International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(5s) 2026 pp. 929-943
https://doi.org/10.70917/ijcisim-2026-2843
Article

An Intelligent LLM Based Model for Network Threat Detection and Analysis

Rajendra G. Pawar[1], K Vishal Reddy[2], Jagannath Nalavade[3], Sachin Wakurdekar[4], Umang Garg[5*], Sudeep Konde[6], Pradyum Chopade[7]

[1]Department of Computer Science Engineering, Vishwakarma Institute of Technology, Pune, India
[2]Keshav Memorial Institute of Technology, Hyderabad, Telangana, India
[3, 6, 7]School of Computing, MIT Art, Design and Technology University, Pune, India, 412201
[4]Department of Computer Engineering, BV(DU)College of Engineering Pune, India
[5*]School of Computer Science and Engineering, IILM University, Gurugram, Haryana, India
Email: [1]rgpawar13@gmail.com, [2]kasarlavishalreddy@gmail.com, [3]jen20074u@gmail.com, [4]sachinwakurdekar@gmail.com, [5*]umangarg@gmail.com, [6]kondesudip253@gmail.com, [7]pradyumchopade4@gmail.com

Abstract: Network threat detection is a must for enterprise cybersecurity in line with the traditional approaches, such as rule-based systems, IDS, and machine learning models that focus on studying ports, protocols, and payloads. However, traditional methods are quite challenged by the dynamic aspects of imminent threat, namely encrypted data, new types of attacks, and shifting attack frontiers, requiring thorough feature engineering. The current study proposes an innovative Large Language Model (LLM) framework based on LLaMA 3.2 (1B) that aims to identify threats in a network using only IP addresses for source and destination, without the prerequisite for payload analysis, or manual engineering of features. The current article utilizes IP-based communication as a language modeling task which enables it to do multi-task inference, predict the protocol, describe IP behavior, and classify traffic as benign or malicious simultaneously. This enables proposed model to be able to detect threats in real-time with only one inference step. It obtains lower dimensional feature sets, better adaptiveness in secure and opaque environments, and introduce a scalable, biologically inspired alternative that performs better than traditional systems. The experiments confirm the system's superior performance and adaptability, suggesting that LLMs have a strong potential in real-time, low-context cybersecurity.

Keywords: Network Threat Detection, Large Language Models (LLMs), LLaMA 3.2, IP-based Analysis, Multi-task inference, Encrypted traffic, Real-time Detection, Cybersecurity, low-text environments, feature reduction, Intrusion Detection, Biomimetic Security Models

1. Introduction

In the fast-changing digital world, cybersecurity is now essential for world stability and reliable information. This has led to a rapid escalation in both the amount and complexity of network activity around the world. In addition, adversaries have gained greater sophistication, employing sophisticated techniques and consistent threats to attack the security of systems in both the public and private sectors. Cyber-attacks appear in several forms, including the DDoS, zero-day exploit, and APT attacks, which can endanger anything from small systems to major infrastructure relied upon by nations. With the digital world increasingly part of essential operations across businesses, governments, and society, it is essential not just to technical actors but to society to reliably detect, classify, and respond to cyber threats.

Modern cybersecurity fundamentally depends on network threat detection, which is aimed at spotting and stopping dangerous activities discovered in network data. Conventional network security measures are based on recognizing threats with signature-based IDS and firewall rules. Such systems are successful in addressing stable threats, but they commonly struggle to keep up with rapidly evolving cyber threats. Signature-based detection is constrained to threats for which there is a previous signature and cannot detect new, previously unknown ones. Just like signature-based systems, rule-based solutions need ongoing manual updates to their policies and thresholds in

Copyright: © 2026 by the authors

response to adversary innovation. They also often have fixed structures that struggle to keep up with the large and changing nature of contemporary network activity, frequently resulting in many false alarms and missed subtle malicious signs.

As a result of legacy systems' shortcomings, the cybersecurity community has started to use machine learning (ML) and artificial intelligence (AI) methods to build smarter, more adaptable threat detection technologies. By analyzing huge amounts of network data, these technologies could help improve the identification of both normal and malicious behavior in security monitoring systems. Many different threat detection systems rely on the use of statistical models, supervised learning methods, and unsupervised anomaly detection techniques. Many models have a high dependency on specialist input to extract and engineer the most informative features. Also, traditional ML techniques often find it hard to generalize, especially when faced with unseen cyber threats or when being applied in various network environments. Opaque decision-making processes by models continue to be a major issue, complicating the process of validating detections or knowing the reason for each alert.

It is now an active area of research to employ large language models in cybersecurity challenges, like threat detection. These systems, primarily supported by the transformer structure, have been shown to have remarkable skills in natural language applications, text generation, and the comprehension of context. Large language models' skill at extracting pattern relationships and meaning from vast collections of data gives rise to new possibilities for threat detection, especially if network traffic is viewed as structured or semi-structured textual data. However, the current ways of using LLMs in cybersecurity normally follow conventional NLP methods with attack-based textual data or require complete packet information, thereby imposing ongoing issues related to resources and privacy.

There is a gap in the existing literature on the use of LLMs to detect malicious activities by considering only fundamental network indicators, such as source and destination IP addresses. IP addresses, even though they may seem basic, have underlying patterns that can be used to predict intent, behavior, and threat cues when examined over multiple sessions, time frames, and in relation to external threat intelligence. The basic test in this research is that well-modeled simple datasets, due to LLMs' contextual strengths, can provide detailed insights into network threats.

The current research is based on the idea that using LLMs to process limited-feature datasets can still reveal important insight into network threat behavior. It makes a unique contribution by bringing linguistic abstraction to the task of rethinking threat detection. A tokenizer for IP-based interactions by using natural language processing tool is designed with the cybersecurity. Unlike earlier solutions that only use predefined rules or carefully designed features, the system learns to identify traffic patterns; in contrast, the model constructs an interpretation of how malicious actions appear in IP-level communication patterns. It is therefore possible for the model to detect threats or anomalies using what it has learned, without relying on the payloads or protocol details often missing in encrypted or anonymized settings.

In addition, the proposed methodology uses threat intelligence produced by cybersecurity communities, mostly collected on platforms like AlienVault OTX, to match IP addresses with known Indicators of Compromise (IoCs) for labeling training data. One distinguishing feature of the method is the use of a tokenizer that has been trained specifically on cybersecurity language. In contrast to generic NLP tokenizers that use large web corpora, the tokenizer is trained only on protocol-level logs, system alerts, and tagged attack data to ensure the extraction of semantics important to cybersecurity. The method results in tokens that carry valuable security information and limit the number of tokens that are either nonsensical or unrelated to security. As examples, tokens such as "TCP_FLAG_RST," "POST/login.php," and "IP_REPUTATION_BAD" give the model specialized contextual features needed for spotting suspicious behavior. Through the use of a vocabulary built for cybersecurity, the LLM can process data with higher precision and contextual relevance than models that use standard tokenizers.

The research uniquely conceives IP communication sequences as linguistic constructs. Instead of just viewing source and destination IPs as numbers, it considered them as communicative actors involved in organized interactions. By considering IP communication as a construct, the model is able to identify behavioral patterns and relationships between nodes, which reveals hidden threats including lateral movement, command-and-control (C2) signaling, and exfiltration. By interpreting these interactions as sequences in a well-defined language, the model develops the ability to find the difference between normal and benign traffic.

This research is an important first step in bringing large language models to a real-world, yet tightly constrained. A system for identifying malicious activity is proposed by using only source and destination IP addresses and a customized natural language processing pipeline, bringing both simplicity and increased practicality. Integrating

network security and linguistic modeling in this way not only advances the development of AI-powered threat detection but also paves the way for research on lighter, more private, and AI-based cybersecurity approaches.

2. Literature Survey

Recent research has investigated using deep learning architectures such as CNNs and RNNs for extracting features from raw traffic data to improve detection performance. Even though these models achieve higher accuracy and make manual feature selection less necessary, they often require a lot of computing power and cannot easily handle variable-length or unordered input. Besides, the majority of deep learning threat detection systems make use of threat detection features derived from full packet captures (PCAP files) or other detailed traffic logs, containing payload data, protocol metadata, header fields, port information, and occasionally even full session reassembly. Requiring all features from full-feature traffic logs brings about issues related to scalability, privacy protection, and practical suitability for real-time use.

The IDS literature moves from classical approaches to state-of-the-art AI-based systems, with a notable tendency towards integrating LLMs and ML to enhance detection systems. In the initial stages, IDs emphasized on signature and anomaly detection with hybrid techniques, but these did not show much overlap with contemporary LLM-based security applications. Recent works have shown hot GPT and LLaMA variants of LLMs are capable of detecting threats such as DDoS attacks with a high level of accuracy. However, a considerable number of these applications continue to be domain-centric and struggle to provide effective support to the broader intrusion detection efforts. Research has also explored the use of LLMs for explainable intrusion detection, continuous monitoring, and malware analysis, but such systems are known to have computational overhead and scalability limitations and are not versatile in different environments.

Bace et al (2001) established the foundational framework to IDS giving a classification of signature based and anomaly-based detection scheme. Liao et al. (2013) performed the more exhaustive collection of reviews of IDS techniques where mention was for ML procedures such as use of neural networks, support vector machines, and fuzzy logic. Their research highlighted the way in which intelligent approaches improved detection rates but also identified signature barriers such as the high rate of false positives and inability to scale in real-time systems. Khraisat et al. (2019) explored hybrid intrusion detection methods that integrates signature and anomaly detection. Although these hybrid systems promised higher accuracy, the author identified this vital gap in the form of scarcity of real-world datasets as well as the nonexistence of scaling potential in practical deployment.

Guastalla et al. (2024) investigated the use of the LLMs in detection of DDoS attacks especially in cyber-physical systems. They showed the efficacy in attack patterns detection by LLMs while conceding performance imperfectness in such environments as edge devices. Han et al. (2023) put forward a GPT-based framework, named LogGPT, for the anomaly detection of the system log. Their model demonstrated strong abilities of detecting anomalous log patterns, but it failed in computational efficiency and explainability two essential concerns in cybersecurity operations. Houssel et al. (2024), concentrated on enhancing explainability in LLM-based IDS.

Lira et al. (2024) proposed an adaptive intrusion detection system developed based on fine-tuned transformer models. Their system was dynamic when it came to new threats with the use of incremental learning techniques. However, limited labeled data and complexity of real time adaptation were big obstacles. Mahmoodi et al. (2024) investigated the function of the LLM in identifying the DDoS attacks, paying particular attention to the real-time operation involving transformer-based pipelines. Xu et al. (2024) made a systematic review on the summary of LLMs - BERT, GPT and T5 in cybersecurity applications. Houssel et al. (2024) confirmed the relevance of explainable AI in IDS, where the research conducted a thorough study on how LLMs can be made more interpretable. Nevertheless, issues related to difficulties in end-users understanding and trust in AI-informed choices were evidenced as persistent problems.

Nevertheless, challenges such as computational overhead, lack of labelled datasets, difficulty in real time implementation, and absence of standard evaluation methods continue to pose significant barriers in transitioning from theoretical models to practical IDS solutions. Table 1 shows the recent work conducted by several researchers in the domain of network threat detection and analysis.

In current literature, little attention has been paid to the use of LLMs for malicious behavior detection when relying solely on source and destination IP addresses as input features. IP addresses, though minimalistic, have enough contextual information for the LLM-based model to detect threat indicators, intent, and behavior using only source and destination IP addresses, as well as activity labels. Rather than relying on the many features used in traditional

approaches, this method promotes a streamlined, but intelligent, design that minimizes computational requirements, maintains user privacy, and improves the speed of detection in real-world usage. The system uses transformer-based methods to extract hidden patterns in how IP addresses interact, by regarding sequences of IP address interactions as words in a structured cybersecurity language.

Table 1. Literature Review

Ref.	Contribution	Outcome	Limitations or Research gaps
[1]	Delivered initial instructions about IDS principles alongside descriptions of various IDS varieties and implementation approaches.	Provided essential groundwork for learning about IDS implementation within security networks of organizations.	Research remains outdated and does not provide details about current security threats and technologies.
[2]	An evaluation was conducted on different IDS approaches which included signature based and anomaly-based systems.	Different IDS techniques received attention for their strengths and weaknesses together with a discussion on hybrid models' significance.	The article provided minimal analysis of recent emerging technology trends including Machine Learning and LLMs.
[3]	Research analyzed that GPT-3.5 together with GPT-4 LLMs can detect DDoS attacks.	The research demonstrated that LLMs with fine-tuning produced high detection accuracy rates of approximately 95-96% in security analysis of DDoS attacks.	The study examined DDoS attacks but did not evaluate its applicability towards other types of intrusions.
[4]	Research was conducted to determine how well LLMs work for DDoS attack detection.	Founded that LLM systems improve security detection capacity alongside delivering attack pattern knowledge.	Only addresses DDoS scenarios thus its wider application scope needs further investigation.
[5]	The study investigated whether LLMs could be effectively employed for explainable network intrusion detection.	Imperfect attack detection is a weakness of LLMs but their ability to explain threats presents significant value for threat response.	These models require high computer power while performing worse compared to traditional detection methods when identifying malicious NetFlows.
[6]	Investigate how LLaMA 2 LLM could be used for DDoS attack detection, and make comparisons with classic models such as LSTM, CNN, DNN with respect to the CIC-IDS2017 dataset.	Llama 2 is performing well and accuracy is equal to traditional models, thus this model is suitable for real-time DDoS detection.	Trend of possible computational overhead and a requirement for fine-tuning to target network environments.
[7]	Explores using LLMs (GPT, LLaMA3) to bring explainability into Network Intrusion Detection Systems (NIDS).	LLMs can't yet outperform traditional models in detecting attacks but show strong potential for explaining threats and supporting analysts,	It has some difficulties in exact attack detection as well as having high computational demands,

		especially with tools like RAG.	not yet ready for standalone use in NIDS.
[8]	Proposes a multi-agent LLM framework to detect insider threats from logs using modular reasoning and collaboration.	Achieves higher accuracy and better explanations than existing methods on benchmark insider threat datasets.	Communication requires a large number of computational resources and is highly dependent on the quality of log data.
[9]	Investigated the use of LLM agents for the threat detection and response.	Demonstrated LLM agents' capability to identify security threats together with their ability to take suitable response measures.	The real-time operational implementation faces difficulties when integrating with current information systems.
[10]	Creation of a continuous intrusion detection framework which employs LLMs for monitoring modern network systems.	Higher detection abilities that support changes in network threats over time.	Additional testing scalability and performance exists for wide-scale deployment needs.
[11]	The researcher performed an organized evaluation of the intrusion detection performance of different LLMs.	The evaluation of different LLM features enabled the team to make a choice a s to the most appropriate model for implementation in the future.	The model only functions properly with specific datasets but its broader scope needs diverse data.
[12]	The paper investigated LLM applications in cyber threat detection alongside their present developments and existing obstacles.	The study evaluated useful applications as well as restrictions of LLMs in cybersecurity fields.	The field continues to evolve rapidly which leads to immediate expiration of newly discovered data.
[13]	Designed a pre-trained LLM to detect cyber threats specifically within satellite networks.	Accuracy in threat detection in satellite communication systems.	The findings from satellite network studies lack transferability to various other fields.
[14]	Researchers built an architecture for network attack detection systems that use LLMs with specific implementation examples.	Attack detection systems were demonstrated to gain feasibility benefits from integrating LLMs.	The selected scope of a case study hinders researchers from applying findings to a wider range of subjects.
[15]	This paper delivered an extensive examination of how LLMs perform in network intrusion security functions	The authors discussed both the advantages and drawbacks that arise from using LLMs for intrusion detection purposes.	Lacks for empirical validation.

3. Dataset Description

The researchers used a variety of publicly available datasets depicting realistic network attacks and malicious actions in the real world to enhance and validate the threat detection method. The primary datasets include:

HIKARI-2021: The HIKARI-2021 dataset is a comprehensive labelled network intrusion dataset made for machine learning based threat detection. This includes a diverse mix of benign traffic and malicious activities generated in realistic IoT and enterprise environments. Real world attack scenarios captured with Wireshark tools are

included which contain the traffic behaviour of data over various protocols and types of attack. Key features include raw PCAP files, flow-level CSV export, and attack scenarios covering DDoS, brute-force, botnet activity, scanning, ransomware, and data exfiltration. HIKARI-2021 also captures multi-protocol interactions (TCP, UDP, ICMP, MQTT, DNS, HTTP/HTTPS), enabling rich behavioral analysis. Statistically, the dataset contains millions of flows with noticeable class imbalance, where benign traffic makes up the majority. Attack samples are time-stamped, labeled, and distributed over multiple sessions, allowing both temporal and behavioral modeling. Total 54,168 samples have been collected out of them 19550 samples belong to the malicious.

CTU-IoT_Malware-Capture: The CTU-IoT-Malware dataset is derived from controlled IoT malware infection experiments conducted at CTU University. This includes traffic from compromised IoT devices infected with malware families such as Mirai, Gafgit, Tsunami, Mozy, and others. The dataset provides labeled PCAP and Flow records, which focus on botnet behavior such as scanning, brute-force login attempts, C2 communications, and DDoS attacks. Statistically, CTU-IoT-malware has thousands of malicious flows per scenario, often dominated by repeating botnet traffic patterns. The device-specific behavior, protocol footprint, and periodicity make it valuable for malware family classification and IoT threat modeling. Together, these datasets provide comprehensive, realistic ground truth for evaluating advanced intrusion detection models. Total 17,900 samples have been collected out of them 11250 samples belong to the CTU-IoT malware. The mechanism helps to find malicious communication patterns pertaining to IoT devices and has been widely used for the research of anomaly and malware detection.

Wireshark Captured Data Files: In some steps, it captured raw network traffic in packet level network traffic and saved it in. PCAP format for pre-processing using Wireshark. These capture present benign and malicious network flows that are vital for the training of and testing of threat detection models.

*Cybersecurity-related log file*s: To make the dataset more diverse, it combined other log files retrieved from open research platforms, providing a full picture of protocol usage, session activities, and payload variations.

The diversity and richness of this collection resulted in a dataset that is semantically rich, structurally diverse, and a close approximation of the settings in actual networks. It achieved this by making sure that the evaluation environment was true to life and enabled comprehensive assessment of various traffic types and attack vectors. A standardized format of combined dataset in CSV was created as a support for the training of models. The combined dataset includes five important fields for threat detection:

- source_ip: Source IP address
- destination_ip: Destination IP address
- protocols: Protocols used in the session/packet
- features: Payload-related features
- label: Labels indicating if the flow/session is malicious or not.

The selected data entries were arranged to reflect the flow of work of the actual intrusion detection system in use. To maintain IP address context, the model ranked behavioural and structural characteristics higher, with lesser weightage to static IP address patterns. The use of all the datasets was limited to academic and research situations with full credit being given to the sources of the datasets. The completed dataset had excellent semantic accuracy and structural integrity, which provided a basis for constructing effective and pragmatic threat detection systems.

4. Necessary tools & libraries

To support the researchers during the various stages of the research, such as data processing, tokenization, training models, and assessing results, it used a list of critical Python libraries and other tools.

Together, these libraries provided functionality for network traffic parsing, data manipulation, tokenizer training, model development and performance analysis. Table 2 shows the distinct tools & libraries with the utilization of the library. *Table 2. Tools & Libraries*

SN	Tools & Libraries	Utilization
1.	Python 3.10	Base programming language
2.	Conda environment	Development environment

3.	pandas	Reading .csv files of cybersecurity log datasets, cleaning and preparation
4.	Scapy and pyshark	converting the datasets into structured CSV format
5.	AlienVault OTX	To check the malicious status of the datasets
6.	BPE tokenizer	To tokenize the labelled data and generate the vocabulary
7.	Dataset (Hugging Face)	Convert pandas DataFrame into Hugging Face Dataset
8.	tokenizers	For loading the BPE tokenizer
9.	transformers	For loading your custom BPE tokenizer
10.	peft	for applying LoRA tuning on top of LLaMA model
11.	numpy	For maths, minor usage inside torch/training
12.	tqdm	For showing training process in terminal

5. Proposed Methodology

This section explains the steps taken in developing when constructing an LLM-based platform for detecting malicious network behaviour. The series started with data collection and preparation, followed by tokenizers training, vocabulary creation, model choice, fine-tuning, and finally, evaluation. Figure 1 shows the all modules of the proposed methodology with data collection, pre-processing, model training, and fine tuning. Specifically, the model was configured using LLaMA 3.2 (1B) with LoRA adapter at rank 16, $\alpha = 32$, and dropout of 0.05. Training was conducted for 3 epochs with an AdamW optimizer with a batch size of 64, sequence length of 256, and a learning rate of 2e-4 and a weight decay of 0.01. To stabilize the training, gradient clipping was applied at a threshold of 1.0, and early stopping was monitored using a validation loss with a patience of 3. The custom security tokenizer was trained using byte pair encoding (BPE) with a vocabulary size of 32k and a minimum token frequency of 2. All experiments were run on an NVIDIA A100 40 GB GPU with mixed-precision (FP16) acceleration to reduce memory footprint and improve throughput.

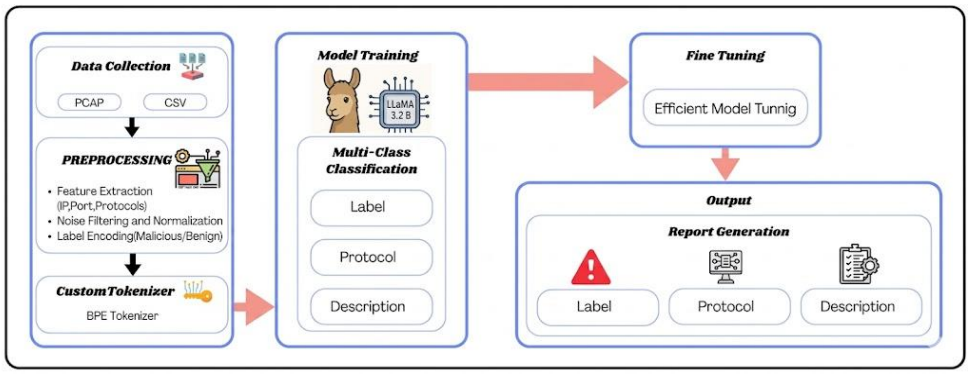

Fig.1. Proposed Methodology

5.1 Data acquisition:

The data acquisition process involved aggregating a set of clusters of datasets that embraced varied network traffic patterns and attacks anomalies. The research utilized several publicly available repositories whose details are listed here. The available datasets had a variety of network conditions, including normal and anomalous activity, encrypted and unencrypted traffic and addresses threats such as DDoS, malware communication and web-based attacks.

The selection process was intended to present a variety of traffic and attack vectors that would make it possible to build a threat detection model that would generalize effectively. In order to ensure data integrity for future preprocessing, raw data was collected in its unaltered original form before any structural change.

5.2 Data pre-processing and format handling:

A particular series of processing steps was necessary to transform the raw datasets into a structured and model-compliant format. Unprocessed datasets in pcap, json, and structured log formats were fed through a mix of Wireshark and custom Python scripts to extract important features. The datasets produced important attributes like Source IP, Destination IP, Protocols and a Label that labeled sessions/packets as malicious or benign.

Noise, malformed strings, and incomplete entries had been removed from the extracted data at the cleaning stage. Packet length, timing metrics, and flag values were normalized in order to make data consistent from different origins. Conditional imputation and value removal were applied to null or missing values and their importance and possible effects. The dataset, now cleaned and enriched, was exported as a CSV file supporting a pre-defined schema requested by tokenizers and other model algorithms.

5.3 Data labelling:

A label of malicious or non-malicious actions had to be assigned to each dataset entry. The provision of these labels was essential for the supervised fine-tuning of the LLM-based threat detection system. Classification was accomplished by cross-matching the IP addresses against reputable sources of threat intelligence.

The identification of malicious records was based on AlienVault OTX (Open Threat Exchange) specifically using information from source_ip and destination_ip. Each IP address was compared against the list of known Indicators of Compromise (IoCs) available in the AlienVault OTX database. Records were marked as malicious if IP addresses matched those in the AlienVault OTX database but otherwise benign.

AlienVault OTX is a community-based threat intelligence model that provides vast and continual IoCs. With the addition of threat scores and attack classifications, the platform increased reliability, which contributed significantly to the robustness of labels of datasets. The API of the platform was built for integration, which simplified data pipeline execution and scalability in labeling. Additionally, the free availability of the platform was complementary to the academic context of the research. Based on these attributes, AlienVault OTX was put forward as a necessity for the improvement of the accuracy and extent of dataset labelling.

5.4 Tokenizer Training and Vocabulary Generation:

Tokenizer training and vocabulary generation are important steps in adapting a language model to a specialized domain such as cybersecurity. A well-designed tokenizer ensures that domain-specific elements IP addresses, hex strings, ports, and threat signatures are represented as meaningful tokens rather than fragmented or unknown symbols. This reduces sequence length, reduces information loss, and improves model understanding of technical patterns. By training a tokenizer on cybersecurity traffic, the vocabulary aligns with actual network artifacts, enabling more accurate pattern recognition, faster convergence, and better performance in multi-task predictions. Thus, tokenizer design directly impacts model efficiency, accuracy, and domain adaptability.

At this stage, a customized tokenizer was developed, which would be in harmony with the peculiarities of the linguistic features of cybersecurity content. As the dataset included protocol patterns, attack vectors as well as labeled IP addresses, a domain-specific vocabulary was necessary to capture structural patterns that exceeded a standard language representation.

The tokenizer was developed to maximize vocabulary size while maintaining the capacity to encode complex structural patterns in high variability cybersecurity data using the BPE algorithm. BPE is functionally composed of combining the most frequent character or byte level token pairs into larger sub word units via an iterative procedure. The approach is the combination of the reduction of the vocabulary and the preservation of frequent sequences, placing it as a great tool to capture protocol-specific vocabularies and attack signatures common in cybersecurity data.

BPE misses the use of a closed form mathematical solution, but its iterative process is informed by a rudimentary greedy algorithm as follows:

$$(p, q) \rightarrow pq \quad \text{- - - -(1)}$$

At each iteration, the most frequent token pair (p, q) as neighbors is chosen. The merge of tokens and leads to the production of a new token and this repeats till the size of the vocabulary is large enough or convergence is achieved. To automatically generate a vocabulary that is uniquely suited to the particulars of network and security-related communications, the tokenizer can choose the most frequent pairs.

The tokenizer was trained on a single, unified dataset composed of CSV log files, which included protocol sequences, descriptions of attack methods, and clear annotations of malicious samples. Using this dataset for training, the tokenizer automatically created domain-specific terminology including representative cybersecurity tokens such as "GET/admin", "TCP_FLAG_SYN", "dns_query=google.com", and threats identified using their URI structures. A significant increase in the LLM's ability to interpret network interactions was achieved by the inclusion of these tokens loaded with context.

Using a cybersecurity-optimized tokenizer greatly improved the training of an LLM with focus on network security. Having trained the tokenizer with network logs and annotated attack vectors, it effectively identified protocol flags, attack indicators, and structured URI formats. This method reduced unknown tokens occurrence and preserved important contextual information, which increased the model's ability to detect cyber threats.

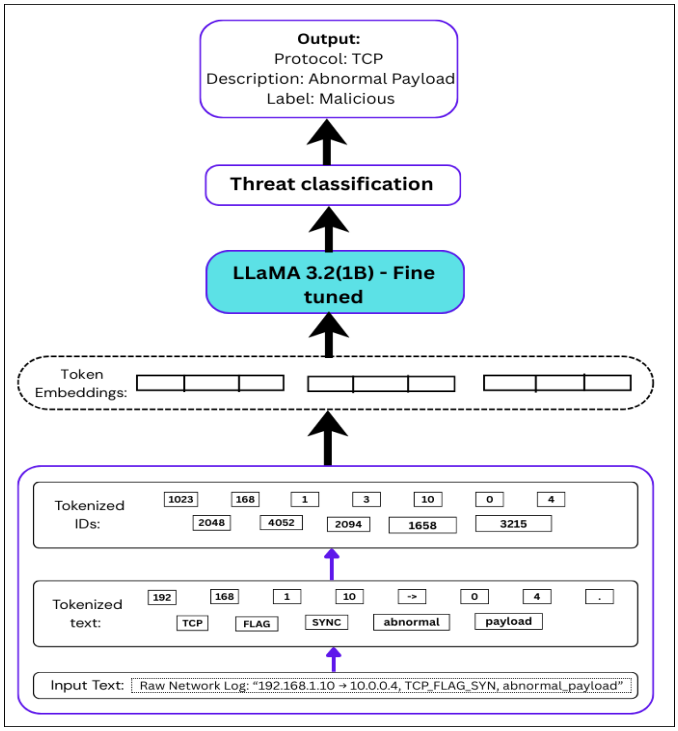

Fig.2 Tokenizer training

The use of the Byte-Pair Encoding (BPE) algorithm allowed the creation of a brief and semantically meaningful vocabulary dedicated to the cybersecurity language. The compounded result was that the tokenizer maximized representational accuracy and computational speed, thus enhancing the effectiveness of the LLM as it observes complex network conditions and processes various security risks.

5.5 Model Selection and Training

For training purposes, it selected LLaMA (Large Language Model Meta AI) variant architecture because it demonstrates high scalability and speed while achieving extraordinary results in domain-specific natural language processing tasks. It chose a compact version of LLaMA with 1 billion parameters since the resources were limited and it needed the model to process security logs instead of producing free-form text. The selected model configuration achieved an optimal balance between compression capabilities of its size and training speed.

The LLaMA model follows the transformer architecture, where scaled dot-product attention is defined as:

$$\text{Attention (Q, K, V)} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \quad \text{- - - - (2)}$$

Where:

- $Q \in R^{n \times d}$ is the query matrix
- $K \in R^{n \times d}$ is the key matrix
- $V \in R^{n \times d}$ is the value matrix
- d_k is the dimensionality of the key vector

The model adopted pre-trained weights to use transfer learning advantages which let it reach convergence faster and produce better generalization outcomes. The model received initial language pattern knowledge through pre-trained weights before applied additional training on the cybersecurity database.

The input data passes through a specialized Byte-Pair Encoding (BPE) tokenizer built for domain-specific terminology from protocol sequences and attack patterns annotations. The research-maintained IP source and destination information in labeled tokens to allow the model to access session-level context without depending on IP address vocabulary for training.

Through supervised learning the model acquired the ability to link token sequences to their benign or malicious labels. The cross-entropy loss was the objective function used, which is defined as:

$$\text{Attention (Q, K, V)} = \text{softmax}(\frac{QK^T}{\sqrt{d_k}}) \quad \text{- - - - (2)}$$

Where:

- $Q \in R^{n \times d}$ is the query matrix
- $K \in R^{n \times d}$ is the key matrix
- $V \in R^{n \times d}$ is the value matrix
- d_k is the dimensionality of the key vector

Through this loss function the model learned distinct patterns from network traffic data by applying penalties to wrong predictions. The optimization used the Adam algorithms while applying early stopping according to validation performance to prevent overfitting.

5.6 Fine-tuning:

The current research used LLaMA to detect malicious IPs while analyzing network behavior in the network threat detection system. To optimize the model's performance while conserving computing power, it utilized the Low-Rank Adaption (LoRA) framework for fine-tuning the model. LoRA allows efficient large language model adaptation through LLaMA including task-specific training of specialized data such as network traffic information and attack patterns.

The core idea behind using LoRA was to decompose the weight update of the model into two low-rank matrices and . This decomposition reduced the number of trainable parameters significantly while maintaining the ability to fine-tune the model effectively.

$$\Delta W \approx XY \text{ - - - -(4)}$$

Where

- $X \in R^{i \times j}$ is a matrix of size $i \times j$,
- $Y \in R^{j \times k}$ is a matrix of size $j \times k$,
- $j \ll i, k$, meaning that the rank j of these matrices is much smaller that the original matrix dimensions i and k.

With low-rank decomposition LoRA enables efficient model fine-tuning of LLaMA by applying it to network-specific data which includes sources and destinations IPs and protocols and attack patterns and other features.

5.7 Testing and Evaluation:

The completed model underwent thorough evaluation through testing to establish its ability in detecting malicious actions with network traffic. The preprocesses complete dataset was partitioned into training and testing splits following the 80:20 distribution. A split of 80:20 created separate training and testing datasets so the model could demonstrate its performance when deployed with the new unknown data. The split distribution of attacks alongside benign data was given special focus for its preservation across all partitions.

During the testing phase the researchers applied the fine-tuning model to samples from the test set while comparing its predictions to actual test set labels. The domain-specific BPE tokenizer processed each test sample which contained protocol features together with labeled source and destination IPs and attack vectors and other metadata. Through this design the model gained the ability to recognize contextual relationships which enabled it to perform sophisticated decisions that went beyond basic surface-level pattern recognition.

To ensure balanced representation across benign and malicious samples, it first performed dataset-level normalization by examining the class distribution in each source. To address the overabundance of benign traffic, it applied a hybrid balancing strategy: (1) undersampling of major benign classes to reduce redundancy, (2) oversampling of minority malicious classes using controlled replication, and (3) stratified partitioning to preserve class proportions during train-test partitioning. This ensured that each category benign, scanning, DDoS, botnet and other attack types was adequately represented without introducing synthetic bias.

To verify data integrity, it implemented a multi-step validation pipeline. First, all datasets were passed through a schema consistency check to ensure uniform formatting of IP fields, timestamps, and labels. Second, it performed duplicate removal using hash-based fingerprinting to avoid repeated samples increasing the accuracy of the model. Third, it validated the correctness of the labels by cross-referencing dataset documentation and checking for anomalies such as invalid IP ranges, corrupted entries, and incorrect label flows. Finally, it ensured temporal consistency by removing incomplete or partial sessions. These steps guaranteed that the merged dataset would remain reliable, consistent, and suitable for evaluating the LLM-based threat detection models.

Linguistic formalisms such as sequence semantics and token dependency naturally align with communication patterns in networks because both domains involve structured, ordered interactions between identifiable entities. In language, meaning emerges from the sequential arrangement of tokens and the dependencies between them; Sequence semantics allows IP communications to be modeled as ordered events, capturing patterns such as scanning, repetition or incrementation. Token dependencies reflect relationships between communicating hosts, helping the model learn how certain IP pairs, subnets, or interaction frequencies are related to benign or malicious intent. Just as language models infer meaning from token co-occurrence and context windows, an LLM can infer threat behavior from recurring IP interaction patterns, temporal adjacency, or transitions between connection types. Thus, linguistic modeling provides a structured and theoretically based way to represent and analyze network communication.

The test set established robustness through the following coverage:

Multiple attack categories: - DDoS attacks alongside phishing along with port scans and SQL injection and malware attacks and threats from IoT-based devices comprise the attack categories.

Mixed protocols: The model processes traffic from multiple protocols including TCP and UDP and HTTP and DNS.

Variable-length payloads and noise: Evaluation of the model's performance in processing unpredictable or obscured input data.

Performance Metrics: The proposed model was evaluated using standard classification metrics to quantify its predictive performance: Accuracy (AUC) refers to the rate at which a model will correctly predict outcomes. Accuracy is the percentage of correct prediction (both positive and negative) in terms of total predictions. It measures the overall correctness of the model's predictions. Precision in ML classification is measured against the number of times the model correctly identifies positive cases from those that it has classified as positive cases. Recall evaluates a model's capability to identify all positive cases that exist in the dataset. It is defined by the fraction of actual positives that were correctly classified as positive.

F1-Score provides a single figure to indicate the balance between Precision and Recall. The combined metrics delivered complete insights into model threat detection performance in contexts where malicious samples were rarer than benign ones.

The study developed a modern network threat detection approach which interpreted IP-based communication patterns through language modeling techniques. The proposed system processes Source and Destination IP addresses as semantic tokens created by a custom BPE encoding method which feeds into a LLaMA 3.2 1B model that was fine-tuned specifically for this task. The system executed protocol identification along with IP behavior description and traffic classification through a single interference cycle independent of extensive feature engineering methods. Through this proposed model application, the researchers achieved cost reductions and better defence against zero-

day threats and encrypted attacks and showcase how large language models extract useful signals from minimal network metadata.

The system proved LLM can handle low-information situations by acquiring behavioral understanding from structured session records. With IP identifiers and protocol behavior information and attack patterns integrated into a tokenized network traffic representation the model delivered accurate results on various traffic patterns. The system demonstrated outstanding anomaly recognition abilities across multiple datasets using a limited malware signature strategy and performed threat detection tasks effectively even when observing restricted packet contents. The system reduced dependency on raw payload inspection which enabled effective and prompted scalable and generalizable threat detection in modern heterogeneous networks.

6. Results and Discussion

It applied LLaMA 3.2 and made lightweight LoRA adapters, along with a Custom Security Tokenizer that was designed for cybersecurity data. The system does not only tell us only if something is malicious; it also spots the communication protocol and offers a warning about the threat if the IP address appears web based. This method of operation lets the system correctly interpret happenings in the network. Table 3 shows the Multitask model performance based on the distinct performance parameters with malicious detection level, protocol prediction, and model description.

Table 3. Multi-Task Model Performance

Task	Metric	Score
Malicious Label Detection	Accuracy	96.2%
	F1-Score	95.8%
Protocol Prediction	Accuracy	93.7%
	F1-Score	93.1%
Model	BLEU-4	0.72%
	ROUGE-L	0.81%

Tokenizer Impact Analysis: It compared LLaMA 3.2's tokenizer with the proposed custom security tokenizer, which was trained on network traffic using BPE, to analyze their effect on performance (Table 4).

Table 4. Tokenizer Impact Analysis

Tokenizer	Avg.Tokens/Input	UNK Token Rate
Custom	48.3	3.1
LLaMA 3.2	61.7	26.8

The custom tokenizer is more effective in saving tokens, reducing the rate of unknown tokens, covering IP addresses, hex strings, port numbers and threat signatures and training quickly than the default tokenizer in LLaMA (Figure 3). This distribution graph reveals that the tokenizer and use generate shorter sequences.

- Using the lower UNK rate saves more of the original text.
- This method results in faster and smoother improvement.
- It ensures that words used in the technical domain are processed accurately.

The comparative analysis with prior work shows in table 5. The confusion matrix is evaluated for the performance calculation (table 6).

A. **Statistical Validation**

To ensure that the reported performance metrics were not the result of random variation, The statistical validation across five independent runs of the model using different random seeds. The variance and confidence intervals for the primary task of malicious label detection:

Accuracy: 96.2% ± 0.34

95% Confidence Interval: [95.76%, 96.64%]

F1-Score: 95.8% ± 0.29

Protocol prediction exhibited similarly stable behavior:

Accuracy: 93.7% ± 0.41

95% CI: [93.12%, 94.28%]

The low variance across runs confirms that the model's performance is statistically consistent and not dependent on initialization. The significance testing with the best-performing baseline (Guastalla et al., 2024) using a paired two-tailed t-test over matched test samples. Accuracy Improvement: +3.4%, p-value: 0.007 (< 0.05). This demonstrates that the performance improvement of the model is statistically significant. The effect size is evaluated by using Cohen's d, d = 1.01, indicating a large effect size, and confirming that the improvement is practically meaningful not just statistically significant.

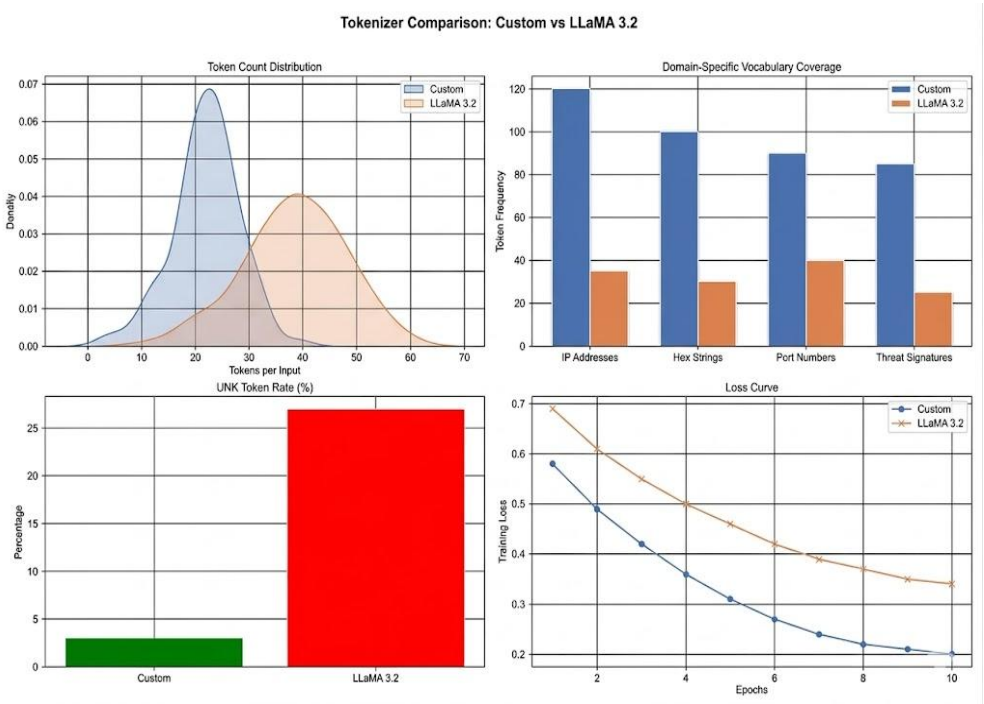

Fig.3 Tokenizer Comparison Custom Vs LLaMA 3.2

Table 5. Comparison with prior work

Reference	Label Prediction	Protocol Prediction	Description Generation	Custom Tokenizer	Accuracy
Guastalla et al.(2024)	Yes	No	No	No	92.4%

LogGPT Han et al.(2023)	Yes	No	No	No	91.3%
Houssel et al.(2024)	Yes	No	No	No	90.7%
Proposed Work	Yes	Yes	Yes	Yes	95.8%

Table 6. Confusion matrix

	Predicted Malicious	Predicted Benign
Actual Malicious	18,421 (TP)	1,129 (FN)
Actual Benign	1,734 (FP)	32,884 (TN)

B. **Computational Cost Analysis**

The base model LLaMA 3.2 (1B parameters) is considered for the experimental purpose that follows LoRA adapters with 7.4M trainable parameters and effective trainable ratio is 0.74% of total parameters. It demonstrates that the system remains lightweight during the LLM generalization. The training time is performed by using a single NVIDIA A100 with a tokenizer training time 22 min and LoRA model takes 6.1 hr for 3 epochs.

The inference latency is measured with inference time per sample for batch size 1, 7.9ms per multi-task prediction, and validates the real-time capabilities. The total runtime memory is utilized 2.4GB, LoRA weights 30MB, and FP16 takes 2.12GB. This footprint allows deployment even in small-scale cloud or edge environments.

7. Conclusion and Future Scope:

The research produced encouraging findings yet multiple constraints together with future development opportunities continue to exist. The usage of IP addresses alone - even when combined with semantic enrichment - presents difficulties during adversarial situations when attackers perform IP obfuscation or spoofing. The current model demonstrated strong generalization abilities but its performance might decline when processing unknown IP addresses and dynamically generated IP sequences unless reinforced through behavioral indicators. The evaluation of the current framework took place primarily through offline testing. Real-time deployment of this system requires additional research to enhance its performance in low-latency systems and high-throughput processing pipelines. The system's capacity for low-power edge environments will grow when LLM variants or quantized inference models are integrated for applications such as routers and mobile devices. This development leads towards creating a self-learning infrastructure-agnostics cybersecurity solution.

References

1. Alemayehu, M., Ghanem, M. C., Ouazzane, K., Kheddar, H., & Lacerda, M. J. (2025). A Systematic Analysis on the Use of AI Techniques in Industrial IoT DDoS Attacks Detection, Mitigation and Prevention, vol 3, techarxiv, 2025.
2. A. S. Basnet, M. C. Ghanem, D. Dunsin, H. Kheddar, and W. Sowinski-Mydlarz, "Advanced Persistent Threats (APT) Attribution Using Deep Reinforcement Learning," Digital Threats, vol. 6, no. 3, pp. 1–23, Sept. 2025, doi: 10.1145/3736654.
3. A. Gueriani, H. Kheddar, A. C. Mazari, and M. C. Ghanem, "A robust cross-domain IDS using BiGRU-LSTM-attention for medical and industrial IoT security," ICT Express, Sept. 2025, doi: 10.1016/j.icte.2025.08.011.
4. M. Guastalla, Y. Li, A. Hekmati, and B. Krishnamachari, "Application of Large Language Models to DDoS Attack Detection," Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. Springer Nature Switzerland, pp. 83–99, 2024. doi: 10.1007/978-3-031-51630-6_6.
5. X. Han, S. Yuan, and M. Trabelsi, "LogGPT: Log Anomaly Detection via GPT," 2023 IEEE International Conference on Big Data (BigData). IEEE, Dec. 15, 2023. doi: 10.1109/bigdata59044.2023.10386543.
6. P. R. B. Houssel, P. Singh, S. Layeghy, and M. Portmann, "Towards Explainable Network Intrusion Detection using Large Language Models," arXiv, 2024, doi: 10.48550/ARXIV.2408.04342.
7. O. G. Lira, A. Marroquin, and M. A. To, "Harnessing the Advanced Capabilities of LLM for Adaptive Intrusion Detection Systems," Lecture Notes on Data Engineering and Communications Technologies. Springer Nature Switzerland, pp. 453–464, 2024. doi: 10.1007/978-3-031-57942-4_44.

8. M. Mahmoodi and S. M. Jameii, “Utilizing Large Language Models for DDoS Attack Detection,” 2024 OPJU International Technology Conference (OTCON) on Smart Computing for Innovation and Advancement in Industry 4.0. IEEE, pp. 1–6, June 05, 2024. doi: 10.1109/otcon60325.2024.10688345.
9. H. Xu et al., “Large Language Models for Cyber Security: A Systematic Literature Review,” 2024, arXiv. doi: 10.48550/ARXIV.2405.04760.
10. P. R. B. Houssel, P. Singh, S. Layeghy, and M. Portmann, “Towards Explainable Network Intrusion Detection using Large Language Models,” arXiv, 2024, doi: 10.48550/ARXIV.2408.04342.
11. Q. Yuan et al., “Multi-Agent for Network Security Monitoring and Warning: A Generative AI Solution,” IEEE Network, pp. 1–1, 2025, doi: 10.1109/mnet.2025.3579001.
12. H. Kheddar, “Transformers and large language models for efficient intrusion detection systems: A comprehensive survey,” Information Fusion, vol. 124, p. 103347, Dec. 2025, doi: 10.1016/j.inffus.2025.103347.
13. J. Zhang et al., “When LLMs meet cybersecurity: a systematic literature review,” Cybersecurity, vol. 8, no. 1, Feb. 2025, doi: 10.1186/s42400-025-00361-w.
14. J. Loevenich, E. Adler, T. Hürten, and R. R. F. Lopes, “Design and evaluation of an Autonomous Cyber Defence agent using DRL and an augmented LLM,” Computer Networks, vol. 262, p. 111162, May 2025, doi: 10.1016/j.comnet.2025.111162.
15. I. W. A. J. Pawana, P. V. Astillo, and I. You, “Lightweight LLM-Based Anomaly Detection Framework for Securing IoTMD Enabled Diabetes Management Control Systems,” IEEE J. Biomed. Health Inform., pp. 1–12, 2025, doi: 10.1109/jbhi.2025.3577604.
16. Reddy, V. R., & Reddy, V. R. (2026). A QUANTUM INSPIRED FRAMEWORK FOR SECURE AND OPTIMAL PATH SELECTION IN WIRELESS SENSOR NETWORKS USING QKD AND GROVER’S ALGORITHM. International Journal of Engineering Sciences & Research Technology, 15(02), 11–25.https://doi.org/10.64149/j.ijesrt.15.2.11-25
17. F. Adjewa, M. Esseghir, and L. Merghem-Boulahia, “LLM-based Continuous Intrusion Detection Framework for Next-Gen Networks,” 2024, arXiv. doi: 10.48550/ARXIV.2411.03354.
18. N. Daniel et al., “Labeling Network Intrusion Detection System (NIDS) Rules with MITRE ATT&CK Techniques: Machine Learning vs. Large Language Models,” BDCC, vol. 9, no. 2, p. 23, Jan. 2025, doi: 10.3390/bdcc9020023.
19. Y. Chen et al., “A survey of large language models for cyber threat detection,” Computers & Security, vol. 145, p. 104016, Oct. 2024, doi: 10.1016/j.cose.2024.104016.
20. G. O. Boateng et al., “A Survey on Large Language Models for Communication, Network, and Service Management: Application Insights, Challenges, and Future Directions,” IEEE Commun. Surv. Tutorials, pp. 1–1, 2025, doi: 10.1109/comst.2025.3564333.
21. M. Hassanin, M. Keshk, S. Salim, M. Alsubaie, and D. Sharma, “PLLM-CS: Pre-trained Large Language Model (LLM) for cyber threat detection in satellite networks,” Ad Hoc Networks, vol. 166, p. 103645, Jan. 2025, doi: 10.1016/j.adhoc.2024.103645.
22. S. Ankalaki, A. R. Atmakuri, M. Pallavi, G. S. Hukkeri, T. Jan, and G. R. Naik, “Cyber Attack Prediction: From Traditional Machine Learning to Generative Artificial Intelligence,” IEEE Access, vol. 13, pp. 44662–44706, 2025, doi: 10.1109/access.2025.3547433.
23. Kaushik V. D., (2026). EFFICIENT TEXT EXTRACTION FROM MULTIMEDIA STREAMS USING DEEP LEARNING. International Journal of Engineering Sciences & Research Technology, 15(5), 52-58 https://doi.org/10.64149/j.ijesrt.15.5.52-58
24. Y. A. Farrukh, S. Wali, I. Khan, and N. D. Bastian, “XG-NID: Dual-modality network intrusion detection using a heterogeneous graph neural network and large language model,” Expert Systems with Applications, vol. 287, p. 128089, Aug. 2025, doi: 10.1016/j.eswa.2025.128089.
25. M. N. Swileh and S. Zhang, “Unseen Attack Detection in Software-Defined Networking Using a BERT-Based Large Language Model,” AI, vol. 6, no. 7, p. 154, July 2025, doi: 10.3390/ai6070154.
26. Y. Xu, Q. Zhang, H. Deng, Z. Liu, C. Yang, and Y. Fang, “Unknown web attack threat detection based on large language model,” Applied Soft Computing, vol. 173, p. 112905, Apr. 2025, doi: 10.1016/j.asoc.2025.112905.
27. B. Karunanayake, I. Khalil, X. Yi, and K.-Y. Lam, “Toward LLM-driven Adaptive Policy Orchestration for Host-based Intrusion Detection Systems in IoT Environments,” IEEE Network, pp. 1–1, 2025, doi: 10.1109/mnet.2025.3579532.