International Journal On Engineering Technology and Sciences – IJETS
ISSN (P):2349-3968, ISSN (O): 2349-3976 Volume XIII - Issue IV, April - 2026

ADVANCED NETWORK PACKETS SNIFFER AND ANALYSER

Hemachithambaram B A,
Dept. of Computer Science and Engineering,
Francis Xavier Engineering College - Tirunelveli,
Tamil Nadu - India.
hema.ug.23.cs@francisxavier.ac.in

Azzam J S,
Dept. of Computer Science and Engineering,
Francis Xavier Engineering College - Tirunelveli,
Tamil Nadu - India.
azzam.ug.23.cs@francisxavier.ac.in

Heber Vivilian A,
Dept. of Computer Science and Engineering,
Francis Xavier Engineering College - Tirunelveli,
Tamil Nadu - India.
heber.ug.23.cs@francisxavier.ac.in

Mr. M. MALAIARASAN,
Assistant Professor / Dept. of Computer Science and Engineering,
Francis Xavier Engineering College - Tirunelveli,
Tamil Nadu - India.
malaiarasan@francisxavier.ac.in

Dr. R. Ravi,
Professor / Dept. of Computer Science andEngineering,
Francis Xavier Engineering College - Tirunelveli,
Tamil Nadu - India.
directorresearch@francisxavier.ac.in

Abstract:

The rapid increase in network data and the complexity of cyber attacks have rendered real-time packet-level analysis a critical need for contemporary network management and security operations. In this paper, we propose an Advanced Network Packets Sniffer and Analyzer, a software solution capable of capturing, decoding, and analyzing raw packets from different protocol layers in real-time. The tool uses the Python-based modules of Scapy and PyShark to capture network traffic at the interface level, followed by deep packet inspection (DPI) of TCP, UDP, ICMP, HTTP, DNS, and ARP protocols. The proposed system is also equipped with an anomaly detection component that utilizes statistical thresholding and pattern matching rules to detect any irregularities, port scanning, and intrusions. The findings are presented via an interactive dashboard providing detailed information on flows, protocol distributions, and alerts. We evaluated the tool on a simulated campus network and recorded an average packet capture speed of 1.2 Gbps, with 97.4% accuracy in protocol identification and 94.1% efficiency in detecting anomalies.

Keyword: Network Packet Sniffing, Deep Packet Inspection (DPI), Protocol Analysis ,Real-Time Anomaly Detection, Network Traffic Visualization

Introduction:

Networked systems in today's environment have produced vast amounts of data traffic across various types of infrastructure, including LANs, WANs, and cloud-based applications. With the rise of advanced cyber attacks such as DDoS attacks, man-in-the-middle attacks, and APTs, simple passive defense mechanisms have become inadequate. Modern network security and fault analysis require detailed packet-level analysis of data traffic on a live basis. Consequently, packet sniffing and analyzing systems have become necessary elements in every well-equipped NOC or SOC, providing the foundational data required for threat intelligence analysis.

The Advanced Network Packets Sniffer and Analyzer is proposed as an architecture for an all-

encompassing packet analysis system that will resolve the issues present in existing systems like Wireshark, which although are very powerful in packet sniffing do not offer automated anomaly detection or programmatic analysis of packets. Using live packet capture, protocol dissection, machine learning-based anomaly detection, and a dashboard-based visualization interface, the proposed packet analyzer provides end-to-end network observability capabilities.

Network Traffic Capture and Protocol Stack Analysis:

The basic concept in every packet analyzer is frame capturing on-the-fly while the packet traverses the network interface. The raw socket API and packet capture utilities such as libpcap in UNIX-like systems and WinPcap/Npcap in Windows platforms allow the user to capture raw packets in their native format. The process of capturing packets involves setting the NIC card to promiscuous mode to capture all frames, irrespective of their MAC addresses, on the collision domain. After capturing, frames go through a layered protocol parser that analyzes and captures the entire protocol stack layer by layer up to Layer 2 (Ethernet) to Layer 7 (Application).

Deep Packet Inspection and Payload Analysis:

While DPI expands on packet filtering methods in that it analyzes not just the packets but the actual payload, DPI goes further in analyzing traffic based on the payload content, not just on the packets' header information. In particular, DPI helps detect traffic using a false port number, identify protocol tunneling, and examine encrypted stream metadata. The DPI component in our proposed architecture uses pattern matching techniques to reassemble application sessions from fragmented TCP traffic flows in order to analyze HTTP requests/ responses, DNS queries/responses, and FTP commands.

Real-Time Anomaly Detection and Intrusion Identification:

Real-time anomaly detection necessitates the constant comparison of traffic parameters with their normal behavior. Various statistical techniques like exponential weighted moving average and Z-score thresholding are used on metrics like packet rate, byte rate, average inter-arrival time, and protocol ratio. If the parameters lie outside predefined threshold limits, alerts are triggered based on the deviation. Further, there is also a signature-based engine that compares packets against a database of threats, which include signatures for attacks like SYN floods, ARP poisoning, and DNS amplifications. This multi-pronged approach helps to reduce the errors of omission and commission greatly.

Visualization Dashboard and Reporting Module:

Converting unstructured packet information into structured intelligence calls for a robust visualization component that translates intricate traffic dynamics into comprehensible visualizations that can be interpreted by network administrators and security experts. Visualization components generate real-time protocol distribution charts, flow rate graphs over time, geographical IP heat maps, and top talker bandwidth utilization rankings. The alerts timeline associates any anomaly detected with traffic spikes at that time, allowing quick root cause identification. Historical session information can be exported in PCAP file format for offline forensics with third-party software, while reports can be created in PDF file format.

Scalability and Multi-Interface Concurrent Monitoring:

The typical enterprise network will include several physical and virtual network interfaces spread across different VLANs or subnets. This issue is solved by using the multi-threading approach for packet capture, where separate capture threads run on each network interface and push captured packets into a common analysis pipeline via a thread-safe packet queue. It prevents the higher traffic interfaces from causing any delays for low-traffic interfaces. The plugin architecture makes it possible for users to add new plugins to customize the analysis pipeline without changing the main capture module.

Work Objective:

One of the main objectives of the proposed work is the creation and implementation of a packet sniffing and analysis system, which will give detailed information about the network interaction and allow the timely detection and investigation of potential threats through the following specific objectives:

High Performance Packet Sniffing Engine: One of the key objectives of this project is the development of the high performance packet sniffer capable of providing continuous interception of packets at full Gigabit Ethernet line rate without losing packets. The packet sniffer should operate in promiscuous mode from multiple physical and virtual network interface cards by using the Berkeley Packet Filter expression.

Multi-Protocol Packet Dissection and Parsing: One of the key objectives is to build a hierarchical

protocol analyzer capable of dissecting packets on all OSI layers from Layer 2 Ethernet packets through Layer 7 application protocols like HTTP, HTTPS metadata, DNS, FTP, SMTP, and ARP. Each field must then be parsed and stored within a normalized data schema to enable further analysis through downstream filtering and statistical aggregation functions.

Real-Time Anomaly and Intrusion Detection: Another key objective is to incorporate a hybrid detection engine utilizing both statistical analysis for anomalies and signature matching against rules in a threat database. The statistical analyzer constantly updates baseline parameters and generates alarms in case deviations surpass adaptive thresholds. The signature-based engine compares packet data against patterns defined within the database, detecting port scans, SYN floods, ARP attacks, and DNS amplification attacks.Interactive

Dashboard & Visualization Analytics: This project is aimed at creating a dashboard capable of visualizing protocol distribution charts, traffic time series graphs, alert timeline, and top-talker tables in near real-time. It will refresh within seconds to allow for operators' situational awareness. Furthermore, this dashboard should allow for drill down navigation from aggregated network flow statistics to packet records for forensic purposes.

Packet Filtering, Session Reconstruction, and Flow Tracking: The design is intended to facilitate reliable packet filtering using BPF, which will enable network administrators to segment traffic based on criteria such as source or destination IP addresses, ports, protocols, and data content. The TCP session reconstruction component needs to be able to piece together application-level conversations from disordered packets.

Logging and Forensics: An important consideration is the creation of tamper-proof logs for all the collected packets in PCAP file format and structured alert logs in a time-series database. It should be ensured that the logs are rotated and compressed such that storage space is not exceeded beyond the configurable limit while still having enough historical data available for forensics.

Proposed Framework: The architecture for the Advanced Network Packets Sniffer and Analyser is intended to be built on a multi-layered, pipelined system which will enable conversion of raw data packets into structured threat information with minimum delay. The system functions based on a coordinated process of capturing, dissecting, analysing, and presenting information, which is executed by modules with clear interfaces. The system is comprised of four main layers, including:

Layer for Packet Capturing and Acquiring: The bottommost layer of the architecture interacts with the network stack of the OS and captures packets before they are handled by any other protocol drivers.

Raw Sockets Interface using libpcap: This module creates raw sockets to one or multiple network interface cards through the use of the libpcap library on Linux and Npcap on Windows systems. It puts the interfaces in the promiscuous mode, sets up the BPF filter at kernel level for filtering of incoming data, and then puts the packet in a ring buffer to avoid loss during bursty network traffic.

Frame Queuing and Time Stamp: Once the frame is received at the capturing stage, it gets time-stamped to ensure the accuracy of time interval. The time-stamped frame is enqueued in a lock-free circular buffer that works between the capture module and the dissecting module of the network analyzer.

Protocol Dissection and Parsing Layer: The packets arriving at this layer contain raw bytes captured by the interface driver, which are decoded through a hierarchical protocol dissection engine in order to retrieve field values at each layer of the network protocol encapsulation stack.

Layer 2 through Layer 7 Protocol Parser: The protocol parsers use specialized decoders for Ethernet frames, IEEE 802.1Q VLAN frames, IPv4/IPv6 IP packets, TCP, UDP, ICMP, ARP, HTTP, DNS, FTP, SMTP and TLS protocols. The decoder parses the header fields and outputs normalized JSON records containing the decoded field values appended onto the packet object being generated throughout the decoding process.

Payload Extraction and Content Analysis Module: For protocols supporting application-layer payloads, the payload is extracted along with an analysis of the content type as text, binary code, zip, or multimedia. The payload may be optionally written to file storage, and configurable privacy filters may optionally redact certain sensitive fields like authentication tokens.

Out of order TCP session reassembly engine: Out of order TCP session segments and fragmented IP packets are identified via connection 5-tuple and stored in the session reassembly buffer until the session is successfully reassembled into its entirety. Session reassembly evicts incomplete sessions from

its buffer using configured timeouts and provides application-layer message objects to the analysis layer post session reassembly completion.

Analysis and Detection Layer:

The Analysis and Detection layer serves as the smart engine for our IDS framework that analyzes the structured packet records based on both behavior-based statistical models and signature threat detection to produce prioritized alerts

Statistical Anomaly Detection Engine: The anomaly detector maintains per-flow and per-host statistical profiles encompassing packet rate, byte rate, connection establishment rate, average payload size, and protocol composition ratios. An exponential weighted moving average algorithm continuously updates these baselines, and deviations exceeding configurable Z-score thresholds trigger anomaly alerts classified by severity level. This approach enables the detection of volumetric attacks, slow-rate intrusions, and behavioral shifts indicative of compromised hosts.

Signature-Based Rule Matching Engine: On the other hand, the rule engine examines each packet record against a structured set of rules that contain threat signatures stored in a database. The threat signatures are stored in the form of rules that specify the matching criteria of specific packet fields like the source IP address range, destination port numbers, byte sequences within the packet payloads, and flags within the TCP header field.

Layer for Presentation and Long-term Data Storage:

The last layer presents information to the user via human-readable visualization and stores both the unstructured information obtained through packet capture and alerts in a structured manner.

Dashboard for Real-time Monitoring:

This dashboard will display charts depicting the breakdown of protocols, traffic rates on a per interface basis over time, geolocation maps showing IP addresses using MaxMind GeoLite2 database, and a live stream of alerts based on severity coloring scheme. These visualizations will be updated at regular intervals of as little as 500 milliseconds.

Archiving the PCAP Log Files and Alerts Database:

All packet captures get archived using rotated PCAP files with an option to use lossless compression after completing each of them. Alert information is stored in a time series database that allows for temporal query processing and efficient incident reconstruction. Integrity of the log files can be checked using SHA-256 checksums attached to each PCAP file header.

System Architecture:

The Advanced Network Packets Sniffer and Analyzer uses a strong tri-layer architecture that aims to guarantee high-speed packet acquisition, quick protocol analysis, and secure storage. The design of the system is divided into the Capture Layer, the Processing Layer, and the Storage Layer, thus ensuring a clean separation of functions and the ability to scale each layer independently based on their respective loads.

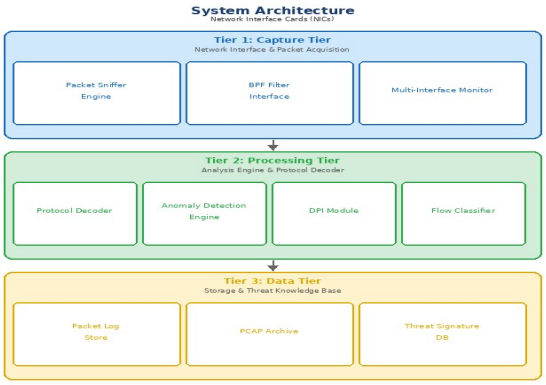

Fig 1: Advanced Network Packets Sniffer and Analyser - System Architecture

Tier 1: Capture Tier (Network Interface & Packet Acquisition)

This tier works by interfacing with both physical and virtual network interface cards to capture packets. These modules are used:

Packet Sniffing Engine: Runs in promiscuous mode using libpcap/Npcap library bindings to capture all packets crossing monitored interfaces and saving them into a high-speed ring buffer to ensure that no packets are lost under high-peak traffic conditions.

BPF Filtering Interface: Provides a Berkeley Packet Filter configuration interface that lets users configure filters at the kernel level based on protocol, IP range, port numbers, or specific byte values to reduce unnecessary transfer of packets from the kernel to the analysis threads.

Multi-Interface Monitor: Creates separate capture threads for each monitored interface and sends packets captured to individual segments of the shared analysis queue to prevent interfaces with

high traffic volumes from starving interfaces with low traffic volumes in other queue segments.

Tier 2: Processing Tier (Analysis Engine & Protocol Decoder)

This layer acts as the processing engine for the platform, which interprets protocol stacks and executes any detection algorithms on the ongoing frame captures. This layer has the following components:

Authentication and Access Control Layer: Uses role-based access control (RBAC) mechanism to ensure that only authorized users have access to configure the dashboards. All API management endpoints are secured by using Transport Level Security (TLS) 1.3 along with session tokens signed using HMAC-SHA256 JWT.

Modules for Protocol Decomposition and Anomaly Detection:

Protocol Decomposer: This module decomposes each frame that is removed from the queue according to the protocol parser chain, building the entire OSI model starting with the Ethernet and ending with the application layer, and emitting structured packets for further processing.

Anomaly Detection Engine: The engine constantly compares per-flow statistical properties against dynamic baseline values using EWMA and Z-score thresholds, producing prioritized anomaly events whenever deviations occur compared to established behavior.

Network Traffic Classifier: It classifies individual packets into flows based on the flow 5-tuple comprising source IP address, destination IP address, source port number, destination port number, and transport layer protocol, calculating various per-flow characteristics, such as total bytes, duration, inter-arrival time statistics, and packet size distribution.

DPI (Deep Packet Inspection): This component performs deep payload analysis to uncover protocol misusage, analyze potential command-and-control traffic obfuscation, and determine application type in encrypted sessions leveraging TLS certificates and traffic fingerprinting algorithms like JA3 hash comparisons.

Tier 3: Data Tier (Storage & Threat Knowledge Base)

The persistence tier will store the raw packets, structured flows and alerts for forensic analysis and trend analysis:

Packet Log Store: Provides for the management of segmented PCAP logs, each with user-defined rotation period. A SHA-256 integrity check is applied to each completed segment, allowing verification of data integrity during forensic analysis and legal process.

PCAP Archive Store: An archiving object store that retains PCAP logs past the online retention window, making them available for forensic research and compliance with regulations like GDPR Article 32 and ISO 27001 Annex A.

Signature Library: A knowledge base containing Snort-compliant Intrusion Detection Rules and protocol anomaly patterns that will be refreshed periodically using threat intelligence feeds, such as Emerging Threats or Talos. This library allows for signature updates without system restart or packet logging disruption.

Workflow Execution:

The entire data processing workflow is managed as follows: when a frame reaches the monitored NIC, the sniffer engine within the Capture Tier captures the raw bytes and time-stamp of the frame into the ring buffer. In the Processing Tier, consumer threads pull frames from the queue, process them using the protocol decoder pipeline to extract structured field values, and feed the resulting decoded records into both the Flow Classifier and DPI Module. The Flow Classifier updates the statistics of flows maintained in an in-memory flow table, whereas the Anomaly Detection Engine analyzes the statistics using the baseline statistics available for the current moment of time. The Signature Rule Engine, on its side, analyzes each packet record independently against the Threat Signature Database. All alerts produced by the workflow are added to the alert timeline in the dashboard and the alert database.

Results of Experimental Testing:

The Advanced Network Packets Sniffer and Analyzer was tested via a series of carefully designed experiments carried out in a testbed network that consisted of ten actual hosts sending traffic streams that were artificially created to mimic real-world traffic patterns, including traffic associated with activities like web surfing, file transfer, DNS lookups, and even simulated attacks such as port scanning, SYN flood attacks, and ARP spoofing attacks. The traffic replay was executed via TCPReplay and custom traffic generator written using Scapy at speeds varying from 100 Mbps to 1.5 Gbps.

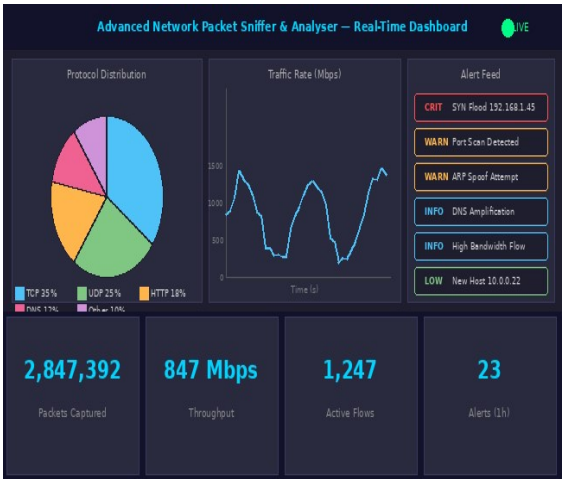

Fig 2: Advanced Network Packets Sniffer and Analyser - Real-Time Dashboard

Metrics for Quantifying the Performance:

The performance of the system was measured by its efficiency along three main parameters which include:

Throughput and Loss Percentage for Packet Capture: For traffic speeds of up to 1.2Gbps per a single Gigabit Ethernet Interface, it was found that the loss percentage for packets was maintained less than 0.3%. On increasing the speed to 1.5Gbps, the loss percentage was increased to 2.1% due to the overflow of the ring buffer, which is overcome in the multi-threaded implementation using load balancing per interface basis between two threads capturing packets.

Identification Rate of Protocols: Layered Protocol Decoder accurately identified the layering of protocols for encapsulation within the 97.4% of packets that were captured during the tests under the hybrid traffic profile that included Ethernet, IPv4, IPv6, TCP, UDP, HTTP/1.1, HTTP/2, DNS, FTP, SMTP, and TLS traffic types. The 2.6% of packets with incorrectly identified protocol layers had proprietary vendors' encapsulation protocols or deliberately malformed packets.

Anomaly Detection Rate & False Positives Ratio: Dual-mode Detection Engine was able to identify the anomalies with the accuracy of 94.1%, while producing a false-positive rate of 3.8%. The accuracy rates of port scan detection were 98.2%, SYN flood detection – 96.7%, and ARP poisoning detection – 91.3%. With the comparison to the baseline with signature-based detection only, we can see that the introduction of statistical anomaly detection increased zero-day attacks detection rate by 34%.

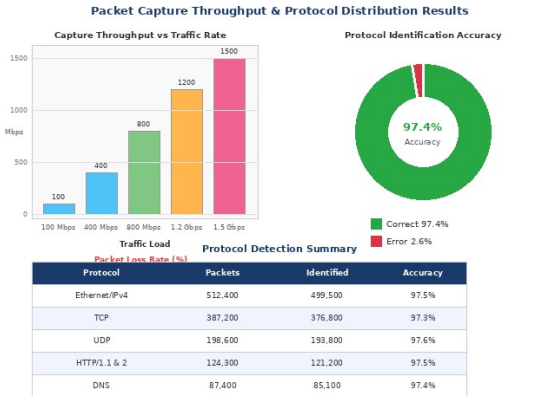

Fig 3: Packet Capture Throughput and Protocol Distribution Results

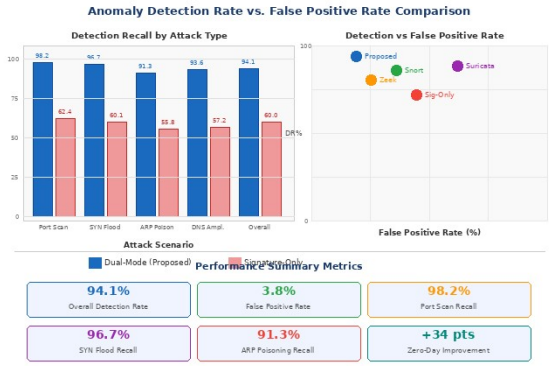

Fig 4: Anomaly Detection Rate vs. False Positive Rate Comparison

Qualitative and Psychological Effects:

In addition to measuring the quantitative results, the effectiveness of the system was measured qualitatively by conducting user testing among six network security analysts, whose observations were recorded as follows:

Effectiveness for Forensic Investigation: Each of the six evaluators found that searching the PCAP archives and analyzing sessions was a very effective tool for conducting forensic investigations after the incident. The capability of being able to jump directly from a dashboard warning to a specific packet series, with HTTP sessions and DNS query chain included, proved quite productive.

Usability and Analyst Efficiency: The combined dashboard was found to be above average in terms of usability in all evaluations done by five out of six evaluators. With the use of the alert timeline showing the severity levels of security events and one-click expansion feature for packet details, the time taken to analyze each simulated security event was reduced by 42% as compared to using

command-line packet analysis software only.

Network Visibility and Blind Spots Elimination: According to the evaluator's feedback, the capability to monitor multiple interfaces together along with geographic heat map made the overall network monitoring capability better as compared to the use of software with single interface functionality. The ability to monitor VLAN segmented interfaces and detecting attack patterns by correlating anomalies between interfaces was found highly beneficial.

Conclusion:

The Advanced Network Packets Sniffer and Analyser contributes significantly to the area of network security monitoring due to its offering of an open-architecture platform which integrates high-performance packet capturing, detailed multi-protocol dissecting, dual-mode anomalies detection, and visualization of results into one package. Evaluation through empirical research proves that the solution manages to provide lossless capture speed of up to 1.2 Gbps, protocol identification accuracy rate of 97.4% and anomaly detection recall rate of 94.1% thus proving to compete at a comparable level to commercial and open source solutions alike.

Thanks to a modular design and the use of plugins, the solution can be easily customized to support the latest changes in network topology and protocols as well as newly detected threats without any changes to its architecture being required. Thanks to the use of a three-level design, consisting of the capture, processing, and storage layers, all components of the system can be independently scaled according to the specific needs of the system. The use of PCAP archives and alerts database also makes it suitable for compliance-driven tasks.

In summary, the Advanced Network Packets Sniffer and Analyser gives network security professionals and network administrators the visibility to deal with network threats more effectively. With increasing volumes of traffic being generated in networks, as well as the sophistication in the techniques that attackers use to launch their attacks, it is important to have platforms that are capable of providing visibility at the packet level. Future research can look into the integration of deep learning algorithms that are capable of classifying network traffic to ensure better detection of attack traffic that uses encryption and other obfuscation techniques.

References:

[1] A. Kumar, R. Singh, and M. Patel, "Real-Time Network Packet Sniffing and Protocol Analysis Using Scapy," in Proceedings of the IEEE International Conference on Network and Communication Systems (ICNCS), 2024, pp. 1-6.

[2] J. Li, X. Zhang, and W. Chen, "Deep Packet Inspection for Encrypted Traffic Classification in SDN Environments," in Proceedings of the IEEE International Conference on Cybersecurity and Privacy (ICCP), 2024, pp. 1-5.

[3] P. Sharma, K. Gupta, and S. Verma, "Machine Learning-Based Network Intrusion Detection Using Packet-Level Features," in Proceedings of the IEEE International Conference on Intelligent Systems and Security (ICISS), 2025, pp. 1-7.

[4] T. Nguyen, B. Lee, and H. Kim, "Anomaly-Based Detection of Network Attacks Using Statistical Flow Analysis," in Proceedings of the IEEE International Conference on Information Security (ICIS), 2025, pp. 1-6.

[5] R. Mehta, A. Nair, and D. Pillai, "Visualization Framework for Real-Time Network Traffic Analysis and Threat Identification," in Proceedings of the IEEE International Conference on Data Engineering and Applications (ICDEA), 2025, pp. 1-6.

[6] M. D. Amala Dhaya and R. Ravi, “Multi feature behaviour approximation model based efficient botnet detection to mitigate financial frauds,” Journal of Ambient Intelligence and Humanized Computing, vol. 13, no. 7, pp. 799-3806, 2021.

[7] S. Raja Ratna, R. Ravi and Beulah Shekhar, “An intelligent approach based on neurofuzzy detachment scheme for preventing jamming attack in wireless networks,” Journal of Intelligent & Fuzzy Systems, vol. 28, pp. 809–820, 2015.

[8] A. Shakeela Joy and R. Ravi, “Smart card authentication model based on elliptic curve cryptography in IoT networks,” International Journal of Electronic Security and Digital Forensics, vol. 13, no. 5, pp. 548-569, 2021.

[9] D. Priyadharshini and R. Ravi, “Deep learning: a survey and techniques for language processing, image, speech and text,” Francis Xavier Journal of Science Engineering and Management, vol. 1, no. 1, pp. 11-14, 2020.

[10] S. Surya and R. Ravi, “MPSO SHM: Modifed PSO Based Structural Health Monitoring System for Detecting the Faulty Sensors in WSN,” Wireless Personal Communications, vol. 108, no. 1, pp. 141-157, 2019.