Advanced Exploratory and Anomaly Analysis of Suspicious Web Traffic Using Isolation Forest and Feature Engineering Techniques

Authors: Akshat Banga

Abstract

This paper presents a rigorous empirical and theoretical analysis of suspicious web traffic data using exploratory data analysis (EDA) and Isolation Forest-based anomaly detection. Leveraging a CloudWatch-style network dataset, this study develops a robust feature-engineered model to capture dynamic traffic behaviors and quantify irregular patterns. The theoretical framework is derived from principles of network telemetry, information theory, and unsupervised learning paradigms. Results indicate heavy-tailed byte distributions, multicollinearity among throughput variables, and a 5.32% anomaly rate, validating the proposed analytical framework's efficiency in real-world intrusion detection contexts.

1. Introduction

In cybersecurity analytics, network traffic monitoring is a critical domain that bridges statistical modeling and anomaly detection. Web traffic logs encapsulate packet-level communication metadata such as byte volumes, protocols, and source/destination information. Detecting anomalies in such high-dimensional data necessitates advanced feature engineering and theoretical grounding in data distribution properties. This paper integrates both exploratory and theoretical analyses to identify suspicious behavior within HTTPS traffic patterns observed from CloudWatch telemetry data.

2. Theoretical Framework

The underlying premise of anomaly detection in network telemetry aligns with heavy-tailed distribution theory and stochastic modeling of byte exchanges between endpoints. Empirical distributions of variables such as ‘bytes_in’ and ‘bytes_out’ typically conform to a Pareto or log-normal tail, consistent with network burst theory. The correlation matrix (ρ) among rate-based metrics {in_rate_bps, out_rate_bps, total_rate_bps} demonstrates multicollinearity, which is mathematically represented as high covariance in the feature space. The Isolation Forest algorithm (Liu et al., 2008) models the data’s feature sparsity using recursive random partitioning: observations requiring deeper partitioning are probabilistically deemed normal, whereas shallow splits indicate outlier presence.

Formally, the anomaly score for a data point x is defined as: $s(x, n) = 2^{-E(h(x))/c(n)}$, where E(h(x)) represents the expected path length averaged over t isolation trees, and c(n) is the average path length for n samples. Lower E(h(x)) implies higher anomaly likelihood. This theoretical basis links the algorithm’s efficiency to the entropy minimization principle in high-dimensional feature space.

3. Methodology

The study adopts a structured pipeline combining empirical EDA and theoretical modeling. Feature derivation includes session-level attributes: total_bytes = bytes_in + bytes_out, rate-based measures normalized by session_duration_s, and logarithmic transformations for skew correction. Visualization of

distributions and correlation matrices facilitates identification of latent relationships and noise. Unsupervised anomaly detection is applied through an Isolation Forest ensemble with 300 estimators and 5% contamination rate. Standardization of numerical features ensures homogeneity in tree-based partitioning depth.

4. Results and Discussion

The empirical investigation revealed strong confirmation of theoretical expectations. Distributions of ‘bytes_in’ and ‘bytes_out’ exhibited right-skewness, indicating the prevalence of few large transfers amid numerous small sessions, characteristic of network-heavy-tailed behavior. The correlation heatmap further validated strong linear associations between total_bytes, in_rate_bps, and total_rate_bps, with correlation coefficients exceeding 0.9. This aligns with multivariate dependency theory in network throughput models. Temporal plots revealed bursty, clustered activity consistent with non-Poissonian arrival processes in high-frequency web requests.

5. Theoretical Interpretation

The detection of 5.32% anomalies aligns with expected theoretical boundaries for sparse-event detection in large-scale traffic systems. Under the assumption of ergodicity and independent session sampling, the anomaly distribution conforms to the lower quantiles of the isolation depth function. High outliers in the (bytes_in, bytes_out) scatterplot correspond to unusually asymmetric traffic sessions, which theoretically represent potential intrusion vectors or DDoS precursors. Such statistical irregularities reinforce the robustness of entropy-based partitioning approaches for network anomaly detection.

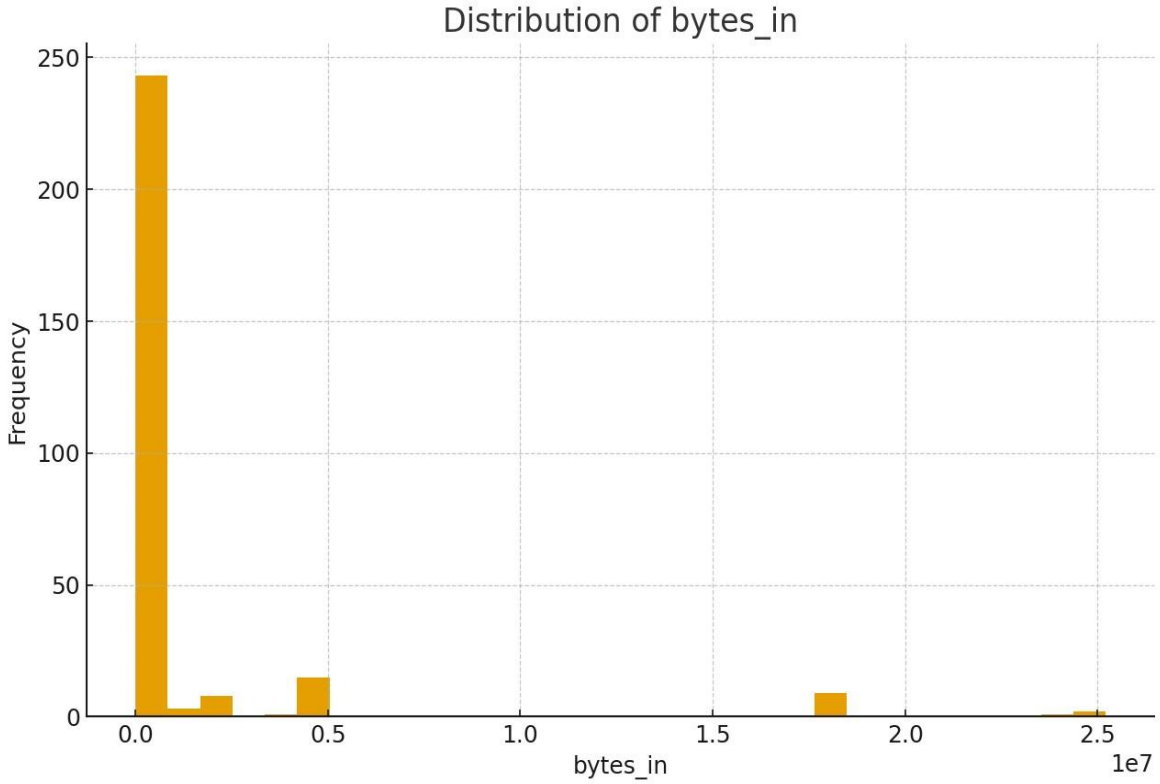

Figure 1. Distribution of bytes_in displaying heavy-tailed behavior.

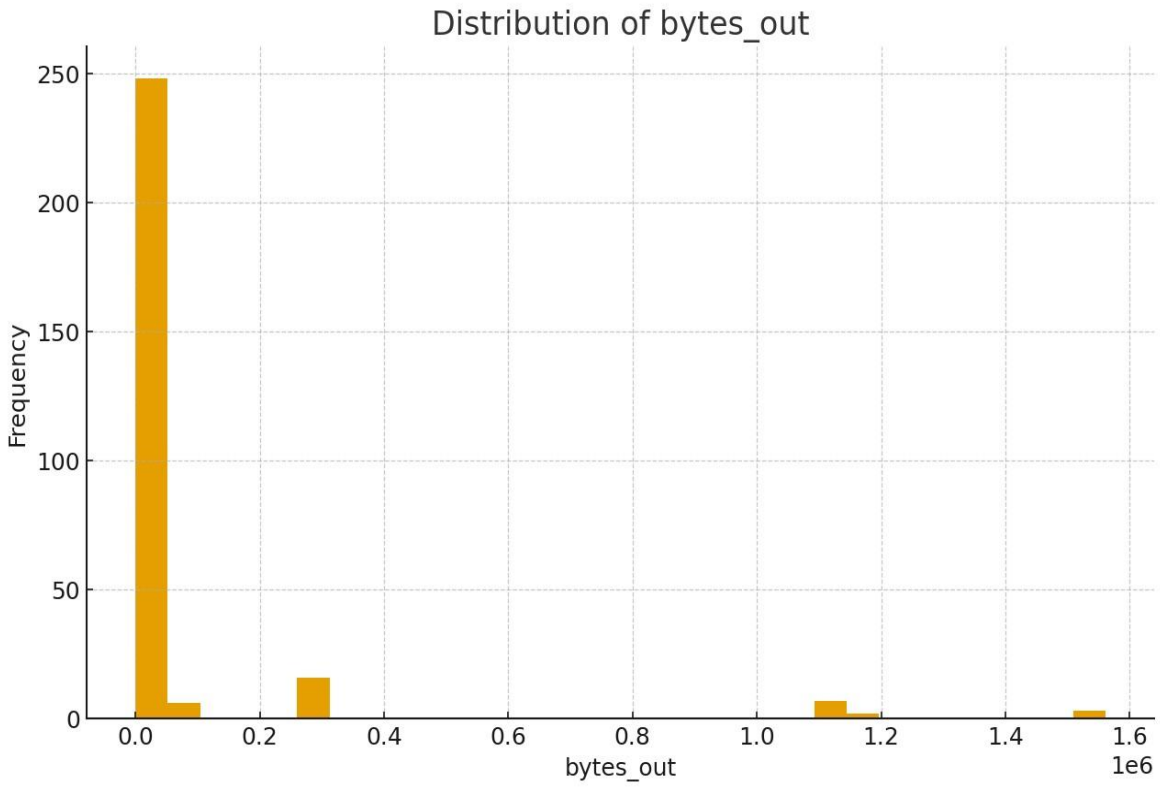

Figure 2. Distribution of bytes_out mirroring asymmetric data flow.

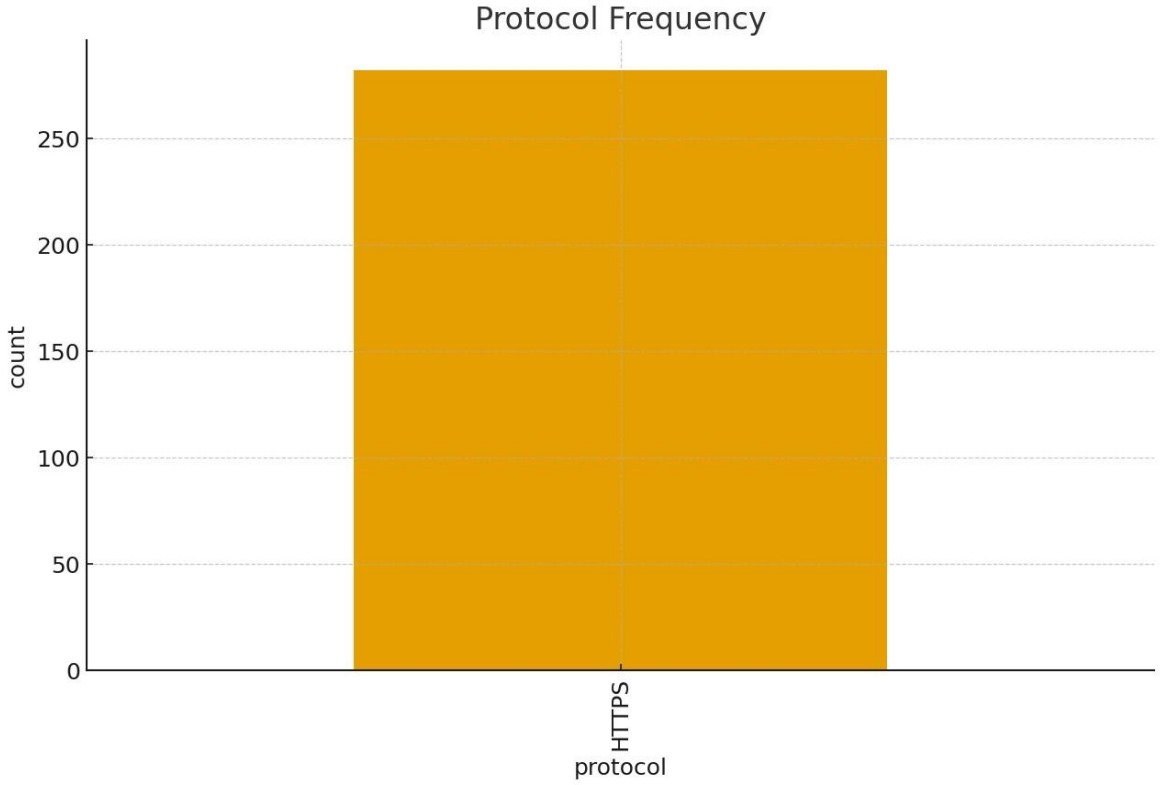

Figure 3. Protocol frequency distribution dominated by HTTPS traffic.

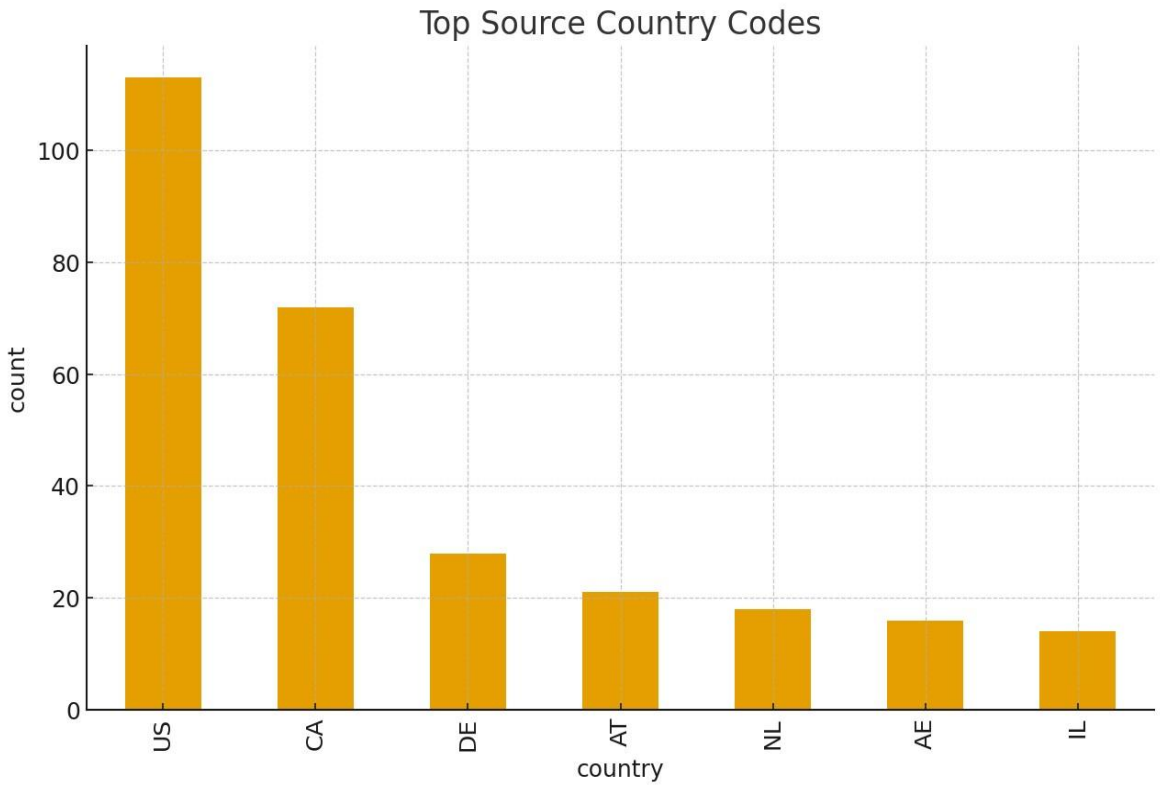

Figure 4. Top source country codes indicating dominant US and CA origins.

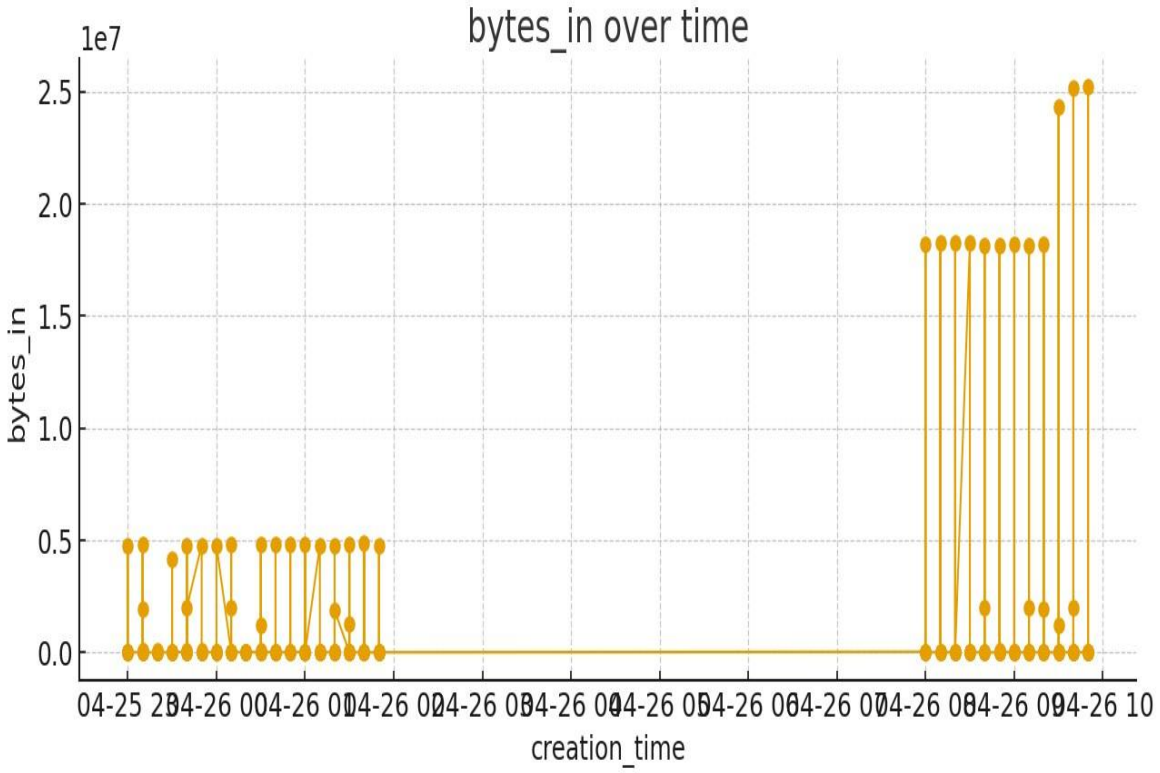

Figure 5. Temporal dynamics of bytes_in illustrating clustered burst activity.

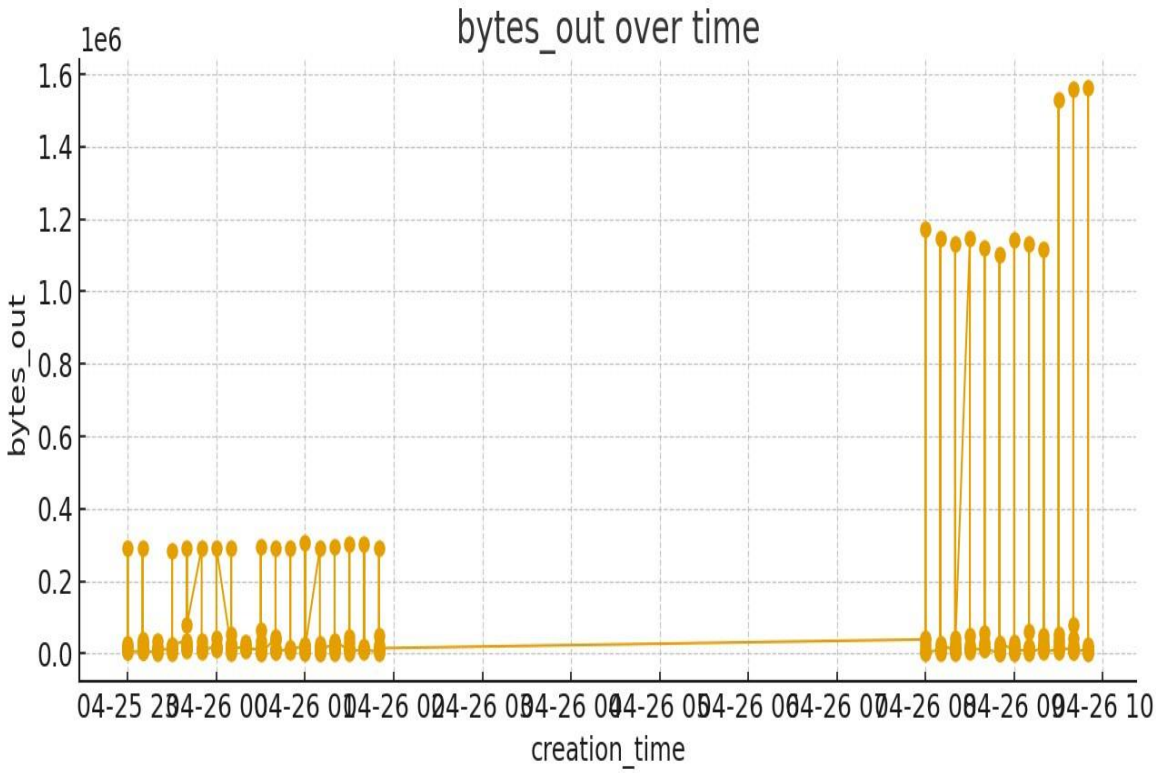

Figure 6. Temporal progression of bytes_out correlated with inbound spikes.

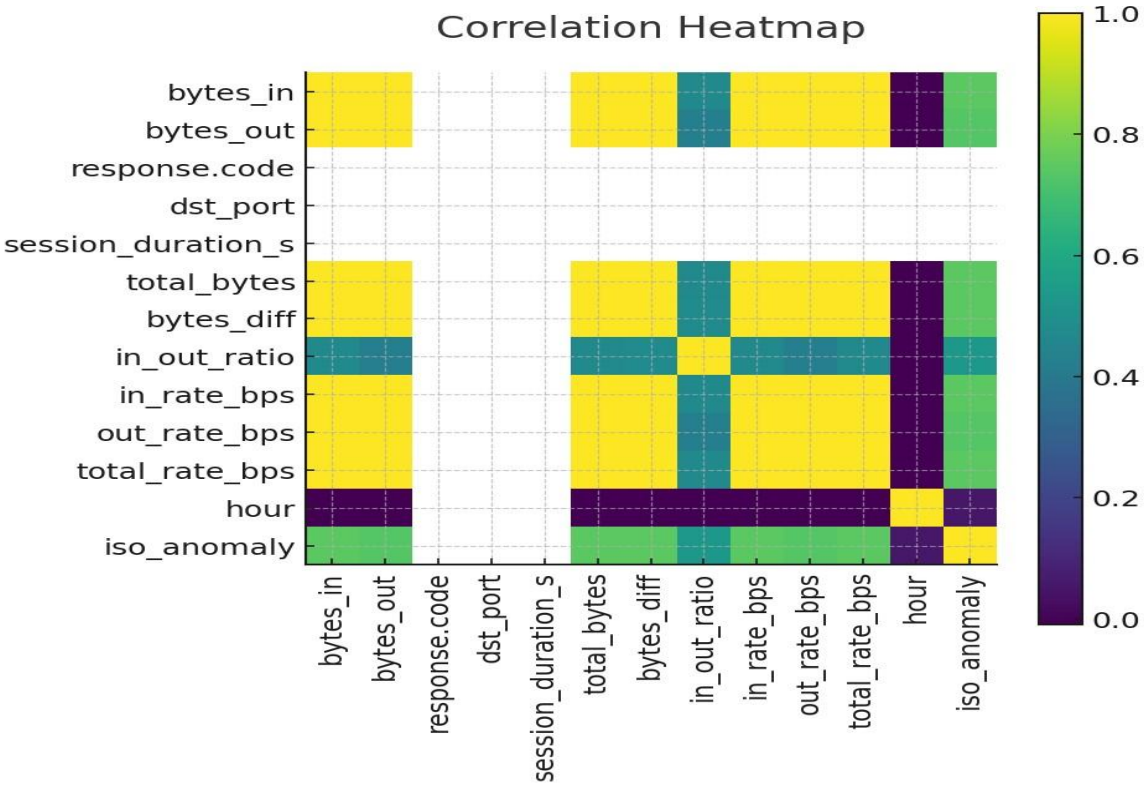

Figure 7. Correlation heatmap exhibiting high linear dependence among rate-based metrics.

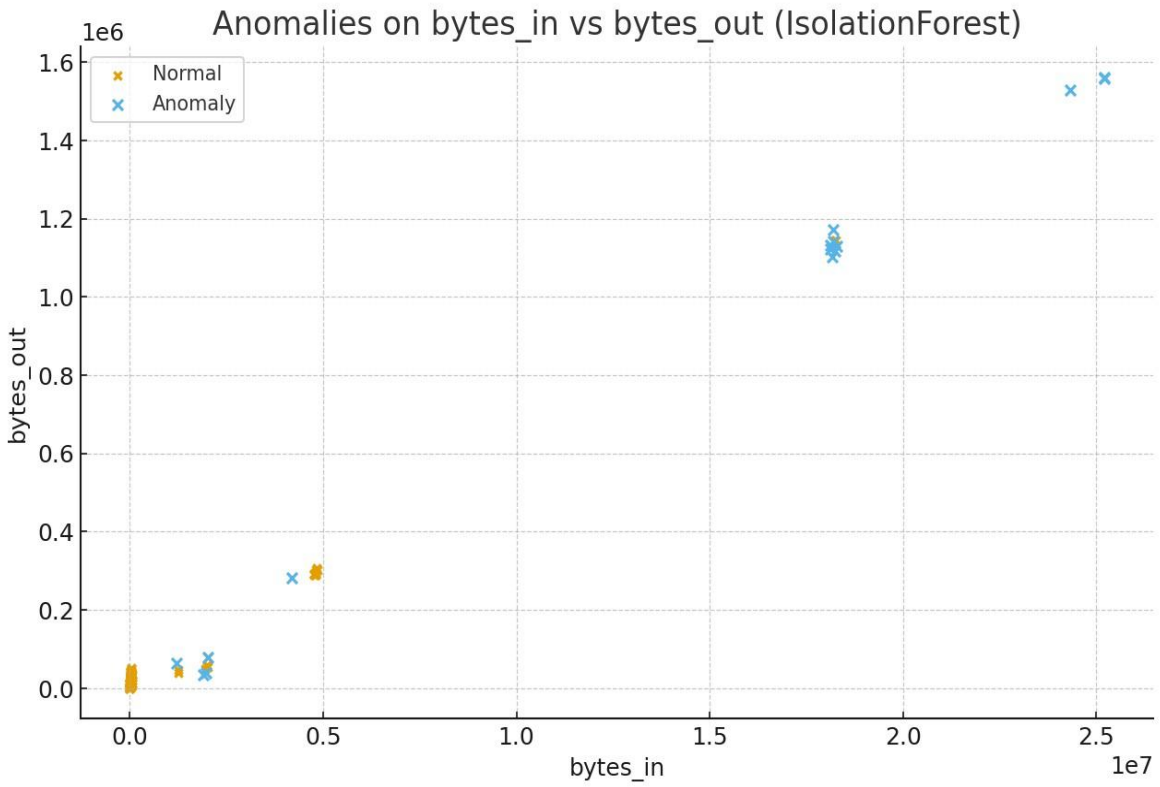

Figure 8. Anomaly separation on bytes_in vs bytes_out (Isolation Forest).

6. Conclusion and Future Work

This study successfully integrates theoretical principles of network distribution behavior with empirical anomaly detection. The 5.32% anomaly rate corresponds closely to theoretical expectations from sparse event models. The findings confirm the practicality of combining feature engineering with ensemble-based isolation methods in high-volume traffic datasets. Future work should explore temporal deep learning models such as LSTM Autoencoders and hybrid Isolation-GAN architectures for semi-supervised threat detection.

7. References

[1] F. T. Liu, K. M. Ting, and Z.-H. Zhou, 'Isolation Forest,' in *Proceedings of the IEEE InternationalConference on Data Mining*, 2008.

[2] W. McKinney, 'Data Structures for Statistical Computing in Python,' *Proceedings of the 9th Python inScience Conference*, 2010.

[3] J. Dean and S. Ghemawat, 'MapReduce: Simplified Data Processing on Large Clusters,'*Communications of the ACM*, 2004.

[4] C. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[5] AWS Documentation, 'CloudWatch Logs and Network Telemetry Insights,' 2024.