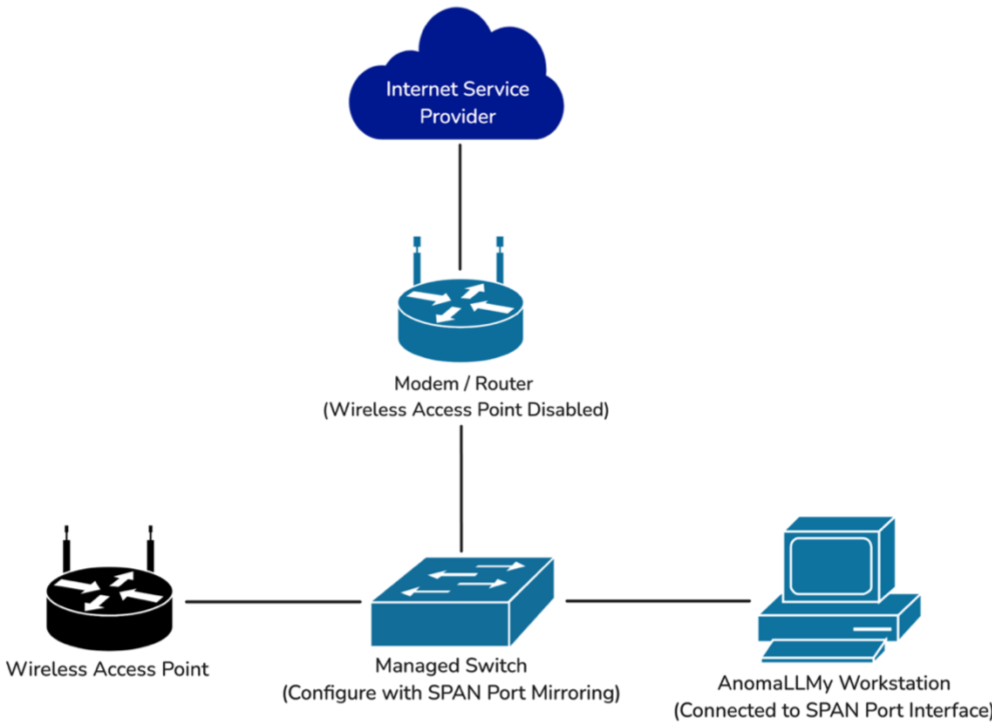
Once the Python scripts are updated with the business's specific baseline information, running them generates the initial baseline data structures. These structures are saved as 'pickle' files, allowing the AnomaLLMy network analysis process to quickly the established baseline during runtime. This loaded baseline is then used to distinguish between normal and potentially abnormal communication as network traffic is analyzed. If communication is flagged as abnormal, an additional comprehensive OUI baseline is consulted to provide further context about the devices involved. This comprehensive baseline utilizes data retrieved from Wireshark's regularly updated 'manuf' file, which contains an extensive list pairing MAC address prefixes (OUIs) with known manufacturer names. It functions as a device manufacturer lookup, using the MAC addresses gathered from the flagged network traffic.

Anomaly Detection

Effective anomaly detection centers on AnomaLLMy having clear visibility into the network's traffic. To achieve this, a specific network configuration is recommended, as shown in **Figure 5**.

Figure 5

Network Configuration for AnomaLLMy



Note. Diagram illustrating the network configuration designed to provide the AnomaLLMy workstation visibility into both wired and wireless network traffic via SPAN port mirroring. Own work.

The core requirement is that the computer running AnomaLLMy must be able to see all the data flowing through the small business network. This is accomplished by using a “Managed Switch”, a central hub for network connections. This type of switch includes a feature called “SPAN Port Mirroring”. When configured, it's like telling the switch: "Copy all the data passing through you and send that copy to this specific port." The AnomaLLMy Workstation connects directly to this designated SPAN ethernet port, allowing it to passively monitor all network conversations without getting in the way.

For full monitoring, the diagram also recommends disabling the Wi-Fi functionality on the main “Modem/Router” provided by the Internet Service Provider. Instead, a separate “Wireless Access Point” connects directly to the Managed Switch. This ensures that traffic from wireless devices (laptops, phones, etc.) also flows through the switch and is mirrored to the AnomaLLMy Workstation. Capturing this wireless communication is critical, as many security incidents can involve mobile or wirelessly connected devices. This setup provides AnomaLLMy with the complete view of both wired and wireless network traffic, creating the footing for the anomaly detection processes described next.

To begin the anomaly detection, the `network_anomaly_detector.py` script must be executed. Running this script requires elevated privileges, such as administrative rights on Windows or using the sudo command on Linux or macOS. These higher permissions are necessary because the script needs system-level access to capture raw network traffic directly from the SPAN port. While monitoring live traffic, the script continuously compares observed activity against the established norms loaded from the two previously generated custom baseline pickle files. This comparison is the primary method used to identify any deviations from the business's specifically approved OUIs and protocols.

If this comparison flags a deviation, particularly one involving an unrecognized device OUI, the script then utilizes the third, comprehensive OUI pickle file for lookup purposes. In this step, the script attempts to identify the potential manufacturer by comparing the device's MAC address against the extensive Wireshark OUI data. The script compiles details about the detected deviation, including any identified manufacturer data for unknown devices, into a CSV file. Within this CSV, related network connections for a single anomaly are grouped together, using blank rows to clearly distinguish these distinct events. This structured CSV format serves as the input for the following LLM analysis task, enabling the LLM to process and report on each grouped anomaly context independently.

By default, anomalous CSV files are produced every 10 minutes, although the interval is can be adjusted by using the `-t` parameter. The CSV files act as the bridge between the initial network detection and the AI-driven analysis. The clear separation of anomaly groups within a CSV is essential for allowing the LLM to focus its analysis effectively on each potential incident.

LLM Analysis

The next step in the AnomaLLMy framework utilizes the `anomaly_ollama_analyzer.py` script to interpret the anomalies detected. When this script begins, it first sets itself up by identifying which AI model to use and where to find the anomaly network reports (CSV files) generated by the detector script. It also holds the detailed set of instructions from the system prompt shown in **Figure 2**. The script then locates the most recent CSV file needing analysis, preparing to read the flagged connection details within it. Finally, it ensures an output directory exists where the final analysis results will be stored.

Once the script identifies the correct anomaly report, it reads the information detailing the suspicious network activity. It processes the data by treating each block of rows separated by a blank line as a different conversation events, ensuring each anomaly context is analyzed independently. For every event group, it combines the internal system prompt with the specific details of that anomaly, including MAC addresses, protocols, and any identified manufacturer information. This merging creates a complete request specifically designed for the chosen model. This careful preparation ensures the model receives both the general instructions for its role and the unique context of the specific data requiring analysis.

This complete request is then sent to the specified model using the Ollama interface running on the host machine. The script patiently waits while the model processes this combined information, applying the prompt's instructions to generate its expert assessment and recommendations. Once the model finishes, the script captures the resulting text-based analysis. It then formats this generated output into a structured report, adding metadata like the source file and timestamps, and saves it as a new text file (.txt) in the declared output directory. Additionally, the script logs performance metrics about the LLM's interaction, such as response time to help evaluate the efficiency for review.

Report Generation

The final stage in the AnomaLLMy workflow involves transforming the raw text analysis generated by a LLM into a more presentable and shareable format. This task is handled by the `create_docx_report.py` module designed to create Microsoft Word (.docx) documents. Its primary purpose is to read the slightly unstructured text files produced by the anomaly analyzer script. It then intelligently parses this text to identify key sections like summary details, performance metrics, and the model's specific analysis for each connection group. The goal is to structure these distinct pieces of information logically within a formatted Word document suitable for review and distribution.

Before a Word document can be created, the script must first understand the content of the input text file. It achieves this by reading the entire text file generated by the LLM analyzer script. Using predefined patterns, it searches through the text to locate specific syntax and headings, such as "Source File:", "Analysis Metrics:", and "Connection Group X Analysis". This allows the script to isolate and extract the different components: the header information, the block detailing performance metrics, and the individual analyses provided by the LLM for each anomalous connection group. This parsing step essentially breaks down the raw text into organized pieces of data that can be placed into the report structure.

The report generation script is designed to be run from the command line, offering users flexibility in how it operates. Users can specify the location of the input analysis files and where the final reports should be saved. Additionally, the script includes modes to process the most recent analysis file, all available analysis files within a directory, or a specific file designated by the user. This allows for automated report generation for the latest findings or batch processing of multiple analysis results as needed. Such options provide convenient control over the reporting process depending on the user's immediate requirements.

It is crucial to understand that the generated reports serve as a valuable first draft and will require human review and refinement. LLMs can vary significantly in their formatted responses, phrasing, and overall structure. While this script attempts to parse common patterns, it cannot flawlessly interpret every variation. Therefore, users should always treat the output as a starting point, where review and edits should be made to the document file to improve readability and achieve a fully professional final appearance. This necessity for human oversight resonates with a quote displayed in an image posted by MIT's Computer Science & Artificial Intelligence Laboratory X account (2022), originating from a 1979 IBM slide: "A computer can never be held accountable, therefore a computer must never make a management decision." Consequently, the AI-generated analysis serves as useful input, but the main responsibility for researching further into the findings and directing actions rests on accountable human IT or security team.

Figure 6

Sample Analysis Section from a Generated Report

Connection Group 1 Analysis

Analysis:

Device Identification:
The devices involved in this connection group are both identified as having an unknown manufacturer, denoted by "UNKNOWN" in the SRCMFG and DSTMFG columns. Their MAC addresses are also listed as "UNKOWN," further emphasizing their obscurity.

Communication Details:
The connections made between these devices utilize the TCP protocol. The source IP address, 192.168.68.68, is associated with both connections. Interestingly, the destination MAC addresses are identical for both connections (f0:09:0d:4d:83:00). However, the destination IP addresses differ slightly, which could indicate the existence of multiple devices behind that MAC address or a potential misconfiguration within the network.

Traffic Volume:
The CNT field indicates that each connection has had six and eight packets sent, respectively. While this traffic volume is not particularly alarming, it may be worth investigating further to determine if these connections represent legitimate communication or potentially malicious activities.

Risk Assessment:
Given the lack of information about the devices involved (unknown manufacturers) and their purpose within the network, it is challenging to assess the risk level accurately. However, using the available data, we can see that both devices are communicating using an unknown MAC address with a slightly different IP address as the destination. This anomaly could indicate unauthorized activity or potential security vulnerabilities.

Recommendations:
Due to the identified risks and the lack of clear context about the devices involved, it is recommended to prioritize further investigation into these connections. Specific actions for security personnel may include blocking the traffic temporarily to prevent any potential harm, closely monitoring the devices' behavior, or conducting a thorough examination of the network for any signs of intrusion or misconfiguration.

Note. Example of the structured analysis provided by the LLM for an anomalous connection group, generated according to the requirements defined in the system prompt. Own work.

Metric Analysis & Token Estimation

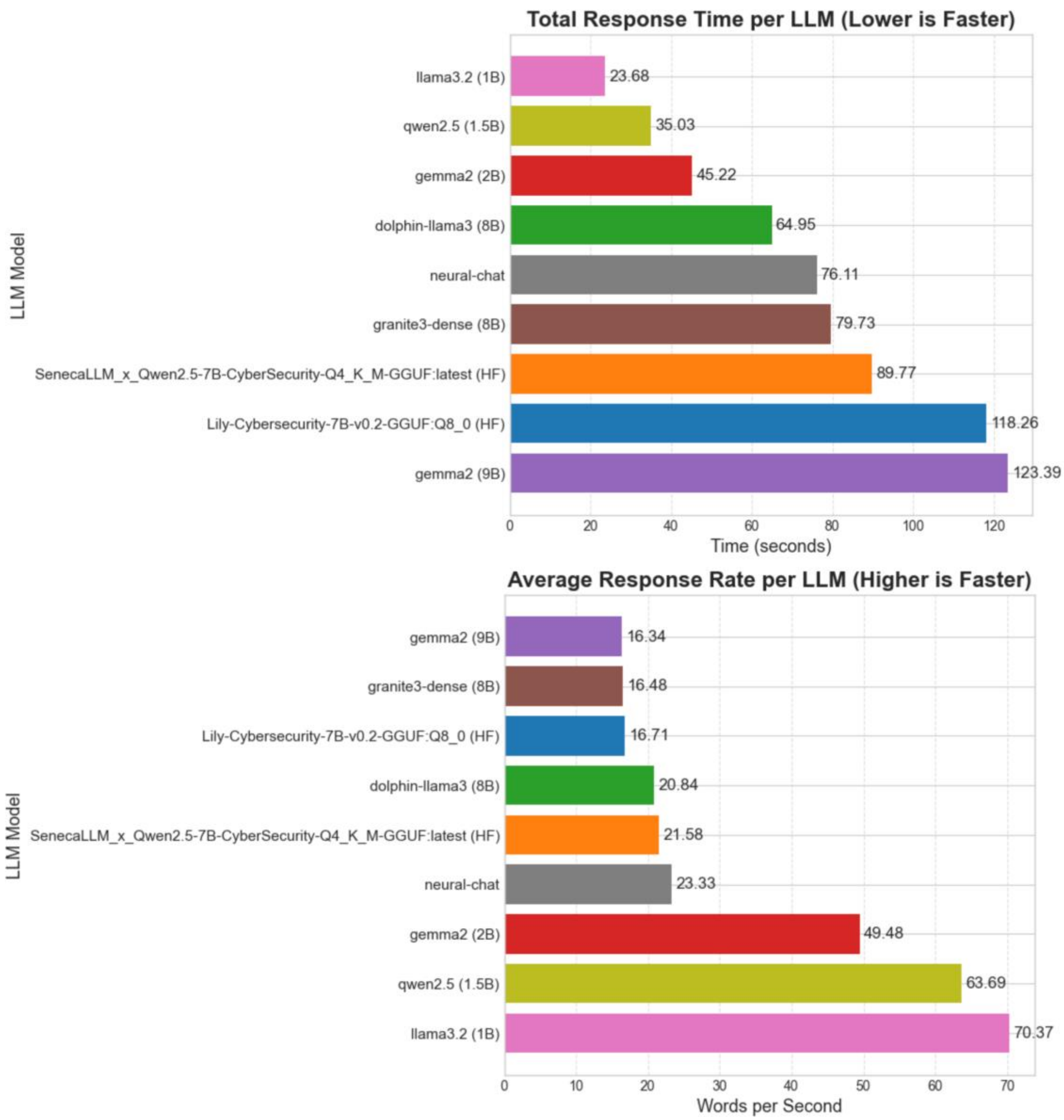
AnomaLLMy provides an exploratory view of models' responses and token input estimations using Jupyter Notebooks. A Jupyter Notebook is an interactive computing environment that supports python code execution, data visualization, and documentation within a single medium. This tool simplifies dynamic experimentation and analysis, as proven by the notebooks that the AnomaLLMy framework provides. The MetricAnalysis notebook examines the text files produced by the LLM analyzer to derive quantitative insights into the models' performance. Its main function is to extract numerical metrics detailing how each LLM responded, going beyond the textual content of the analysis itself.

It searches through the analysis text files found in the designated output directory. Within these files, it looks for specific reported values such as the total time taken for the response, the number of words generated, and measures of vocabulary richness. This extracted data provides objective measures of the LLM's output characteristics for comparison. Once extracted, the notebook organizes these various performance metrics into a structured table using the popular Pandas data analysis library. If multiple analysis files exist for the same AI model from different runs or anomaly files, the script calculates average performance figures for that specific model.

A key outcome of this notebook is the automatic generation of several bar charts using Matplotlib, visualizing each metric across the different LLMs tested. These charts offer a clear, comparative view, enabling users to easily assess differences between models regarding their speed, verbosity, and linguistic complexity. The Matplotlib visuals are helpful for understanding model behavior and selecting the most suitable one for network anomaly detection within a business environment. The results of these metrics are subject to change depending on the hardware resources of the machine running the AnomaLLMy framework.

Figure 7

LLM Performance Metrics: Response Time and Response Rate Comparison



Note. Two plots provided by the MetricAnalysis Jupyter Notebook. The top plot compares the total time (seconds) taken by each LLM to complete the analysis task. The bottom plot compares the average processing speed (words per second) for each LLM, with consistent color coding per model. Own work.

Complementing the performance metrics, the project also includes the TokenCounter Jupyter Notebook for estimating potential AI usage. This notebook focuses specifically on tokens – the fundamental units that LLMs process, which directly relate to usage limits and the potential costs associated with commercial LLM services like OpenAI’s ChatGPT. Understanding token counts is crucial for anticipating resource consumption and planning data submission strategies before running extensive analyses. The notebook loads the same standard instruction prompt used by the actual analyzer (**Figure 2**) and accesses the raw anomaly data from the CSV files generated by the detector. For accurate estimations, it uses the Tiktoken library, reflecting common tokenization methods used by OpenAI’s leading models.

The TokenCounter notebook estimates token usage for processing a given network anomaly file using two methods. The first method calculates the total estimated token count required if the entire CSV file were sent with the instruction prompt in one large request to an LLM. The second method processes the CSV data group-by-group, calculating the separate token count needed for sending the prompt plus each individual anomalous connection group's data. The notebook clearly displays these estimations, including minimum, maximum, and average token counts for the per-group scenario, often alongside data snippets for context. This dual analysis allows users to understand how many tokens a CSV file represents both in its entirety and when sectioned by group analysis, providing practical insights into potential token usage and costs on efficiency. Crucially, the total bulk estimate helps users assess if processing the entire file at once might exceed an LLM's context window limit, which could otherwise lead to processing errors or request failures. This insight informs decisions on whether large anomaly CSVs need to be split into smaller and more manageable files before being sent for analysis.

Although they are optional steps in the workflow, the MetricAnalysis and TokenCounter notebooks are recommended additions to the AnomaLLMy framework. They deliver objective metrics regarding LLM performance and anticipated resource usage. These metrics are instrumental in evaluating the trade-offs between different models concerning speed, resource cost, and output style. Ultimately, these tools provide the necessary data for users to tailor the process effectively, ensuring the chosen model and analysis approach align well with their SMB requirements.

Current Day Solutions vs AnomaLLMy

Many cybersecurity tools, like traditional antivirus software or firewalls, primarily work like security guards with a list of known criminals (or malicious software signatures and blocked websites). They excel at stopping threats that have been seen and documented before. If something matches their "wanted list," they block it. However, they can struggle with brand-new threats or unusual activities that aren't on their list yet. AnomaLLMy takes a different approach, acting more like a security guard who knows the normal routine of a building particularly well. Instead of just looking for known troublemakers, it watches for any activity that deviates significantly from the baseline. This focus on detecting deviations from a learned normal is the core of anomaly detection.

While sophisticated anomaly detection exists in high-end enterprise security systems – such as complex Security Information and Event Management (SIEM) platforms – these often represent the cutting edge of AI application in the field. As Parisi (2019) notes, "With the introduction of AI techniques to the field of NIDS [Network Intrusion Detection Systems], it is now possible to evolve traditional IDS toward more advanced detection solutions, exploiting supervised and unsupervised learning algorithms, as well as reinforcement learning and deep learning" (p. 129). However, leveraging these advanced machine learning approaches typically requires significant investment, dedicated security analysts to interpret the complex models, and extensive configuration, placing such solutions out of reach for many small businesses. AnomaLLMy aims to bridge this gap, utilizing the accessibility of open-source tools and the distinct interpretive power of LLMs, rather than complex statistical learning algorithms, to analyze and contextualize potential anomalies identified against a customized baseline.

Another key difference lies in the output. Where some tools might automatically block traffic based on rigid rules or signatures, and others might flood an administrator with cryptic alerts requiring expert interpretation, AnomaLLMy is designed to use the LLM as an analytical assistant. The goal isn't just to flag an anomaly, but to generate a preliminary human readable explanation of why something was flagged and what its potential implications might be. This generated report is intended to accelerate understanding for an IT generalist or business owner but should be treated as a starting point for human review and decision making, aligning with the principle that accountability requires human judgment, not just automated output.

Conclusions

Initiated seven months ago, the AnomaLLMy project aimed to harness the capabilities of LLMs to address a critical need in today's digital world: the vulnerability of small and medium-sized businesses facing advanced cyber threats with limited resources and expertise. This journey sought to establish that accessible AI models could provide a view at suspicious network events. Witnessing the framework evolve has been surprising; as new models surfaced via open platforms like Hugging Face and became usable through Ollama local interface, it became clear that applying these technologies was apparent. Incorporating these models with careful prompt refinement resulted in tangible anomaly analysis responses– reinforcing the belief that such technology can be applied in resource and technical known constrained environments.

This progress is particularly substantial given the dual nature of AI as it is a powerful tool for humanity that can also be weaponized by threat actors, creating environments where SMBs are at risk. AnomaLLMy was intentionally designed with an open-source and modular philosophy specifically for these businesses. The modularity allows for reframing to unique networks without requiring deep expertise, while the open-source approach removes cost barriers, directly addressing the core challenges SMBs face. Its effectiveness still lies on user engagement in maintaining baselines updated and exploring the framework functionality, but it aims to lower the barrier to entry for proactive network monitoring.

As the development phase concludes, AnomaLLMy transitions from an academic project into a resource for the others to use. The open-source code and methodologies are available (see Appendix A), offering a foundation for others to build upon or adapt to their own networks or scenarios. While my active development under this project structure sunsets here, the hope is that AnomaLLMy stands as a practical demonstration fulfilling the initial thesis: offering a feasible open-source solution that empowers small and medium businesses to bolster their cybersecurity posture against rising threats, even with minimal expertise. The creation process has been a rewarding learning experience, affirming the potential for applied AI technologies like LLMs to help level the playing field in the significant domain of cybersecurity.

Lastly, I encourage novice security researchers and business IT staff alike to explore the inner workings of these changing technologies as they are becoming embedded into our future. The constantly shifting landscape drives advancements in AI capabilities, which represent the next frontier. Key developments integral to LLM inference are already being integrated, as Liu et al. (2025) observe: "Parallel computing, model compression, memory scheduling, and specific optimizations for transformer structures, all integral to LLM inference, have been effectively implemented in mainstream inference frameworks" (p. 13). Understanding and integrating such advancements into frameworks like AnomaLLMy will be key for maintaining effective solutions. This commitment to continuous learning is essential if we are to apply AI—not as an infallible decision maker, but as a collaborative partner in the ongoing process to secure our digital world, particularly SMBs who need practical and accessible solutions the most.

References

IBM Security. (2024). *Cost of a Data Breach Report 2024*. https://www.ibm.com/downloads/documents/us-en/107a02e94948f4ec

Kaloudi, N., & Li, J. (2021). The AI-Based Cyber Threat Landscape: A Survey. *ACM Computing Surveys*, *53*(1), 1–34. https://doi.org/10.1145/3372823

Kotb, H. M., Gaber, T., AlJanah, S., Zawbaa, H. M., & Alkhathami, M. (2025). A novel deep synthesis-based insider intrusion detection (DS-IID) model for malicious insiders and AI-generated threats. *Scientific Reports*, *15*(1), 207–214. https://doi.org/10.1038/s41598-024-84673-w

Liu, Y., He, H., Han, T., Zhang, X., Liu, M., Tian, J., Zhang, Y., Wang, J., Gao, X., Zhong, T., Pan, Y., Xu, S., Wu, Z., Liu, Z., Zhang, X., Zhang, S., Hu, X., Zhang, T., Qiang, N., … Ge, B. (2025). *Understanding LLMs: A comprehensive overview from training to inference*. *Neurocomputing, 620*, Article 129190. https://doi.org/10.1016/j.neucom.2024.129190

MIT's Computer Science & Artificial Intelligence Laboratory [@MIT_CSAIL]. (2022). *An IBM slide from 1979*. [Image attached] [Post]. X. https://x.com/MIT_CSAIL/status/1604884273789603842

National Institute of Standards and Technology. (2016). *Small Business Information Security: The Fundamentals (NISTIR 7621 Rev. 1)*. https://nvlpubs.nist.gov/nistpubs/ir/2016/NIST.IR.7621r1.pdf

Pan, Zhiwen. (2017). *A Context Aware Anomaly Behavior Analysis Methodology for Building Automation Systems.* https://repository.arizona.edu/handle/10150/625624

Parisi, A. (2019). *Hands-on artificial intelligence for cybersecurity : Implement smart AI systems for preventing cyber attacks and detecting threats and network anomalies* (1st edition). https://ebookcentral.proquest.com/lib/uaz/detail.action?pq-origsite=primo&docID=5847212

QuantFactory. (2024). *Lily-Cybersecurity-7B-v0.2-GGUF* [Large language model]. Hugging Face. https://huggingface.co/QuantFactory/Lily-Cybersecurity-7B-v0.2-GGUF

Sreejith, V., Reddy, P. U., Rao, B. H., & Balakrishnan, S. A. (2022). Hybrid Network Security Model for Small and Mid Sized Enterprises. *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, 622–626. https://doi.org/10.1109/ICICICT54557.2022.9917896

Verizon. (2024). *2024 Data Breach Investigations Report*. https://www.verizon.com/business/resources/T212/reports/2024-dbir-data-breach-investigations-report.pdf

Appendix A

Code Repository Access

The complete source code, documentation, setup instructions, and supplementary materials for the AnomaLLMy project described in this paper are publicly available at the following GitHub repository: https://github.com/pet6r/AnomaLLMy