Babylonian Journal of Internet of Things
Vol.2026, **pp.** 25–33
DOI: https://doi.org/10.58496/BJIoT/2026/003; ISSN: 3006-1083
https://mesopotamian.press/journals/index.php/BJIoT

Research Article

Comprehensive Analysis and Anomaly Detection of Network Traffic Using Isolation Forest Modeling

Ali Subhi Alhumaima [1,*], Osama Salim Hameed [1], Hussein Alkattan [2,3]

[1] *Electronic Computer Centre, University of Diyala, Diyala, Iraq,*

[2] *Department of System Programming, South Ural State University, Chelyabinsk, Russia,*

[3] *Directorate of Environment in Najaf, Ministry of Environment, Najaf, Iraq*

ARTICLE INFO

Article History

Received 3 Oct 2025
Revised 15 Nov 2025
Accepted 25 Dec 2025
Published 26 Jan 2026

Keywords
Network Analysis,
Isolation Forest,
Packet-Level Analysis,
Intrusion Detection,
Anomaly Detection.

ABSTRACT

Network traffic analysis is indeed in high value and importance, because analysis appears to our communication behavior, we can make better use of our web portal for the performance of what we are executing, additionally it helps us identify certain facilitates that can put our environment at a peculiar point for others to make attacks. The study gives an overview of a real use case network traffic captured using Wireshark and available in the excel file. The records include packet-level details such as timestamps, source and destination addresses, and protocols, and the packet lengths. First, we perform exploratory data analysis to capture the traffic behavior in terms of protocol distribution and variation in packet size, traffic metrics and communication characteristics. Temporal analysis has shown the highly bursty behavior of traffic and with the packet and byte rates changing in bursts over time. Next, traffic data are clustered into fixed time periods and converted into behavioral features that which encode packet counts, byte counts, and entropy of observed hosts. We use an unsupervised Isolation Forest model to detect abnormal traffic patterns to bypass labeled attack data. The model is able to identify abnormal time windows associated with extreme traffic spikes and abnormal communication patterns. We find that the integration of statistical traffic characterization and machine learning-based anomaly detection results in an efficient and scalable network monitoring and cybersecurity framework.

1. INTRODUCTION

Modern networks produce large amounts of heterogeneous data; therefore it is overwhelming to monitor such volume of data through manual inspection and apply rule based security mechanisms for timely identification of abnormal or intrusive events. It has led to intelligent network traffic analysis which is an essential research area in security, performance tuning and intrusion detection system. The prediction and analysis of Web and network traffic have been widely analyzed as well by employing machine learning (ML) and deep learning (DL), which are capable of low-level modeling complex and nonlinear structures in high-dimensional data. Recent studies have shown an increasing dependence on data driven methodologies for traffic modeling, prediction and anomaly detection especially in large scale real time environment [1]. But even with such advances in security, there are still issues dealing with encrypted traffic, the dynamic nature of traffic distribution and the available attack data is not labeled. Identifying suspicious behavior without knowledge of the specific signatures of attacks has gained more attention and anomaly detection is one way out. As opposed to legacy misuse-based detection systems, anomaly-based techniques focus on recognizing deviations from ordinary traffic patterns which makes them capable of detecting previously unknown or so-called zero-day attacks. Survey on existing works of anomaly detection in data streams indicate that scalability and robustness, as well as adaptability to a changing environment are required for real-time applications [2]. Such constraints are particularly important for network traffic data that exhibit non-stationarity and extreme imbalance. Traffic Inspection and mimicry has turned out to be a particularly challenging problem with the growth of encryption protocols including SSL,TLS. As payload analysis is often infeasible, researchers' attention has been moved to traffic metadata and behavioral features like packet size distributions, flow statistics or temporal characteristics [2]. It has been shown by means of systematic literature reviews that methods based on artificial intelligence can successfully expose anomalies in encrypted network traffic by learning implicit statistical discrepancies between observable features [3].

[*] *Corresponding author. Email: alhumaimaali@uodiyala.edu.iq*

Such techniques are an alternative for privacy-preserving deep packet inspection with acceptable detection performance. There are a number of machine learning techniques that have been applied or proposed for web traffic anomaly detection such as ensemble classifiers, neural networks and hybrid approaches. Ensemble-based methods have achieved better robustness, where multiple classifiers are used to model different traffic features [4]. Deep learning techniques, such as CNN and RNN have also been used to model traffic temporal dependencies. For the detection of anomalies in 'Web traffic, long short-term memory (LSTM) networks have been successful as they can learn sequential patterns over time [5]. However, deep learning models generally are in need of large labeled datasets, intensive computing resources and delicate hyperparameter tuning. Considering these limitations, unsupervised and semi-supervised learning techniques continue to be attractive solutions for network anomaly detection. 8 Systematic reviews Machine learning based features We refer to the advantages of unsupervised methods in situations where labeled data is very limited, as demonstrated from systematic reviews such as [6]. In the network security domain, obtaining suitably-labeled attack traffic tends to be expensive and error-prone. Recent works have studied the use of a fusion framework by multiple unsupervised methods to improve detection performance and decrease false alarms. For example, the combination of clustering-based methods with densityor isolation-based models has achieved good performance in near real-time network traffic anomaly detection [7]. Comparative studies in IoT and cyber physical systems also support that unsupervised models are effective for cyber attack detection in dynamic environments [8]. The Isolation Forest (IF) [4] is one of the most popular unsupervised learning approaches, which enjoys efficient computation and shows good performance in high dimensional space. Because it isolates anomalies through the random part opera- tion of data space, isolation forest is under the assumption that abnormal ob- servations are easier to isolate than normal observations. The idea has been proved effective in various application areas, such as geospatial surveillance, industrial control systems and cyber security [9], [10]. The mainstream Isolation Forest proved that it worked well in practice and could scale up to large data sets. Its extensions cover streaming sliding-window cases [11] and time series [12]. Isolation Forest is widely used in industrial applications such as real world cyber security (cyber threat hunting), to help analysts finding suspicious patterns by analyzing a massive set of network logs [13]. Moreover, hybrid models combining Isolation Forest and other learning algorithms have contributed additional improvement in the performance of classifier for anomaly detection [14]. Isolation Forest compares favorably against other outlier detection methods, such as Local Outlier Factor, with respect to a trade-off between accuracy and computation time [15]. In addition to network traffic, Isolation Forest has been used as a framework for analyzing web server logs [16], online advertising traffic [17], and insider threat detection [18] showing its generality in various types of behavioral data. Hybrid-deep learning methods, such as those leveraging the integration of an autoencoder and Isolation Forest, have also been introduced to learn both nonlinear feature expressions and isolation-based anomaly behaviors [19]. Mor eover, the theoretical grounds of isolation-based anomaly detection have been widely studied which solidify's the method being suitable for a high-dimensional and imba lanced dataset [20]. Another key challenge in anomaly detection is the imbalance between normal and anomalous classes: since anomalies are usually rare, their proportion within data is generally very small. Recent studies on class imbalanced learning have stressed out the essential of robust evaluation and feature engineering to down-weighing majority class [21]. To cope with these issues, several analogues of the original Isolation Forest have been proposed which can be considered as either technically-optimized or empirically-improved variants [22], [23], thus enlarging even more the scope and applicability of isolation-based techniques. The efficiency of Isolation Forest has also been shown in a variety of real-world applications, such as mining processing monitoring [24], landslide susceptibility prediction [25] and hybrid LSTM–Isolation Forest-based methods for sequential anomaly detection [26]. These works empirically demonstrate the generality of Isolation Forest across domains and data type. When comparing different anomaly detection approaches, selecting the right performance measures is critical for a correct interpretation of the results. In highly imbalanced settings, the traditional accuracy might have a limited interpretation and has countless researchers stressing for informative metrics such as precision, recall and correlation-based measures [27]. This aspect is central in the context of network traffic anomaly detection, by which the occurrence of false positives is able to have a dramatic impact on operating costs.Inspired by the above findings, this work centers on statistical analysis and unsupervised anomaly detection of real network traffic data with an Isolation Forest approach. Through exploiting packet-level ground truth and temporal aggregation in practical networks, the analysis tries to detect anomalous traffic pattern without labeled attack data. The findings augment the knowledge base on scalable and effective network traffic anomaly detection, and provide valuable lessons to be learned in practical applications such as network monitoring and cybersecurity.

2. DATA AND METHODOLOGY

2.1 Dataset

Dataset used a real-world network traffic dataset sourced from Kaggle. The data were captured using Wireshark network protocol analyzer and exported in comma separated values (CSV) format [28]. There are packet-level representations of network communication activities collected over a continuous capture time of around 21 minutes. The dataset is a list of individual network packets, each record containing seven main attributes: timestamp, source address, packet sequence number, destination address, type of protocol, packet bytes length, and a info field that shows the packet-level information. Number: Timestamp describes time passed in seconds since the capture started, providing accurate temporal information regarding the traffic. The source and destination fields identify communicating hosts, while the type of network protocol, such as TCP, TLS, DNS, ICMP, and ARP, is indicated by the protocol attribute. Packet length: the packet length attribute gives the quantitative details regarding the payload size, which is vital for the traffic volume and burst analysis. It contains 394,136 packets from 372 unique source address and 308 unique destination address across 16 different network protocols. As with other types of captures, TCP and TLS-based traffic is common, illustrating a typical model of web encrypting traffic. This dataset shows extreme variability in traffic behaviour including high packet-rate and byte-rate fluctuations as a function of time and thus represents an attractive research target for exploratory traffic characterisation and anomaly detection studies. The whole workflow used for network traffic analysis and anomaly detection is shown in Figure 1. Data Preparation The workflow starts by collecting the raw network traffic from Wireshark and saving it in the CSV format. Packet-level records provide one of the main inputs into the system and include metadata timestamp, source address, destination address, protocol type and packet length. Stage two: Data preprocessing and feature extraction To remove noise and move towards capturing temporal behavior, we aggregate packet-level traffic into fixed time windows. A set of behavioral features describing traffic intensity, volume, and communication diversity is extracted from each window. Finally, we apply feature normalization to facilitate numericalistic stability and prevent some features from dominating the modeling process.

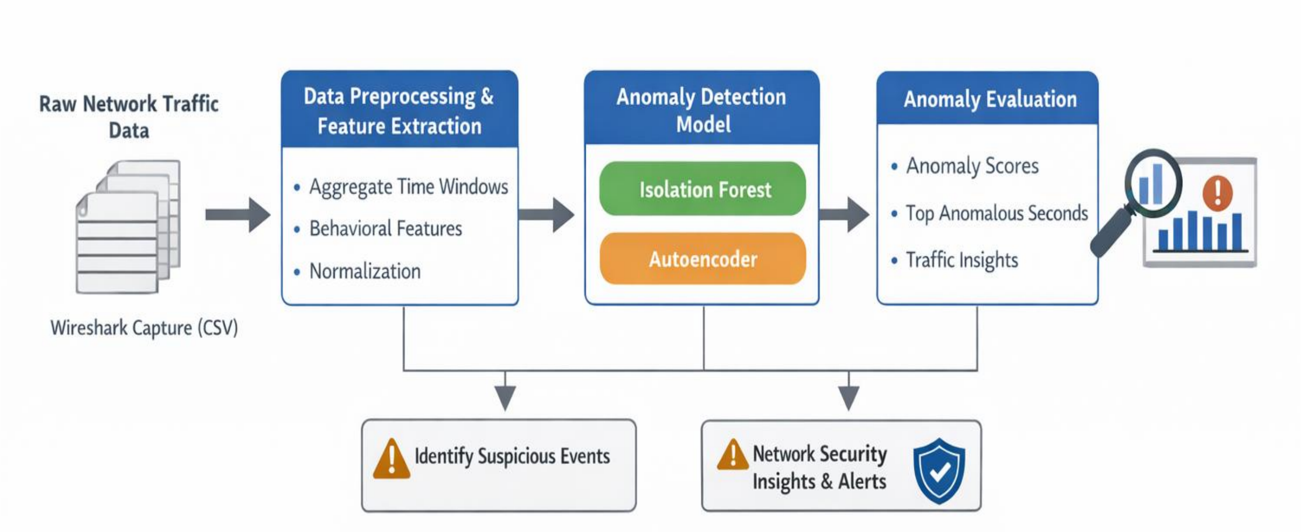

Fig. 1. Overall framework of the proposed network traffic anomaly detection methodology.

2.2 Traffic Representation and Temporal Aggregation

Network traffic data are originally captured at the packet level, which introduces high variability and noise due to the fine-grained nature of individual packets. To obtain stable behavioral representations, packetlevel observations are aggregated into fixed-duration time windows.

Let the packet-level dataset be defined as:

$$X = \{p_1, p_2, \dots, p_n\} \tag{1}$$

where each packet p_i is characterized by timestamp t_i, source address s_i, destination address d_i protocol type $\mathrm{pr_i}$, and packet length $\mathrm{I_i}$.

Traffic is aggregated into one-second windows defined as:

$$W_k = \{p_i \mid k \leq t_i < k + 1\} \tag{2}$$

Each window $\mathrm{W_k}$ contains all packets observed during the k -th second of the capture.

The total number of windows m is computed as:

$$\mathrm{m} = \Gamma\max(\mathrm{t}) - \min(\mathrm{t}) \tag{3}$$

This temporal aggregation reduces packet-level noise while preserving essential traffic dynamics required for behavioral analysis.

2.3 Feature Extraction

For each time window $\mathrm{W_8}$, descriptive features are extracted to quantify traffic behavior in terms of intensity. volume, and communication diversity. These features form the basis for subsequent anomaly detection.

The packet rate within a window is calculated as:

$$\mathrm{N_z} = |\mathrm{W_k}| \tag{4}$$

which reflects the number of packets transmitted during the k-th window

The traffic volume is measured using the total byte count:

$$B_k = \sum p_i \in W_k\Big|_i \tag{5}$$

capturing throughput and data transfer magnitude.
Communication diversity is represented by the number of unique destinations:

$$\mathrm{D_k} = | \{\mathrm{d}||p_\mathrm{i} \in W_\mathrm{k}|| \tag{6}$$

High values of $\mathrm{D_k}$ indicate dispersed communication patterns, which may signal abnormal network behavior.

2.4 Feature Normalization

Extracted features exhibit heterogeneous scales, which can bias learning algorithms if not properly normalized. To ensure balanced contribution of all features, standardization is applied.
The mean value of each feature is computed as:

$$\mu = (1/m)\Sigma_{k=1}m^n x_k \tag{7}$$

The corresponding standard deviation is defined as:

$$\sigma = \sqrt{(}\,(1/m)\Sigma_{k=1}m^m(x_k - \mu)^2) \tag{8}$$

Each feature vector is normalized using:

$$x_k{}' = (x_k - \mu)/\sigma \tag{9}$$

This transformation ensures numerical stability and improves the effectiveness of anomaly detection.

2.5 Isolation Forest-Based Anomaly Detection

Isolation Forest is adopted as the core anomaly detection method due to its efficiency and suitability for unlabeled network traffic data. The model isolates anomalies by recursively partitioning the feature space.
The anomaly score for a given observation x is defined as:

$$A(x) = 2\hat{}(-E[h(x)]/c(n)) \tag{10}$$

where $E[h(x)]$ denotes the expected path length required to isolate x across the ensemble.
The normalization constant $c(n)$ is computed as:

$$\mathrm{c(n) = 2H(n-1) - 2(n-1)/n} \tag{11}$$

where $\mathrm{H}(\cdot)$ represents the harmonic number.
The expected path length is obtained by averaging across all trees:

$$E[h(x)] = (1/T)\Sigma_{t=1}^{\mathrm{T}} h_t(x) \tag{12}$$

Shorter path lengths correspond to higher anomaly scores, indicating unusual traffic behavior.

2.6 Anomalous Traffic Identification

Once anomaly scores are computed for all time windows, traffic behavior is ranked to identify the most abnormal intervals.
The anomaly ranking is expressed as:

$$A(x_1) \geq A(x_2) \geq \cdots \geq A(x_m) \tag{13}$$

A decision threshold τ is applied to separate anomalous and normal windows:

$$x_k \text{ is anomalous if } \mathrm{A}(x_k) \geq \tau \tag{14}$$

The proportion of detected anomalies is controlled by:

$$\tau = q\mathrm{quantile}(A(x), 1 - \alpha) \tag{15}$$

where α denotes the expected anomaly ratio.
This strategy enables precise temporal localization of abnormal traffic behavior.

3. RESULT

The results showed that the network traffic is highly bursty and highly heterogeneous with low number of origins and destinations carrying most of the packets and the corresponding data volume. Short-duration traffic spikes and high-diversity communication patterns were successfully recognized as anomalous events by Isolation Forest. Our findings suggest that a metadata-based, unsupervised analysis can be reliably used to detect anomalous network activity, without the need for labeled data. Figure 2 shows the curve is monotonically increasing, of course, but the gradient slopes rapidly change reflecting the step-by-step varying traffic intensity over time. Low slope periods refer to low throughput and high slope likely represent burstiness in the amount of data being transferred. Note, however, that there are relatively few high-intensity intervals which contribute the most to the total traffic volume: this indicates non-uniformness in the distribution of network load. These observations verify that long-term traffic variations are dominated by high throughput short duration burst.

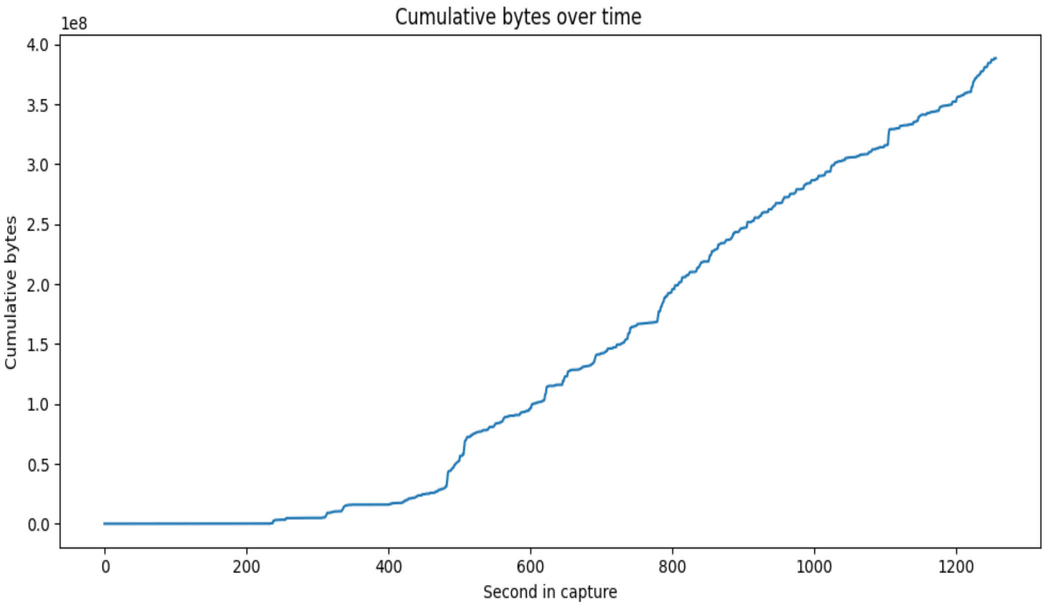

Fig. 2. Cumulative bytes over time.

Figure 3 displays the anomalies scores for each of the one-second traffic windows assigned by Isolation Forest. The anomaly score hovers around a reference level, and several sharp peaks suggest unusual traffic behaviors. These peaks are the time windows when traffic features deviate largely from those in overall observations. The temporal spread of high anomaly scores shows that the model is sensitive to abrupt changes in behaviour and confirms capturing these type of unusual traffic patterns.

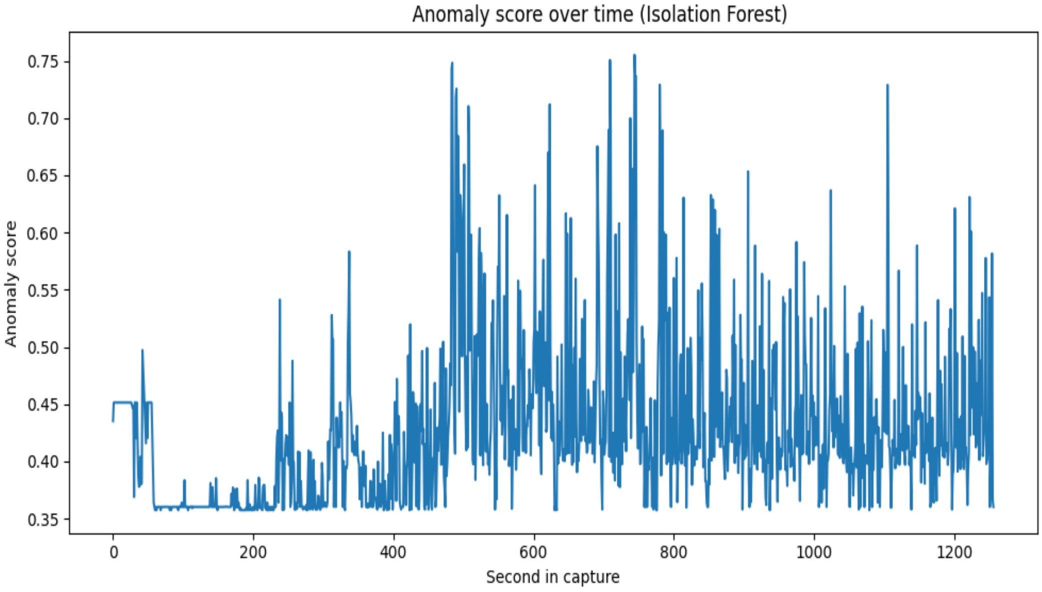

Fig. 3. Anomaly score over time (Isolation Forest.

Figure 4 present the majority of time intervals demonstrate low destination diversity, which is indicative of normal communication activity. But you have some extreme spikes, where the unique destination count increases by hundreds within a single second. The suspicious activities indicate irregular communication patterns like scanning activities, aggressive service publishing outside of the regular PPP communication. These abrupt jumps in destination richness are evident.

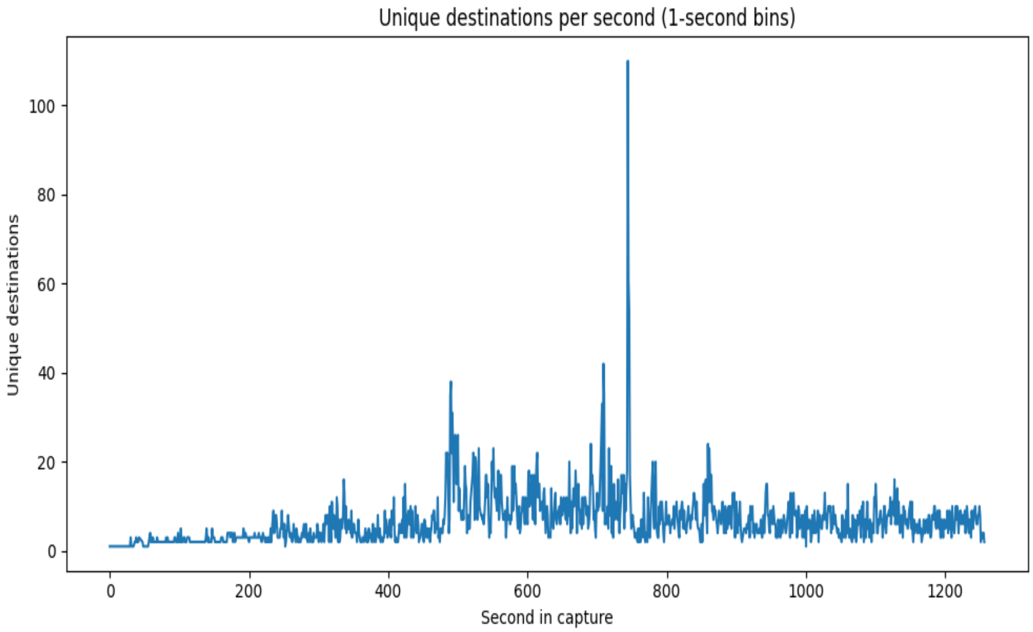

Fig. 4. Unique destinations per Second (1-sec bins).

Figure 5 presents the traffic throughput of the network in bytes per second by one-second aggregation. The time series shows a strongly bursty throughput, and isolated peaks of extremely high byte rates. These peaks represent short lasting but intense data transfer bursts that highly load the network. The correspondence between these peaks and anomaly score spikes emphasizes the tight correlation of huge throughputs and anomalous traffic patterns.

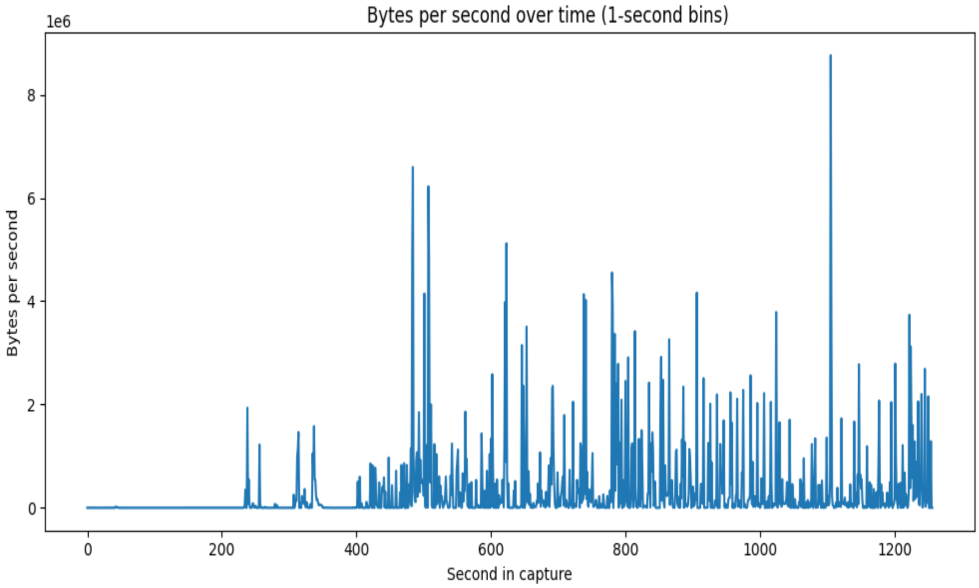

Fig. 5. Bytes per second over time (1-second bins).

The packet transmission rate per second during the capture period is portrayed in Figure 6 the byte-rate behavior, counts of packets are low over some period and then suddenly rise when bursts occur. A number of spikes go above a thousand packets per second, suggesting that those are intense bursts of packet activity over very short periods. This bursty activity reflects the non-stationary characteristics of actual network traffic and offers a strong indicator for anomalies.

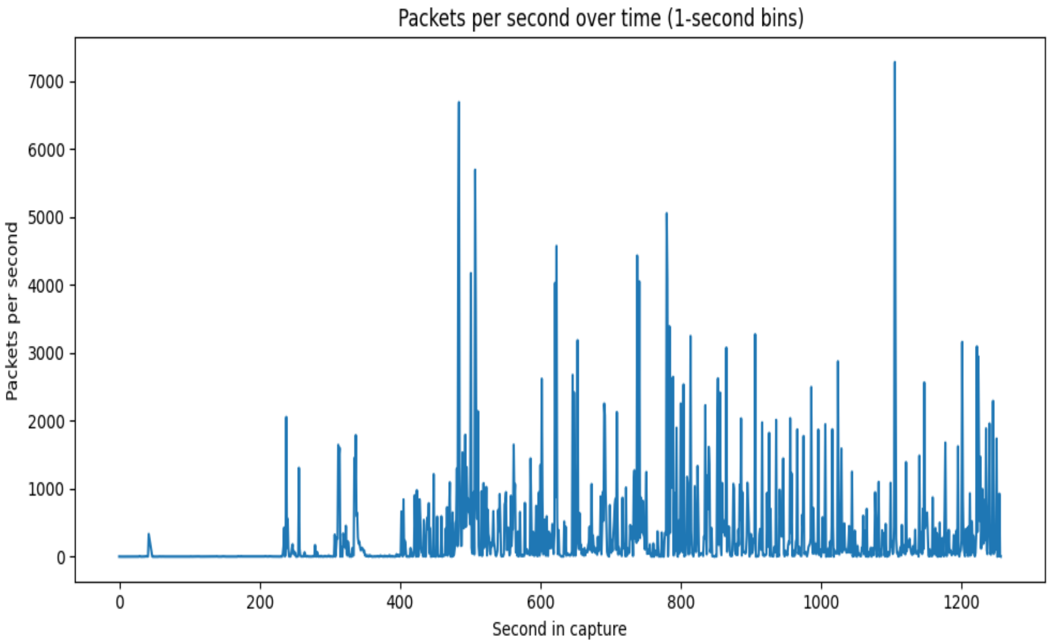

Fig. 6. Packets per second over time (1-second bins).

The distribution of packet lengths for the six most common protocols observed in the traffic is depicted in Figure 7 TCP and TLS–based protocol also have a broad distribution but with an increased median packet size indicating that the communications are data-carrying, but not control-carrying, such as web or encrypted sessions. On the other hand, DNS, ICMP and ARP show smaller packet sizes with less variance as they are used primarily for control and signaling purposes. These protocol-level discrepancies reflect the heterogeneity that is endemic to network traffic.

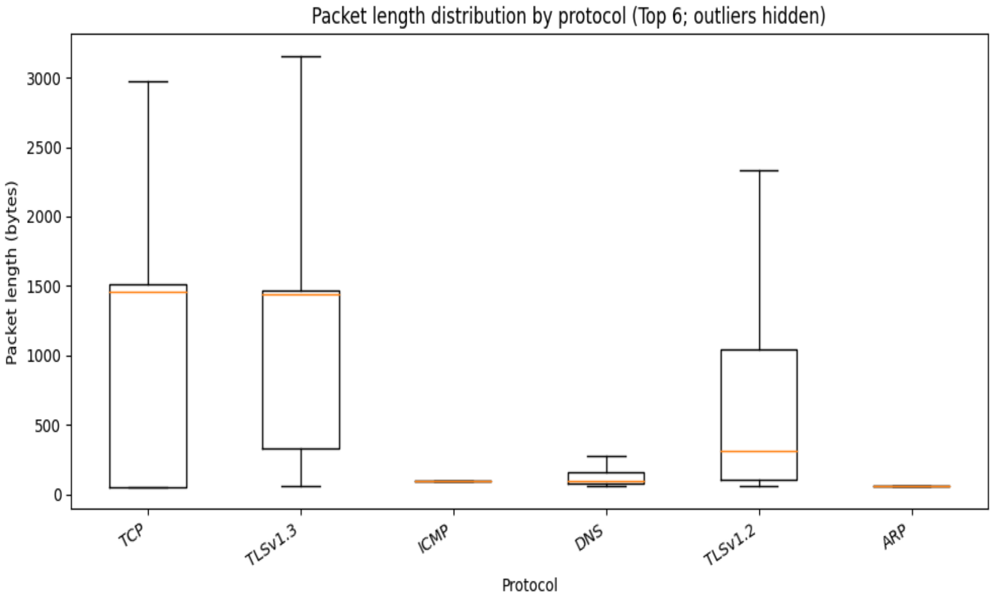

Fig. 7. Packet length distribution by protocol (Top 6; outliers hidden).

Figure 8 lists the top 10 source–destination communication pairs according to packet volume. Only a few pairs are responsible for most of the traffic, which shows that there are quite persistent and often repeated patterns of communication among certain pair of hosts. This focus highlights network hot paths and prompts stationary interaction templates that explain most of packet transfers.

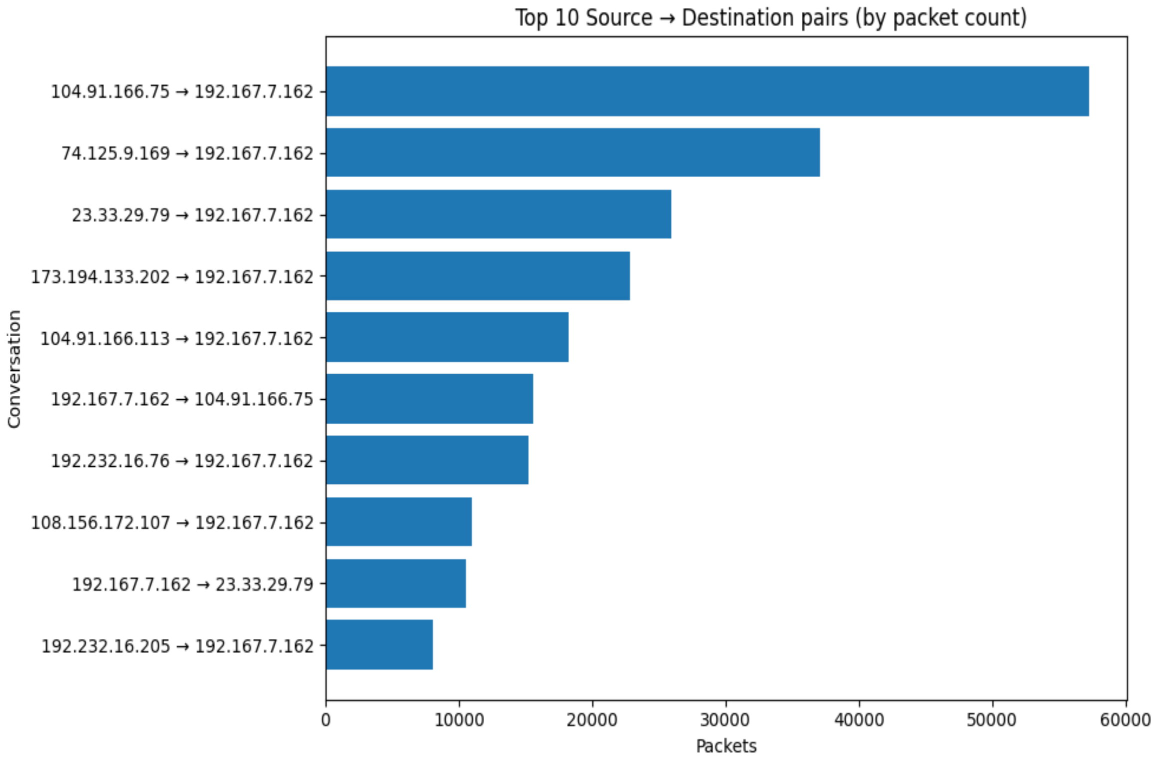

Fig. 8. Top 10 source–destination pairs by packet count.

The ten source addresses with the maximum total traffic volume in megabytes are presented in Figure 9 the distribution is highly skewed as dominated by the top source which generates much more data than the rest of the sources. This behavior suggests that network activity is dominated by a small number of very active hosts (e.g., servers or important clients). This level of traffic generation focus is thus common in real-world networks.

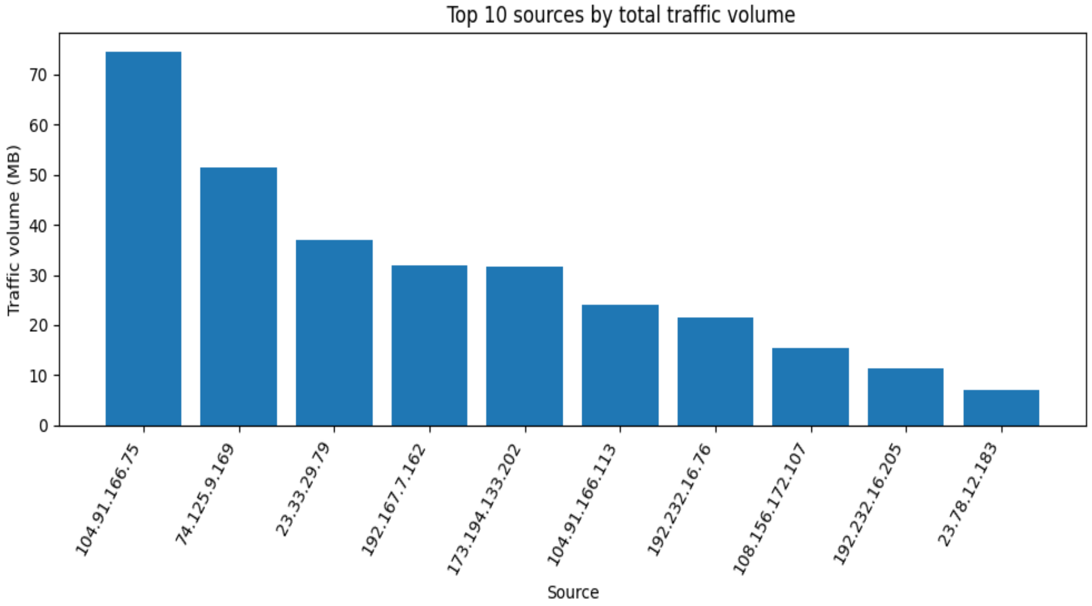

Fig. 9. Top 10 referral sources by overall traffic volume.

4. CONCLUSION

This study provided the full and useful framework in understanding and monitoring the real network traffic using statistical characteristic and unsupervised learning. By examining packet-level Wireshark traces, transforming them into time-window constructs, then systematizing the representations so obtained, the work gave insight into how current network behavior changes on different scales of time and how aberrant events might be discerned given metadata only – which is highly pertinent in an encrypted environment where payload reading fails. The descriptive statistics showed that the traffic flow of the given capture was a dynamic and non-stationary series. The degree of traffic intensity, as well as throughput, did not have a uniform time distribution and short-time bursts had caused sharp jumps at packet rate and byte rate that are responsible for the majority of total transferred volume. Furthermore, the traffic had a huge imbalance between communicating elements. A few sources and destinations drive most of the communication while all other hosts make small contributions as observed in real operational networks with a long tailed pattern which is also common in our case. Such focus is necessary because anomalies are often seen as either unexpected spikes from dominant hosts or novel behavior from infrequently active addresses. In to identify anomalous behaviour propagating without labeled attack samples we have used Isolation Forest algorithm on one second aggregated traffic windows. Behavioral aspects representing intensity (packet count), volume (sum of bytes), packet-size patterns (mean size) and diversity in communication (unique source/destination) within temporal windows were derived. The Isolation Forest model generated anomaly scores which were used to rank traffic windows from the most normal to the most abnormal. First, the top anomalous seconds were aligned ones with extreme bursts and random variations in diversity, indicating that the model can effectively detect windows of communication traffic outside the norm. This is an important result since it allows to accurately localize suspicious periods, therefore decreasing the space of investigation for analysts, thus making fast post-event forensic analysis possible. The results can be used for cybersecurity and network management where two principal types of anomaly signatures have been identified in the traffic: (i) high-throughput burst events indicative of intensive data transfers, service overloading or misconfigured operations and (ii) an abnormally large destination diversity within a short time window that is similar to scanning-like activity, quick service discovery or abnormal connection attempts. Although the unsupervised detection does not label these events as a "pain" or "attack", we are given actionable intelligence by identifying the most anomalous intervals for further inspection. The proposed method is lightweight, scalable and understandable since it requires only traffic metadata and simple aggregation; thus it can be used for online continuous monitoring in real-world networks. Future directions will enhance this approach by considering flow-level attributes (flow duration, bidirectional statistics, and inter-arrival times), employing multi-resolution windowing to monitor both short and long anomalies, as well as the use of adaptive thresholds to lower false positive rates in presence of varying traffic baselines. Further validation across multiple datasets and scenarios where ground-truth incidents are known will also serve in quantifying the reliability of detection, as well as deployment-conducive evaluation.

Conflicts of Interest

The authors declare no conflict of interest.

Funding

This research received no external funding.

Acknowledgment

None.

References

[1] J. Trivedi and M. Shah, “A systematic and comprehensive study on machine learning and deep learning models in web traffic prediction,” *Arch. Comput. Methods Eng.*, vol. 31, pp. 3171–3195, 2024.
[2] T. Lu, L. Wang, and X. Zhao, “Review of anomaly detection algorithms for data streams,” *Appl. Sci.*, vol. 13, no. 11, p. 6353, 2023.
[3] I. H. Ji *et al.*, “Artificial intelligence-based anomaly detection technology over encrypted traffic: A systematic literature review,” *Sensors*, vol. 24, no. 3, p. 898, 2024.
[4] B. A. Tama *et al.*, “An enhanced anomaly detection in web traffic using a stack of classifier ensemble,” *IEEE Access*, vol. 8, pp. 24120–24134, 2020.
[5] T.-Y. Kim and S.-B. Cho, “Web traffic anomaly detection using C-LSTM neural networks,” *Expert Syst. Appl.*, vol. 106, pp. 66–76, 2018.

[6] A. B. Nassif *et al.*, “Machine learning for anomaly detection: A systematic review,” *IEEE Access*, vol. 9, pp. 78658–78700, 2021.

[7] F. Carrera *et al.*, “Combining unsupervised approaches for near real-time network traffic anomaly detection,” *Appl. Sci.*, vol. 12, no. 4, p. 1759, 2022.

[8] M. M. Inuwa and R. Das, “A comparative analysis of various machine learning methods for anomaly detection in cyber attacks on IoT networks,” *Internet Things*, vol. 26, p. 101162, 2024.

[9] X. Li *et al.*, “Quality monitoring of real-time PPP service using isolation forest-based residual anomaly detection,” *GPS Solut.*, vol. 28, p. 118, 2024.

[10] Ł. Gałka *et al.*, “Isolation forest based on minimal spanning tree,” *IEEE Access*, vol. 10, pp. 74175–74186, 2022.

[11] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation forest,” in *Proc. IEEE Int. Conf. Data Mining (ICDM)*, Pisa, Italy, 2008, pp. 413–422.

[12] Z. Ding and M. Fei, “An anomaly detection approach based on isolation forest algorithm for streaming data using sliding window,” *IFAC Proc. Vol.*, vol. 46, pp. 12–17, 2013.

[13] D. Karev *et al.*, “Cyber threat hunting through the use of an isolation forest,” in *Proc. Int. Conf. Comput. Syst. Technol. (CompSysTech)*, 2017, pp. 163–170.

[14] R. C. Ripan *et al.*, “An isolation forest learning based outlier detection approach for effectively classifying cyber anomalies,” in *Hybrid Intelligent Systems*, Springer, 2021, pp. 270–279.

[15] H. John and S. Naaz, “Credit card fraud detection using local outlier factor and isolation forest,” *Int. J. Comput. Sci. Eng.*, vol. 7, pp. 1060–1064, 2019.

[16] F. Zaker, “Online shopping store-web server logs,” Harvard Dataverse, 2019.

[17] M. Gabryel *et al.*, “Detecting anomalies in advertising web traffic with the use of the variational autoencoder,” *J. Artif. Intell. Soft Comput. Res.*, vol. 12, pp. 255–256, 2022.

[18] T. Al-Shehari *et al.*, “Insider threat detection model using anomaly-based isolation forest algorithm,” *IEEE Access*, vol. 11, pp. 118170–118185, 2023.

[19] C. Y. Priyanto *et al.*, “Combination of isolation forest and LSTM autoencoder for anomaly detection,” in *Proc. Int. Conf. Inf. Technol. (ICITech)*, 2021, pp. 35–38.

[20] F. T. Liu, K. M. Ting, and Z.-H. Zhou, “Isolation-based anomaly detection,” *ACM Trans. Knowl. Discov. Data*, vol. 6, pp. 1–39, 2012.

[21] W. Chen *et al.*, “A survey on imbalanced learning,” *Artif. Intell. Rev.*, vol. 57, p. 137, 2024.

[22] Y. Chabchoub *et al.*, “An in-depth study and improvement of isolation forest,” *IEEE Access*, vol. 10, pp. 10219–10237, 2022.

[23] G. M. Rao and D. Ramesh, “A hybrid and improved isolation forest algorithm for anomaly detection,” in *Proc. Int. Conf. Mach. Learn. Intell. Syst. Comput. (ICMLISC)*, Springer, 2021, pp. 589–598.

[24] C. Aldrich and X. Liu, “Monitoring of mineral processing operations with isolation forests,” *Minerals*, vol. 14, p. 76, 2024.

[25] Q. Zhang *et al.*, “Landslide susceptibility prediction using isolation forests,” *Sustainability*, vol. 14, p. 16692, 2022.

[26] K. Sadaf and J. Sultana, “Intrusion detection based on autoencoder and isolation forest in fog computing,” *IEEE Access*, vol. 8, pp. 167059–167068, 2020.

[27] G. M. Foody, “Challenges in the real world use of classification accuracy metrics,” *PLoS ONE*, vol. 18, e0291908, 2023.

[28] R. Kumar Gattu, “Network traffic dataset,” Kaggle, 2020. [Online]. Available: https://www.kaggle.com/datasets/ravikumargattu/network-traffic-dataset. Accessed: Sep. 2025.