Advanced Network Anomaly Detection Using Machine Learning

George Smith
Department of Computer Science
College of Charleston
Charleston, SC
smithgr1@g.cofc.edu

***Abstract*— In the domain of cybersecurity, the rapid detection and mitigation of network anomalies are critical for maintaining the integrity and performance of network infrastructures. Traditional detection systems often struggle with modern cyber threats' dynamic and increasingly sophisticated nature, mainly distributed denial-of-service attacks. This study evaluates Neural Networks' efficacy as a network anomaly detection and identification method. Utilizing the CIC-DDoS2019 dataset developed by the Canadian Institute of Cybersecurity at the University of New Brunswick, this project implemented a comprehensive and rigorous methodology encompassing data cleaning, preprocessing, and feature selection aimed at building a Neural Network that is capable of network anomaly detection and later network anomaly identification. My conclusions demonstrate that neural networks are a way of detecting network anomalies. This paper details my experimental setup, model configurations, and a comparative analysis of the performance metrics. It discusses the implications of my findings in the broader context of network security, highlighting the potential of neural networks to advance anomaly detection systems.**

Keywords— Network Anomaly Detection, Neural Networks, Machine Learning, Data Preprocessing, Model Evaluation, CIC-DDoS2019

I. Introduction

In the realm of cyber cybersecurity, network anomalies represent deviations from standard traffic patterns that may indicate malicious activities, such as Distributed Denial of Service (DDoS) attacks, unauthorized access, and other security threats [1]. Unlike traditional cyber threats like viruses, worms, or Trojan horses, which directly harm systems' integrity, confidentiality, and availability, network anomalies might not always signify malicious intent. However, they can still undermine network performance and stability.

The differentiation between network anomalies and conventional cybersecurity threats lies in their manifestations and impacts. Traditional threats often involve explicit malicious payloads designed to execute harmful operations. At the same time, network anomalies could be symptomatic of malicious attacks and benign irregularities caused by system faults or unusual but legitimate usage patterns [1]. Consequently, accurately identifying and classifying network anomalies is crucial for maintaining robust network security without unduly affecting system performance through false positives.

The evolution of threat detection technology illustrates a dynamic shift from simple, rule-based systems that could only detect known threats to more sophisticated AI-driven approaches that learn and adapt over time. Initially, in the 1970s, threat detection relied on these rule-based systems, which were rigid and often failed against new, evolving cyberattacks. By the 1980s, the signature-based approach improved automated threat detection by matching known patterns but faltered against zero-day exploits. The late 1980s and early 1990s introduced heuristic-based threat detection, which allowed for identifying zero-day cyber threats by examining suspicious code properties rather than relying solely on known signatures [2].

Anomaly detection systems further evolved in the late 1990s, utilizing statistical models to monitor network traffic and system activities to establish a baseline of normal behavior and flag deviations as potential threats. This shift marked a significant advancement in reducing manual monitoring and improving the speed and accuracy of threat detection. Today, artificial intelligence (AI) and machine learning (ML) powered solutions dominate the landscape [2].

The increasing complexity and frequency of network attacks, cyber-attacks, and cybercrime, which have evolved to bypass traditional detection systems, underscore the importance of this research [3-4]. As cyber-attacks grow in sophistication, the underlying mechanisms for intrusion detection and network monitoring must advance correspondingly. With its ability to leverage statistical methods to discern patterns and anomalies from vast quantities of data that would be incomprehensible to human analysts, machine learning offers a potent toolset in this arms race.

This study focuses on applying machine learning techniques, particularly Neural Networks (NN), to enhance the detection of network anomalies. Neural networks present a promising alternative with their ability to model complex nonlinear relationships and learn incrementally. They can significantly improve detection rates and reduce false positives, making them a compelling choice for network anomaly detection.

II. Dataset Description

A. CIC-DDOS2019 Dataset

The CIC-DDoS2019 dataset was developed by the Canadian Institute for Cybersecurity (CIC). It responds to the critical need for comprehensive, realistic datasets that align closely with real-world data dynamics. The dataset encompasses a blend of benign and contemporary DDoS attacks, mirroring the nuanced behaviors of network interactions in genuine settings. This dataset was crafted by capturing a mix of benign activities and DDoS attacks in a controlled environment, utilizing the CICFlowMeter-V3 for detailed traffic analysis [5-6].

The traffic generation for CIC-DDoS2019 was designed to simulate real human interactions using the B-Profile system, which profiles the abstract behavior of human activities across various protocols such as HTTP, HTTPS, FTP, SSH, and email. CIC developed a simulation environment that included various operating systems and network configurations to ensure the dataset's complexity and applicability to real-world scenarios.[5] The dataset recorded detailed network interactions over specified attack periods, capturing both the raw network packets (PCAPs) and derived traffic features (CSV files), providing a dual perspective on network dynamics [5].

The dataset extensively represented reflective and exploitation-based DDoS attacks, encompassing various attack vectors, such as MSSQL, UDP, SYN, and UDP-Lag. These attacks were executed to stress-test network resilience and evaluate detection systems' effectiveness under varied and complex attack scenarios [5].

B. *Dataset Characteristics*

The Total Entries and Features: The merged dataset comprises a substantial volume of data entries, reflecting a broad spectrum of network traffic scenarios. It contains approximately 70,427,637 entries spread across 87 different feature columns, totaling 31 Gigabytes of network traffic.

Feature Diversity: The dataset includes a wide array of features capturing various aspects of network traffic, such as source and destination IPs, port numbers, protocol types, flow durations, packet counts, and sizes. These features are essential for identifying patterns indicative of normal or malicious activities within the network.

Label Distribution: Each entry in the dataset is labeled as either benign or representative of one of the multiple types of DDoS attacks, such as UDP, TCP, ICMP floods, and others. This labeling supports supervised learning approaches, which aim to train models that can accurately differentiate between benign and attack traffic.

Figure 1 shows the distribution of captured network traffic to visualize the amount of attack traffic captured by the data label.

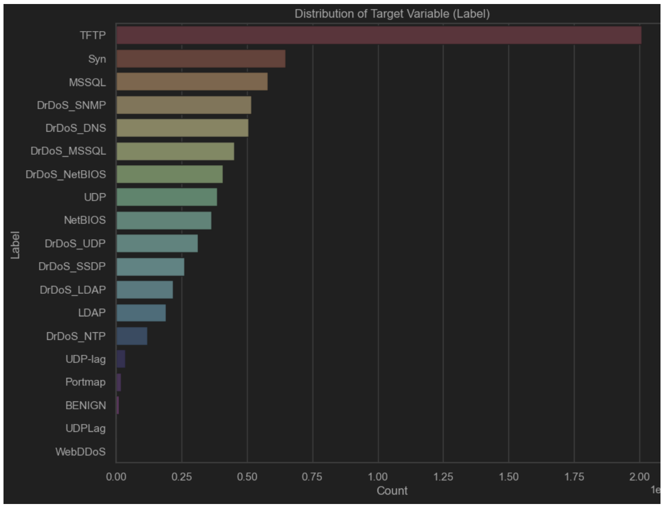

Figure 1: Distribution of Labels for CIC-DdoS2019

C. *Dataset Usage*

The CIC-DDoS2019 dataset originally consisted of 17 separate CSV files, each tailored to represent different network scenarios, including various DDoS attacks and benign traffic. These files were intended to isolate specific conditions or attack types for focused analytical scrutiny and to facilitate a detailed understanding of distinct attack vectors or benign behaviors under controlled circumstances. These 17 files were consolidated into one CSV file to streamline analysis and model training, pooling all DDoS attack traffic into a single dataset.

Using this comprehensive and unified dataset in my project facilitates a more nuanced understanding of network behaviors under various conditions. By training my machine learning models on such a diverse and extensive dataset, I aim to enhance the accuracy and reliability of my NN mode.

III. Data Preprocessing and Reduction Techniques

In preparing the CIC-DDoS2019 dataset for analysis, several data preprocessing steps were undertaken to refine and optimize the data. Here's a breakdown of the critical preprocessing techniques used:

A. *Combining Data Files:*

Initially, the dataset was split into 17 separate CSV files. Each file represented different segments of network traffic data, including various types of DDoS attacks and benign traffic. These files were merged into one complete dataset to facilitate comprehensive analysis and modeling. This consolidation was critical for maintaining continuity in data analysis and ensuring that the models were trained on the full spectrum of data.

B. *Handling Missing Values:*

Early in the preprocessing stage, a thorough examination of the dataset revealed missing or incomplete data entries. To address this, missing values were filled using appropriate imputation techniques based on the nature of the data. In some cases, where the missing data was too extensive, or the imputation could bias the results, the affected rows were removed.

C. *Data Cleaning:*

Redundant rows were identified and removed, simplifying the dataset and focusing model training on relevant features for identifying DDoS traffic. This process reduced the dataset from nearly 70,000,000 rows to 62,339,818, with 5,910,769 duplicate rows identified. These duplicates likely resulted from errors in data collection or conversion from PCAP files to CSV using CICFlowMeter-V3. Despite these issues, the dataset was reasonably clean for analysis.

D. *Normalization and Scaling:*

Normalization techniques were applied to ensure that the numerical values across different features contributed equally to the analysis. This step is crucial for models sensitive to input data scale, such as neural networks, as it ensures that the range of input values doesn't skew the model's performance.

E. Encoding Categorical Variables:

Some features in the dataset, like protocol types and attack categories, were categorical. These were encoded into numerical formats using methods like one-hot encoding. This conversion is essential for processing by machine learning algorithms that require numerical input.

F. Feature Selection:

Feature selection techniques were employed to reduce dimensionality and enhance model performance by retaining features that significantly influenced the target variable, which is the type of network traffic in this context. We started with 87 features and refined this down to 45, focusing on those most correlated with network flows, as these proved to be the strongest indicators for anomaly detection. For instance, features like source IP addresses were removed to prevent the neural network from overly simplifying its decision-making, such as incorrectly identifying a specific IP address as consistently associated with DDoS attacks. Figure 2 illustrates the correlations among the retained features, highlighting their relevance in the training of our model.

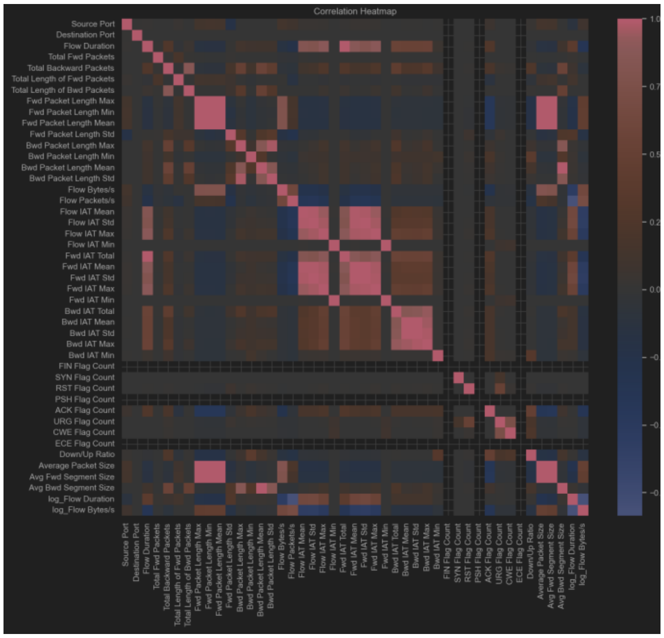

Figure 2: Remaining feature correlation.

G. My Final Dataset Characteristics:

Following these steps, the dataset was condensed to 62,339,818 rows and 45 features. This refined dataset now includes a balanced mix of critical features, which are essential for accurately modeling and detecting DDoS activities, thus boosting the performance and efficiency of the developed machine-learning models. These preprocessing measures ensured the data was clean, relevant, and optimally structured, facilitating effective learning and prediction. Additionally, the reduction in data size and refinement of features contributed to more efficient management of computational resources, vital for handling large datasets.

IV. Setup, Data Analysis and Methodologies

A. Setup:

For this research project, the experimental setup was primarily a workstation equipped with an 11th Generation Intel(R) Core(TM) i7-11390H CPU, operating at a maximum clock speed of 3.40 GHz, supported by 32 GB of RAM, which at times had trouble handling of large datasets and simultaneous processes. This system was also outfitted with a dual GPU setup, including an NVIDIA GeForce RTX 3050 Laptop GPU and Intel(R) Iris(R) Xe Graphics, providing substantial computational power necessary for training and evaluating the neural network models. The operating system used was Microsoft Windows 11 Pro, which offered a stable and secure platform for developing and testing the software components. The data and scripts were managed across several Jupyter notebooks, which were instrumental in coherently organizing the code, visualizations, and iterative testing sequences. Employed in code were several Python libraries such as Pandas, NumPy, Scikit-learn, and PyTorch, which were instrumental in development and visualization.

The IDE used for the project was Dataspell, which is specifically tailored for data science workflows and provided seamless integration with Jupyter notebooks, enhancing productivity and efficiency.

B. Phase 1: Preliminary Data Exploration

Objective: The initial phase aimed to explore a subset of the data for basic understanding and preparation.

Dataset: Initially, I focused on the UDPlag CSV from the CIC-DDoS2019, representing a specific DDoS attack type. This was one of the 17 CSVs developed by CIC. This subset consisted of 725,165 rows and 87 features.

Activities: Data exploration included fundamental statistical analysis, identification of missing values, and preliminary visualization to understand data distributions. Standard cleaning techniques (outlined before), such as handling missing values, normalizing data ranges, and encoding categorical variables for machine-learning compatibility, were used.

Outcome: This phase confirmed the data's integrity and usability for further modeling. The data was found to be well-balanced within this subset, which was not necessarily representative of the entire dataset. Figure 3 shows the distribution of benign (regular) traffic vs. the UDPLag attack traffic. This inspection led me to believe this balance would remain for the complete 17 CSV files.

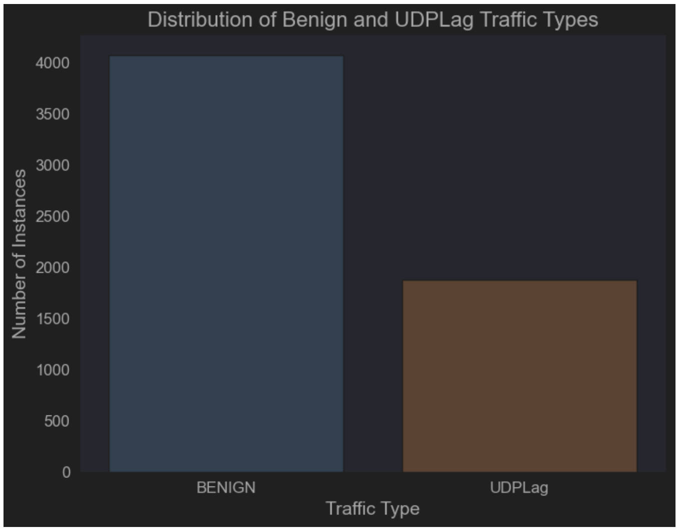

Figure 3: Distribution of Benign vs. UDPlag traffic from the UDPLag sub-dataset from CIC

C. *Phase 2: Model Development and Preliminary Testing*

Objective: To develop a machine learning model based on the cleaned subset and evaluate its performance.

Modeling Approach: A Basic Neural Network (BNN) was meticulously designed and trained for this purpose. The architecture of the BNN consisted of an input layer matched to the number of features in the UDPlag dataset, several hidden layers that enhance the model's ability to learn non-linear relationships, and an output layer designed to classify the network traffic as either benign or malicious. The model utilized ReLU activation functions for non-linearity and a softmax output to handle the binary classification. I employed a batch size of 64 and a learning rate of 0.001, with the Adam optimizer chosen for its efficiency in handling sparse gradients and its adaptability in adjusting learning rates.

Training Details: Training was conducted over ten epochs, with early stopping implemented to prevent overfitting. This was complemented by dropout layers incorporated at strategic points in the network to mitigate further the risk of overfitting by randomly omitting subsets of features during the training process.

Evaluation: The BNN's performance was rigorously assessed based on its accuracy and generalization ability from the training data. The evaluation metrics included accuracy, precision, recall, and F1-score, providing a holistic view of model performance:

Table I. BNN RESULTS

Accuracy of BNN: 0.8368			
	Precision	Recall	F1-Score
	0.84	.083	0.84

TABLE II. CIC PUBLISHED RESULTS

Algorithm	Precision	Recall	F1-Score
ID3	.078	0.65	0.69
RF	0.77	0.56	0.62
Naïve Bayes	0.41	0.11	0.05
Logistic Regression	0.25	0.02	0.04

Outcome: The BNN demonstrated promising results, significantly improving all metrics compared to established algorithms. This success provided the impetus to scale the modeling approach to a more comprehensive dataset in subsequent phases. The comparison of the BNN results with the CIC published results, as illustrated in Tables I and II, underscores the BNN's potential efficacy in anomaly detection within this controlled subset. However, it was recognized that this subset might not fully represent the complexity of the entire dataset.

D. *Phase 3: Comprehensive Data Integration and Analysis*

Objective: To integrate all CSV files into a unified dataset for a complete analysis. Reuse our BNN to evaluate against the entire dataset.

Challenges: The whole dataset integration highlighted significant imbalances between different types of traffic, particularly an overwhelming prevalence of attack traffic compared to regular traffic. Figure 4 will clearly show the gap between benign traffic and attack traffic.

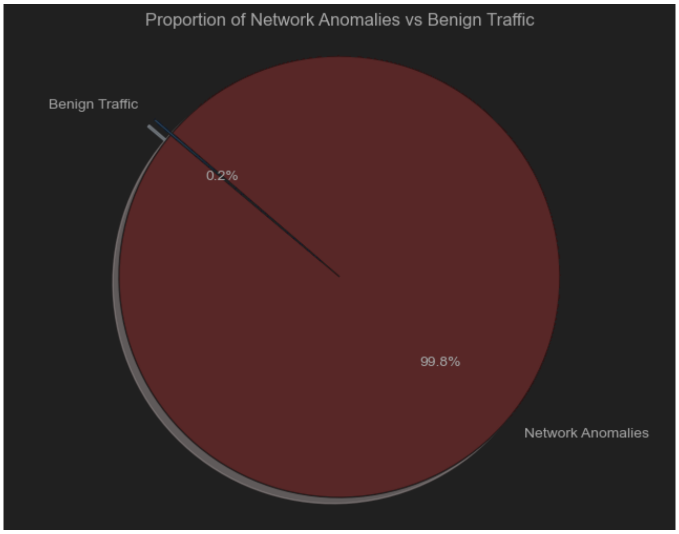

Figure 4: Total Traffic Difference

We can see the gap in the amount of traffic. 99.8% of the traffic was labeled as Network Anomalies, while only .2% was benign traffic. Based on Phase 1 subset exploration, this is not what I expected.

Data Handling: Attempts to manage these imbalances without artificially augmenting the dataset (e.g., through synthetic data generation or duplication) were unsuccessful, impacting model performance.

Outcome: This phase was crucial in understanding the limitations posed by natural data distributions and led to a strategic pivot in how the dataset was approached in subsequent phases. Even though we had an imbalance, I used the same BNN as Phase 1 to try to detect network anomalies. As expected, we did not achieve nearly the same results as in Phase 1.

TABLE III. BNN RESULTS ON FULL DATASET

Accuracy of BNN: 0.2527			
	Precision	Recall	F1-Score
	.0656	0.2527	0.1019

Phase 4: Data Refinement and Focused Analysis

Objective: The aim was to refine the dataset by selectively focusing on specific types of attacks alongside normal traffic, thereby improving data balance and enhancing model accuracy.

Data Selection: To achieve a more balanced dataset, selecting significant attack vectors that frequently occur and have substantial impacts was necessary. The selected labels for this phase were MSSQL, DrDOS_SSDP, and combined benign traffic, representing a focused subset of the data. This selection allowed for a direct comparison between all benign traffic and a specific subset of attacks, aimed at creating a more manageable and representative sample for analysis.

Model Re-evaluation: The same BNN architecture was reused to evaluate this new subset of data. The evaluation indicated some improvement in results; however, challenges persisted due to the still significant imbalance between network anomaly traffic and benign traffic. Specifically, we analyzed 8,238,471 rows with 45 features. Figure 5 illustrates the distribution of retained labels, clearly showing that the imbalance remains considerable, with 98.7% of the traffic being network anomalies and only 1.3% benign.

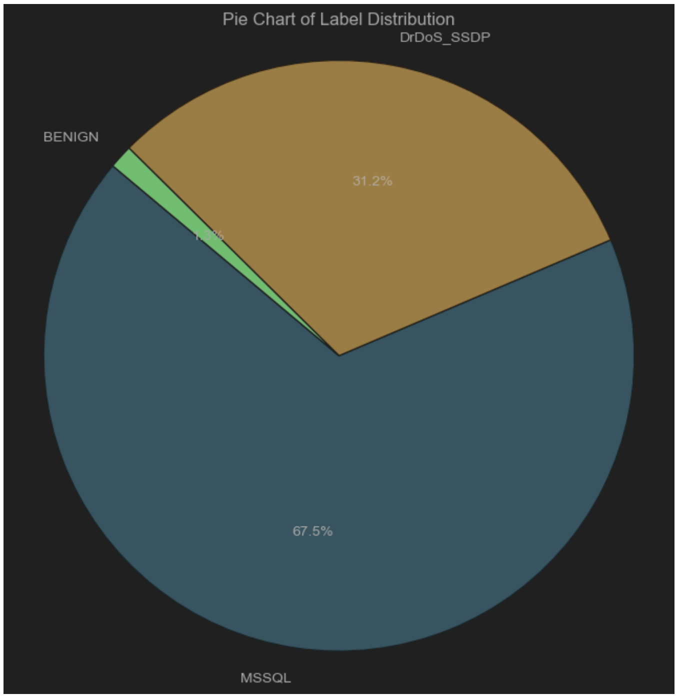

Figure 5: Subset 2 vs. Benign traffic.

Here, 98.7% of traffic was Network anomaly traffic vs. 1.3% Benign traffic. While benign traffic increased 10x, it was nowhere near balanced.

Outcome: Despite the persistent challenges, the focused analysis yielded better results than earlier, broader approaches. This phase underscored the inherent complexities in accurately classifying and differentiating DDoS traffic, even with a more targeted dataset.

TABLE 4: BNN RESULTS ON A PARTIAL DATASET WITH A SLIGHT INCREASE IN BALANCE

Accuracy of BNN: 0.68			
	Precision	Recall	F1-Score
	.4557	0.6751	0.5441

Phase 5: Advanced Anomaly Detection Techniques

Objective: The next step was to reevaluate the entire dataset with an adjusted focus on differentiating among various types of attack traffic, moving beyond the mere binary classification of normal versus attack traffic.

Methodological Adjustment: The focus shifted towards multi-class classification of various attack types, using the same BNN framework to determine if distinct attack signatures could be effectively identified.

Challenges and Innovations This phase involved intricate data labeling and required modifications in the model architecture to accommodate multiple classes effectively. Techniques such as one-vs-all classification were employed to isolate better and identify specific attack patterns. However, the data distribution, as shown in Figure 6, indicates that some attacks were more frequently represented than others, leading to imbalances that affected model performance.

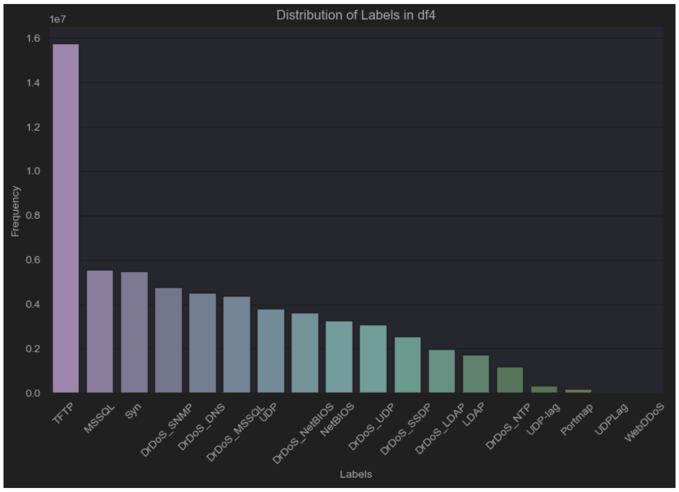

Figure 6: Final distribution by label.

Outcome: This advanced phase broadened the project’s scope from simple detection to a more nuanced classification, offering deeper insights into the distinct characteristics of various DDoS attacks. Unfortunately, the imbalance among the labels continued to pose significant challenges, impacting the overall effectiveness of the detection system.

TABLE IV. BNN IDENTIFICATION BY ATTACK TYPE

Class	**Precision**	**Recall**	**F1-Score**	**Support**
0	0	0	0	905,482
1	0	0	0	396,497
2	0	0	0	875,214
3	0	0	0	239,071
4	0	0	0	721,602
5	0	0	0	954,411
6	0	0	0	513,671
7	0	0	0	618,725
8	0	0	0	343,960
9	0	0	0	1,112,618
10	0	0	0	656,169
11	0	0	0	35,433
12	0	0	0	1,099,173
13	0.25	1	0.4	3,150,149
14	0	0	0	757,976
15	0	0	0	66,005
16	0	0	0	314
17	0	0	0	88
accuracy		0.25		
macro avg	0.01	0.06	0.02	
weighted avg	0.06	0.25	0.10	

Throughout these phases, the iterative approach allowed for continuous learning and adaptation, crucial in cybersecurity, where threats evolve rapidly. Each phase builds upon the lessons learned from previous steps, refining the strategies and methodologies to meet the project's objectives better.

V. Results and interpretation

The project's progression through its phases delineated an intricate narrative of analytical discovery and data-driven adaptation. The initial promise was glimpsed in Phase 1, where the Basic Neural Network (BNN) presented an accuracy of 83.68% on the UDPLag sub-dataset, illustrating the model's potential. Balancing data within this controlled subset permitted clear model learning pathways, yielding precision and recall scores above 0.8, as seen in Table 1. Yet, the optimism cultivated here was met with the complex reality of significant data imbalance in subsequent phases.

In Phase 3, the expansion to the entire dataset introduced a stark imbalance: 99.8% attack traffic to 0.2% benign, leading to a sharp decline in the BNN's efficacy, with accuracy plummeting to 25.27%. Precision and recall suffered alongside, illustrating the skewing impact of imbalance on the model's learning capacity. The deliberation to refine the dataset in Phase 4 saw only a slight improvement in balance—98.7% attack to 1.3% benign traffic—resulting in moderate gains: the BNN's accuracy rose to 68%, and the precision to just over 0.45, as detailed in Table 4.

Phase 5 underscored the variability of performance across different attack types. The BNN, when faced with a multi-class classification task, exhibited varied precision and recall values, as seen in Table 5, further highlighting the disparity in the model's predictive capabilities across the spectrum of attack traffic. Through the iterative and phased approach, the project evolved in terms of complexity, analytical depth, and precision of its outcomes. The numerical insights derived from each phase—whether promising or challenging—served as a beacon, guiding subsequent methodological refinements and ultimately contributing to a nuanced understanding of the model's capabilities and limitations.

VI. Discussion

A. *Initial Promises and Subsequent Challenges*

The initial phase of the project was promising. By focusing on a controlled subset of the data, specifically the UDPLag sub-dataset, the BNN model demonstrated good performance metrics. This success was primarily due to the balanced nature of the dataset within this constrained environment, which provided a clear pathway for the model to learn and make accurate predictions. This phase provided an optimistic outlook on the potential of using neural networks for network anomaly detection. However, as the project expanded in scope during the subsequent phases, the initial optimism was tempered by the emerging challenge of significant data imbalance in the comprehensive dataset. Integrating all CSV files into a single dataset in Phase 3 highlighted a dominant presence of attack traffic over regular traffic. This imbalance skewed the model's learning process, leading to a bias toward predicting attacks, which was not ideal for a system designed to detect various traffic types accurately.

B. The Imbalance Dilemma

The challenge of data imbalance was not merely a statistical hurdle but a fundamental issue that affected the reliability of the detection system. In network security, accurately detecting regular traffic is as critical as identifying malicious activities. The skewed results from Phase 3 onward revealed that the model's effectiveness was compromised, with a diminished ability to generalize from the training data to real-world scenarios. This misrepresentation led to a reassessment of the initial results, which had seemed promising but were later understood to be overfitted to the specific characteristics of the balanced subset rather than indicative of the model's performance on a naturally imbalanced dataset.

C. Why Not Synthetic Data?

Addressing data imbalance typically involves techniques like resampling or synthetic data generation. However, these methods were deemed inappropriate in the context of network security, mainly when dealing with network flows. Network flows, which represent sequences of packets exchanged between sender and receiver during a session, are crucial for understanding the context and behavior of network traffic. Simply duplicating flows or generating synthetic ones using methods like SMOTE could introduce substantial noise into the dataset, distorting the real-world dynamics it is meant to represent. Such artificial augmentation could lead the model to learn from inaccurate or misleading patterns, further impairing its performance and applicability.

VII. Conclusion

This research journey has navigated through intricate phases of data exploration, model development, and rigorous testing, highlighting the potent yet challenging role of Neural Networks in network anomaly detection. The project commenced with promising results from a controlled subset (Phase 1), where the Basic Neural Network (BNN) achieved an accuracy of 83.68%. This early success, although encouraging, presented a skewed perspective due to the inherently balanced nature of the dataset used.

As the study progressed to encompass the full dataset in Phase 3, it encountered substantial data imbalances, with attack traffic overwhelming benign traffic at a ratio of 99.8% to 0.2%. This led to a significant drop in model accuracy to 25.27%, sharply illustrating the impact of data imbalance on neural network performance. Phase 4 attempted to address these issues by focusing on specific attack types against normal traffic, which marginally improved the data balance and enhanced model performance, boosting accuracy to 68% and precision to 45.57%. However, these improvements, while notable, underscored that even modest attempts at rebalancing are critical yet insufficient on their own for achieving robust anomaly detection.

The exploration of advanced anomaly detection techniques in Phase 5 further broadened the project's scope, moving from basic binary classifications to nuanced multi-class identifications of attack types. Despite the sophisticated approach, the performance variability across different attack types revealed persistent challenges, with continued imbalances leading to low accuracy and precision.

A. Key Insights and Future Directions:

Dataset Composition and Model Training: This project's findings stress the critical need for well-balanced datasets in training machine learning models, especially in dynamic fields such as network anomaly detection. Future research should explore novel neural network architectures that are more resilient to data imbalances. Data imbalances should be expected in this field. No network should be expected to be the same, so the algorithm must be robust.

Potential of Neural Networks: Despite the challenges, the potential of neural networks to significantly improve network anomaly detection remains clear. This study lays the groundwork for further exploration into how these models can be optimized and effectively implemented in real-world scenarios.

Realistic Data Representation: The decision against using synthetic data generation techniques like SMOTE was pivotal in maintaining the integrity of network flows, emphasizing the importance of realistic data representation in training models that are truly effective in practical applications.

B. Final Remarks

Through the phases of this project, it has become evident that while machine learning, particularly neural networks, offers considerable promise for advanced anomaly detection, realizing this potential is contingent upon overcoming significant challenges related to data quality and model training. The adaptability required in the face of evolving cyber threats and a steadfast commitment to enhancing network resilience underscores the ongoing need for continuous research and development in this field. While neural networks show potential, their performance is intricately tied to the quality and distribution of the data they learn from. Through this project's progression, it becomes evident that while machine learning holds the promise of advanced anomaly detection, the path to realizing its full potential is meticulously marked by rigorous data assessment and continuous model evolution. The numbers speak to the adaptability required in the face of evolving cyber threats and the unwavering pursuit of enhanced network resilience.

All code developed during this project has been version-controlled and is publicly accessible on GitHub, ensuring transparency and facilitating collaboration. This setup supported the rigorous demands of network anomaly detection research and ensured that the findings and methodologies were reproducible and scalable, paving the way for future enhancements and studies in this domain.

References

[1] PingPlotter. "A Beginner's Guide To Anomaly Detection and its Role in the Network." PingPlotter, 2024. [Online]. Available: https://www.pingplotter.com/wisdom/article/anomaly-detection-role-in-networks/. [Accessed: April 24, 2024].

[2] "Palo Alto Networks. 'What is the Role of AI in Threat Detection?' Accessed April 28, 2024. https://www.paloaltonetworks.com/cyberpedia/ai-in-threat-detection."

[3] M. Bruce, J. Lusthaus, R. Kashyap, N. Phair, and F. Varese, "Mapping the global geography of cybercrime with the World Cybercrime Index," PLoS ONE, vol. 19, no. 4, Art. no. e0297312, Apr. 2024. [Online]. Available: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0297312. [Accessed: Apr. 29, 2024]..

[4] Law Library of Congress, Global Legal Research Directorate, "Cybercrime," United States, 2002.

[5] Canadian Institute for Cybersecurity, "CIC-DDoS2019 Dataset," Canadian Institute for Cybersecurity, University of New Brunswick, 2019. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html. [Accessed: April 29, 2024].

[6] I. Sharafaldin, A. H. Lashkari, S. Hakak and A. A. Ghorbani, "Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy," *2019 International Carnahan Conference on Security Technology (ICCST)*, Chennai, India, 2019, pp. 1-8, doi: 10.1109/CCST.2019.8888419.
keywords: {Computer crime;Taxonomy;IP networks;Protocols;Cloud computing;Feature extraction;Tools;DDoS;IDS;DDoS Dataset;DDoS taxonomy;Network Traffic}.