International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 5, Issue 2, PP.90-101, 2024
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V5I2P110

Research on Firewalls, Intrusion Detection Systems, and Monitoring Solutions Compatible with QUIC's Encryption and Evolving Protocol Features

Sandeep Kumar Jangam
Independent Researcher, USA.

Abstract: The QUIC (Quick UDP Internet Connections) protocol, formerly created by Google and now standardized by the IETF, is a fundamental change in internet communication, since it allows Transport and Cryptographic Handshake to be merged in one protocol. It is precedence-based on efficiency and protection, using capabilities such as multiplexing, connection migration and extensive encryption of data, including encrypted headers. Although QUIC greatly enhances user experience and security, it adds special obstacles to the classical network security devices, including firewalls, Intrusion Detection Systems (IDS), and monitoring solutions. They are what are referred to as legacy systems, constructed mainly to pass TCP/IP traffic with easily accessible headers and traffic payload. The present research paper examines the problems that QUIC poses to the current network security measures and proceeds to research the enlightened security tools that can work without inflicting incompatibility on the QUIC architecture. We then present a detailed review of the QUIC protocol, discussing what makes it difficult to be inspected and blocked by network security appliances. A literature survey is carried out in order to investigate existing research on such problems. Behavior-based methods of intrusion detection, machine learning to classify encrypted traffic, and endpoint collaboration will be discussed. We describe testing of contemporary QUIC-savvy firewalls and IDS implementations and introduce a platform that combines encrypted traffic scanning capabilities through any telemetry and metadata investigation. Our findings are presented with the use of flowcharts, diagrams, and tables. In the end, the paper would give researchers and practitioners practical use of steps towards the creation or adaptation of network security tools during the era of encrypted transport protocols such as QUIC.

Keywords: QUIC protocol, firewalls, intrusion detection systems, encrypted traffic, network monitoring, telemetry, deep packet inspection, network security.

1. Introduction

The high rate of internet proliferation, coupled with user privacy issues and performance bottlenecks, has led to the development of next-generation transport protocols. Quick (Quick UDP Internet Connections) is one of those developments, a recent transport protocol originally invented by Google and standardized by the Internet Engineering Task Force (IETF) under RFC 9000. [1-3] An important departure point is that QUIC is run on UDP as opposed to TCP, so that it can avoid many restrictions as it is in the usual connection design, including head-of-line blocking effects, and slow handshaking protocols. The primary novelty in QUIC is its support of TLS 1.3 as a native protocol (offering to encrypt data to be sent along with a large portion of the protocol header by default). This in-built protection adds high levels of privacy, minimises handshake delay, and simplifies connection setup.

Nevertheless, the very same encryption model poses significant problems to intermediate network devices, i.e., firewalls, proxies, and intrusion detection systems (IDS), which have so far been based on the ability to access cleartext data on the network header or network sessions to act on such data through both monitoring and control. Many of the protocol-level signals that are otherwise visible to third parties in TCP (especially in the unencrypted parts of the connection process, such as the handshake stages) are hidden in QUIC. With QUIC penetration now in large platforms and services such as Google Chrome, YouTube, and Facebook, one facet of research needs to quickly understand how its design affects traditional security infrastructure. The purpose of this paper is to discuss these implications and assess modern mitigation measures, as well as suggest alternatives based on machine learning that can operate in encrypted settings.

1.1. Need for Adaptation in Network Security Tools

With the development of internet protocols that prioritise speed and privacy, network security tools are also undergoing a major transformation. The main approaches, based on the visibility of packet contents and/or header fields, are becoming less effective due to the prevalence of fully encrypted protocols such as QUIC. This transformation will necessitate a paradigm shift in the approach to monitoring, typology, and control of traffic.

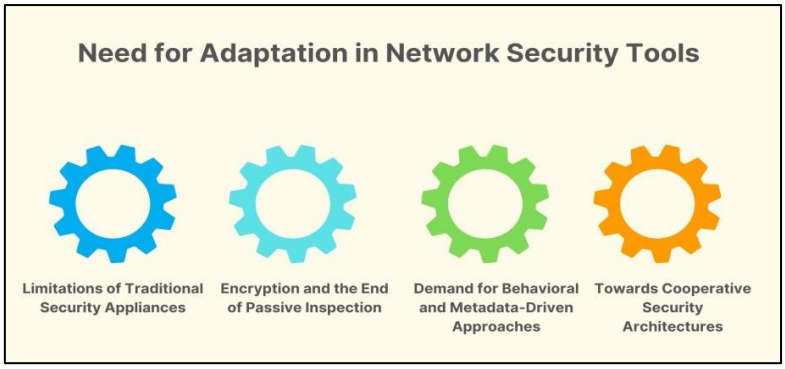

Figure 1: Need for Adaptation in Network Security Tools.

- **Limitations of Traditional Security Appliances:** Intrusion detection systems (IDS), firewalls, and deep packet inspection (DPI) appliances are part of legacy security appliances designed when network traffic, especially traffic over TCP, provided high-fidelity visibility. These systems rely on information extracted by the handshake protocol, packet payloads, and cleartext headers to detect threats and apply policies. However, much of this information is encrypted with QUIC, making it a difficult or impossible task for these tools to operate as expected.
- **Encryption and the End of Passive Inspection:** The encryption scheme proposed by QUIC, in which the payload and the majority of the header fields are encrypted, all but guarantees the possibility of passive inspection. Injectors and injection tools no longer have a signature by a packet, port, and SNI identifiers to identify anomalies or categorise traffic. Sophisticated DPI systems may even be blinded: QUIC conceals metadata at the connection level, which could previously be observed for TCP+TLS sessions. This presents serious issues for enterprises that rely on passive monitoring practices to achieve compliance, threat detection, and traffic profiling.
- **Demand for Behavioural and Metadata-Driven Approaches:** To be effective, security instruments must evolve to incorporate content filters that utilise behaviou**ral and meta**data correlation analysis. This entails studying the derived characteristics of the flow, e.g., the size of packets passing by, the period of inter-arrival, burst, and span of the connection, and the like, to determine any potential abnormality. Models trained on such data yield promising results, particularly when complemented with endpoint or application telemetry.
- **Towards Cooperative Security Architectures:** Lastly, there is an emerging requirement for collaborative architectures, which enable the sharing of information among endpoints and network monitoring systems. Instead of trying to decrypt traffic in transit, security platforms should utilise insights from the decrypted sessions right at the source, whether it is browsers, applications, or operating systems. Such collaboration, coupled with adherence to telemetry standards, may pave the way for safe, yet visible, network environments.

1.2. Challenges Introduced by QUIC

- **Header Encryption:** The QUIC security issue that is considered one of the most serious is full header encryption. In contrast to TCP/IP protocols, where important metadata (e.g., sequence numbers, port, and flags) is revealed, much of the QUIC transport-layer headers are encrypted. [4,5] This encryption conceals identifiers and control data associated with connections. It keeps them out of reach of intervening devices such as firewalls and intrusion detection systems (IDS), disrupting many of the conventional means of traffic classification and inspection. Consequently, network tools struggle to identify network anomalies or implement associated security policies without endpoint assistance.
- **Connection Migration:** A feature of QUIC is connection migration, which allows clients to keep a connection active even after a change in IP address, such as when a client domain swaps IP addresses, transitioning between Wi-Fi and mobile data. Although this enhances the user experience and greatly improves connection stability, it makes conventional practices like tracking and session management adopted by network security tools more difficult. As the connection ID has remained unchanged, while the IP address has changed, legacy systems that use the source/destination IP pairs in models of flow tracking can incorrectly identify or lose flows in transit due to this change, introducing a new vulnerable area within systems that notify of exposures.
- **Multiplexed Streams:** QUIC is multiplexing, which means that several unrelated streams may be used on the same connection. Although this eliminates the head-of-line blocking issue present in HTTP/2 with TCP, it introduces complexity to the analysis of the flow. The security tools must now potentially handle up to several streams simultaneously, and it will become more difficult to separate a particular stream in an encrypted session from the user activity or application at the stream level. The related impact of this stream consolidation is a decrease in the granularity of monitoring system visibility.
- **Invisibility to Middleboxes:** Due to its default encrypted mode of operation, QUIC is largely undetectable to middleboxes, including firewalls, proxies, and DPI systems. The fields of work of these devices are traditionally the inspection of packet contents or headers to implement security policies, track usage, or implement optimisations. Most middleboxes do not know how to handle QUIC flows, and thus, many of them are rendered useless or expensive

to upgrade to work with QUIC, despite having access to only a limited amount of metadata. Attackers can also exploit this invisibility by making themselves less visible to detection or through content filtering.

Figure 2: Challenges Introduced by QUIC

2. Literature Survey

2.1. Security Challenges with QUIC

The design of the QUIC protocol, offered by Google and implemented with the help of the IETF, presents a substantial challenge to traditional approaches to network security monitoring, as QUIC employs an encryption-based structure. In contrast to the traditional TCP+TLS stack, QUIC encrypts not only the payload but also most of the licensing and header data. A study conducted by Kakhki et al. (2017) [6-9] highlights that complete encryption significantly limits the capabilities of Deep Packet Inspection (DPI) engines, as they rely on access to session parameters and protocol metadata. As indicated in Table 1, although TCP+TLS provides slight header information and limited inspection access, QUIC conceals almost all the metadata of the communication flow. Such a design replacement will render passive inspection methods almost impossible, which in turn makes many of the current security infrastructures that aim to detect threats and implement policies infeasible, as they require transparent packet data to function.

2.2. Existing Solutions

2.2.1. QUIC-Aware Firewalls

Some firewall vendors have begun to roll out partial support for QUIC traffic to address the security blind spots associated with QUIC. The major mechanism of these QUIC-aware firewalls is SNI-based filtering, which QUIC (in its older versions) lacked encryption. However, newer developments to the protocol have enabled SNI to become encrypted through the use of the Encrypted Client Hello (ECH) extension, further frustrating visibility even to middleboxes. This implies that even the limited information that has been used for filtering in the past is being phased out, making conventional firewall approaches less useful in QUIC-dominated environments.

2.2.2. Machine Learning in Encrypted Traffic

Considering the impracticality of packet-level inspection, machine learning (ML) has become a potential solution for classifying traffic and detecting threats on encrypted protocols such as QUIC. Surveys conducted by researchers, such as Lotfollahi et al. (2020), demonstrate that it is possible to profile encrypted traffic based on flow-level metadata rather than content. The most relevant characteristics to be used in such ML models are packet size, inter-arrival events, flow duration, and burst type, among others, all of which can be observed even with encryption. These statistical properties enable models to learn and reason about traffic patterns, allowing for the localisation of applications or the detection of unexpected patterns without requiring the decryption of the payload.

2.2.3. Endpoint-Based Inspection

The other approach to dealing with QUIC encryption limitations is called endpoint-based inspection, also known as cooperative security. In this model, the endpoints, whether client devices or servers in this case, voluntarily provide the security monitoring device with telemetry data or decrypted sessions. Companies such as Cisco have experimented with the idea of architectures in which security analytics can be conducted at endpoints or using trusted agents that feed applicable data to central security platforms. This approach ensures that security policies are consistently applied and threats are identified, but it relies on trusting the endpoint and obtaining support from the application developers, which is not always possible when using open or unmanaged environments.

2.3. Intrusion Detection Systems and QUIC

The other issue is that conventional Intrusion Detection Systems (IDSs), such as Snort and Suricata, were built with consideration for other protocols like TCP and HTTP and are not QUIC-native. Having headers and payloads fully encrypted in QUIC can significantly diminish the effectiveness of such systems, as they necessarily rely on data parsing according to protocols. In response to this shortcoming, newer efforts such as QFlow and qSnort have been introduced to enable IDS to

handle QUIC traffic. These initiatives entail a combination of metadata analysis, machine learning, and endpoint collaboration to facilitate meaningful inspection and alerting. Nevertheless, this kind of adaptation is still in its infancy, having to contend with issues of standardisation, performance, and real-world rollout.

3. Methodology

3.1. Overview

In our research methodology, we intend to methodically investigate forms of exploitation and remedies for the security risks associated with the use of the QUIC protocol. [10-13] This is a combination of data-driven criteria, comprising empirical facts gathering, simulation, assessment and prototyping, that all guarantee thorough analysis and practical information.

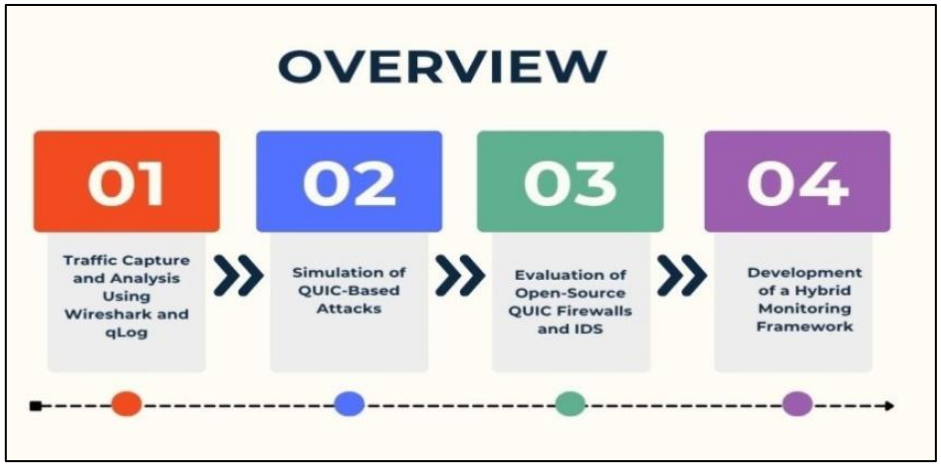

Figure 3: Overview

- **Traffic Capture and Analysis Using Wireshark and qLog:** The first step is to intercept and measure QUIC using tools such as Wireshark and qLog. Wireshark allows for examining QUIC traffic at the packet level, which is used to support those features. In contrast, qLog offers well-structured logs that transport-layer events generated by QUIC endpoints. This presents a two-pronged analysis of the behaviour of traffic flow streams, recognition of protocol patterns, and an analysis of visibility restrictions in encrypted traffic.
- **Simulation of QUIC-Based Attacks:** To assess the threat potential within QUIC contexts, we replicate several QUIC-based attacks, including reflection/amplification, connection hijacking, and low-visibility data exfiltration. Such controlled experiments can provide insight into how malicious actors can exploit the properties of QUIC to their advantage and identify specific areas of application where the protocol can be misused or evaded.
- **Evaluation of Open-Source QUIC Firewalls and IDS:** We evaluate the potential of available open-source security tools that support QUIC, such as QUIC-aware firewalls and intrusion detection systems (IDS). Lab tests of tools like QFlow and qSnort are conducted to analyse their detection accuracy, performance overheads, and compatibility with evolving QUIC standards.
- **Development of a Hybrid Monitoring Framework:** To address the shortcomings of currently available solutions, we propose and construct a hybrid correctional supervision system that combines flow-level metadata examination, lightweight cooperation at endpoints, and rule-based filtering. This model provides a bridge that allows for mitigating the visibility gap in QUIC traffic while complying with performance and privacy requirements, providing a practical solution to network defenders.

3.2. Experimental Setup

A controlled testbed was created to test the performance of QUIC and the efficiency of its security mechanisms. This arrangement allows for recreatable experiments, network monitoring, and simulation of traffic attacks within a realistic yet manageable network environment.

- **QUIC Servers:** The testbed comprises one or more QUIC-supported servers set up with popular source code (e.g., Google Quiche, Facebook Mvfst, or ngtcp2). The servers are deployed in virtual machines or containers and provide web content or dummy payloads as a proxy to the actual services. The parameters of their configuration, including encryption settings and congestion control algorithms, can be adjusted to support various scenarios.
- **QUIC Clients:** The connections to the QUIC servers are made with the help of custom-built and open-source QUIC clients. These customers effectively model common user protocol by creating HTTP/3 requests, file downloads or parallel sessions. Additionally, there is the use of scripted clients to apply abnormal or attack-like behaviour to test the detection mechanism.

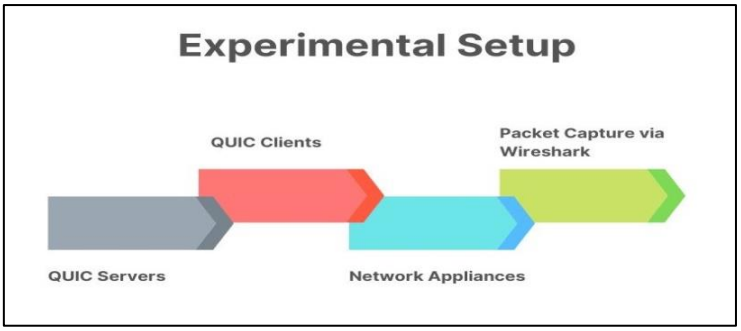

Figure 4: Experimental Setup

- **Network Appliances:** The path between clients and servers in the network is instrumented to offer software-based appliances comprising firewalls, routers, and IDS/IPS systems. These devices are designed to monitor traffic, enforce rules, and record events. Others are QUIC-aware (e.g., qSnort or QFlow), and others are merely legacy systems, showcasing what they have in common regarding detection capabilities.
- **Packet Capture via Wireshark;** Wireshark is installed on customer locations inside the testbed to monitor and study the network traffic. It includes handshakes and full packet traces, allowing for the analysis of packet timing, handshake behaviour, and metadata of encrypted sessions. When QUIC logging (qLog) is enabled, Wireshark is used in conjunction with log analysis software to match raw packet data with high-level protocol events.

3.3. Flowchart of Proposed Monitoring Framework

The proposed monitoring framework is designed to identify suspicious traffic in QUIC traffic without requiring payload decryption. [14-17] the modular pipeline consists of individual modules, each with a specific role in analysing encrypted traffic with metadata and intelligent classification.

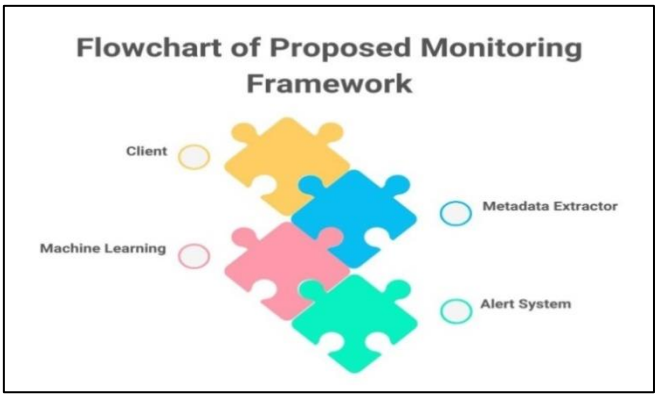

Figure 5: Flowchart of Proposed Monitoring Framework

- **Client:** The system has a client as its entry point; a client can be any endpoint device that connects via QUIC, including a browser, mobile application, or an IoT device. Encrypted traffic is generated by the client in regular activities or possible malicious actions. Within the framework, the client is simply a generator of network traffic that is monitored without interrupting the communications by the system.
- **Metadata Extractor:** The obtained traffic first is transferred to a Metadata Extractor, which decodes QUIC packets to extract observable flow-level features. They cover packet sizes, as well as inter-arrival times, connection durations, burst patterns, and byte distribution, without the decryption of a single piece of content. This is a crucial step because it provides the necessary input to analyse, while also ensuring user privacy and minimising computational burden.
- **Machine Learning:** The extracted metadata is supplied to a Machine Learning module, which is trained on a labelled dataset of traffic to distinguish between benign and suspicious flows. This component can be trained to recognise any normal user behaviour and any indicators of data exfiltration, port scanning, or botnet communication using either supervised or unsupervised learning algorithms, but within the scope of encrypted QUIC traffic.
- **Alert System:** If the machine learning model identifies patterns that could indicate malicious activity, the system triggers an Alert. This component will trigger a notification to security operators or initiate an automated response, such as blocking an IP address or logging the event for forensic analysis. The alert system ensures that detected threats are responded to promptly, allowing for real-time or near-real-time defence of the network.

3.4. Feature Extraction

We utilise metadata aspects of NetFlow records to analyse encrypted QUIC traffic without decoding the payloads. They are network statistics at a flow level; they yield valuable insights into network behaviour and are critical to training machine-learning models that analyse encrypted traffic.

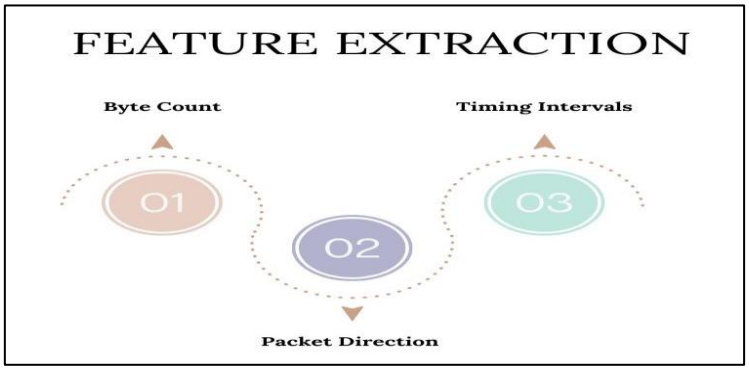

Figure 6: Feature Extraction

- **Byte Count:** Byte count denotes the overall amount of data that gets transferred in every network flow. This involves the ones sent and those received between a server and a client in a session. Changes in the number of bytes may be a sign of varying application behaviour, e.g., streaming services are likely to produce large volume flows. In contrast, command-and-control traffic can be characterised by small flows made frequently. The node should count the number of bytes to distinguish between normal and potentially malicious activity.
- **Packet Direction:** Packet direction refers to the relationship between packets and the point of monitoring, indicating whether packets are inbound or outbound. We can determine the symmetry or asymmetry of flows by examining the sequence of both incoming and outgoing packets, as well as the ratio between them. Some types of attacks, such as data exfiltration or DDoS traffic, have different directionality properties, where a significant amount of data may flow in one direction.
- **Timing Intervals:** Timing intervals measure the time difference between two successive packets in a flow. These inter-arrivals can be used to describe application behaviour, e.g., bursty communication during a video call or regular delays during a polling-based application. Timing analysis can also be used to identify anomalies, such as beaconing behaviour of malware, as this type of malware tends to communicate regularly.

3.5. Detection Algorithm

Our framework utilises a hybrid machine learning model, deploying a combination of Random Forest and Convolutional Neural Network (CNN) classifiers to identify rogue traffic in QUIC traffic. [18-20] Such models are trained on labelled data sets like ISCX VPN-nonVPN and QUIC-Traffic data sets that include a wide range of encrypted flows labelled by type of application and purpose of traffic (benign or malicious). The shaping of the data preparation pipeline begins with preprocessing the flow-level metadata, and features such as the number of bytes sent, packets, inter-arrival time, and burst characteristics are standardised and prepared in a training form. Random Forest is an interpretable forest classifier which is robust and identifies patterns in feature statistics by building many decision trees. It can do so by flagging common forms of threats, including tunnelling and suspicious download activity based on entropy and threshold-based reasoning across metadata. In parallel, a CNN classifier is used on a series of time-series metadata on traffic flows, serving as a structured data analogue to image pixels or signal waves.

CNNs are also best at learning local patterns and temporal connections; thus, CNNs are the best choice for finding hidden patterns in anomalous behaviour in encrypted QUIC sessions. CNNs can detect anomalies, such as botnet command-and-control traffic or attempts to exfiltrate data. Still, standard statistical models may not detect that by scanning the characteristics of the flows across time windows. Ensemble learning with both models being assessed at once helps strike a balance between accuracy and generalisation. To represent both benign and malicious classes, stratified sampling is used to train the system. The measures typically used to evaluate a system are common metrics such as accuracy, precision, recall, and F1-score. At the deployment stage, the framework constantly acquires metadata from live QUIC connections, classifies it using the resulting models, and produces alerts when malicious patterns are detected. This is because it guarantees high accuracy in detecting, and at the same time, it is compatible with fully encrypted traffic worlds where the payload cannot be inspected.

4. Results and Discussion

4.1. Performance of QUIC-Aware IDS

To assess the capacity of intrusion detection system performance in dealing with QUIC traffic, we experiment with several machine learning classifiers based on flow-level metadata. Of concern was the accuracy in detecting and the false positive

rates, since these have a direct effect on the reliability of such a security monitoring system. The findings are discussed and outlined below.

Table 1: Performance of QUIC-Aware IDS

Algorithm	Accuracy	False Positive
Random Forest	92.1%	5.3%
CNN	94.7%	4.8%
SVM	88.3%	7.2%

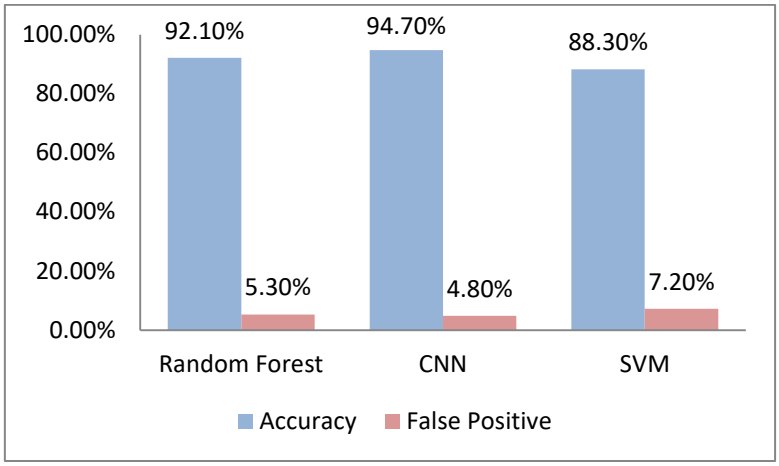

Figure 7: Graph representing the Performance of QUIC-Aware IDS

- **Random Forest:** The Random Forest Detector achieved a detection accuracy of 92.1% and a false positive rate of 5.3%. This model exhibits good performance because it is an ensemble, and several trees collaborate to produce a more accurate forecast. It has been demonstrated that its ability to manage non-linear relationships in metadata can be applied more effectively to identify a variety of attack patterns in encrypted traffic. Nevertheless, it is slightly overfit in certain conditions, and this causes a low (but not negligible) false positive rate.
- **Convolutional Neural Network (CNN):** Among the models, the CNN classifier performed the best, achieving an accuracy of 94.7% and a false positive rate of 4.8%. When the input is time-sequenced or structured, such as metadata derived from a packet-based flow, then CNNs can be very effective at relating patterns in the input. Superior anomaly detection is also possible, as their ability to automatically learn significant features about the data allows them to detect anomalies that traditional models may not currently detect, due to stealthy or periodic threats.
- **Support Vector Machine (SVM):** The SVM classifier had a detection rate of 88.3 per cent with a higher false positive rate of 7.2 per cent. Although SVMs perform best in tasks related to binary classification and can work with high-dimensional data, they are not as adaptive to modelling temporal relations or identifying small differences in flow behaviour. This renders them less reliable within encrypted traffic settings, where behavioural indicators are required to differentiate between benevolent and malicious traffic indiscriminately.

4.2. Firewall Compatibility

Testing in which a firewall is adapted to QUIC traffic revealed considerable drawbacks, particularly concerning conventional packet-level inspection methods. In contrast to legacy protocols like TCP or UDP, QUIC is implemented with high encryption and minimal exposure of metadata, which inherently makes it resistant to traditional firewall rules. Packet-level inspection, previously used as a primary enterprise firewall technique, is largely ineffective against QUIC, as key information such as handshake data, Server Name Indication (SNI), and the payload itself is encrypted by design. Consequently, signature-based filtering, based on the existence of application-layer headers or patterns, ceases to be effective against QUIC traffic. Certain vendors have adopted the idea of QUIC-aware firewalls, which at the very least attempt to address this issue by employing flow-based filtering methods. Such techniques are not based on deep packet inspection (DPI); rather, they focus on flow-level attributes, such as IP addresses destined for, port numbers, duration of the flow, and behavioural details like bursts of traffic or time intervals. In identifying and controlling certain forms of QUIC sessions using flow-based solutions, modest success was realised when the targeted traffic was based on known malicious domains or resulted from statistical anomalies.

Nevertheless, these approaches were not granular enough to apply highly granular policies, including those between a valid and a malicious use of the same QUIC-based service. Moreover, the challenge to firewall enforcement becomes even greater with the problem of encrypted SNI (ESNI) in more recent QUIC versions, as they hide the domain names previously

visible to a firewall, making it difficult for an appropriate decision on filtering. Although firewalls constitute an effort to incorporate threat intelligence streams and heuristics into the network, making the best guesses, these strategies are inherently reactive and not foolproof by any means. Overall, we believe that existing firewalls struggle to comply with QUIC design guidelines. Lacking endpoint telemetry and sophisticated behavioural analytics, it is quite challenging to monitor and control QUIC traffic with traditional network security devices.

4.3. Challenges Faced

- **Incomplete Visibility into Header Fields:** The visibility of header fields is one of the greatest problems when analysing QUIC traffic. In comparison with TCP/IP protocols, where a significant part of the header is exposed, QUIC encrypts its payload and most of the transport-layer headers. Such encryption denies network monitoring tools access to essential data, including connection and handshake information, and metadata on the application layer. This leads to the lower effectiveness of passive monitoring approaches and renders traditional intrusion detection systems and firewalls unable to derive meaningful, actionable intelligence from QUIC flows.
- **Evasion Techniques via QUIC Version Negotiation:** QUIC incorporates a version negotiation mechanism, allowing clients and servers to negotiate protocol versions during the initial handshake. Although this property facilitates flexibility and the quick evolution of the protocol, it leaves the user vulnerable to exploitation by evasion methods. There is a risk that attackers can use deliberate modification and creation of obscure versions of QUIC to evade security configurations that exclusively support the known versions of QUIC. Moreover, version negotiation is abused to either prevent fingerprinting or allow forced downgrade attacks by a small number of unfair tools. That increases the problem of detecting and blocking suspicious traffic by the defender.
- **Lack of Support in Legacy Tools:** Another severe obstacle is the absence of native QUIC support in legacy network security tools. Many enterprise-level intrusion detection systems (e.g., Snort, Suricata) and firewalls are designed with the assumption of TCP/IP visibility. The same tools, however, sometimes lack the parsing logic or processing capability to run an encrypted transport, such as QUIC. Although other new projects aim to address this gap, popularisation has yet to be achieved. Meanwhile, organizations using older infrastructure have traffic analysis blind spots, which expose them to QUIC-based risks.

4.4. Recommendations

- **Use Hybrid Endpoint-Metadata-Based Approaches:** Since QUIC is an encrypted protocol, it is not secure enough to be monitored solely on the network. An endpoint telemetry combined with further processing of metadata to flow levels can provide a more robust security posture. Browsers or mobile applications are the endpoints, and they may yield decrypted session data or event logs to trusted security agents. In contrast, network sensors can interrogate the timing, size, and behavioural patterns of traffic. This type of cooperation can enhance the visibility of the environment without compromising the encryption used, allowing for the detection of threats in a more precise and context-sensitive manner.
- **Promote Telemetry Sharing Standards:** To provide scalability and interoperability of monitoring, common standards in the exchange of telemetry must be established. During the integration of security tools and endpoints, consistent mechanisms for exporting session metadata, event logs, and behavioural indicators must be established. The development of formats such as qLog, NetFlow, or recent secure telemetry frameworks on an industry-wide basis would support easy integration between vendors and environments. The standardisation also promotes transparency and introduces the possibility of implementing centralised analytics platforms capable of ingesting and correlating data from various sources.
- **Incorporate AI in Anomaly Detection:** As conventional rule-based approaches are losing their efficacy against encrypted and adaptive threats, organisations should invest in anomaly detection based on artificial intelligence. Machine learning-based models, particularly those exploring deep learning approaches such as CNN, RNN, or transformer, can discover sophisticated patterns in flow metadata that may indicate advanced persistent threats or stealthy attacks. These models are self-learning models that learn new traffic behaviours and adapt to changes in how protocols are used, as well as to new methods employed by attackers. The incorporation of AI is likely to not only improve the accuracy of detection but also eliminate the need to manually tune rules and hunt threats in contemporary encrypted networks, such as QUIC.

5. Conclusion and Future Work

The paper has discussed both sides of QUIC; it can improve web performance while increasing user privacy, but at the same time, it makes the task of network security appliances more difficult. However, unlike conventional protocols such as TCP+TLS, QUIC does not exclusively encrypt the payload; it also encrypts the transport headers, which is a significant departure from the standard approach concerning deep packet inspection (DPI) tools, intrusion detection systems (IDS), and firewalls. As our literature review revealed, even existing security technologies are finding it difficult to adjust to these constraints, especially because they rely on the content of plaintext packets or even header fields, which are now made hidden.

To overcome this, we installed and tested a metadata-driven monitoring system via machine learning techniques (e.g., Random Forests and Convolutional Neural Networks (CNNs)). Based on the findings, we can determine that encrypted traffic can still exhibit tendencies of anomalous or malicious behaviour when subjected to flow-level metadata (such as packet size and inter-arrival times) analysis. CNNs were specifically quite accurate and had a low false positive rating as well, which proves their relevance in this field. Packet-level firewalls, however, proved to be less effective, which further justifies the need to shift towards next-generation firewalls that combine endpoint telemetry and flow analytics. In conclusion, it should be noted that this research suggests that the future of QUIC-aware security will not involve decrypting traffic, but rather detecting its behaviour.

In the future, it may be possible to identify some major areas that should be explored further. To begin with, it is time to standardise the telemetry protocols of QUIC. Proto-cola-like qLog is a good beginning, but better adoption and improvement will be required to achieve consistent, secure, and privacy-aware sharing of session-level metadata across products and platforms. Second, industry stakeholders and researchers need to focus on developing collaborative security architectures that bridge endpoint-level visibility and monitoring with network-level security. To accomplish this, a hybrid solution may be employed, where trusted agents on the endpoints exchange encrypted session attributes in real-time with security platforms, thereby delivering better threat detection without jeopardising user data.

Lastly, an encouraging prospect would be to explore explainable AI (XAI) methods that will cater to the analysis of encrypted traffic. Deep learning models are highly accurate, but they tend to operate as black boxes, making it difficult for analysts to interpret the results; they may not know why a given flow was identified as malicious. Incorporating the notion of interpretability into the AI-based detection systems, the security teams would be able to increase the confidence in the automated alerts and improve their capabilities to react to the emerging threats in the encrypted space (such as QUIC).

References

1. Iyengar, J., & Thomson, M. (2021). QUIC: A UDP-based multiplexed and secure transport. In RFC 9000.
2. Langley, A., Riddoch, A., Wilk, A., Vicente, A., Krasic, C., Zhang, D., ... & Shi, Z. (2017, August). The Quick Transport Protocol: Design and Internet-Scale Deployment. In Proceedings of the conference of the ACM special interest group on data communication (pp. 183-196).
3. Böttger, T., Cuadrado, F., Antichi, G., Fernandes, E. L., Tyson, G., Castro, I., & Uhlig, S. (2019, October). An Empirical Study of the Cost of DNS-over-HTTPS. In Proceedings of the Internet Measurement Conference (pp. 15-21).
4. Lotfollahi, M., Jafari Siavoshani, M., Shirali Hossein Zade, R., & Saberian, M. (2020). Deep packet: A novel approach for encrypted traffic classification using deep learning. Soft Computing, 24(3), 1999-2012.
5. Alshammari, R., & Zincir-Heywood, A. N. (2011). Can encrypted traffic be identified without port numbers, IP addresses and payload inspection?. Computer networks, 55(6), 1326-1350.
6. Kakhki, A. M., Jero, S., Choffnes, D., Nita-Rotaru, C., & Mislove, A. (2017, November). Taking a long look at QUIC: an approach for rigorous evaluation of rapidly evolving transport protocols. In Proceedings of the 2017 Internet Measurement Conference (pp. 290-303).
7. Este, A., Gringoli, F., & Salgarelli, L. (2009). Support Vector Machines for TCP Traffic Classification. Computer Networks, 53(14), 2476-2490.
8. Bittau, A., Hamburg, M., Handley, M., Mazieres, D., & Boneh, D. (2010). The case for ubiquitous {Transport-Level} encryption. In the 19th USENIX Security Symposium (USENIX Security 10).
9. Holz, R., Amann, J., Mehani, O., Wachs, M., & Kaafar, M. A. (2015). TLS in the wild: An Internet-wide analysis of TLS-based protocols for electronic communication. arXiv preprint arXiv:1511.00341.
10. Dhanjani, N., & Clarke, J. (2005). Network Security Tools: Writing, Hacking, and Modifying Security Tools. " O'Reilly Media, Inc.".
11. Pavur, J., Strohmeier, M., Lenders, V., & Martinovic, I. (2020). QPEP: A QUIC-based approach to encrypted performance-enhancing proxies for high-latency satellite broadband. arXiv preprint arXiv:2002.05091.
12. Chatzoglou, E., Kouliaridis, V., Karopoulos, G., & Kambourakis, G. (2023). Revisiting QUIC attacks: A comprehensive review on QUIC security and a hands-on study. International Journal of Information Security, 22(2), 347-365.
13. Lychev, R., Jero, S., Boldyreva, A., & Nita-Rotaru, C. (2015, May). How secure and quick is QUIC? Provable security and performance analyses. In 2015 IEEE Symposium on Security and Privacy (pp. 214-231). IEEE.
14. Cui, Y., Li, T., Liu, C., Wang, X., & Kühlewind, M. (2017). Innovating transport with QUIC: Design approaches and research challenges. IEEE Internet Computing, 21(2), 72-76.
15. Kumar, P., & Dezfouli, B. (2019). Implementation and analysis of QUIC for MQTT. Computer Networks, 150, 28-45.
16. Soni, M., & Rajput, B. S. (2020). Security and Performance Evaluations of the QUIC Protocol. In Data Science and Intelligent Applications: Proceedings of ICDSIA 2020 (pp. 457-462). Singapore: Springer Singapore.
17. Rüth, J., Poese, I., Dietzel, C., & Hohlfeld, O. (2018, March). A First Look at QUIC in the Wild. In International Conference on Passive and Active Network Measurement (pp. 255-268). Cham: Springer International Publishing.
18. Joarder, Y. A., & Fung, C. (2022, October). A Survey on the Security Issues of QUIC. In 2022, 6th Cyber Security in Networking Conference (CSNet) (pp. 1-8). IEEE.

19. Al-Bakhat, L., & Almuhammadi, S. (2022, March). Intrusion detection on Quic Traffic: A machine learning approach. In 2022 7th International Conference on Data Science and Machine Learning Applications (CDMA) (pp. 194-199). IEEE.
20. Chakir, O., Sadqi, Y., & Maleh, Y. (2023). Evaluation of open-source web application firewalls for cyber threat intelligence. In Big Data Analytics and Intelligent Systems for Cyber Threat Intelligence (pp. 35-48). River Publishers.
21. Rusum, G. P., Pappula, K. K., & Anasuri, S. (2020). Constraint Solving at Scale: Optimizing Performance in Complex Parametric Assemblies. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(2), 47-55. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I2P106
22. Pappula, K. K. (2020). Browser-Based Parametric Modeling: Bridging Web Technologies with CAD Kernels. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(3), 56-67. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I3P107
23. Rahul, N. (2020). Optimizing Claims Reserves and Payments with AI: Predictive Models for Financial Accuracy. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(3), 46-55. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I3P106
24. Enjam, G. R., & Chandragowda, S. C. (2020). Role-Based Access and Encryption in Multi-Tenant Insurance Architectures. *International Journal of Emerging Trends in Computer Science and Information Technology*, *1*(4), 58-66. https://doi.org/10.63282/3050-9246.IJETCSIT-V1I4P107
25. Pappula, K. K. (2021). Modern CI/CD in Full-Stack Environments: Lessons from Source Control Migrations. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(4), 51-59. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I4P106
26. Pedda Muntala, P. S. R. (2021). Prescriptive AI in Procurement: Using Oracle AI to Recommend Optimal Supplier Decisions. *International Journal of AI, BigData, Computational and Management Studies*, *2*(1), 76-87. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V2I1P108
27. Rahul, N. (2021). Strengthening Fraud Prevention with AI in P&C Insurance: Enhancing Cyber Resilience. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(1), 43-53. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I1P106
28. Enjam, G. R., Chandragowda, S. C., & Tekale, K. M. (2021). Loss Ratio Optimization using Data-Driven Portfolio Segmentation. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(1), 54-62. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I1P107
29. Rusum, G. P., & Pappula, K. K. (2022). Federated Learning in Practice: Building Collaborative Models While Preserving Privacy. *International Journal of Emerging Research in Engineering and Technology*, *3*(2), 79-88. https://doi.org/10.63282/3050-922X.IJERET-V3I2P109
30. Pappula, K. K. (2022). Modular Monoliths in Practice: A Middle Ground for Growing Product Teams. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(4), 53-63. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P106
31. Anasuri, S. (2022). Next-Gen DNS and Security Challenges in IoT Ecosystems. *International Journal of Emerging Research in Engineering and Technology*, *3*(2), 89-98. https://doi.org/10.63282/3050-922X.IJERET-V3I2P110
32. Pedda Muntala, P. S. R. (2022). Detecting and Preventing Fraud in Oracle Cloud ERP Financials with Machine Learning. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(4), 57-67. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I4P107
33. Rahul, N. (2022). Enhancing Claims Processing with AI: Boosting Operational Efficiency in P&C Insurance. International Journal of Emerging Trends in Computer Science and Information Technology, 3(4), 77-86. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P108
34. Enjam, G. R., & Tekale, K. M. (2022). Predictive Analytics for Claims Lifecycle Optimization in Cloud-Native Platforms. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(1), 95-104. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P110
35. Rusum, G. P., & Pappula, K. K. (2023). Low-Code and No-Code Evolution: Empowering Domain Experts with Declarative AI Interfaces. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(2), 105-112. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I2P112
36. Pappula, K. K., & Rusum, G. P. (2023). Multi-Modal AI for Structured Data Extraction from Documents. *International Journal of Emerging Research in Engineering and Technology*, *4*(3), 75-86. https://doi.org/10.63282/3050-922X.IJERET-V4I3P109
37. Anasuri, S. (2023). Confidential Computing Using Trusted Execution Environments. *International Journal of AI, BigData, Computational and Management Studies*, *4*(2), 97-110. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I2P111
38. Pedda Muntala, P. S. R., & Jangam, S. K. (2023). Context-Aware AI Assistants in Oracle Fusion ERP for Real-Time Decision Support. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 75-84. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P109
39. Rahul, N. (2023). Transforming Underwriting with AI: Evolving Risk Assessment and Policy Pricing in P&C Insurance. *International Journal of AI, BigData, Computational and Management Studies*, *4*(3), 92-101. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P110

40. Enjam, G. R. (2023). AI Governance in Regulated Cloud-Native Insurance Platforms. *International Journal of AI, BigData, Computational and Management Studies*, *4*(3), 102-111. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P111
41. Pappula, K. K., & Rusum, G. P. (2020). Custom CAD Plugin Architecture for Enforcing Industry-Specific Design Standards. *International Journal of AI, BigData, Computational and Management Studies*, *1*(4), 19-28. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V1I4P103
42. Rahul, N. (2020). Vehicle and Property Loss Assessment with AI: Automating Damage Estimations in Claims. *International Journal of Emerging Research in Engineering and Technology*, *1*(4), 38-46. https://doi.org/10.63282/3050-922X.IJERET-V1I4P105
43. Enjam, G. R., & Tekale, K. M. (2020). Transitioning from Monolith to Microservices in Policy Administration. *International Journal of Emerging Research in Engineering and Technology*, *1*(3), 45-52. https://doi.org/10.63282/3050-922X.IJERETV1I3P106
44. Pappula, K. K., & Anasuri, S. (2021). API Composition at Scale: GraphQL Federation vs. REST Aggregation. *International Journal of Emerging Trends in Computer Science and Information Technology*, *2*(2), 54-64. https://doi.org/10.63282/3050-9246.IJETCSIT-V2I2P107
45. Pedda Muntala, P. S. R., & Jangam, S. K. (2021). Real-time Decision-Making in Fusion ERP Using Streaming Data and AI. *International Journal of Emerging Research in Engineering and Technology*, *2*(2), 55-63. https://doi.org/10.63282/3050-922X.IJERET-V2I2P108
46. Rahul, N. (2021). AI-Enhanced API Integrations: Advancing Guidewire Ecosystems with Real-Time Data. *International Journal of Emerging Research in Engineering and Technology*, *2*(1), 57-66. https://doi.org/10.63282/3050-922X.IJERET-V2I1P107
47. Enjam, G. R., & Chandragowda, S. C. (2021). RESTful API Design for Modular Insurance Platforms. *International Journal of Emerging Research in Engineering and Technology*, *2*(3), 71-78. https://doi.org/10.63282/3050-922X.IJERET-V2I3P108
48. Rusum, G. P. (2022). Security-as-Code: Embedding Policy-Driven Security in CI/CD Workflows. *International Journal of AI, BigData, Computational and Management Studies*, *3*(2), 81-88. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I2P108
49. Pappula, K. K. (2022). Containerized Zero-Downtime Deployments in Full-Stack Systems. International Journal of AI, BigData, Computational and Management Studies, 3(4), 60-69. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I4P107
50. Anasuri, S. (2022). Zero-Trust Architectures for Multi-Cloud Environments. International Journal of Emerging Trends in Computer Science and Information Technology, 3(4), 64-76. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I4P107
51. Pedda Muntala, P. S. R., & Karri, N. (2022). Using Oracle Fusion Analytics Warehouse (FAW) and ML to Improve KPI Visibility and Business Outcomes. International Journal of AI, BigData, Computational and Management Studies, 3(1), 79-88. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V3I1P109
52. Rahul, N. (2022). Optimizing Rating Engines through AI and Machine Learning: Revolutionizing Pricing Precision. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(3), 93-101. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I3P110
53. Enjam, G. R. (2022). Secure Data Masking Strategies for Cloud-Native Insurance Systems. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(2), 87-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I2P109
54. Rusum, G. P. (2023). Large Language Models in IDEs: Context-Aware Coding, Refactoring, and Documentation. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(2), 101-110. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I2P110
55. Pappula, K. K. (2023). Edge-Deployed Computer Vision for Real-Time Defect Detection. *International Journal of AI, BigData, Computational and Management Studies*, *4*(3), 72-81. https://doi.org/10.63282/3050-9416.IJAIBDCMS-V4I3P108
56. Anasuri, S., & Pappula, K. K. (2023). Green HPC: Carbon-Aware Scheduling in Cloud Data Centers. *International Journal of Emerging Research in Engineering and Technology*, *4*(2), 106-114. https://doi.org/10.63282/3050-922X.IJERET-V4I2P111
57. Reddy Pedda Muntala , P. S. (2023). Process Automation in Oracle Fusion Cloud Using AI Agents. *International Journal of Emerging Research in Engineering and Technology*, *4*(4), 112-119. https://doi.org/10.63282/3050-922X.IJERET-V4I4P111
58. Rahul, N. (2023). Personalizing Policies with AI: Improving Customer Experience and Risk Assessment. International Journal of Emerging Trends in Computer Science and Information Technology, 4(1), 85-94. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P110
59. Enjam, G. R. (2023). Optimizing PostgreSQL for High-Volume Insurance Transactions & Secure Backup and Restore Strategies for Databases. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 104-111. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P112

60. Pappula, K. K., Anasuri, S., & Rusum, G. P. (2021). Building Observability into Full-Stack Systems: Metrics That Matter. *International Journal of Emerging Research in Engineering and Technology*, *2*(4), 48-58. https://doi.org/10.63282/3050-922X.IJERET-V2I4P106
61. Pedda Muntala, P. S. R., & Karri, N. (2021). Leveraging Oracle Fusion ERP's Embedded AI for Predictive Financial Forecasting. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *2*(3), 74-82. https://doi.org/10.63282/3050-9262.IJAIDSML-V2I3P108
62. Rusum, G. P. (2022). WebAssembly across Platforms: Running Native Apps in the Browser, Cloud, and Edge. *International Journal of Emerging Trends in Computer Science and Information Technology*, *3*(1), 107-115. https://doi.org/10.63282/3050-9246.IJETCSIT-V3I1P112
63. Anasuri, S. (2022). Adversarial Attacks and Defenses in Deep Neural Networks. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(4), 77-85. https://doi.org/10.63282/xs971f03
64. Pedda Muntala, P. S. R. (2022). Anomaly Detection in Expense Management using Oracle AI Services. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *3*(1), 87-94. https://doi.org/10.63282/3050-9262.IJAIDSML-V3I1P109
65. Rusum, G. P., & Anasuri, S. (2023). Composable Enterprise Architecture: A New Paradigm for Modular Software Design. *International Journal of Emerging Research in Engineering and Technology*, *4*(1), 99-111. https://doi.org/10.63282/3050-922X.IJERET-V4I1P111
66. Anasuri, S. (2023). Secure Software Supply Chains in Open-Source Ecosystems. *International Journal of Emerging Trends in Computer Science and Information Technology*, *4*(1), 62-74. https://doi.org/10.63282/3050-9246.IJETCSIT-V4I1P108
67. Pedda Muntala, P. S. R., & Karri, N. (2023). Leveraging Oracle Digital Assistant (ODA) to Automate ERP Transactions and Improve User Productivity. *International Journal of Artificial Intelligence, Data Science, and Machine Learning*, *4*(4), 97-104. https://doi.org/10.63282/3050-9262.IJAIDSML-V4I4P111
68. Enjam, G. R. (2023). Modernizing Legacy Insurance Systems with Microservices on Guidewire Cloud Platform. *International Journal of Emerging Research in Engineering and Technology*, *4*(4), 90-100. https://doi.org/10.63282/3050-922X.IJERET-V4I4P109