

AI-Based Wi-Fi Traffic Profiling and Anomaly Detection System

Real-Time Traffic Analysis, Visualization, and Threat Detection in Wireless Networks

Micael Zeesen Saragih*

Faculty of Mathematics and Sciences
Universitas Negeri Medan
Medan, Indonesia

micaelzeesen.4243250056@mhs.unimed.ac.id

Dedy Kiswanto

Faculty of Mathematics and Sciences
Universitas Negeri Medan
Medan, Indonesia

dedykiswanto@unimed.ac.id

Jhon Gabriel Simarmata

Faculty of Mathematics and Sciences
Universitas Negeri Medan
Medan, Indonesia

rielsimarmata06@gmail.com

Bryant Tinambunan

Faculty of Mathematics and Sciences
Universitas Negeri Medan
Medan, Indonesia

bryanttinambunan12@gmail.com

Abstract— Network traffic monitoring plays an important role in maintaining the availability, reliability, and performance of modern network infrastructures. The increasing volume and complexity of network traffic require monitoring solutions that provide real-time visibility, traffic profiling, anomaly detection, and comprehensive usage analysis. This study proposes an AI-based Wi-Fi network traffic monitoring and analysis system that integrates ntopng for traffic collection, Python-based analytical modules, Prometheus for metrics aggregation, Grafana for visualization, K-Means Clustering for traffic profiling, and Isolation Forest for anomaly detection. The system was implemented in a wireless network environment using real network traffic data collected through the ntopng REST API. The collected traffic data were analyzed to identify traffic patterns, assess traffic severity, categorize network usage activities, and generate monitoring metrics for real-time visualization. The results demonstrate that the proposed system successfully performs traffic clustering, anomaly detection, usage categorization, dashboard-based monitoring, and automated alert integration. Furthermore, the generated analytical information provides insights into network utilization patterns that may support future Quality of Service (QoS) implementation, bandwidth allocation strategies, and intelligent network management. The integration of machine learning techniques with modern monitoring platforms contributes to more effective and centralized network monitoring.

Keywords— network monitoring; network traffic analysis; ntopng; Prometheus; Grafana; K-Means clustering; Isolation Forest; anomaly detection; traffic profiling; machine learning.



© 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution ShareAlike (CC BY SA) license (<https://creativecommons.org/licenses/by-sa/4.0/>).

on computer networks to support their daily operational activities. The growing number of devices connected to the network and the high volume of data traffic make the need for network monitoring systems increasingly vital. Network monitoring not only serves to oversee the condition of devices and services but also assists administrators in understanding network usage patterns, identifying disruptions, and supporting real-time data-driven decision-making [1].

As the concept of observability in modern information technology infrastructure evolves, monitoring systems are required not only to display device status but also to provide information that can be used to comprehensively analyze network conditions. The utilization of observability technology enables the process of collecting, storing, and visualizing network data to be carried out more effectively, thereby facilitating the monitoring and management of network infrastructure [2].

Furthermore, the development of network monitoring technology also indicates an increase in the use of open-source platforms such as Prometheus and Grafana because they can provide real-time metric visualization and are easily integrated with various monitoring systems. Various studies have been conducted to develop network monitoring systems using Prometheus and Grafana. The implementation of network and server infrastructure monitoring shows that the integration of these two platforms can provide real-time system condition information, thereby facilitating the monitoring and maintenance of network devices [3]. Other studies also indicate that Prometheus and Grafana can be applied in virtualization environments to continuously monitor system performance [4]. Additionally, the integration of Prometheus, Grafana, and Alertmanager has been used to build network monitoring systems capable of sending automatic notifications when disruptions occur in the monitored services [5]. In the aspect of network security, visualizing logs and network activities using Grafana has also proven to help administrators identify potentially suspicious activities more quickly [6].

I. INTRODUCTION

The development of information and communication technology has increased the reliance of various organizations

On the other hand, recent research developments show that Artificial Intelligence (AI) and machine learning are increasingly being utilized to support computer network analysis. The application of machine learning allows systems to automatically identify network traffic patterns, detect anomalies, and assist in data-driven decision-making processes [7]. Machine learning approaches have also been used to conduct large-scale, real-time network traffic monitoring with analysis capabilities that are more adaptive to changing network conditions [8]. Moreover, the development of modern observability technology indicates that the utilization of adaptive monitoring mechanisms can improve the efficiency of network data collection and analysis without imposing a significant burden on the observed systems [9]. The use of machine learning algorithms such as Isolation Forest has also proven effective in detecting anomalies in network traffic without relying on previously documented attack patterns [10]. The capability of this anomaly detection algorithm is also excellent when implemented in highly dynamic and heterogeneous network traffic, such as the Internet of Things (IoT) ecosystem [11].

In addition to using monitoring platforms, the implementation of modern observability systems also extensively utilizes containerization technology through Docker. Docker allows applications to run in isolated and consistent environments, simplifying the deployment process, service management, and the integration of various monitoring system components. The use of containerization also supports portability, scalability, and infrastructure management efficiency because each service can be run independently without depending on different operating system configurations [12]. The use of this technology has become crucial in securing and isolating services in the modern era [13].

Therefore, the use of Docker is a relevant approach in implementing the Prometheus and Grafana-based monitoring systems in this study. Although various studies have discussed the implementation of network monitoring using Prometheus and Grafana, most research still focuses on monitoring infrastructure performance, servers, virtualization, and network data visualization. Furthermore, the use of monitoring software such as Ntop is also important to identify users who consume the most bandwidth in a dual-stack based network [14]. Its latest generation, NTOPNG, also possesses highly beneficial features and superior ease of use as a web-based monitoring system when compared to other monitoring applications [15]. However, the integration of network traffic analysis using ntopng, the utilization of Artificial Intelligence for traffic clustering and anomaly detection, and the visualization of analysis results using Prometheus and Grafana is still relatively limited.

Hence, this research is conducted to develop a network traffic monitoring and analysis system capable of integrating the processes of traffic data collection, AI-based analysis, and real-time visualization into a single integrated platform. The purpose of this study is to implement an Artificial Intelligence-based network traffic monitoring and analysis system using ntopng, Prometheus, and Grafana. The developed system is expected to collect traffic data in real-time, cluster traffic characteristics using the K-Means Clustering algorithm, detect anomalous activities, and present analysis results in the form of interactive

dashboards to assist network administrators in monitoring and decision-making.

Unlike previous studies that primarily focus on infrastructure monitoring or dashboard visualization, this study integrates real-time traffic collection using ntopng, traffic profiling through K-Means Clustering, anomaly detection using Isolation Forest, risk-based severity assessment, Grafana-based visualization, and Telegram alerting within a unified monitoring framework. This integration enables both operational monitoring and intelligent traffic analysis within a single platform.

II. RESEARCH METHODOLOGY

This study implements a network traffic monitoring and analysis system using ntopng, Prometheus, Grafana, and Artificial Intelligence (AI) algorithms. The system was developed to perform real-time network traffic collection, traffic characteristic analysis, anomaly detection, and visualization of monitoring results through an interactive dashboard.

A. Research Stages

Based on Fig. 1, this study was conducted through several stages as follows. the study began with a literature review to establish the theoretical foundation related to network monitoring, network traffic analysis, ntopng, Prometheus, Grafana, and the application of Artificial Intelligence (AI) in computer networks. The literature review was conducted by examining relevant journals, scientific articles, and official documentation related to the research topic.

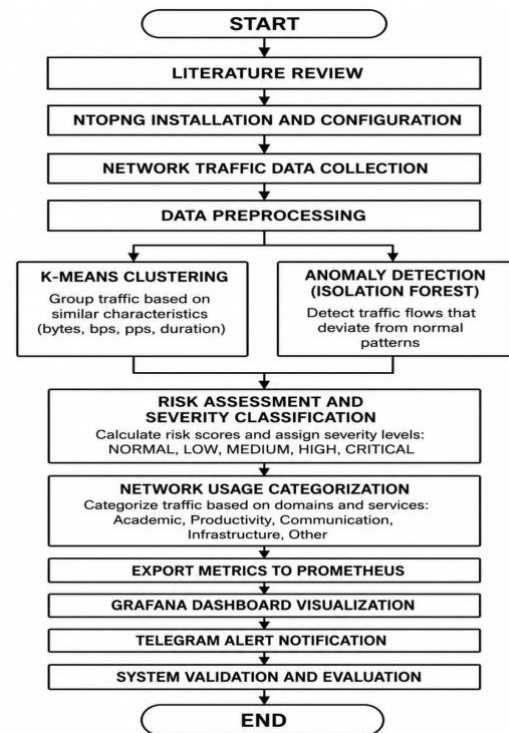


Fig. 1. Research Flow Diagram

The next stage involved the installation and configuration of ntopng as the primary software used to monitor network activities. The configuration process ensured that ntopng was able to capture and store network traffic data in real time. After the monitoring system had been successfully configured, network traffic data were collected through the REST API provided by ntopng. The collected data included source IP addresses, destination IP addresses, destination domain names, network protocols, transmitted data volume, data transfer rates, packets per second, and connection duration.

The collected data then entered the preprocessing stage. This stage was performed to clean the data, identify missing values, standardize data formats, and select attributes to be used in the analysis process. Subsequently, K-Means Clustering was applied to group network traffic based on characteristics with similar patterns. The clustering process utilized bytes, bps, pps, and duration as the primary analytical features.

Following the preprocessing stage, two analytical processes were performed independently. K-Means Clustering was applied to group network traffic based on similar characteristics using bytes, bps, pps, and connection duration as the primary features. At the same time, the Isolation Forest algorithm was utilized to identify traffic flows that exhibited abnormal behavior compared to normal traffic patterns. Activities detected as anomalies were categorized as potential network threats.

The anomaly detection results were subsequently processed through a risk assessment mechanism to calculate traffic risk scores and assign severity levels, including NORMAL, LOW, MEDIUM, HIGH, and CRITICAL categories. This process provided additional contextual information for identifying potentially harmful network activities and supported automated alert generation. In addition to clustering and anomaly detection, network usage categorization was performed based on the domains and services accessed by users. The categories included Academic, Productivity, Communication, Infrastructure, and Other.

All analysis results were subsequently converted into monitoring metrics using the Prometheus Client Library and exported to Prometheus. Prometheus and Grafana were deployed using Docker containerization technology to simplify monitoring service management and visualization. These metrics were then visualized through Grafana in the form of an interactive dashboard that enabled real-time network monitoring. The final stage consisted of system testing to ensure that all components operated according to their intended functions, including data collection, traffic analysis, metric storage, dashboard visualization, and Telegram notification delivery.

B. System Architecture

The system architecture developed in this study is illustrated in Fig. 2. The figure illustrates the architecture of the network traffic monitoring and analysis system developed in this study. The system consists of several main components, namely ntopng as the network traffic data source, Python-based processing

modules for data analysis, Prometheus as the monitoring metric collector and storage system, Grafana as the dashboard visualization platform, and Telegram as the automated notification service.

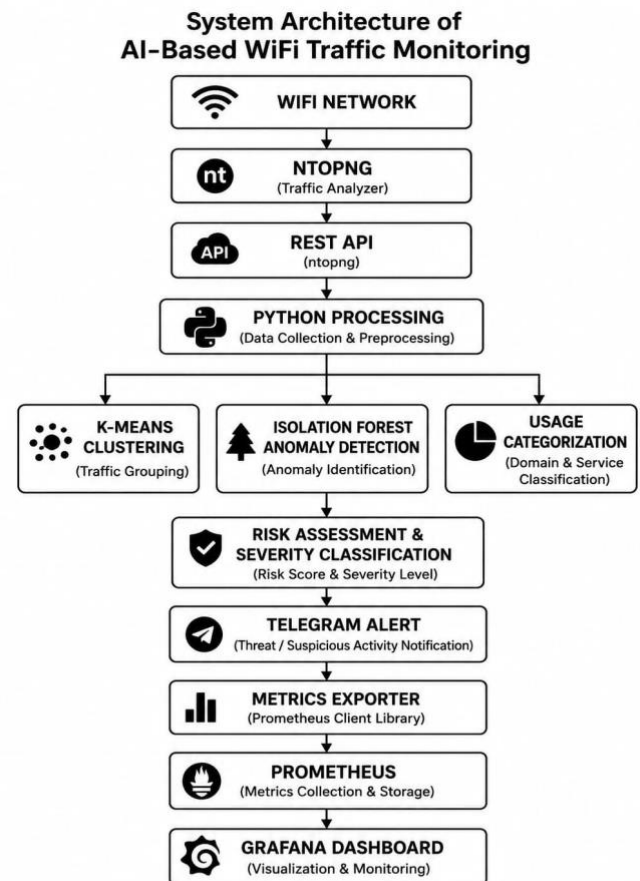


Fig. 2. Network Traffic Monitoring and Analysis System Architecture

Wi-Fi network activities were monitored using ntopng, which functioned as a network traffic analyzer. The traffic data obtained from ntopng were accessed through its REST API and processed using a Python-based application. The Python modules performed data collection through the REST API, data preprocessing, traffic clustering using K-Means Clustering, anomaly detection using Isolation Forest, network usage categorization, and risk assessment generation based on traffic characteristics and anomaly detection results.

Furthermore, anomaly detection results and generated risk scores were used as the basis for severity classification and Telegram notification delivery whenever activities categorized as suspicious or potentially harmful traffic were identified. The outputs generated from traffic clustering, anomaly detection, severity assessment, risk scoring, network usage categorization, and domain activity analysis were converted into monitoring metrics using the Prometheus Client Library and exported to Prometheus running within a Docker container environment. Grafana was subsequently used to present all monitoring and

analysis results through an interactive dashboard that could be accessed in real time.

C. Network Traffic Data Collection

The research data were obtained from Wi-Fi network traffic monitored using ntopng. Data collection was performed periodically through the REST API, enabling the system to acquire network information in real time. The dataset used in this study consisted of network traffic flows collected during the observation period. The number of records may vary depending on network activity because the monitoring system captures traffic dynamically in real time through the ntopng REST API. The attributes used in this study are presented in TABLE 1.

TABLE 1. Network Traffic Data Attributes

No	Attribute	Description
1	client_ip	Source IP address
2	server_ip	Destination IP address
3	server_name	Destination domain name
4	protocol_l4	Layer 4 Protocol
5	protocol_l7	Layer 7 Protocol
6	bytes	Total data transmitted
7	bps	Data transfer speed
8	pps	Number of packets per second
9	duration	Connection duration

Before analysis, the collected traffic data were preprocessed to remove incomplete records, normalize numerical attributes, and select relevant features. The preprocessing stage ensured that the traffic data were suitable for clustering and anomaly detection processes. Among the available attributes, bytes, bps, pps, and duration were selected as the primary features for network traffic analysis because they effectively represent the characteristics of observed network activities.

D. Artificial Intelligence-Based Traffic Analysis

1) K-Means Clustering

K-Means Clustering was employed to group network traffic based on similarities in traffic characteristics. The clustering process utilized bytes, bps, pps, and duration as analytical features. In this study, three clusters were selected to represent low, medium, and high traffic characteristics observed during the monitoring period. The selected cluster configuration was intended to provide a practical traffic profile for exploratory network traffic analysis and monitoring purposes. The clustering results were later exported as monitoring metrics and visualized through Grafana to provide real-time insight into traffic distribution patterns.

To justify the selection of the number of clusters, a Silhouette Score evaluation was performed for K values ranging from 2 to 6. The evaluation produced silhouette scores of 0.7584, 0.5749, 0.6215, 0.6263, and 0.6584 for K=2, K=3, K=4, K=5, and K=6, respectively. Although K=2 achieved the highest score, K=3 was selected because it provided a more interpretable representation of network traffic profiles, consisting of low-traffic, long-duration, and high-bandwidth traffic groups. Furthermore, the obtained Silhouette Score for K=3 remained

within the reasonable clustering quality range and provided better practical insights for network traffic monitoring. The Silhouette Score evaluation measures how similar an object is to its own cluster compared to other clusters, and it is calculated using the following equation:

$$S(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))} \quad (1)$$

The clustering results were used to identify traffic groups with similar characteristics in terms of data volume, transfer speed, packet count, and connection duration. This grouping assists network administrators in understanding network activity patterns observed during the monitoring process.

The distance between data points and centroids was calculated using the Euclidean Distance formula, as shown in Equation (2).

$$d(x, c) = \sqrt{\sum_{i=1}^n (x_i - c_i)^2} \quad (2)$$

where (1), $d(x, c)$ represents the distance between a data point and a centroid, x_i represents the value of the i-th data attribute, c_i represents the value of the i-th centroid attribute, and n denotes the number of attributes used in the clustering process.

2) Anomaly Detection

Anomaly detection was performed using the Isolation Forest algorithm on the preprocessed network traffic data. This method was utilized to identify activities that deviated from normal traffic patterns based on the observed network traffic characteristics. Isolation Forest was selected because it is capable of detecting anomalies without requiring labeled data (unsupervised learning). The algorithm operates based on the principle of data isolation, where observations that differ significantly from the majority of the dataset are isolated more quickly than normal observations. Such easily isolated activities are subsequently categorized as anomalies or potential threats.

This approach is particularly suitable for network traffic analysis because malicious or suspicious activities generally exhibit patterns that differ from normal traffic and occur less frequently. In addition, Isolation Forest can simultaneously analyze multiple network parameters, including bytes, bps, pps, and duration, making it appropriate for anomaly detection in this study. In addition to anomaly classification, the severity classification was determined using a composite risk scoring mechanism that combined Isolation Forest anomaly scores, reputation analysis, country risk assessment, IOC matching results, and destination novelty scores. The final risk score ranged from 0 to 100 and was categorized into five severity levels: NORMAL (0–20), LOW (21–40), MEDIUM (41–60), HIGH (61–80), and CRITICAL (81–100). These severity levels were subsequently used for dashboard visualization and automated alert generation.

To ensure a transparent and quantifiable risk assessment, the composite risk score (R) was formulated by assigning specific weights to each evaluation metric, as defined in the following equation:

$$R = 0.30S_{AI} + 0.25S_{rep} + 0.15S_{country} + 0.15S_{IOC} + 0.15S_{novelty} \quad (3)$$

Where S_{AI} is the anomaly score generated by the Isolation Forest model, S_{rep} is the IP/domain reputation score, $S_{country}$ is the geographical risk score, S_{IOC} represents the Indicator of Compromise matching score, and $S_{novelty}$ is the destination novelty score. This weighted formulation ensures that severity justification is scientifically structured and not solely reliant on machine learning anomaly outputs. The detection results were classified into two categories: NORMAL and THREAT. These classifications were used as early indicators of activities requiring administrator attention and as the basis for Telegram notification delivery.

3) Network Usage Categorization

This study also performed network activity categorization based on the domains and services accessed by users. The categories used were Academic, Productivity, Streaming, Infrastructure, and Other. The categorization process grouped network activities into several categories, including Productivity, Streaming, Infrastructure, Development, Messaging, AI Services, Cloud Services, Operating System, VPN, Monitoring, and Other. The categorization was performed using domain and service information collected from network traffic records.

E. Monitoring Implementation and Visualization

The clustering results, anomaly detection outputs, and network usage categorization results were converted into monitoring metrics using the Prometheus Client Library and exported to Prometheus for real-time monitoring. These metrics were subsequently visualized using Grafana, enabling administrators to monitor network conditions in real time. The developed dashboard displayed traffic cluster distributions, threat statistics, severity distributions, average risk scores, network activity categories, user activity distributions, medium-risk destinations, and the most frequently accessed domains. This visualization provided a comprehensive overview of network behavior and potential security threats detected by the system. The presented information assists administrators in understanding network conditions more quickly and effectively. In this study, Prometheus and Grafana services were deployed using Docker containerization technology to simplify deployment and management of monitoring services.

F. System Testing

Testing was conducted on all system components to ensure that each module operated correctly according to its intended function. The testing parameters used in this study are presented in TABLE 2. The testing process focused on functional validation of each system component to ensure successful integration between traffic collection, Artificial Intelligence processing, metric generation, visualization, and alert delivery. The objective of the testing stage was to verify the operational capability of the proposed monitoring framework rather than to compare machine learning performance with benchmark datasets.

TABLE 2 presents the parameters used during system testing. The system was considered successful if all components operated according to their intended functions, generated monitoring data in real time, performed network traffic analysis, and presented analytical results through a dashboard that could be effectively utilized by network administrators.

TABLE 2. System Testing Parameters

No	Component	Indicator
1	ntopng	Traffic data successfully collected
2	REST API	Traffic data successfully retrieved
3	K-Means Clustering	Traffic clusters successfully generated
4	Isolation Forest	Anomaly status, risk score, and severity level successfully generated
5	Prometheus	Metrics successfully exported and stored
6	Grafana	Dashboard displays monitoring data in real time
7	Telegram Alert	Alert notifications successfully delivered

Testing of the Artificial Intelligence modules was conducted by verifying the traffic clusters generated by K-Means Clustering and evaluating the ability of Isolation Forest to identify anomalous activities. The generated outputs included anomaly status, risk scores, and severity levels, which were subsequently exported to Prometheus and visualized through Grafana dashboards for real-time monitoring. TABLE 3 presents the software platforms and Artificial Intelligence methods utilized in the implementation of the network traffic monitoring and analysis system developed in this study.

TABLE 3. Research Environment

Component	Information
Operating Systems	Windows 11
Programming Languages	Python 3.12
Traffic Monitoring	Ntopng
Database Metrics	Prometheus
Visualization	Grafana
Notification	Telegram Bot
AI Method	K-Means Clustering and Isolation Forest
Containerization	Docker

III. RESULTS AND DISCUSSION

A. System Implementation

The proposed system was implemented by integrating several main components, namely ntopng as the network traffic data collector, Python as the data processing and artificial intelligence analysis module, Prometheus as the metrics storage system, Grafana as the visualization dashboard, and Telegram as the notification platform. Wi-Fi network traffic data were collected in real time through the ntopng REST API and subsequently processed using the K-Means Clustering and Isolation Forest algorithms. The analysis results were exported to Prometheus for visualization in Grafana dashboards and were also used as the basis for sending Telegram notifications whenever anomalous activities were detected. The implemented

system was deployed in a Docker-based environment to simplify service management and integration among ntopng, Prometheus, Grafana, and the Artificial Intelligence modules. The architecture enabled automated traffic collection, analysis, visualization, and alert generation within a unified monitoring framework.

B. Network Traffic Data Collection Results

Network traffic data which can be seen in Fig. 3 were collected using ntopng connected to a Wi-Fi network environment. The collected data consisted of client IP addresses, destination server addresses, packets per second (pps), bits per second (bps), connection duration, and communication protocol information. These data were stored in CSV format and used as input for the subsequent analysis stages. The monitoring system successfully collected network traffic data from a Wi-Fi network environment through the ntopng REST API. During the observation period, the collected traffic consisted of communication flows generated by web browsing activities, cloud services, messaging platforms, Artificial Intelligence services, and software development tools. The acquired traffic information included source and destination addresses, communication protocols, throughput, packet rates, and connection duration.

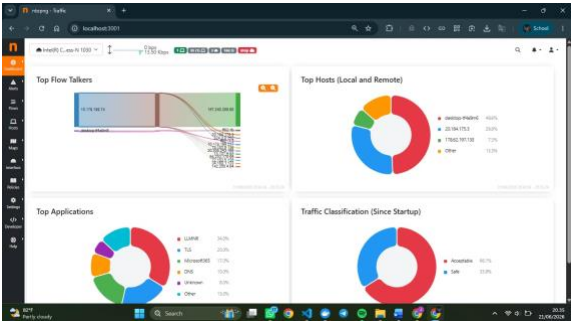


Fig. 3. Real-time traffic monitoring by ntopng, showing multiple hosts, applications, and communication flows

C. Traffic Clustering Results

Traffic clustering was performed using the K-Means algorithm to group network traffic based on similar traffic characteristics. The clustering process utilized throughput (bps), packets per second (pps), byte size, and connection duration as the primary analytical features. The clustering results indicate that the traffic data were successfully grouped into three clusters. The distribution of traffic records in each cluster is presented in TABLE 4.

TABLE 4. Traffic Clustering Distribution

Cluster	Data Amount
0	20
1	31
2	2

The clustering results indicate that the observed network traffic was successfully grouped into three distinct traffic profiles. Cluster 1 contained 31 traffic flows (58.5%) and represented low-traffic activities characterized by short connection durations and relatively low data transfer volumes. Cluster 0 contained 20 traffic flows (37.7%) and mainly represented persistent network connections with long communication durations. Cluster 2 contained only 2 traffic flows (3.8%) and represented high-bandwidth traffic characterized by significantly larger data volumes, higher throughput, and higher packet rates compared to other clusters. These findings demonstrate that K-Means Clustering was able to distinguish different network usage patterns within the monitored Wi-Fi environment. The detailed average metrics representing the specific characteristics of each traffic profile are summarized in TABLE 5. To evaluate the performance and quality of the K-Means clustering model, several statistical metrics were recorded during the analysis process, as presented in TABLE 6.

TABLE 5. Cluster Characteristics

Cluster	Traffic Profile	Average Duration (s)	Average Bytes	Average bps	Average pps
0	Long Duration	1355.90	27980.80	374.06	0.73
1	Low Traffic	27.84	12613.87	582.82	0.67
2	High Bandwidth	32.50	312856.00	23203.87	4.46

TABLE 6. Clustering Evaluation Metrics

Metric	Value
Number of Clusters	3
Silhouette Score	0.5749
Inertia (WCSS)	58.5603
Iterations	3

D. Anomaly Detection Results

Anomaly detection was conducted using the Isolation Forest algorithm to identify network activities that significantly deviated from normal traffic patterns. This algorithm was selected because it operates without requiring labeled datasets, making it suitable for dynamic network environments. In addition to anomaly classification, the proposed system generated risk scores and severity levels to provide a more comprehensive assessment of network activities. The severity assessment categorized traffic into NORMAL, LOW, MEDIUM, HIGH, and CRITICAL levels based on calculated risk indicators. Based on the monitoring results, the system analyzed a total of 52 network traffic flows. The severity distribution generated by the anomaly detection module is presented in TABLE 7.

Fig. 4 indicate that the majority of observed traffic activities were categorized as NORMAL, while the remaining traffic flows were classified as LOW severity. No activities were identified within the MEDIUM, HIGH, or CRITICAL severity categories during the observation period. These findings suggest that the monitored Wi-Fi environment generally operated under normal conditions and did not exhibit significant anomalous behavior requiring immediate administrative intervention.

TABLE 7. Anomaly Detection Results

Status	Data Amount
NORMAL	30
LOW	22
MEDIUM	0
HIGH	0
CRITICAL	0

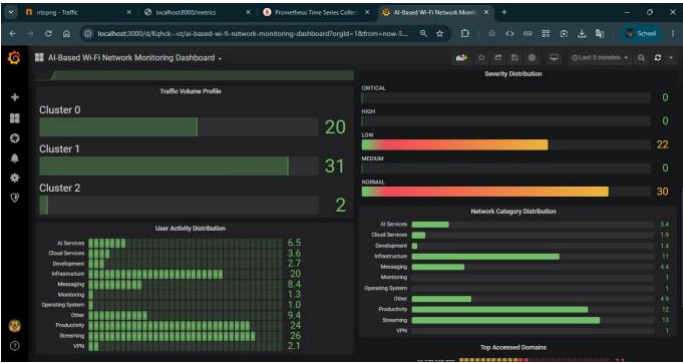


Fig. 4. Grafana dashboard showing traffic severity distribution, detected anomalies, and network activity

E. Network Usage Categorization Results

Network usage categorization was performed to analyze user activity patterns based on the services and domains accessed during the observation period. The system classified network traffic into several categories, including AI Services, Cloud Services, Development, Infrastructure, Messaging, Monitoring, Operating System, Other, Productivity, Streaming, and VPN. The analysis results indicate that Streaming, Productivity, and Infrastructure represented the dominant categories of network activity during the observation period. The distribution of network activity categories is shown in TABLE 8.

The categorization results revealed that Streaming, Productivity, and Infrastructure activities accounted for most of the observed traffic during the monitoring period. Meanwhile, smaller portions of traffic were associated with AI services, cloud services, development platforms, messaging applications, monitoring services, operating system activities, and VPN connections. These findings demonstrate that the proposed system is capable of identifying and classifying diverse network usage patterns, thereby providing administrators with more

detailed insights into user behavior and network resource utilization.

TABLE 8. Network Activity Category Distribution

Category	Data Amount
AI Services	3.4
Cloud Services	1.9
Development	1.4
Infrastructure	11
Messaging	4.4
Monitoring	1.0
Operating System	1
Other	4.9
Productivity	12
Streaming	13
VPN	1.0

F. Monitoring Visualization and Notification Results

The results which can be seen in Fig. 5 generated by the traffic clustering, anomaly detection, and network usage categorization modules were exported to Prometheus and visualized through Grafana dashboards. The developed dashboard provided real-time information regarding traffic cluster distributions, severity levels, network usage categories, top accessed domains, average risk scores, and threat statistics. These visualizations enabled administrators to observe network conditions through a centralized monitoring interface and obtain insights into user activity patterns.

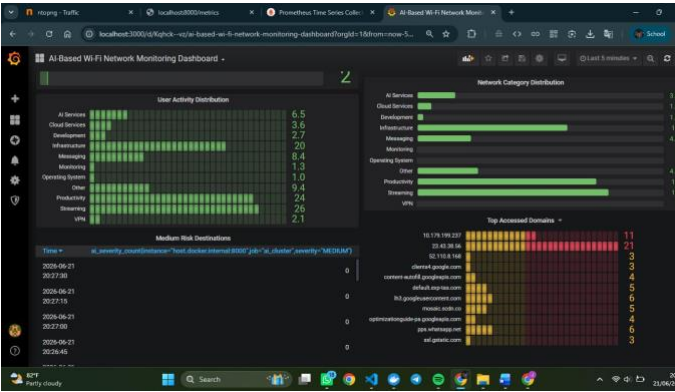


Fig. 5. Grafana dashboard integrating traffic clustering, anomaly detection, traffic categorization, and domain monitoring

In addition to dashboard visualization, the system was equipped with an automated Telegram notification mechanism which can be seen in Fig. 6. Whenever traffic activities exceeded predefined risk criteria or were categorized as suspicious, the system automatically generated warning messages and transmitted them to administrators through Telegram. This mechanism allowed important events to be delivered immediately without requiring continuous dashboard observation.

G. Discussion

The results demonstrate that the proposed architecture successfully integrates network traffic collection, machine learning-based analysis, visualization, and notification services into a unified monitoring environment. The combination of ntopng, Prometheus, Grafana, and Python-based analytical modules enables real-time monitoring while providing additional insights beyond conventional infrastructure monitoring systems.

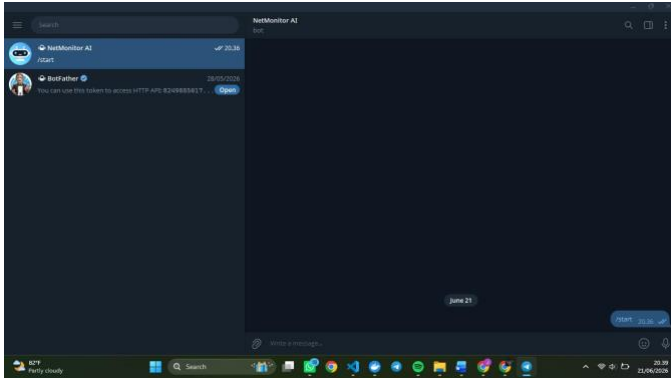


Fig. 6. Telegram bot interface integrated with the proposed network monitoring system for real-time alerts

The clustering results generated by the K-Means algorithm indicate that network traffic can be grouped according to similar communication characteristics. This capability provides administrators with a simplified representation of network behavior, making it easier to identify dominant traffic patterns and distinguish unusual communication profiles. Meanwhile, the Isolation Forest algorithm complements the clustering process by assigning risk levels to observed traffic activities and highlighting activities that deviate from normal network behavior.

The developed Grafana dashboard further improves monitoring effectiveness by consolidating traffic profiling, severity assessment, network usage categorization, and domain monitoring into a single interface. Such integration enables administrators to observe network conditions more efficiently and supports faster decision-making compared to relying on multiple monitoring tools separately.

The network usage categorization results also demonstrate the potential of the proposed system to provide insights into user behavior and resource utilization patterns. Information regarding dominant traffic categories, frequently accessed services, and communication trends can serve as a foundation for future network optimization strategies, including bandwidth allocation, Quality of Service (QoS) implementation, and application prioritization policies. Although the experiments were conducted in a limited Wi-Fi network environment, the proposed architecture demonstrates the feasibility of integrating artificial intelligence techniques with modern monitoring platforms. Future research may extend the implementation to larger network infrastructures, incorporate additional traffic features, evaluate alternative anomaly detection algorithms, and investigate the effectiveness of AI-driven monitoring for supporting intelligent network management and optimization.

IV. CONCLUSION

Based on the conducted research, an AI-based Wi-Fi network traffic monitoring and analysis system was successfully developed by integrating ntopng, Prometheus, Grafana, K-Means Clustering, and the Isolation Forest algorithm. The proposed system is capable of collecting network traffic data in real time, profiling traffic patterns through clustering, assessing traffic severity, detecting anomalous activities, and categorizing network usage based on user activity types.

The implementation results demonstrate that the system can provide centralized monitoring through Grafana dashboards and support automated alert delivery through Telegram whenever network activities exceed predefined risk thresholds. The integration of all system components operated successfully, enabling real-time traffic analysis, visualization, and monitoring within a unified environment.

In addition to monitoring functionality, the generated traffic profiles and usage categorization results provide valuable insights into network utilization patterns. Such information can support future network optimization strategies, including Quality of Service (QoS) implementation, bandwidth allocation policies, and application prioritization mechanisms in educational institutions, enterprise environments, and public Wi-Fi infrastructures. Future research may extend the implementation to larger-scale network environments, incorporate additional traffic features and machine learning algorithms, and evaluate the effectiveness of AI-driven monitoring for supporting intelligent network management and optimization.

V. REFERENCES

- [1] M. D. Elradi, "Prometheus & Grafana : A Metrics-focused Monitoring Stack," vol. 03, no. June, pp. 28–39, 2025.
- [2] D. Soldani, P. Nahi, H. Bour, S. Jafarizadeh, M. F. Soliman, L. D. Giovanna, F. Monaco, G. Ognibene, and F. Risso, "eBPF : A New Approach to Cloud-Native Observability , Networking and Security for Current (5G) and Future Mobile Networks (6G and Beyond)," vol. 11, no. June, 2023.
- [3] M. D. Saputra, Husain, and K. Marzuki, "Desain dan Implementasi Monitoring Infrastruktur Perangkat Jaringan dan Server Menggunakan Grafana dan," vol. 12, no. 3, 2025.
- [4] Sahimin, I. P. Hariyadi, and K. Marzuki, "PENERAPAN PROMETHEUS TERINTEGRASI GRAFANA UNTUK MONITORING VIRTUALISASI BERBASIS PROXMOX," no. September, pp. 365–375, 2025.
- [5] R. A. Putrayana, I. Akbar, and W. Haryono, "Implementasi Sistem Monitoring Jaringan Rumah Sakit Menggunakan Blackbox Exporter, Prometheus, Grafana, dan Alertmanager," vol. 2, pp. 114–122, 2025.
- [6] A. H. Sukma, D. Kiswanto, F. G. Sembiring, and S.

- Ashillah, "Rancang Bangun Sistem Logging Jaringan Berbasis Web dengan Visualisasi Interaktif Grafana untuk Analisis Aktivitas IP Mencurigakan," vol. 14, no. 1, pp. 31–39, 2026.
- [7] E. Zeydan and J. Mangués-Bafalluy, "Recent Advances in Data Engineering for Networking," *IEEE Access*, vol. 10, no. D1, pp. 34449–34496, 2022, doi: 10.1109/ACCESS.2022.3162863.
- [8] M. Seufert, K. Dietz, N. Wehner, S. Geibler, J. Schuler, M. Wolz, A. Hotho, P. Casas, T. Hobfeld, and A. Feldmann, "Marina : Realizing ML-Driven Real-Time Network Traffic Monitoring at Terabit Scale," vol. 21, no. 3, pp. 2773–2790, 2024, doi: 10.1109/TNSM.2024.3382393.
- [9] S. Magnani, F. Risso, and D. Siracusa, "A Control Plane Enabling Automated and Fully Adaptive Network Traffic Monitoring With eBPF," *IEEE Access*, vol. 10, no. September, pp. 90778–90791, 2022, doi: 10.1109/ACCESS.2022.3202644.
- [10] K. A. Nugroho, T. Hariguna, and A. S. Barkah, "Deteksi Anomali Trafik Jaringan dan Aktivitas Pengguna Menggunakan Isolation Forest untuk Meningkatkan Keamanan Jaringan Network Traffic and User Activity Anomaly Detection Using Isolation Forest to Improve Network Security," vol. 5, no. 5, pp. 1365–1376, 2025.
- [11] A. Luthfi and Emigawaty, "Isolation Forest-Based Anomaly Detection in IoT Smart Home Network Traffic," vol. 18, no. 1, pp. 43–57, 2026.
- [12] M. Waseem, A. Ahmad, P. Liang, M. A. Akbar, A. A. Khan, I. Ahmad, M. Setala, and T. Mikkonen, "Containerization in Multi-Cloud Environment : Roles , Strategies , Challenges , and Solutions for Effective Implementation," pp. 1–52.
- [13] G. Thiyagarajan and P. Nayak, "Docker under Siege : Securing Containers in the Modern Era".
- [14] R. Riani, M. A. F. Ridha, and R. P. Sari, "Implementasi Monitoring Lalu Lintas Jaringan Dengan Ntop pada Jaringan Dual Stack Implementation of Network Traffic Monitoring With Ntop on DualStack Networks," vol. 17, no. 4, pp. 424–432, 2018.
- [15] A. P. Situmorang, T. Wati, and A. B. Pangaribuan, "Analisis Perbandingan Sistem Monitoring Jaringan Berbasis Web Menggunakan NTOPNG dan Zabbix di SMP N 1 Tamansari," vol. 6, pp. 8652–8666, 2022.