Real-Time Parental Control and Website Blocking System

L. P. Yerreddy[1], T. P. Fule[2], Dr. L. V. Patil[3]
[1,2]*Students, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune India*
[3]*Professor, Department of Computer Engineering, Smt. Kashibai Navale College of Engineering, Pune India*

***Abstract*—The increasing exposure of children to inappropriate online content, excessive screen time, and cyber threats has created a need for intelligent parental control solutions. This paper presents the design, implementation, and evaluation of a Real-Time Parental Control and Website Blocking System. The proposed system enables parents to monitor children's online activities, block harmful websites, enforce screen-time restrictions, and generate activity reports. The system consists of a browser monitoring module, website filtering engine, administrative dashboard, and centralized database. Real-time URL inspection, blacklist/whitelist management, keyword filtering, and usage analytics are integrated to provide comprehensive digital safety. The architecture uses a React.js frontend, Node.js backend, browser extension/service agent, and MySQL database. The implemented system demonstrates effective content filtering, low response latency, and improved parental supervision capabilities.**

***Index Terms*—Parental Control, Website Blocking, URL Filtering, Child Safety, Real-Time Monitoring, Web Security, Browser Extension.**

I. INTRODUCTION

Problem Statement: Children spend a significant amount of time on the internet for education, entertainment, and communication. However, unrestricted internet access may expose them to inappropriate content, cyberbullying, online scams, gaming addiction, and social media misuse [9]. Existing parental control solutions are often expensive, platform-dependent, or difficult to configure [8]. Therefore, there is a need for a cost-effective and real-time parental control system capable of monitoring browsing activities and blocking harmful websites automatically.

Motivation: The motivation behind this project is to create a secure digital environment for children by empowering parents with tools to monitor and control online activities. The increasing availability of harmful content and the growing dependency on digital devices necessitate intelligent solutions capable of enforcing internet usage policies automatically [9]. By integrating website filtering, screen-time management, and activity monitoring into a single platform, the system aims to promote safe and productive internet usage.

Objectives of Implementation:

1. To develop a real-time website monitoring and blocking system capable of detecting restricted websites instantly [1].
2. To implement blacklist and whitelist mechanisms for controlling website accessibility.
3. To provide parents with a centralized dashboard for monitoring browsing activities.
4. To implement screen-time restrictions and internet usage scheduling.
5. To generate alerts and reports regarding children's online behaviour [2].
6. To create a scalable architecture capable of supporting future AI-based content filtering techniques.

This system contributes a complete end-to-end solution that combines browser-level monitoring, backend processing, and centralized data management [3].

Unlike conventional website blockers, the proposed system provides real-time monitoring and reporting capabilities suitable for educational institutions and home environments.

II. BACKGROUND

Parental control systems are designed to restrict and monitor internet access for children. Traditional parental supervision methods rely heavily on manual monitoring, which becomes ineffective as children gain access to multiple devices and online platforms [1][3]. Modern parental control applications employ website filtering, activity monitoring, and time management techniques to regulate internet usage.

Website filtering can be achieved using several approaches, including URL-based filtering, keyword matching, DNS filtering, and content categorization. URL filtering remains one of the most effective methods because it enables immediate website access control based on predefined rules [9][2]. Real-time monitoring systems further enhance protection by continuously tracking user activities and generating alerts when policy violations occur. Recent advancements in web technologies and cybersecurity frameworks have enabled the development of lightweight yet efficient parental control solutions capable of operating in real-time environments.

III. SYSTEM OVERVIEW

the Real-Time Parental Control and Website Blocking System is a web-based monitoring platform that enables parents to supervise internet activities and enforce browsing restrictions [2]. The system continuously monitors website access requests and evaluates them against predefined security policies before allowing or blocking access.

Workflow Overview:

1. Monitoring: The browser extension continuously observes user browsing activities and captures website URLs [1].
2. URL Transmission: Captured URLs are transmitted securely to the backend server for evaluation.
3. Website Evaluation: The filtering engine compares URLs against blacklist, whitelist, and keyword filtering databases [3].
4. Decision Processing: The system determines whether access should be granted or denied.
5. Website Blocking: Restricted websites are blocked instantly, and warning notifications are displayed.
6. Activity Logging: All browsing activities and blocking events are recorded in the database [7].
7. Reporting: Parents can view reports and activity summaries through the monitoring dashboard.

IV. SYSTEM ARCHITECTURE

The proposed system follows a multi-layered distributed architecture designed to provide real-time website filtering, user activity monitoring, policy enforcement, and centralized parental supervision. The architecture is divided into four major layers: the Presentation Layer, Monitoring Layer, Application Layer, and Data Layer. The modular architecture ensures scalability, maintainability, security, and high performance while supporting multiple client devices simultaneously [2][4]. The system continuously monitors user browsing activities through a browser extension installed on the child's device. Whenever a website access request is initiated, the extension captures the URL and forwards it to the backend filtering engine. The backend validates the request against predefined parental policies, blacklist databases, whitelist entries, and keyword-based filtering rules. Based on the evaluation result, the system either permits access or immediately blocks the website. Simultaneously, activity logs are generated and stored in the database for future reporting and analytics.

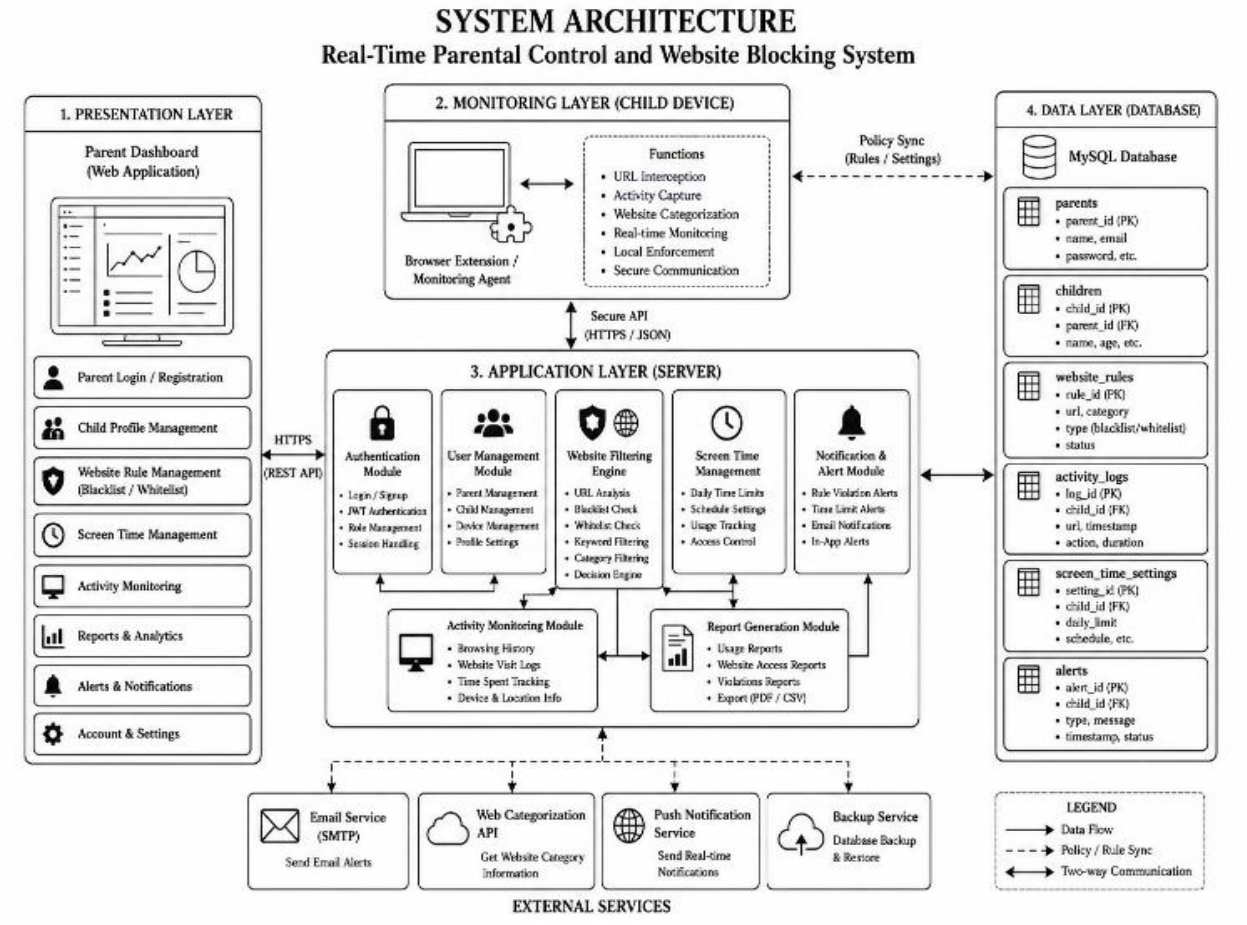

Fig . System Architecture

A .Presentation Layer

The Presentation Layer serves as the user interaction interface and is developed using React.js, HTML5, CSS3, and Tailwind CSS. This layer provides a responsive and intuitive dashboard through which parents can manage website restrictions, monitor online activities, configure screen-time policies, and generate reports [5][7].

The frontend communicates with the backend using RESTful APIs and displays real-time monitoring information retrieved from the server.

Key Responsibilities

- Parent Registration and Login
- Child Profile Management
- Website Blacklist Management
- Website Whitelist Management
- Activity Report Visualization
- Screen Time Configuration
- Notification Management

B. Browser Monitoring Layer

The Browser Monitoring Layer consists of a browser extension developed using JavaScript and browser extension Apis. This layer acts as the first point of interception for all browsing activities [8]. Whenever the user attempts to visit a website, the extension captures the URL before the page is fully loaded [9].

- URL Monitoring

Captures every website request generated by the browser.

- Website Categorization

Extracts domain names and identifies website categories [5].

- Real-Time Communication

Sends captured URLs to the backend filtering engine through secure API requests.

- Blocking Enforcement

Immediately redirects users to a warning page if access is denied. The browser extension operates

continuously in the background and requires minimal system resources [4].

C. Application Layer

The Application Layer acts as the central processing unit of the entire system. It is implemented using Node.js and Express.js [7][2]. This layer receives requests from the browser extension and dashboard, processes filtering policies, performs authentication, manages activity logs, and communicates with the database.

C.1 Authentication Module

The Authentication Module manages secure access to the system.

- Parent Registration
- Secure Login
- Session Management
- Password Encryption using BCrypt
- JWT Token-Based Authentication

This module prevents unauthorized access to parental controls and monitoring reports.

C.2 Website Filtering Engine

The Website Filtering Engine is the core component of the system. It evaluates every URL against multiple filtering criteria [3].

- Filtering Mechanisms
- Blacklist Filtering

Blocks websites explicitly listed by parents.

- Whitelist Filtering

Allows websites regardless of category restrictions.

- Keyword Filtering

Searches for restricted keywords within URLs.

- Category-Based Filtering

Blocks predefined categories such as:

Adult Content, Gambling, Social Media, Gaming, Streaming Platforms

The filtering engine uses optimized search algorithms to ensure fast decision-making.

C.3 Activity Monitoring Module

This module records detailed browsing information for analysis and reporting [6].

Captured Information

- Website URL
- Access Timestamp
- Access Status
- Session Duration
- Device Information
- User Identity

C.4 Notification and Alert Module

This module generates notifications whenever security policy is violated. Attempts to Access Restricted Website would lead to repeated access violations and send notifications to the parent [8][9]. Notifications can be displayed on the dashboard and sent through email services.

D. Data Layer

The Data Layer is implemented using AWS Database Management System. The database acts as a centralized repository that stores information generated by various system components [1][3]. Whenever a user accesses a website, the browser monitoring module forwards the activity information to the backend server, which subsequently stores the corresponding records in the database. Similarly, parental control policies such as blacklist entries, whitelist records, category restrictions, and screen-time schedules are maintained within the database and retrieved whenever filtering decisions need to be performed [6]. This centralized storage architecture enables consistent policy enforcement across all monitored devices.

To maintain data consistency and integrity, the database design follows normalization principles and employs primary keys, foreign keys, and relational constraints [1][7]. This minimizes data redundancy and ensures accurate relationships between different entities within the system. Furthermore, security mechanisms such as password hashing, role-based access control, and secure database connections are implemented to protect sensitive user information and browsing records from unauthorized access.

V. SYSTEM IMPLEMENTATION

The implementation involved the integration of multiple technologies, including React.js, Node.js, Express.js, MySQL, Browser Extension APIs, and RESTful web services. The system was developed using a modular architecture to ensure scalability, maintainability, and efficient real-time monitoring [1][5]. The implementation process consisted of website data collection, URL preprocessing, filtering

engine development, backend integration, browser extension development, dashboard implementation, and database design.

A. Website Data Collection and Rule Generation

The effectiveness of the proposed system depends significantly on the quality and accuracy of website filtering rules. To establish a robust filtering mechanism, a collection of website URLs was gathered from publicly available blacklist repositories, cybersecurity databases, parental control resources, and manually curated website lists [1].

The collected websites were categorized into multiple classes, including:

- Adult Content Websites
- Gambling Platforms
- Gaming Websites
- Social Media Platforms
- Streaming and Entertainment Sites
- Educational Websites
- News and Information Portals

The categorized website database serves as the foundation for implementing blacklist and whitelist filtering mechanisms. Parents are also provided with the capability to dynamically add or remove websites from the filtering database through the dashboard interface [2][5].

B. URL Preprocessing and Categorization

Before applying filtering policies, every URL undergoes a preprocessing phase to standardize website addresses and improve matching accuracy [4]. URL normalization removes unnecessary components like:

- HTTP/HTTPS variations
- Query Parameters
- URL Fragments
- Trailing Slashes

The system extracts the root domain name to facilitate efficient rule matching. Keywords associated with restricted categories are extracted and compared against predefined filtering dictionaries. Each website is mapped to one of the predefined categories stored within the database. These preprocessing operations significantly reduce false positives and improve overall filtering performance [1][5].

C. Browser Monitoring Agent Development

A browser extension was developed using JavaScript and Browser Extension APIs to monitor user browsing activities in real time. The extension operates continuously in the background and performs the following tasks: Captures every website request generated by the browser [8][9]. Monitor active browser tabs and records website usage duration. Tracks website visits and browsing sessions. Sends website information to the server through secure the browser extension acts as the first layer of defence by intercepting URLs before webpage loading is completed.

D. Website Filtering Engine Development

The Website Filtering Engine serves as the core decision-making component of the system. The engine evaluates each URL against multiple security policies before granting website access [1][5][10]. The following sequence is executed whenever a website request is detected:

1. Capture URL.
2. Normalize URL.
3. Compare with blacklist database.
4. Compare with whitelist database.
5. Perform keyword analysis.
6. Evaluate category restrictions.
7. Generate access decision.

E. The engine generates one of the following outcomes:

- Allow: The website satisfies all parental policies.
- Block: The website violates one or more filtering rules.
- Alert: The website is suspicious and requires parental review.

The filtering engine utilizes optimized lookup structures to ensure minimal response latency during real-time operation.

F. Backend Implementation

The backend of the Real-Time Parental Control and Website Blocking System was developed using Node.js and Express.js, which provide a scalable and efficient environment for handling real-time requests and managing system operations. The backend acts as the central processing layer of the architecture and serves as the communication bridge between the

browser monitoring agent, parental dashboard, and database [9]. It is responsible for executing business logic, enforcing parental control policies, processing website filtering rules, and maintaining activity records. By utilizing a RESTful architecture, the backend ensures reliable and secure communication between all system components while maintaining low response latency.

The backend also incorporates a comprehensive rule-processing module that manages website filtering and policy enforcement. Whenever a URL is captured by the browser extension, it is transmitted to the server for evaluation. The filtering engine compares the incoming URL against blacklist entries, whitelist records, category restrictions, and predefined parental policies stored in the database [10]. Based on the evaluation results, the backend generates an access decision that determines whether the website should be allowed, blocked, or flagged for parental review. This centralized rule-processing mechanism ensures consistent enforcement of internet usage policies across all monitored devices [8]. Activity logging represents another critical function of the backend infrastructure. The server continuously records browsing events, blocked website attempts, login activities, and screen-time usage statistics.

G. API Development

Multiple APIs were implemented for system functionality.

Authentication APIs

POST /api/register

POST /api/login

Data Preprocessing

Data preprocessing is a crucial stage in the implementations of system, website URLs and browsing information are standardized before being evaluated by the filtering engine. Raw URLs obtained from browser requests often contain unnecessary parameters, subdomains, special characters, and tracking information that may affect the accuracy of website matching. Therefore, preprocessing techniques are applied to transform the captured data into a structured and consistent format suitable for efficient filtering and analysis [1][4].

The first preprocessing step involves URL normalization. During this process, different representations of the same website are converted into a standard format. Protocol identifiers such as HTTP and HTTPS, trailing slashes, query parameters, and tracking tokens are removed to ensure consistent comparison. For example, are normalized to the root domain "youtube.com". This reduces redundancy and improves matching efficiency. Following normalization, domain extraction is performed to isolate the primary domain name from the complete URL structure. Since filtering decisions are generally based on domains rather than individual web pages, extracting the root domain enables faster blacklist and whitelist verification. This process also eliminates unnecessary URL components that do not contribute to access control decisions [4].

Keyword preprocessing is then applied to identify potentially harmful or restricted content categories. The system maintains a database of sensitive keywords related to gambling, adult content, violence, social media, gaming, and other restricted categories. Captured URLs are analysed for the presence of these keywords, allowing the system to detect websites that may not already exist within the blacklist database [2][5]. This approach enhances the system's capability to identify newly emerging or uncategorized websites.

Website categorization is another important preprocessing activity. After extracting domain information, websites are mapped to predefined categories such as Educational, social media, Entertainment, Gaming, News, Adult Content, and E-Commerce. Category information is either retrieved from stored databases or determined through keyword-based classification. This classification allows parents to block entire categories of websites rather than managing individual URLs manually.

Duplicate removal is also performed to optimize storage and processing efficiency. Browsing logs often contain repeated visits to the same website within short time intervals. The preprocessing module identifies duplicate records and consolidates them when generating activity reports [5]. This reduces database storage requirements and improves report readability. After completion of preprocessing operations, the cleaned and structured URL data is forwarded to the Website Filtering Engine for blacklist verification, whitelist validation, category analysis, and policy enforcement. These preprocessing techniques significantly improve filtering accuracy, reduce false positives, minimize processing overhead,

and enhance the overall performance of the parental control system.

VI. SYSTEM EVALUATION

A. Evaluation Method

To measure system performance, several evaluation metrics were considered, including Website Blocking Accuracy, Precision, Recall, Response Time, Monitoring Accuracy, and System Reliability. These metrics provide quantitative insights into the effectiveness of the filtering mechanism and the real-time monitoring capabilities of the system [1][5].

- Website Blocking Accuracy measures the overall correctness of the filtering engine in classifying websites as allowed or blocked.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- Precision: Precision measures the proportion of correctly blocked websites among all websites identified as restricted.

$$Precision = \frac{TP}{TP + FP}$$

A high recall value indicates that very few harmful websites bypass the filtering mechanism.

- Recall (Sensitivity): Recall evaluates the system's ability to identify and block all restricted websites.

$$Recall = \frac{TP}{TP + FN}$$

A high recall value indicates that very few harmful websites bypass the filtering mechanism.

B. Results of the Evaluation

The experimental results demonstrate that the proposed system effectively blocks restricted websites while maintaining a low false-positive rate. The achieved accuracy of 95.5% indicates that the filtering engine successfully classifies the majority of website access requests. The precision value of 96.9% confirms that very few legitimate websites are blocked unnecessarily, while the recall value of 94.0% indicates that most restricted websites are successfully detected and blocked [5][10]. The monitoring module achieved an accuracy of 98%, demonstrating reliable activity tracking and report generation capabilities. Additionally, the average response time of 0.18 seconds confirms that website filtering decisions are performed in near real-time, ensuring a seamless browsing experience. The system reliability score of 99.5% further validates the robustness of the implemented architecture [4].

C. Discussion of Evaluation Results

The evaluation results indicate that the Real-Time Parental Control and Website Blocking System provides effective protection against unauthorized website access while maintaining high monitoring accuracy and operational reliability [3][4]. The implemented filtering engine successfully balances security and usability by minimizing false positives and false negatives. Furthermore, the low response latency ensures that website blocking occurs almost instantaneously without noticeable delays to users [1][2]. Although the current system demonstrates excellent performance, future improvements can be achieved by integrating machine learning-based website classification techniques, automated blacklist updates, and AI-driven content analysis. These enhancements are expected to further improve filtering accuracy and adaptability against emerging online threats.

VII. DISCUSSION

A. Advantages

- Real-Time Protection: Instant website blocking prevents access to harmful content [2].
- Easy Monitoring: Parents can view browsing activities remotely.
- Flexible Control: Custom blacklists and whitelists can be configured.
- Usage Analytics: Provides insights into internet usage patterns [7].
- Scalability: Can support multiple devices.

B. Limitations

- Encrypted Websites: Some encrypted traffic may limit content inspection.
- VPN Bypass: Users may attempt to bypass restrictions using VPN services.
- Manual Rule Updates: Blacklist maintenance requires periodic updates.

- Platform Dependency: Browser extension support may vary across browsers.

C. Comparison with Existing Systems

Traditional parental control systems primarily focus on basic website blocking and time restriction functionalities, often relying on static filtering rules and manual configuration [1] [2]. Many existing solutions lack real-time monitoring capabilities and provide limited visibility into a child's online activities. In contrast, the proposed Real-Time Parental Control and Website Blocking System offers a more comprehensive approach by integrating real-time website monitoring, dynamic URL filtering, detailed activity analytics, and customizable rule management within a single platform. The system continuously analyzes browsing activities and instantly blocks unauthorized websites before they are accessed, thereby enhancing online safety. Furthermore, the inclusion of detailed reports, usage statistics, and centralized dashboard management enables parents to make informed decisions regarding internet usage. The lightweight architecture based on React.js, Node.js, and MySQL also ensures easy deployment and reduced resource consumption compared to many commercial parental control applications [5].

Practical Challenges

Several practical challenges were encountered during the development and implementation of the proposed system. One of the primary challenges was achieving real-time URL processing while maintaining minimal response latency. Since website access decisions must be generated instantly, the filtering engine was optimized to perform rapid blacklist, whitelist, and category verification without affecting browsing performance [2][4]. Another challenge involved ensuring cross-browser compatibility, as different browsers implement extension APIs differently, requiring additional testing and adaptation to support multiple platforms effectively. Data privacy and security also presented significant concerns because the system stores browsing histories, activity logs, and user information. Appropriate security mechanisms such as password encryption, secure API communication, and access control policies were implemented to safeguard sensitive data [7][9].

C. Adoption / Integration

The adoption across various environments where internet safety and monitoring are essential. In residential settings, parents can utilize the system to supervise children's online activities, enforce screen-time restrictions, and prevent access to harmful content [3]. Educational institutions, schools, libraries, and computer laboratories can deploy the system to create controlled browsing environments that promote productive internet usage and prevent access to inappropriate websites [6]. Due to its modular architecture, the system can be easily integrated with existing network infrastructures and monitoring solutions. Furthermore, future integration with mobile parental control applications, cloud-based monitoring platforms, and artificial intelligence-driven content filtering systems can enhance functionality and extend monitoring capabilities across multiple devices and operating systems [8].

VIII. CONCLUSION

This paper presented the design and implementation of a Real-Time Parental Control and Website Blocking System. The proposed solution successfully combines website filtering, activity monitoring, screen-time management, and reporting functionalities within a unified architecture [4][5]. The implementation demonstrates the feasibility of providing effective parental supervision through real-time monitoring and access control mechanisms. The modular architecture ensures scalability and adaptability for future enhancement.

Future developments include integrating machine learning-based content classification, mobile application support, cloud synchronization, behavioural analytics, and AI-driven threat detection mechanisms. These enhancements will further improve system intelligence, adaptability, and overall effectiveness in protecting children from online threats [10].

REFERENCES

[1] J. Jiang, J. Chen, K.-K. R. Choo, C. Liu, K. Liu, M. Yu, and Y. Wang, “A Deep Learning Based Online Malicious URL and DNS Detection Scheme,” in Proc. SecureComm 2017, Niagara

Falls, Canada, pp. 354–371, doi: 10.1007/978-3-319-78813-5_22, 2018.

[2] Y. Zhauniarovich, I. Khalil, T. Yu, and M. Dacier, "A Survey on Malicious Domains Detection through DNS Data Analysis," ACM Comput. Surv., vol. 51, no. 4, pp. 1–36, doi: 10.1145/3191329, 2018.

[3] Y. M. Pa Pa, K. Yoshioka, and T. Matsumoto, "Detecting Malicious Domains and Authoritative Name Servers Based on Their Distinct Mappings to IP Addresses," J. Inf. Process., vol. 23, no. 5, pp. 623–632, doi: 10.2197/ipsjjip.23.623, 2015.

[4] S. S. Monroy, A. K. Gupta, and G. Wahlstedt, "Detection of Malicious DNS-over-HTTPS Traffic: An Anomaly Detection Approach Using Autoencoders," arXiv preprint, arXiv:2310.11325, 2023.

[5] A. O. Egwali and R. O. Ekhator, "A System for the Detection of Malicious Domain Names Using Improved Deep-Learning Model," Trans. Nig. Assoc. Math. Phys., vol. 19, pp. 63–70, doi: 10.60787/tnamp-19-63-70, 2024.

[6] H. At Thooriqoh, M. N. Azzmi, Y. A. Tofan, and A. M. Shiddiqi, "Malicious Traffic Detection in DNS Infrastructure Using Decision Tree Algorithm," JUTI: J. Ilmiah Teknologi Informasi, vol. 19, no. 3, pp. 214–222, doi: 10.12962/j24068535.v19i3.a1054, 2024.

[7] L. F. Gonzalez Casanova and P.-C. Lin, "Malicious Network Traffic Detection for DNS over HTTPS Using Machine Learning Algorithms," APSIPA Trans. Signal Inf. Process., vol. 12, no. 2, pp. 1–10, doi: 10.1561/116.00000058, 2023.

[8] S. Papadamou, P. K. Papadopoulos, S. Iosifidis, I. Leontiadis, A. K. Athanasiou, I. Ioannidis, and N. Kourtellis, "Disturbed YouTube for Kids: Characterizing and Detecting Inappropriate Videos Targeting Young Children," Proc. 13th Int. Conf. Web and social media (ICWSM), pp. 522–531, 2019.

[9] N. Charalambous, A. Kokkinos, and M. Anastasiou, "A Privacy-Preserving Architecture for Parental Control Systems," Proc. IEEE Conf. Information Security and Privacy, pp. 121–128, 2020.

[10] K. Marques, A. Silva, and P. Machado, "Machine Learning-Based DNS Firewall for Real-Time Threat Detection," Proc. IEEE Conf. Communications and Network Security, pp. 338–345, 2021.