Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

# Advanced Fraud Threats and Detection in Browser, Android, and iOS Environments

**Jeffrie Joshua Lazarus George**

Sardine.AI

**Abstract**

As digital technologies evolve, so do the methods employed by cybercriminals to exploit vulnerabilities within browser and mobile environments. This paper explores advanced fraud threats such as VPN detection evasion, browser spoofing, mobile device factory resets, jailbroken phones, user-initiated app cloning, and Android emulators and tampering. We discuss how these threats are misused and outline advanced detection techniques to mitigate risks. The paper emphasizes the importance of implementing robust security measures to protect against sophisticated fraud tactics in browser, Android, and iOS platforms.

## Introduction

In recent years, the digital landscape has undergone a profound transformation, marked by the massive proliferation of smartphones, tablets, and web-based applications. The convenience brought by these devices has fundamentally changed how we interact with the digital world, streamlining activities such as banking, shopping, and social networking. However, this unprecedented level of connectivity comes with risks. Cybercriminals have honed their skills, creating highly sophisticated fraud schemes that exploit weaknesses in browsers and mobile operating systems (OS).

For organizations, particularly those operating in sensitive sectors such as finance, healthcare, and e-commerce, understanding the scope of these threats is critical. Traditional fraud detection mechanisms, such as static rule-based systems, are no longer sufficient to combat modern threats, as attackers leverage dynamic tools like VPNs, which allow them to evade detection. Therefore, this paper seeks to uncover the most pressing fraud challenges in browser, Android, and iOS environments, with an emphasis on developing robust detection strategies tailored to each platform’s unique vulnerabilities.

## VPN Detection and Its Significance

### What is a VPN?

At its core, a VPN offers privacy and security by creating a secure tunnel between the user's device and the destination server. This tunnel encrypts all data traffic, ensuring that no third parties can intercept or monitor the activity. VPNs are especially useful for users accessing sensitive information, such as financial data or personal records, over unsecured networks like public Wi-Fi. Beyond individual users, corporations and governments rely heavily on VPNs to protect their internal networks from external threats. However, this powerful technology also opens doors for misuse, as it allows bad actors to conceal their identity and location.

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

VPNs enable cybercriminals to engage in illegal activities while maintaining anonymity, such as hacking, data theft, and money laundering. Additionally, VPNs have been known to facilitate click fraud schemes, where malicious actors use fake clicks to generate advertising revenue or inflate engagement metrics. This anonymity, while beneficial for legitimate privacy needs, poses a significant challenge for security teams attempting to track and identify fraudulent behavior.

**Misuse of VPNs**

While VPNs offer privacy benefits, they can also be exploited for malicious activities:

- **Identity Concealment**: Cybercriminals use VPNs to hide their true location and identity, making it difficult to trace fraudulent activities. This allows them to perform illegal actions like hacking, data breaches, and illicit financial transactions without being detected.
- **Bypassing Geo-Restrictions**: Fraudsters can exploit regional pricing differences or access services not available in their region, allowing them to manipulate markets, engage in arbitrage, or abuse online service terms. For instance, accessing lower-cost digital goods intended for other regions can significantly harm the business models of online services.
- **Click Fraud**: Using VPNs to artificially inflate clicks on online ads, leading to significant financial losses for advertisers. Ad networks, reliant on user engagement, can see reduced ROI for advertisers when fraudulent clicks drive up the cost of campaigns.

**VPN Detection Methods**

**Database Validation**

These databases rely on constantly updated records of VPN exit nodes and known IP ranges associated with VPN providers. Although they are a helpful first line of defense, private VPNs and rapidly rotating IP addresses challenge this method. Fraudsters often exploit lesser-known VPN services or set up self-hosted VPNs that evade detection by popular databases, making them harder to track. Additionally, some providers offer "stealth VPNs," which mask their traffic to look like regular encrypted internet traffic, further complicating detection efforts.

**Time Zone Mismatch**

Time zone mismatches are commonly flagged by fraud detection systems as potential indicators of VPN usage, especially when combined with other suspicious signals. However, savvy fraudsters are increasingly aware of this detection method and may spoof the time zone settings on their devices or use sophisticated VPN services that automatically synchronize time zones to match the IP address location. In response, security teams are developing more advanced techniques, such as combining time zone data with device geolocation, browser behavior patterns, and historical login records to spot anomalies more accurately.

**TCP/IP Fingerprinting**

TCP/IP fingerprinting involves analyzing how packets are structured and transmitted over the network. Each operating system and network stack handles traffic in a slightly different way, which can be detected by tools like p0f. VPNs can obscure this information by introducing layers of encryption, but anomalies in packet size, sequence numbers, and response times may still expose the use of a VPN. Sophisticated attackers might attempt to emulate normal traffic patterns to evade detection, making it essential for security teams to use tools capable of identifying even subtle inconsistencies.

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

**MTU Analysis**

MTU refers to the largest packet size that can be transmitted over a network connection. Different VPN protocols—such as OpenVPN, L2TP, or IPSec—alter the default MTU values due to the encryption overhead added by the VPN tunnel. By monitoring for atypical MTU values, network administrators can identify traffic passing through a VPN. This method is particularly effective when used in conjunction with packet inspection tools to reveal specific patterns associated with VPN protocols. However, attackers can manipulate MTU settings to match legitimate traffic, necessitating a multi-layered approach to detection.

**Importance of VPN Detection**

For companies that operate across multiple regions, accurate VPN detection can be essential for enforcing geo-restrictions, maintaining pricing integrity, and preventing fraudulent activities like click fraud and credential stuffing attacks. VPN detection helps ensure that only legitimate users are accessing services, enhancing trust between the organization and its customer base. Additionally, early detection of VPN misuse can prevent data breaches, financial losses, and reputational damage, making it a critical component of modern cybersecurity frameworks.

**Browser Spoofing and Tampering**

**Understanding Browser Spoofing**

Browser spoofing is a technique where a user alters their browser attributes, such as the user agent string, to disguise the browser's identity. While some users do this to enhance privacy or bypass restrictions, it can also be exploited for malicious purposes to evade detection. Spoofing allows cybercriminals to appear as though they are using a different browser, operating system, or device than they actually are. This can be leveraged to bypass security mechanisms or make it harder for website administrators to track malicious activities.

Spoofing is increasingly used in conjunction with other methods such as VPNs or proxies, making it difficult for security systems to identify suspicious behavior. As a result, cybercriminals can conduct large-scale fraud by imitating different devices and browsers without being detected. This form of deception is particularly harmful in sectors like e-commerce and online banking, where identity verification is critical.

**Common Tampered Attributes**

- **User Agent String**: The user agent string provides information about the browser, device, and operating system. Tampering with it can mislead websites into thinking the user is on a different device or browser. Fraudsters can use this tactic to bypass security features that block specific user agents.
- **JavaScript Capabilities**: JavaScript is used to check browser functionality and support. Spoofing these capabilities allows users to hide or fake their browser's capabilities, making it difficult for websites to detect anomalies.
- **Screen Resolution and Time Zone**: Changing the screen resolution or time zone can help fraudsters mislead websites about the geographical location of the user, making it appear as though they are accessing the site from a different device or region.
- **Installed Fonts and Canvas API**: Websites often rely on browser fingerprinting to track users,

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

which includes analyzing installed fonts and canvas rendering. By altering these attributes, fraudsters can disrupt fingerprinting efforts, making it difficult to uniquely identify a device.

**Detection Techniques**

- **Inconsistency Analysis**: One of the most effective ways to detect browser spoofing is through inconsistency analysis. This involves comparing reported browser data with actual behavior to identify anomalies. For example, if a user claims to be using a mobile browser but exhibits behavior consistent with desktop usage, it can raise red flags.
- **Statistical Modeling**: By collecting historical data from users, security systems can develop models of typical browser behavior. Statistical modeling helps identify deviations from normal behavior that may indicate spoofing.
- **Machine Learning Algorithms**: Modern detection systems employ machine learning algorithms to predict and identify browser tampering based on learned patterns. These algorithms can detect subtle changes in behavior or inconsistencies in browser data that would be difficult for manual systems to catch.

**Implications**

Browser tampering creates significant challenges for security teams by complicating user identification and risk assessment. Fraudsters can exploit this technique to evade detection, causing significant financial and reputational damage to businesses. For example, an attacker might manipulate a browser to bypass security protocols, commit click fraud, or evade region-based restrictions. Detecting and mitigating browser spoofing is critical for maintaining the integrity of online systems, especially in industries that rely on accurate user identification, such as finance and healthcare.

**Mobile Device Factory Resets as Fraud Indicators**

**What is a Factory Reset?**

A factory reset restores a mobile device to its original settings, erasing all user data, installed applications, and system configurations. While this is commonly used for troubleshooting or preparing a device for resale, frequent factory resets can be a red flag for fraudulent behavior. Cybercriminals often use factory resets to wipe away traces of their malicious activities and start anew, making it harder for security systems to track or link their actions to previous behavior.

**Fraudulent Use Cases**

- **Account Creation Abuse**: Fraudsters reset devices to create multiple accounts in quick succession, exploiting signup bonuses, referral programs, or free trials. Each reset wipes the slate clean, allowing them to bypass account creation restrictions that prevent multiple signups from a single device.
- **Anonymity for Malicious Activities**: Repeated factory resets can help fraudsters remain anonymous by concealing previous activities. This makes it harder for security teams to link suspicious behavior to the same user or device, allowing them to fly under the radar.

**Detection Strategies**

- **Monitoring Reset Frequency**: One of the most effective ways to detect fraudulent behavior associated with factory resets is by tracking how often a device is reset. Unusually frequent resets, especially when coupled with suspicious activity, can indicate fraud.
- **Analyzing Usage Patterns**: Analyzing patterns before and after factory resets can reveal suspicious

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

behaviors, such as the creation of new accounts immediately after a reset or unusual transaction activity. By correlating these patterns, security teams can flag devices that are likely being used for fraud.

## Risks Associated with Jailbroken Devices

### Understanding Jailbreaking

Jailbreaking involves removing the software restrictions placed on a device by the operating system, typically to gain root access. This allows users to install unauthorized apps, customize the device, and alter its core functionality. However, jailbreaking also exposes the device to significant security risks, as it bypasses many of the built-in safeguards designed to protect users from malicious software and attacks. For fraudsters, jailbroken devices are a goldmine, offering enhanced privileges that can be exploited to bypass security measures and commit fraud.

### Fraudulent Exploitation

- **Malware Installation**: Jailbroken devices are more vulnerable to malware, as they are no longer protected by the operating system's built-in security protocols. Fraudsters can install malicious software that steals sensitive information or hijacks the device for further fraudulent activities.
- **Security Bypass**: With root access, fraudsters can bypass security measures such as app sandboxing, encryption, and permission checks. This enables them to tamper with apps, extract sensitive data, or modify system files.
- **Account Takeovers**: The enhanced privileges provided by a jailbroken device make it easier for fraudsters to take over user accounts. For example, they can use root access to intercept communications or manipulate authentication mechanisms.

### Detection Methods

- **File System Checks**: Scanning the device's file system for changes indicative of jailbreaking, such as the presence of unauthorized apps or modifications to system files, can help detect jailbroken devices.
- **API Usage Monitoring**: Monitoring the use of APIs that are commonly accessed on jailbroken devices can also signal tampering. Unauthorized API calls or modifications can alert security systems to potential fraud.
- **Integrity Verification**: Verifying the integrity of the operating system and applications by checking for modifications or alterations to core components can help ensure that the device has not been tampered with.

## User-Initiated App Cloning and Its Threats

### What is App Cloning?

App cloning refers to the process of duplicating an application on a single device, allowing users to run multiple instances of the same app. While app cloning can be used for legitimate purposes—such as maintaining separate work and personal accounts—it is often exploited for fraudulent activities. Cybercriminals use cloned apps to circumvent app limitations, create multiple accounts, or engage in promotion abuse. Cloning tools make it easy for users to run multiple copies of an app without the need for additional devices, which opens the door to various forms of misuse.

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

Fraudsters can manipulate app cloning to scale their attacks by generating multiple accounts for malicious purposes. In many cases, they use cloned apps to evade restrictions imposed on single accounts, such as voting limits, referral bonuses, or time-sensitive promotions. By distributing these cloned apps across several devices or running multiple instances on a single device, they can generate significant financial or reputational damage for companies.

**Fraudulent Activities Enabled by App Cloning**

- **Promotion Abuse**: One of the most common uses of app cloning is to exploit signup bonuses or referral programs by creating multiple accounts. Fraudsters clone apps to register numerous fake users and claim rewards that were intended for new, genuine customers. This leads to inflated costs for businesses and can distort user growth metrics.
- **Bypassing Usage Limits**: Apps often have built-in restrictions, such as daily usage limits, to prevent abuse. Fraudsters can use cloned apps to bypass these limits and gain unfair advantages in gaming apps, stock trading, or cryptocurrency platforms, where access to higher usage levels offers greater financial benefits.
- **Ban Evasion**: If a user is banned from an app for violating terms of service, they may use a cloned version of the app to bypass the ban and continue accessing the platform under a new account. This is particularly common in social media apps or online marketplaces, where user behavior is heavily monitored.
- **Vote Manipulation**: Fraudsters can use app cloning to create multiple accounts and manipulate polls, reviews, or ratings by casting multiple votes or submitting numerous reviews. This can lead to skewed results and false feedback, undermining the credibility of platforms reliant on user input.

**Detection Techniques**

- **Device Signal Correlation**: One of the key ways to detect app cloning is through composite identifiers that track multiple device signals. By analyzing unique device fingerprints, such as MAC addresses, IMEI numbers, and network configurations, systems can detect when multiple app instances are being run on the same device, raising red flags for potential fraud.
- **Behavioral Analysis**: Another effective method for detecting app cloning is through behavioral analysis. By monitoring user behavior patterns—such as login times, usage frequency, and transaction volumes—security teams can identify anomalous activity indicative of cloned apps. For example, if multiple accounts exhibit similar interaction patterns within a short time frame, this could indicate cloned app activity.
- **Consistency Checks**: Systems can also identify discrepancies between device data and app data. For instance, apps running on cloned instances may not report the same device information as the original app. Detecting these inconsistencies, such as different geolocations for the same device or mismatched app versions, can help uncover cloned app fraud.

**Android Emulators and Device Tampering**

**Understanding Android Emulators**

Android emulators are software applications that replicate the functionality of Android devices on computers. Originally designed for app developers to test applications across various device configurations, emulators have also become popular tools for fraudsters. By simulating multiple Android

Advanced International Journal of Multidisciplinary Research
E-ISSN: 2584-0487 editor@aijmr.com
Volume 2, Issue 5, September-October 2024
CrossRef DOI: 10.62127/aijmr.2024.v02i05.1101

devices on a single machine, fraudsters can engage in large-scale fraudulent activities without the need to purchase physical devices. Emulators allow cybercriminals to manipulate app behavior, automate actions, and scale their fraudulent efforts with minimal cost.

Emulators enable fraudsters to bypass app restrictions, create fake accounts en masse, and commit fraudulent activities at a much higher scale than would be possible with a single mobile device. For instance, they can be used to automate actions such as clicking ads, filling out forms, or interacting with apps in ways that generate revenue or manipulate metrics.

**Fraudulent Use of Emulators**

- **Click Fraud**: One of the most prevalent forms of fraud associated with emulators is click fraud, where fraudsters automate the clicking of online ads to generate revenue from pay-per-click advertising schemes. By running multiple emulators simultaneously, they can inflate click counts and siphon advertising funds from legitimate businesses.
- **Mass Account Creation**: Emulators allow cybercriminals to create and manage thousands of fake accounts across different platforms, such as social media, gaming, or financial apps. These accounts can then be used for spam campaigns, boosting follower counts, or engaging in coordinated fraudulent activities like review manipulation or fake feedback.
- **Bypassing Security Measures**: Emulators provide an easy way for fraudsters to mimic real devices while avoiding security measures. Fraud detection systems may struggle to differentiate between emulated and real devices, allowing fraudsters to bypass verification processes and continue operating undetected.

**Detection Strategies**

- **Property Anomalies**: One way to detect the use of emulators is by identifying anomalies in device properties. Emulators often present generic device information that does not correspond to specific physical devices. By cross-referencing device model numbers, operating system versions, and hardware characteristics, security systems can detect when an emulator is being used to simulate device behavior.
- **Behavioral Monitoring**: Emulated devices often exhibit unnatural usage patterns, such as rapid-fire clicks or actions performed with machine-like precision. Behavioral monitoring tools can flag these patterns as suspicious, particularly when actions occur in quick succession or from multiple instances at the same time.
- **Network Analysis**: Another effective method for detecting emulator-based fraud is through network traffic analysis. Emulators running on the same machine often share a common IP address or network signature, making it possible to detect when large numbers of requests are originating from a single source. This is especially useful for identifying click fraud or mass account creation schemes.

**Android Tampering**

**What is Android Tampering?**

Android tampering refers to the unauthorized modification of the Android operating system or applications in order to exploit vulnerabilities or alter functionalities. Tampering can occur through a variety of methods, including rooting the device, reverse-engineering apps, and injecting malicious code

into otherwise legitimate apps. Tampered devices are a significant threat to mobile app security, as they allow fraudsters to bypass security features and gain access to sensitive data or functionality.

**Common Tampering Methods**

- **App Cloning and Repackaging**: Fraudsters clone legitimate apps and repackage them with malicious code. These tampered apps can then be distributed to unsuspecting users via unofficial app stores or phishing campaigns, allowing attackers to steal user credentials or compromise the device.
- **Code Injection and Reverse Engineering**: By reverse-engineering an app, attackers can modify its source code to change its behavior or extract sensitive information. Code injection allows them to introduce malicious features, such as keyloggers or spyware, into legitimate apps, compromising the security of the app and its users.

**Detection and Prevention**

- **Security Audits**: Regularly auditing Android applications for vulnerabilities is a key step in detecting tampering. Security teams can use tools to analyze the app's code, monitor for unauthorized changes, and ensure that the app's integrity remains intact.
- **Integrity Checks**: Verifying the integrity of the app by checking its digital signature and comparing it to the original version can help identify tampered apps. Tampered apps often have altered code that no longer matches the original signature, making it easier to flag them as malicious.
- **User Education**: Educating users on the dangers of downloading apps from unofficial sources and encouraging them to only install apps from trusted app stores can reduce the risk of tampering. Additionally, informing users about the signs of tampered apps—such as excessive permission requests or abnormal app behavior—can help them avoid falling victim to malicious software.

## Conclusion

Advanced fraud threats leveraging VPNs, browser spoofing, mobile device resets, jailbroken phones, app cloning, and Android emulators present significant challenges to cybersecurity. The continually evolving nature of these fraud tactics requires organizations to adopt more sophisticated detection methods, incorporating machine learning, behavioral analysis, and network monitoring to stay ahead of cybercriminals. Furthermore, businesses must continuously update their security practices, conducting regular audits and integrating real-time monitoring to detect fraud as it happens. As fraudsters refine their methods, it is imperative that organizations maintain a proactive approach to fraud prevention, ensuring that their systems can adapt to the ever-changing threat landscape in browser and mobile environments.

## References

1. Alshboul, A., & Omar, K. (2020). VPN Traffic Detection Techniques: A Survey. *IEEE Access*, 8, 136879-136892.
2. MaxMind. (2023). *GeoIP2 Databases*. Retrieved from https://www.maxmind.com
3. Nikiforakis, N., et al. (2013). Cookieless Monster: Exploring the Ecosystem of Web-based Device Fingerprinting. *IEEE Symposium on Security and Privacy*, 541-555.
4. Owens, R., & Matthews, P. (2019). An Analysis of Android Device Security: Android vs. iOS. *Journal of Information Security*, 10(2), 123-133.

5. Ren, J., et al. (2016). ReCon: Revealing and Controlling PII Leaks in Mobile Network Traffic. *ACM Symposium on Systems and Information Security*, 361-372.
6. Starov, O., & Nikiforakis, N. (2017). XHOUND: Quantifying the Fingerprintability of Browser Extensions. *IEEE Symposium on Security and Privacy*, 941-956.
7. Wang, Z., & Stavrou, A. (2010). Exploiting Smart-phone USB Connectivity for Fun and Profit. *IEEE Symposium on Reliable Distributed Systems*, 4-13.
8. Zhauniarovich, Y., & Gadyatskaya, O. (2016). Small Changes, Big Changes: An Updated View on the Android Permission System. *International Conference on Risks and Security of Internet and Systems*, 81-98.
9. Zheng, M., & Sun, J. (2015). DroidTrace: A Ptrace Based Android Dynamic Analysis System with Forward Execution Capability. *IEEE Symposium on Security and Privacy Workshops*, 121-126.
10. Zimmeck, S., et al. (2017). Automated Analysis of Privacy Requirements for Mobile Apps. *NDSS Symposium*, 1-15.
11. Preyaa Atri, "Design and Implementation of High-Throughput Data Streams using Apache Kafka for RealTime Data Pipelines", International Journal of Science and Research (IJSR), Volume 7 Issue 11, November 2018, pp. 1988-1991, https://www.ijsr.net/getabstract.php?paperid=SR24422184316
12. Pei, Y., Liu, Y., Ling, N., Ren, Y., & Liu, L. (2023, May). An end-to-end deep generative network for low bitrate image coding. In 2023 IEEE International Symposium on Circuits and Systems (ISCAS) (pp. 1-5). IRRELEVANT.
13. Leng, Q., & Peng, L. Medical Image Intelligent Diagnosis System Based on Facial Emotion Recognition and Convolutional Neural Network.
14. Priya, M. M., Makutam, V., Javid, S. M. A. M., & Safwan, M. AN OVERVIEW ON CLINICAL DATA MANAGEMENT AND ROLE OF PHARM. D IN CLINICAL DATA MANAGEMENT.
15. Zhizhong Wu, Xueshe Wang, Shuaishuai Huang, Haowei Yang, Danqing Ma, Research on Prediction
16. Recommendation System Based on Improved Markov Model. Advances in Computer, Signals and Systems (2024) Vol. 8: 87-97. DOI: http://dx.doi.org/10.23977/acss.2024.080510.
17. Preyaa Atri, "Optimizing Financial Services Through Advanced Data Engineering: A Framework for Enhanced Efficiency and Customer Satisfaction", International Journal of Science and Research (IJSR), Volume 7 Issue 12, December 2018, pp. 1593-1596, https://www.ijsr.net/getabstract.php?paperid=SR24422184930
18. Ma, D., Wang, M., Xiang, A., Qi, Z., & Yang, Q. (2024). Transformer-Based Classification Outcome Prediction for Multimodal Stroke Treatment. arXiv preprint arXiv:2404.12634.
19. Preyaa Atri, "Enhancing Big Data Interoperability: Automating Schema Expansion from Parquet to BigQuery", International Journal of Science and Research (IJSR), Volume 8 Issue 4, April 2019, pp. 2000-2002, https://www.ijsr.net/getabstract.php?paperid=SR24522144712
20. Preyaa Atri, "Unlocking Data Potential: The GCS XML CSV Transformer for Enhanced Accessibility in Google Cloud", International Journal of Science and Research (IJSR), Volume 8 Issue 10, October 2019, pp. 1870-1871, https://www.ijsr.net/getabstract.php?paperid=SR24608145221
21. ang, H., Wang, L., Zhang, J., Cheng, Y., & Xiang, A. (2024). Research on Edge Detection of LiDAR Images Based on Artificial Intelligence Technology. arXiv preprint arXiv:2406.09773.

22. Wang, L., Cheng, Y., Xiang, A., Zhang, J., & Yang, H. (2024). Application of Natural Language Processing in Financial Risk Detection. arXiv preprint arXiv:2406.09765.
23. Atri, P. (2024). Enhancing Big Data Security through Comprehensive Data Protection Measures: A Focus on Securing Data at Rest and In-Transit. International Journal of Computing and Engineering, 5(4), 44–55. https://doi.org/10.47941/ijce.1920