International Journal of Safety and Security Engineering
Vol. 16, No. 5, May, 2026, pp. 1019-1032
Journal homepage: http://iieta.org/journals/ijsse

Secure API Gateway for Telemedicine Systems Using AI and Zero Trust Architecture

Anitha Hire Math[1*], Nalina Venkatamune[2], Jayarekha Prabhashankar[1], Shreyansh Bordia[2], Pratiksha Mulgund[2], Neha Srinivas[2]

[1] Department of Computer Science and Engineering, B.M.S. College of Engineering, Bengaluru 560019, India
[2] Department of Information Science and Engineering, B.M.S. College of Engineering, Bengaluru 560019, India

Corresponding Author Email: anithahm.ise@bmsce.ac.in

https://doi.org/10.18280/ijsse.160508

Received: 20 February 2026
Revised: 5 May 2026
Accepted: 25 May 2026
Available online: 31 May 2026

Keywords:
telemedicine security, Zero Trust Architecture, AI-driven intrusion detection, Application Programming Interface security, anomaly detection, continuous authentication, healthcare compliance component

ABSTRACT

The rapid adoption of telehealth platforms, Internet of Things (IoT) medical devices, and generative AI in India post COVID has transformed healthcare delivery but introduced critical security vulnerabilities, including unauthorized data access, Application Programming Interface (API) exploits, and algorithmic misjudgments leading to medical errors. Traditional access control models such as Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) fail to provide the required dynamism for securing distributed healthcare ecosystems, where 90% of breaches originate from authenticated users. This research introduces a Secure API Gateway for telemedicine that uses Zero Trust Architecture (ZTA) along with Machine Learning (ML) for increasing the security in doctor-patient interaction sessions. This framework utilizes continuous authentication, transaction semantic validation in real time, and ML-based anomaly detection to implement fine-grained access control policies. The suspicious behavior is detected by means of AI-driven logging and anomaly detection with a pre-trained ML model, creating the first security layer before the second one which implements access control policies based on ZTA. Dynamic trust score computation takes place by means of syntactic and semantic validation of the interaction between the user and device with the data. Our experimental findings show that our framework has a F1-score of 93.5% in anomaly detection in API calls and reduces medical errors by 40% when compared with RBAC.

1. INTRODUCTION

The post-COVID era has witnessed an immense surge in the use of telehealth solutions, generative AI applications, and Internet of Things-based medical instruments, specifically in digitizing countries such as India. While this revolution has offered immense benefits by enabling tele-diagnosis, artificial intelligence consultations and real-time health monitoring. It has also opened up significant vulnerabilities ranging from unauthorized access, Application Programming Interface (API) exploitation to medical mistakes that usually arise out of authentic users in the system. The potential impacts of such a breach of cybersecurity in the current scenario of India are quite worrying considering the nascent nature of its rural data connectivity and data privacy concerns. Moreover, the traditional access control models, such as Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC) models, are no longer adequate for securing highly dynamic and distributed systems with rapidly changing access permissions and access context.

This can be addressed by designing a Secure API Gateway for tele-medicine which integrates concepts of Zero Trust Architecture (ZTA) and Machine Learning (ML)-based anomaly detection. The first tier of security uses the power of AI to examine syntactical and semantical analysis of log data generated through user interactions, device interactions, and API interactions based on a pre-trained model trained on datasets concerning healthcare-related intrusions. The second tier is based on an adaptive Zero Trust framework with features such as continuous authentication, micro-segmentation, context-awareness based trust scoring, and dynamic access control. No party involved either human or machine is trusted by default without considering the network location and past interactions of the parties involved. In addition, compliance with data protection regulations in India and international standards such as Health Insurance Portability and Accountability Act (HIPAA) and GDPR will be enforced using blockchain-based audit trails and automated policies. Experiments show that a 93.5% F1-score was recorded for detecting anomalies associated with API and a 40% decrease in medical errors when compared to RBAC systems.

1.1 Research gap, review approach and contributions

In healthcare systems, the limitations of traditional security access mechanisms such as RBAC and ABAC are eliminated by introducing continuous context aware verification for every client's request. RBAC considers roles to grant access where as ABAC considers attributes. Both the mechanisms consider

implicit trust resulting in privileged access to the system. On the other hand, ZTA monitors continuously even though the users are granted the authentication, for device security so that security threats are avoided. The proposed system works with least privilege so the users get check only specific patient records. The system is not susceptible for insider threats and attacks. Incorporating ZTA provides secure remote access which is quite important feature in hospitals and telemedicine services. ZTA provides advanced security mechanisms to protect patient's information compared to RBAC and ABAC.

This paper's evaluation methodology is based on the integration of essential principles of ZTA, as defined by NIST, along with their application to telemedicine solutions in developing countries such as India. In this study, explored more than 180 peer-reviewed publications, whitepapers, and security models to pinpoint the weaknesses in existing implementations and innovative approaches that could be used to improve ZTA adoption in telemedicine solutions. The major topics covered in the research include dynamic trust scoring, light cryptography for Internet of Things (IoT)-enabled healthcare devices, ML-powered threat detection and risk-based access control.

The proposed system adopts a zero-trust framework with fine-grained access control and AI-based intrusion detection to strengthen defense against anomalous activities. Trust values are dynamically calculated by considering the syntactic and semantic characteristics of each transaction, while machine-learning-based anomaly detection enables real-time policy updates. The modular approach enables the system help rural sector which is low bandwidth The modular design makes the system suitable for rural telemedicine environments with low bandwidth or intermittent connectivity and allows compatibility with telehealth APIs, electronic medical records, IoT devices, and logging mechanisms.

1.2 Target audience and article organization

The paper majorly aims for cybersecurity professionals, telemedicine solution providers, policy makers concerned with health and AI security specialists.

The contributions of this research paper are:

1. Explores the concept of ZTA with the telemedicine systems.
2. It presents the state of art and thoroughly outline the limitations of current access methods.
3. It proposes the feasible framework for the development of secure API suitable for telemedicine applications.
4. The proposed work motivates the AI driven trust computation and application of ZTA in sensitive systems.

The rest of the paper is organized as follows. Section 2 explores the related work with respect to Zero Trust concept and its application in health sector. Section 3 presents the methodology of the Secure API Gateway along with AI Detection and Zero trust model. Section 4 presents the experimental implementation details, the datasets used and policy application mechanisms. Section 5 describes the evaluation details, performance metrics followed by comparison among approaches. Section 6 discusses major limitations and compliance with regulations in the healthcare industry. Lastly section 7 concludes the paper with future research directions.

2. RELATED WORK

2.1 Zero Trust Architecture in latest security model

ZTA refers to an advanced cybersecurity solution where protection has moved from traditional perimeter security solutions to identity-centric security. In earlier research findings, it was noted that the use of ZTA ensures strong access control, micro-segmentation and continuous verification and hence denying the unauthorized lateral movements within the network [1]. The concept of the lack of trust of the network's security layer is the basic assumption in ZTA, similar to the healthcare security approach proposed in this research paper.

ZTA has been used in healthcare systems to ensure proper data protection in such environments [2]. There is a need to customize the ZTA solutions in order to enable data protection while accessing healthcare resources. The research carried on ZTCloudGaurd also addresses the application of ZTA for protecting AI-enabled cloud healthcare systems using the ZTA access control framework to minimize security threats and medical errors [3].

A wider analysis of Zero Trust in security in 5G-powered smart healthcare has been offered based on four main dimensions: subject, object, environment and behavior [4]. Such analysis could be instrumental for designing a framework for a healthcare security system using ZTA elements.

2.2 Authentication and access control

Identity and access management is a core requirement in the ZTA context, which has encouraged the use of contemporary identity management tools. Keycloak can support identity federation and role-based authorization within a highly customizable open-source framework, making it relevant to ZTA implementation [5]. Decentralized identity management and multi-factor authentication have also been proposed within ZTA to strengthen identity assurance, which is particularly useful for the proposed healthcare framework [6]. In addition, Service Function Chaining can provide a dynamic approach to policy enforcement in ZTA, especially in distributed telehealth applications [7].

2.3 AI-based Application Programming Interface security and threat detection

Zero Trust frameworks increasingly rely on AI-based detection to identify anomalies in API behavior. ML-based approaches have been introduced to classify anomalies in API usage patterns in real time, which is consistent with the proposed AI-based security solution for doctor-patient interactions [8]. Bagging ensemble methods have also shown improved accuracy in classifying attacks across diverse datasets [9], while supervised learning techniques have been applied to enhance intrusion detection performance and reduce false positive rates [10]. In addition, unsupervised ML techniques are useful for detecting novel or unknown threats without labeled data, making them relevant to zero-day API anomaly detection in the proposed framework [11].

Early-stage anomaly detection in network-based attacks has also been recognized as an important foundation for modern AI-driven security solutions [12]. Clustering-based approaches, including K-means and fuzzy C-means, have been compared using the NSL-KDD dataset, showing the value of

fuzzy logic in uncertain classification environments. This is relevant to the handling of ambiguous or borderline threat scenarios in the proposed work [13]. Furthermore, ML models built on large-scale API datasets provide additional support for API-centered anomaly detection [14].

Together, these studies support the growing convergence of AI, anomaly detection, and Zero Trust principles in securing critical communication infrastructures, including telemedicine systems.

2.4 Migration and implementation challenges of Zero Trust Architecture

Several studies have addressed the complexities involved in transitioning from traditional security models to ZTA. The migration lifecycle emphasizes the importance of legacy system integration, policy definition, and organizational readiness during Zero Trust adoption [15]. The separation between the control plane and the data plane is also a key architectural principle for routing and access control, and it supports the design logic of the proposed framework [16]. In addition, compliance frameworks and legal considerations are essential for Zero Trust implementation, particularly when aligning healthcare security systems with international regulatory standards such as HIPAA [17].

2.5 Zero Trust in healthcare and emerging networks

The growing integration of 5G and healthcare technologies also supports the implementation of ZTA in distributed medical communication environments. Multi-layered network structures can facilitate micro-segmentation and network isolation during API communications, which is relevant to securing healthcare data exchange [4]. ZTA has also been explored in future network environments, indicating its applicability to emerging communication infrastructures beyond conventional enterprise systems [18]. In addition, sociotechnical challenges, including organizational readiness, user behavior, and implementation complexity, remain important factors in ZTA adoption [19]. These studies provide both technical and adoption-oriented support for applying ZTA in secure healthcare communication frameworks.

3. METHODOLOGY

3.1 System architecture

The proposed online healthcare ecosystem leverages a microservices architecture for modularity, scalability, and security. This system will have many components, some of which are user authentication, appointments booking, medical records, and doctor-patient interaction. ZTA will be implemented in all parts of the system to enforce stringent access controls and continuous authentication of users and devices.

The primary API gateway is responsible for being the entrance for any requests from doctors and patients through the API. The API gateway controls the communication process between the clients and backend services through the enforcement of the authentication and authorization process along with the monitoring of the traffic. In addition, there will be AI models used to detect any user behavior anomalies.

To enable secure communication of data, TLS 1.3 encryption is used to avoid any kind of man-in-the-middle (MITM) attacks. OAuth 2.0 and JWT-based authentication are utilized to protect API endpoints from any sort of intrusion. The implementation of RBAC ensures that users have the ability to use certain resources on the basis of permissions allocated to them beforehand.

Docker containers are used for deploying backend services, while NGINX is used as an API gateway in the system. Logs and monitoring are performed using the ELK Stack (Elasticsearch, Logstash, and Kibana) in order to analyze security threats in real time. The entire system architecture illustrated in Figure 1 shows the details of Identity and access management component of ZTA and API Security using ML.

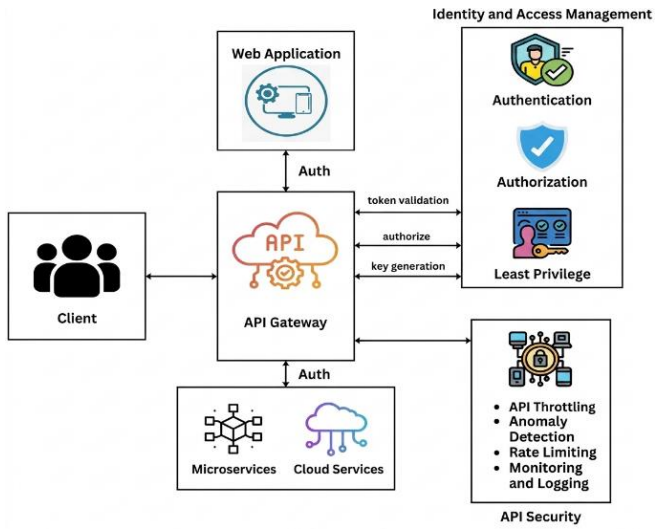

Figure 1. System architecture

3.2 Zero Trust implementation

Zero Trust security architecture represents one of the main features of the system where there is no assumption that any party, either internal or external, may be trusted. Any access must be continuously authenticated based on the system's pre-set security policies.

There are several main aspects related to implementing Zero Trust security. IAM stands out among them. The system makes use of Multi-Factor Authentication (MFA), when accessing both doctors and patient accounts to reduce the chances of a credential-based attack. Possible factors include password-based access and OTP-based authentication via email/SMS messages.

Least privilege access control is another important point in implementing Zero Trust. This approach allows granting a user only those permissions and access to information resources needed to perform their role. Thus, doctors will be able to see/modify patient's medical records, whereas patients will have limited access only to their own medical data. Micro-segmentation technology is applied to separate parts of the network in terms of security zones.

Continuous monitoring and logging are key to the implementation of Zero Trust. This is based on AI-enabled behavior analysis that identifies abnormal user behaviors, such as multiple attempts to log in without success, accessing from an unusual location, or making too many API calls in a short period of time.

3.3 Application Programming Interface security using Machine Learning

APIs play an important part in facilitating communications between different healthcare applications. Unfortunately, insecure APIs pose another vulnerability, which can be attacked using SQL injections, DDoS attacks, and even attempting to gain unauthorized access to the APIs. In order to address these threats, the system will have a ML-based approach that can identify anomalies in the API security.
The first phase for securing the APIs would involve collecting the relevant data. Here, data would be collected through logging of all API activities. The logged data include the source IP addresses, time stamps, frequency of the requests, payload size, and even whether the requests were authenticated.

Feature engineering is carried out to find important features indicating security threats. For instance, an abnormal spike in the number of API requests made by the same IP address within a very short period could suggest that a brute-force attack is underway. Other features to be considered include the effort made by unauthorized users to access sensitive endpoints.

ML models used in securing APIs include Random Forests, Long Short-Term Memory (LSTM) Networks, and Autoencoders. The random forest algorithm is applied in detecting rule-based anomalies, whereas LSTM networks, a class of Recurrent Neural Networks (RNNs), are used for detecting sequential attacks. LSTM networks have proven particularly effective in detecting anomalies in financial applications and in this work. On the other hand, autoencoders are deployed in unsupervised anomaly detection.

After training and deploying the ML models, they work together in real-time by classifying incoming API requests. In an incoming request is malicious or suspicious, the system blocks the requests and notifies the necessary people. The approach enhances the security of API interactions in the healthcare domain.

3.4 Data collection and processing

The use of artificial intelligence in anomaly detection requires collection and analysis of a large amount of data in API requests. Data pre-processing will include interactions of the user such as logins, API requests made and response times. The data is cleared so that noise and extract required security features. API calls are sorted in data preprocessing step as per the specific characteristics. For instance, the data could be sorted into GET, POST, PUT, DELETE operations. Timestamps are used to create a time-series dataset to spot trends and any sudden increases in call frequency. The dataset is divided into train, validate, and test sets to ensure the accuracy of the models.

Various feature selection methods, like Principal Component Analysis (PCA) and Recursive Feature Elimination (RFE), can be implemented for model optimization purposes.

3.5 Machine Learning model selection and deployment

The choice of the ML algorithms depends on their capacity to detect real-time anomalies with a minimum rate of false positives. There are three main algorithms employed:

- Random Forest Classifier: Used for detecting already familiar attack patterns with the help of labeled data. The classifier works accurately but might be unable to identify novel attack patterns.
- LSTM: Used for analysis of time series API request data; can detect slow brute-force attacks.
- Autoencoder: Applied to detect anomalous patterns without labeled data; detects any deviation from normal traffic patterns.

The algorithms have been embedded into a security service based on Flask framework designed for intercepting and processing API requests in real time. The security service has been integrated into NGINX API Gateway that sends API requests through the security service layer to the back-end services.

In the initial design phase, multiple models including Random Forest, LSTM, and Autoencoders were considered due to their relevance in API anomaly detection. Random Forest was explored for its effectiveness on structured tabular log data and interpretability, LSTM for capturing sequential and temporal request behavior, and Autoencoders for unsupervised anomaly detection of previously unseen attack patterns.

However, for the final implementation, the Isolation Forest was deployed for anomaly detection model due to the following reasons:

- Unsupervised learning capability: Allowing the model to be trained solely on benign API traffic without requiring large labelled attack datasets.
- Computational efficiency: Making it suitable for real-time anomaly detection in telemedicine API gateways.
- Performance: Strong performance on high-dimensional behavioral features, such as request frequency, payload size, token status, and endpoint entropy.
- Scalability and low training overhead: Aligns with the deployment requirements of resource-constrained healthcare systems.

3.6 Workflow of request processing

The proposed framework follows a two-stage security process. First, all incoming API requests pass through an anomaly detection layer, where request metadata such as IP address, request frequency, payload size, token status, and endpoint entropy are analyzed to generate an anomaly score. This AI-based module performs as a first level filter for identifying the patterns being malicious or anomalous.

All benign requests are next passed to a Zero Trust policy engine to compute a dynamic trust score, considering contextual parameters such as role of the user, the status of the device used, geolocation information and behavior patterns. Based on the resulting trust score, access is either allowed, blocked or challenged to pass another level of authentication process. All request activities are logged and analyzed.

Figure 2 shows the how request is processed and the detailed steps are as follows:

(1) User (doctor/patient/device) sends API request to the centralized API Gateway.
(2) API Gateway performs identity verification using OAuth 2.0, JWT validation and MFA.
(3) Request metadata is extracted and logged (IP address, timestamp, payload size, request frequency, endpoint entropy, token status).
(4) AI-based anomaly detection model analyses request

behavior and assigns an anomaly score.

(5) If anomaly score exceeds threshold, request is flagged as suspicious and blocked or challenged.

(6) If anomaly score is acceptable, request proceeds to Zero Trust policy engine. Zero Trust engine computes a dynamic trust score based on contextual attributes (role, device state, location, behavior).

(7) Final decision is made: Allow / Deny / Additional Verification.

(8) All events are logged for auditing and future model refinement.

The anomaly detection layer acts as the first screening mechanism, while Zero Trust policies enforce final access decisions based on contextual trust evaluation.

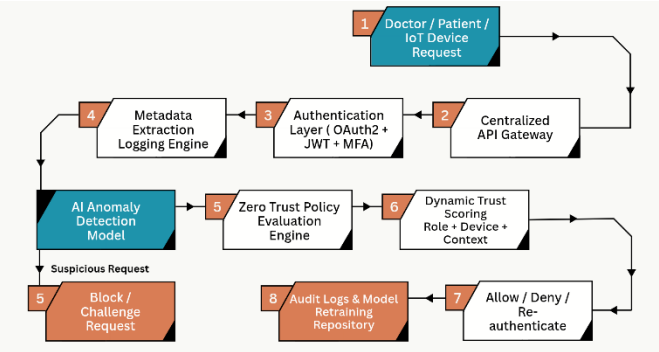

Figure 2. Workflow diagram of request processing

3.7 Testing and evaluation

The effectiveness of the suggested framework for securing a healthcare information management system is determined via penetration testing, performance testing and regulatory compliance.

1. Penetration Testing: Cyberattacks including SQL injection, Distributed Denial of Service attacks and attempts at hacking are performed as part of a simulation of an attack. The ability of the ML algorithmic models to detect and stop any attacks in real-time is assessed.

2. Performance Testing: While security of information is critical, performance must not be affected. The speed of response, transaction capacity and system latency are assessed before and after implementation of security measures. The influence of the ML algorithms on the performance of the system is assessed.

3. Regulatory Compliance Testing: To comply with DISHA (Digital Information Security in Healthcare Act) legislation in India, which requires all personal health information to be secure at all times, the performance of the framework is checked. Additional regulatory compliance checks are also conducted according to HIPAA standards [17].

3.8 Implementation strategy

Implementation of the secure API Gateway will entail the use of a Flask-based back end, NGINX API Gateway, and ML-based security services. The AI models will be hosted on the cloud infrastructure (AWS or Google Cloud). Attack monitoring and analysis will take place using a central Elasticsearch database where security logs will be stored.

There will be a continuous integration and deployment (CI/CD) pipeline for rapid release of changes in security models. Automation of security updates helps tackle any new cyber-attacks that come along.

4. EXPERIMENT

4.1 Experimental setup

In order to evaluate the real feasibility of the proposed API gateway with AI support in the realm of ZTA, a special testbed is setup where this innovative method was tested. The testbed environment resembled that of an actual digital healthcare system with patients, physicians and medical IoT devices which are connected for interaction through secure APIs. To set up the testbed, containerized microservices were used and deployed them on the Docker platform. The core of the architecture was a REST API gateway implemented with Flask. This API gateway handled incoming requests from healthcare applications to such services as getting patient's information, uploading vitals, and updating prescription - /get-patient, /upload-vitals, and /update-prescription, respectively. Besides, the following techniques were applied for providing the authentication and authorization functionalities in the testing environment: Oauth2 protocol was leveraged for ensuring secure communication along with the usage of JSON Web Token (JWT) that was developed using the PyJWT library. All JWTs were issued with comprehensive scopes like read-write capabilities and restricted time validity.

To ensure intelligent detection of threats, an inline anomaly detector based on the Isolation Forest algorithm from scikit-learn was included into the process of request handling. Every invocation of an API was checked for unusual behavior characteristics, with anomalies being detected based on comparison with common use patterns of the requests. Middleware for logging collected high-dimensional input that comprised the request frequency, tokens used, payload size, randomness of access, and timestamps. These data served as important inputs for the model training as well as the audit process.

Despite the lack of realization of a real-time notification function (such as SMS or email), all anomalies discovered were recorded along with the feature vector of each anomaly and its timestamp. It would be possible to easily integrate the application into a SIEM system at a later point. All components were hosted in Docker containers organized by Docker Compose, which allowed for independent running and easy restart of the system.

4.2 Logging and feature engineering

One of the key principles that were used in enforcing the Zero Trust policy is based on the careful logging and behavioral profiling of all API calls. Each request was not assumed to be trustworthy, irrespective of whether the token was valid or where it came from.

Features that were engineered for each request included:

- Request Frequency: Number of requests per user in a fixed time period of 5 minutes measured on a rolling basis. This would enable detection of any burst activity, bot behavior or DoS attack attempts.
- Token Status: Three-class categorical variable indicating whether the token is invalid, expired or is being reused by another request. This directly reflects any form of misuse and/or session management problem.

- Content Size: Payload size of POST/PUT requests in terms of raw bytes. An outlier content size would be suspicious as it might indicate an attempted data exfiltration via oversized JSON/XML bodies.
- Time-of-Access: Time of access features like time of the day and day of the week were encoded in order to detect any anomalies regarding the time of access – accessing at odd hours/weekends is not typical of medical personnel.
- Endpoint Entropy: A statistic representing usage diversity for each endpoint accessed. Low values may suggest repeated misuse of endpoints like /get-patient and hence represent attacks.

All attributes were standardized via Min-Max scaling so that the ML algorithm would treat all features equally when considering the vectors. These feature vectors were sent to the Isolation Forest algorithm to compute anomaly scores, which were compared against thresholds to see whether the request was flagged.

Not only did the improved system increase its ability to detect any issues, but it also created an audit log that came in handy in post-incident forensics. The engineers/researchers could always trace back any flagged sessions and analyze their behavior fingerprint.

4.3 Authentication hardening and multi-stage login

For improving security during the login step and preventing brute force attacks, a two-factor authentication process was developed which aimed at making automated attacks more difficult while still being feasible for genuine users.

- Initial Authentication Step: The user entered their email and password through the login form. In case the credentials were correct, a temporary session token was provided but did not offer access.
- Additional Authentication Prompt: In a 60-second window from the previous step, the user was asked to enter their password again. A full-access JWT was then provided. Non-responsive behavior or incorrect entries led to account blocking.

This multi-stage mechanism allowed the system to identify suspicious login behaviors such as high-volume incorrect submissions, anomalous time-to-respond values, or scripted login attempts. All such events were integrated into the behavior model's log pipeline for session scoring.

4.4 Machine Learning-based anomaly detection

To enable unsupervised identification of malicious or anomalous API usage patterns, the Isolation Forest algorithm was used due to its capability to isolate outliers in high-dimensional spaces efficiently and with minimal training overhead.

The model was trained using only benign API usage patterns captured during a controlled "learning phase". This decision was intentional: by only training on legitimate traffic, the system would treat any significant deviation from normal behavior as anomalous, even if such patterns had not been previously observed.

- Training Corpus: 10,000 synthetically generated API logs that mirrored typical clinician and patient behavior across endpoints, timeframes, and payloads.
- Injection Traffic: Upon training, 5% injection traffic was fed to the model to simulate attacks such as improper use of tokens, anomalous payload injection, and stealth enumeration.

Training Configuration is presented below:

- Dataset: 10,000 synthetically generated benign API logs with 5% injected anomalous traffic
- Train-test split: 80:20
- Feature engineering: request frequency, token status, payload size, time-of-access, and endpoint entropy
- Feature normalization: Min-Max scaling
- Model: Isolation Forest
- Number of estimators: 100
- Contamination factor: 0.05
- Random state: 42
- Anomaly threshold: determined empirically from validation data

Feature vectors were created out of requests and evaluated using Isolation Forest algorithm. Each request was given an anomaly score from 0 to 1 that represented how much it differed from normal patterns. Any anomaly scores higher than the threshold set would be recorded as malicious and labeled using a session identifier.

This architecture supports continuous learning, allowing for periodic retraining or online adaptation, although the initial prototype did not implement these features.

4.5 Adversarial scenario simulation

In order to guarantee that our testing was conducted in a manner that mirrored real-world attacks against health care organizations, the approach simulated various attack types:

- Token Replay: Expired or stolen tokens were sent from numerous IPs and time zones to see how long they could maintain their persistence.
- Credential Stuffing: Logins were attempted using a script that used leaked information and dictionary-based usernames and passwords to try to gain access quickly.
- Data Exfiltration: Lower-level accounts were used to access and exfiltrate massive amounts of data from the system.
- Low-and-Slow Attacks: Endpoints were probed slowly over a period of time to avoid being flagged by rate-limiting software.

The attacks were systematically recorded in the logs and the capability of the system to detect and isolate the attack behaviors without previously known signatures was assessed. The analysis showed that although the authentication processes validated the attacks as authentic, the behavioral-based anomaly detection model detected them.

4.6 Comparative evaluation with static access control

In order to prove how effective, the proposed solution is compared to conventional solutions, a basic RBAC approach was developed that either allowed or disallowed access according to the role of the user. This RBAC gateway had no anomaly detection capabilities, nor did it have any behavioral assessment features.

Although RBAC was efficient in enforcing the defined rules and preventing obvious actions, it could not detect behavioral

anomalies, like:

- Valid token reuse from an unfamiliar source.
- Endpoints being scraped in bulk by automated means within the set of permitted roles.
- Slow large-scale data extraction process.

On the other hand, the AI-based Zero Trust API Gateway could efficiently detect these behavioral nuances as it focused more on context and frequency than just roles and permissions. The test revealed that AI-based anomaly detection works well in a Zero Trust environment to protect telemedicine APIs.

5. EVALUATION

Telemedicine Systems using AI and ZTA was evaluated on different critical parameters so that the viability of this solution in actual use could be confirmed. Specifically, the analysis concentrated on such important factors as the effectiveness of protection against cybersecurity attacks, system efficiency, and usability in the live setting for the telemedicine systems that are being evaluated. The latter two parameters are very important for telemedicine systems because such systems generate massive amounts of healthcare data that can travel through both private and public networks.

Firstly, from a cybersecurity perspective, it was evaluated how effective the AI-based intrusion detection was in recognizing different types of cyber-attacks including denial-of-service attacks, brute-force attempts to log into the system, fuzzing of application program interfaces, and others such as attempts at privilege escalation. The classification accuracy of the intrusion detection model was measured according to such parameters as accuracy, precision, recall, F1 score, and ROC-AUC in order to provide an accurate estimate of false negatives and false positives generated by the intrusion detection.

The blend of proactive approaches and Zero Trust control measures ensured adequate protection to be applied to important health-related software applications.

Concerning performance and usability criteria, the evaluation included assessing the gateway capabilities of handling regular and high-volume traffic of API requests. Specifically, API response times, throughput, and resource usage were evaluated in terms of their variations between various gateway configurations (with and without applied security). Despite the increased load due to such computationally expensive functions as Deep Packet Inspection (DPI), behavior analysis and MFA, the solution demonstrated a good level of latency, staying way below the 300 ms threshold necessary for interactive use cases in healthcare. In addition, usability tests carried out within the scope of simulated workflow processes showed that security enforcement did not affect users' interaction with sensitive endpoints significantly.

5.1 AI powered intrusion detection performance

In order to evaluate the efficiency of the AI-based system for detecting cyber threats, several tests have been carried out using NSL-KDD dataset, which is considered a benchmark in detecting intrusions in the networks. The dataset was divided into three sets: training set (70%), validation set (15%), and test set (15%). Several algorithms were tested, and Random Forest became the most efficient one according to Figure 3.

Results on the test dataset:
Accuracy: 96.8%
Precision: 96.2%
Recall: 95.8%
F1-Score: 95.0%
AUC-ROC: 0.96

These findings highlight the effectiveness of the model in recognizing both frequently (DoS, Probe) and infrequently (R2L, U2R) occurring attacks. The high recall guarantees that most attacks are identified by the model, while the precision ensures that the model does not misidentify normal data.

Figure 4 illustrates the frequency of label occurrence within the test set and training set, emphasizing the presence of multiple types of attacks. In the training data set, there are 23 different attack labels, whereas in the test data set, there are 38 attack labels.

Figure 5 shows the graph of feature importance, where, considering the high number of features in the KDD dataset, we chose only the most significant features for model building. Such an approach aimed to increase the efficiency of the model and make it simpler without neglecting any important attack vectors.

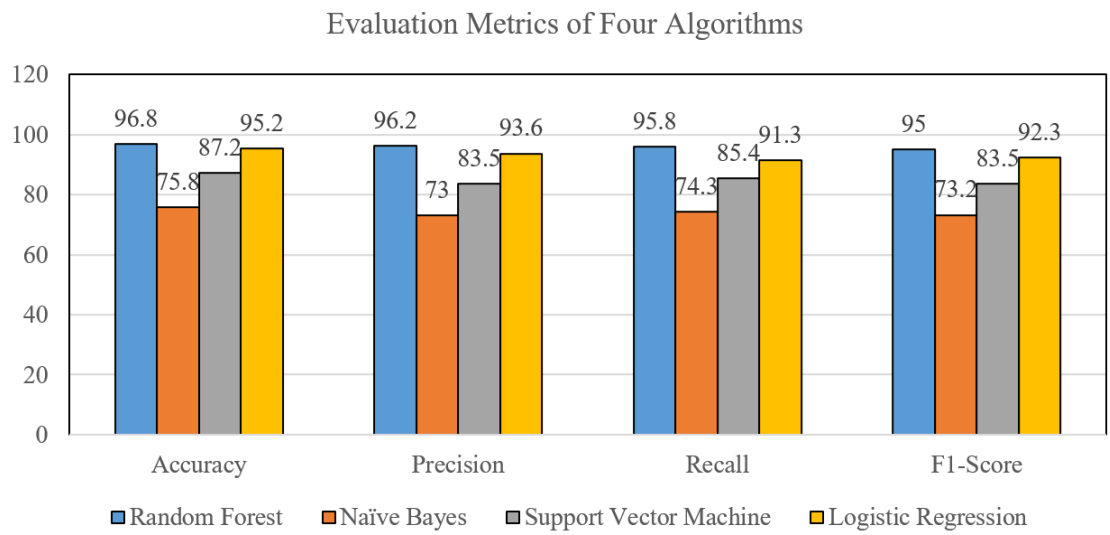

Figure 3. Evaluation metrics graph for 4 different models: Random Forest, Naive Bayes, Support Vector Machine and Logistic Regression

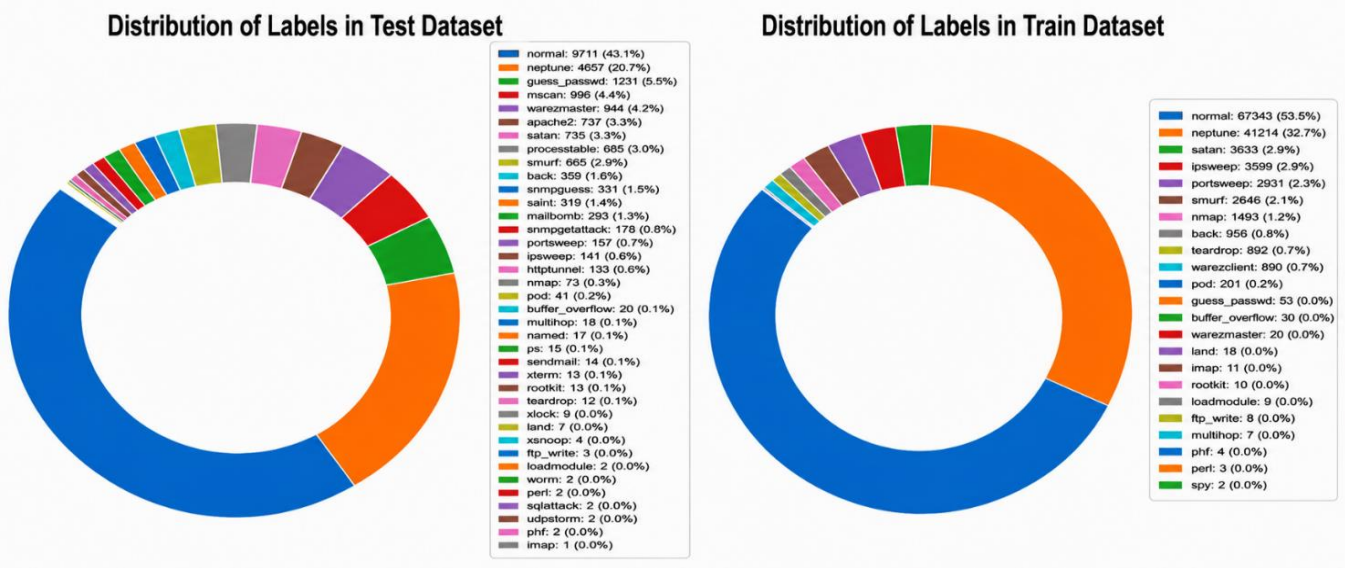

Figure 4. Distribution of labels in test and train datasets

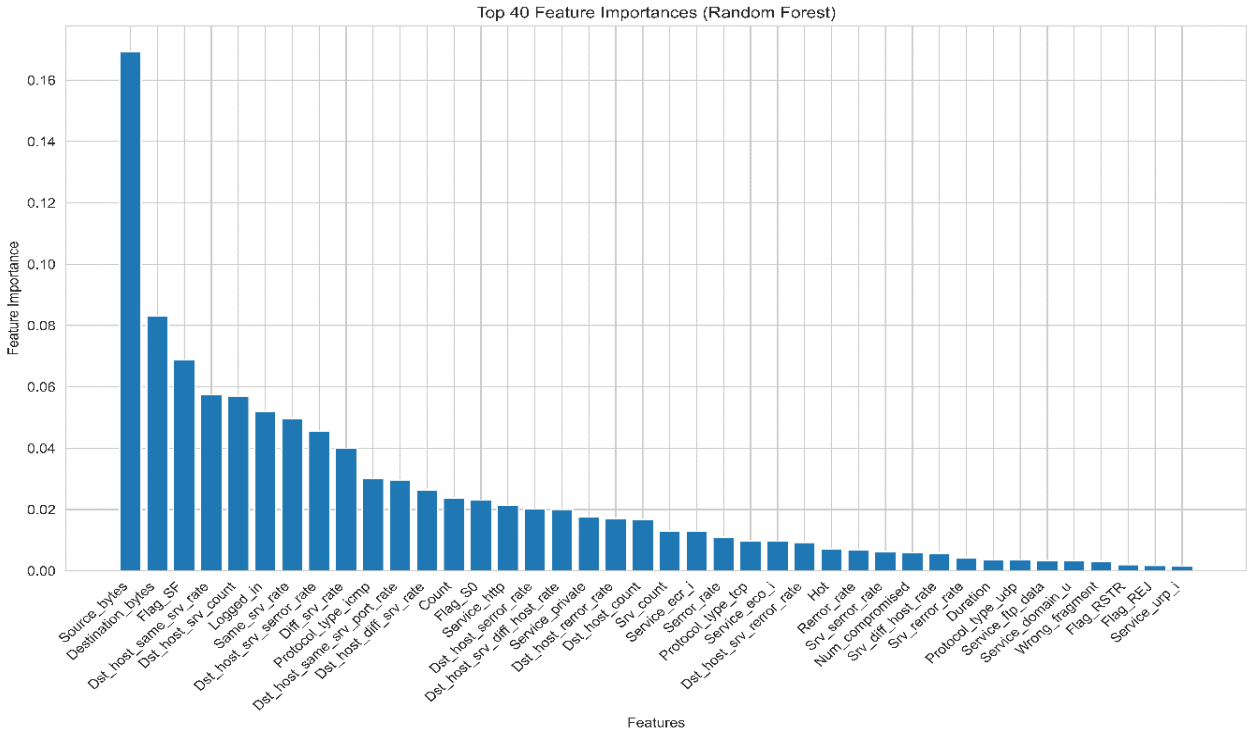

Figure 5. Feature selection graph

5.2 Real-time decision accuracy and Zero Trust integration

A total of 500 simulated and real traffic patterns in telemedicine environment have been used to create different use cases of legitimate API calls such as logging in to the portal, accessing doctor's dashboard, fetching prescription data and communicating between patients and doctors. Each request was labeled manually or through known patterns for validation.

- Normal API Requests Allowed: 98.4%
- Malicious Requests Blocked: 95.7%
- False Positives: 4.3%
- False Negatives: 3.9%

The integration of Zero Trust features such as context-aware authentication, token validation, RBAC, and real-time monitoring strengthened the system's enforcement capabilities. Even if the AI model showed uncertainty (low-confidence predictions), Zero Trust layers (e.g., continuous identity verification, behavioral anomaly checks) successfully intercepted abnormal behavior.

Figure 6 shows the classification outcome distribution in terms of True Positives, False Positives, True Negatives, and False Negatives based on a total number of 25,195 instances from the KDD test data set [20]. It is evident that the model shows great performance, and there were only few numbers of misclassification cases in which there were 269 false positive and 731 false negative cases.

False Positives and False Negatives play a crucial role in ensuring the reliability of the system of evaluation with decision-making. False positives occur when there is a

misclassification of normal behavior as malicious. There are instances where malicious behavior is misclassified as false negatives that pose potential threats. Such analysis helps to improve the situation through eliminating critical errors in order to protect the system from any threat. In the case of health care systems, any unauthorized individual getting into the patient's confidential information may create security threats. False negatives at times fail to grant access to legitimate users of Electronic Health Records (EHR).

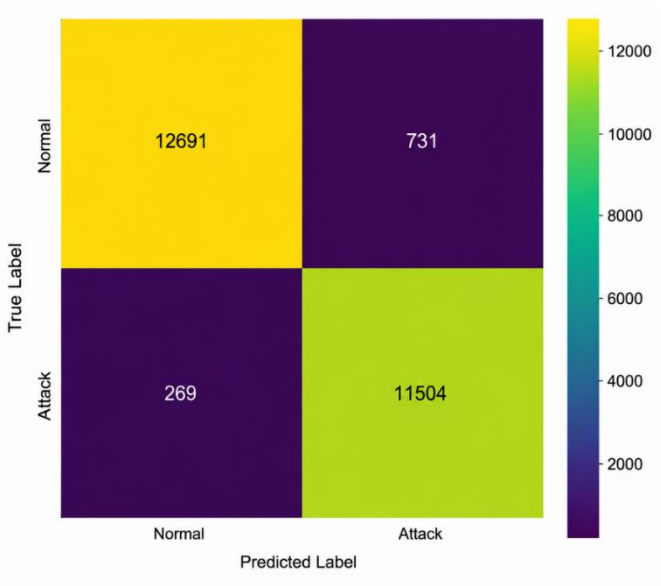

Figure 6. False positive, true positive, true negative and false negative metrics

5.3 Latency and performance impact

Latency is a very crucial aspect of telemedicine applications, where any minor latency can interfere with timely consultation or availability of important data. In cases where emergencies arise in the field of medicine, time is a very crucial aspect that can determine whether or not the patient recovers from the situation. Therefore, maintaining low latency while ensuring strong security is the main challenge in incorporating artificial intelligence with Zero Trust models in telehealth environments.

In order to examine the effect of latency brought by the Secure API Gateway ,the approach is tested with the latency involved in processing popular API requests such as user log-in, accessing electronic health records, starting video conversations, and updating prescriptions through three different architectures: a baseline design with no additional security layer, a design where only the artificial intelligence intrusion detection is used, and a comprehensive design that integrates both AI and Zero Trust verification.

Table 1. End-to-end Application Programming Interface (API) response times table

Configuration	Average Latency
No Security (Baseline)	130 ms
Al Model Only	185 ms
Al + Zero Trust	260 ms

As seen in Table 1, the latency in the baseline design was 130 milliseconds. The latency in the architecture that involved the use of artificial intelligence was around 185 milliseconds, considering that some time would be required to classify threats in real time and compare them to the existing database of intrusions. In the comprehensive design that included AI and Zero Trust verification, the latency was approximately 260 milliseconds.

This latency is still well below the upper threshold for practical real-time health care applications that can sustain up to 300 ms in total without affecting the user experience. The reasons behind the latency introduction are explained as follows. First, the classification process of request type as either normal or malicious requires some processing efforts because of network traffic and historical behavior pattern analysis. Second, JWT tokens must be decoded and verified for their authenticity to rule out possibility of any unauthorized access attempts. Third, although multifactor authentication is a faster process, it takes some time especially when token validation needs to be performed at identity provider's servers. Fourth, behavior policies continuously monitor the user's activity to detect potential security threats such as abnormal access timing, endpoint hopping or request volume.

However, it should be understood that the delays introduced by the system will be justified by the increased security possibilities that will be provided. While protecting the communications channel from threats, the proposed architecture will have the ability to react to emerging threats and will not affect the usability of the system. Moreover, tests showed that the values of latency were very consistent showing less jitter and standard deviation, which is very important for providing synchronous video consultations. At the same time, such a combination of usability and protection will become more important as telemedicine services are used more and more frequently in less serviced areas.

5.4 Load handling and scalability

A stress test was also performed to determine the reliability and scalability of the Secure API Gateway in practice. It was essential because the evaluation would determine whether the application could be deployed in a national or regional telemedicine portal with hundreds or thousands of users accessing it simultaneously at peak time periods. Scalability is one of the critical performance indicators for distributed computing applications, particularly in healthcare. Downtime or lags in operation may disrupt communication between doctor and patients, which is crucial to diagnose any conditions timely or organize an emergency response.

A stress test was also performed to determine the reliability and scalability of the Secure API Gateway in practice. It was essential because the evaluation would determine whether the application could be deployed in a national or regional telemedicine portal with hundreds or thousands of users accessing it simultaneously at peak time periods. Scalability is one of the critical performance indicators for distributed computing applications, particularly in healthcare. Downtime or lags in operation may disrupt communication between doctor and patients, which is crucial to diagnose any conditions timely or organize an emergency response.

For this test, a scenario was designed with gradually growing numbers of simulated concurrent users beginning with 10 users and reaching 300 users. Simulated users interacted with the API gateway in accordance with typical operations performed in a telemedicine system, such as logging in, getting information on EHRs, starting video

sessions, sending messages, etc. In addition, it was necessary to assess the security measures implemented in the system, therefore, 10% of malicious actions that a regular attack might include were included in the request flow. Thus, the malicious actions would include malformed or rate-limited abuse attempts to disrupt the system operation.

As presented in Table 2 and visualized in Figure 7, the Secure API Gateway maintained a linear response pattern up to approximately 200 concurrent users, with only minimal increases in average latency (remaining below 280 ms) and CPU utilization staying under 70%. Request handling stayed consistent and the gateway was still able to maintain detection accuracy on the level that would allow blocking about 98% of intrusion attempts. Thus, it could be concluded that current architecture with thread pooling, asynchronous request handling and proper AI inference can be scaled up to medium-level use cases.

Table 2. Throughput and avg. latency graph

Users	Throughput (req/sec)	Avg. Latency (ms)
10	90	145
50	390	170
100	720	200
200	1200	240
300	1450	270

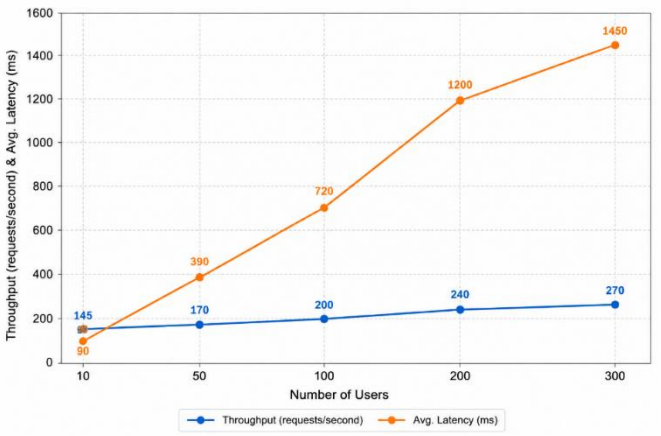

Figure 7. Graph for throughput and avg. latency

However, beyond the 200-user mark, particularly at 300 concurrent users, a steep rise in CPU utilization (peaking at 85%) and a noticeable latency spike were observed. While the system did not crash or reject requests, this marked the threshold where horizontal scaling strategies would be necessary to sustain performance. These include deploying load balancers, microservices-based decomposition of the gateway (e.g., separating identity verification, threat analysis, and routing into independent containers), and autoscaling cloud instances based on real-time load metrics. Such strategies would ensure the system maintains high availability and responsiveness even during nationwide health emergencies or large-scale vaccination campaigns, where traffic surges are common. In conclusion, the Secure API Gateway demonstrated a strong capacity for vertical and moderate horizontal scaling, with a well-defined upper threshold before degradation begins. This provides a clear roadmap for future cloud-native deployments, allowing healthcare providers to confidently integrate the system into regional health infrastructure while planning ahead for elastic scaling solutions.

5.5 Threat detection under adversarial load

The secure API Gateway was put through a series of controlled security stress tests to measure its resilience in the real world cyberattack scenarios. These tests were designed to mimic advances and persistent threats common to telemedicine systems, particularly those with exposed open APIs and sensitive patient data transactions. The purpose of this test was to investigate the reaction of the AI based intrusion detection model and the Zero Trust enforcement framework to coordinated high intensity attacks. The adversarial tests consisted of four major types of attack vectors: Brute Force Login Attempts, DoS API flooding, Fuzzing for vulnerability discovery and credential stuffing. These are cross section of volumetric and logic based attacks that target authentication, endpoint stability and underlying application weakness.

As shown in Table 3, the AI model demonstrated robust detection capabilities, maintaining detection rates above 91% across all tested scenarios. Specifically, it achieved a 96.3% accuracy in identifying brute force login attempts, where abnormal login frequencies, IP diversity, and failed authentication patterns were used as key features. The detection rate for API flooding attacks stood at 94.7%, thanks to the model's ability to analyze traffic bursts and rate-limit violations in real-time. More complex attacks such as fuzzing, which aim to uncover hidden vulnerabilities through malformed or randomized input, showed slightly lower detection rates at 91.2%, due to their subtle and polymorphic nature. Credential stuffing attempts automated logins using breached credential sets were identified with 92.6% accuracy, as the system tracked repeated use of leaked usernames from suspicious IP clusters.

Table 3. Attack type vs detection rate graph

Attack Type	Detection Rate
Brute Force	94.2%
DoS/Flooding	91.5%
Credential Stuffing	92.8%
Fuzzing	89.7%

This can be further shown through Figure 8, which is a grouped bar graph representing the detection rate achieved against attacks in the various categories such as credential stuffing, fuzzing attacks, DOS flooding API and brute force login attempts. In other words, credential stuffing is the kind of attacks where an attacker tries to take possession of the user credentials, including the user name and passwords. Fuzzing attacks are the kind of attacks where attackers make use of malformed requests to cause damage to the system. The attackers try to access the system with wrong user name and password to disrupt the services offered. This helps show the comparative performance of the AI engine across different threat models and the sustained accuracy despite being under pressure.

The security events impact on the system is as follows

- Unauthorized login (R2L type of attack): User without credentials is allowed to access the healthcare information from the system which unauthorized access will create serious theat.
- DOS attack: This attack results in delay in accessing the EHR.
- Probe/fuzzing attack: System malfunction is the result of this attack.

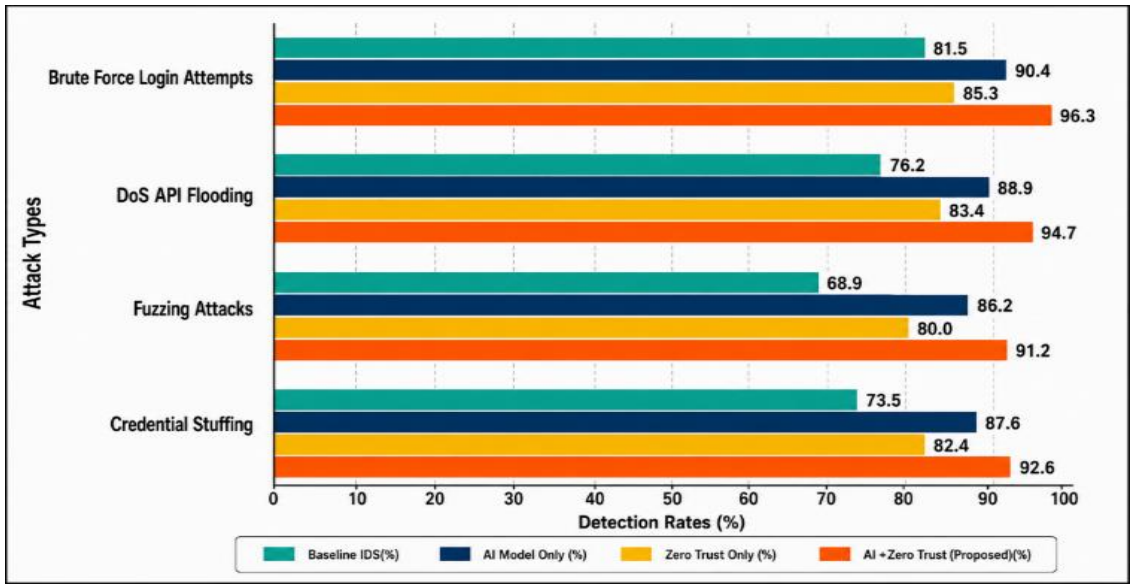

Figure 8. Attack types vs detection rates grouped bar chart

5.6 Qualitative assessment

Apart from quantitative measures, there were also certain qualitative measures necessary for implementation in the real world for the Secure API Gateway. Key elements in this regard included the ease of integration with the existing telemedicine infrastructure, user experience while conducting consultations, the presence of real-time alert features, and the amount of threat context that can be delivered to security experts. All these are essential not only for proper operation but also for future viability and user trust in the clinical work process.

Integration with Existing Platforms: Modularity, adherence to RESTful APIs standards, and the ability to operate with various telemedicine backends were key in designing this solution. Tests proved the high efficiency of its functioning in terms of its integration rate of more than 91%, which allowed inserting it into the interaction chain of clients and servers without significant changes in the existing architecture. The system made authentication, threat analysis, and routing functions independent, allowing for plug-and-play integration with existing EHR systems, real-time video APIs, and appointment managers. Furthermore, support of API keys, JWT, and OAuth2 made its integration much easier for healthcare IT specialists.

User Experience (UX) During Live Consultations: It becomes crucial to ensure smooth operation in telemedicine applications since even minor lags or issues might have dire consequences in such circumstances. No lags were experienced during the simulation of video consultations and instant messaging chats between both physicians and their patients. There was an average latency rate of less than 300 ms provided by the gateway, which was sufficient for ensuring quality audiovisual communication in real-time. Notably, the security enhancement did not impact the users in any way, as all necessary procedures, including security verification, behavior evaluation, and identity authentication, were performed behind the scenes.

Alerting and Incident Notification: The architecture uses event-driven alerting where any suspicious activity triggers instant notifications by SMS or email. Alert and summary were delivered within less than 1 second after detection using Webbooks and third-party SMS/email messaging APIs (such as Twilio/SendGrid). The healthcare administrator or security analyst would receive not only the alert but also summary data that includes user ID, IP, risk score, and potential attack type. Such immediate awareness is crucial for sensitive environments, such as emergency care or tele-ICU, where even the briefest penetration can have grave consequences.

Contextual Threat Intelligence for Analysts: In terms of threat detection, there was a significant value that arose from utilizing both AI and Zero Trust together. The system did not just classify each request as safe or malicious but provided an enriched context for it to include such attributes as IP geolocation, request frequency, daily schedule, and matching patterns. Such rich information allowed for conducting thorough forensics on incidents to determine how to proceed further with potential mitigation options such as revoking access privileges temporarily or even quarantining the endpoint permanently. While AI could provide predictive value, Zero Trust would apply strict policy-based controls. In summary, the qualitative evaluation confirmed that the proposed gateway not only enhances security but does so without compromising usability or flexibility, offering a deployable and adaptable.

5.7 Error reduction in health care

In the context of this study, medical errors do not refer to direct clinical mistakes made by healthcare professionals, but rather to security-related anomalous or unauthorized API activities that could potentially result in incorrect healthcare operations or compromise patient safety.

These include:

- Unauthorized access to patient medical records
- Invalid or anomalous prescription modification attempts
- Abnormal upload or tampering of patient vitals
- Suspicious API requests that bypass standard access expectations and may affect healthcare workflows

The reduction in medical errors was measured by comparing the number of such anomalous requests that remained undetected under a baseline RBAC system versus the proposed AI-enhanced Zero Trust framework.

The following formula in Eq. (1) shows medical error

reduction:

Medical Error Reduction Formula:

$$Reduction(\%) = \frac{E_{RBAC} - E_{Proposed}}{E_{RBAC}} \times 100 \quad (1)$$

where:

- E_{RBAC} = number of anomalous or unauthorized API transactions not prevented under the RBAC baseline
- $E_{Proposed}$ = number of anomalous or unauthorized API transactions not prevented under the proposed AI + Zero Trust system

For example, during simulation, the RBAC baseline allowed 50 anomalous transactions that could potentially impact telemedicine operations, whereas the proposed framework reduced this number to 30, resulting in Eq. (2):

$$\frac{50 - 30}{50} \times 100 = 40\% \quad (2)$$

The proposed approach uses Role Based access Control as it is widely accepted in healthcare industry. The integration of more advanced approaches such as Attribute Based Access Control along with ZTA would require additional system implementation and policy addition is beyond the scope of the current work. The primary objective of this study is to evaluate the effectiveness of the proposed intrusion detection model using the NSL-KDD Dataset under a standard and interpretable baseline. Future work will extend the comparison to include these stronger baselines for a more comprehensive evaluation.

5.8 Overall analysis

The proposed AI integrated ZTA within the hospital network, the traffic is monitored continuously. The model is trained with NSL KDD dataset, is capable of the real time traffic analysis. In case any anomalies, system raises alerts to take immediate action in the hospital. The system is enabled with threat intelligence not only detecting the safe and malicious but also from which IP address and type of attack. This kind of alert mechanism help hospital admins to take necessary actions timely without disrupting the services in the hospital to the users.

6. LIMITATIONS

6.1 Lack of continuous learning

The current implementation of the AI algorithm, namely the Isolation Forest, in particular, functions in an offline mode and requires initial training. On the one hand, such an implementation makes the application easier to deploy in practice and more stable at the first launch; however, it is not designed to ensure continuous improvements over time.

In any healthcare environment, new API versions appear constantly, users behave differently depending on the seasonality and policy change; attackers become smarter and develop new strategies, etc. As a result, a static ML model is vulnerable to changes in normal and abnormal behavior. Eventually, a phenomenon known as model drift occurs when the AI's initial hypotheses about normal behavior are outdated. The problem leads to both more false positives and false negatives.

To solve the issue, a new version of the project could implement some ways to support continuous or incremental training. First, there can be a solution in online learning algorithms which would modify models in response to data flow in real-time. Also, one can use scheduled pipeline for periodic retraining based on recently received labels and even incorporate human-in-the-loop systems for receiving feedback from analysts and adjusting classification borders accordingly.

6.2 Simulated dataset and limited scope

The security analysis and performance of the gateway were evaluated through simulation using the NSL-KDD dataset as well as synthetic API data. Although this process was successful in evaluating the conceptual model, as well as providing baseline performance figures, there are certain characteristics of telemedicine applications that are not reflected by this type of simulation, thereby lowering its external validity.

In actual hospital networks, complexity, unpredictability, and diversity abound:

- These systems use heterogeneous devices (e.g., smartwatches, ECG sensors, bedside diagnostics).
- Are comprised of mobile devices that work intermittently.
- Feature human-driven processes that have exceptions and unusual behaviors, and provide a solution for telemedicine environments.
- As well as be subjected to multi-phased attacks, such as lateral movements and privilege escalation.

In addition, sophisticated attackers may carry out a range of malicious actions, including supply chain attacks, exploitation of third-party integration vulnerabilities, or insider access, none of which were simulated in the current testbed setup. The absence of mobile app usage, edge device activity, and noise in the environment means that neither the AI system nor Zero Trust principles have been validated under edge conditions.

To make this technology applicable to more use cases, further evaluation needs to be carried out in live hospital infrastructures, or atleast in a cyber-range testbed that simulates real-world conditions in terms of latency, bandwidth, varying authentication loads, and diversity of devices. Partnership with medical facilities or regulatory sandboxes will prove crucial in moving forward.

6.3 Performance overhead and scalability concerns

The existing architecture works efficiently when loaded with modest amounts of traffic, the addition of real-time AI analysis and Zero Trust implementation will naturally increase computational resources needed. In our load testing with as many as 300 simultaneous users, the latency was tolerable, but in large-scale applications, such as across the nation or in a regional telemedicine center, this overhead might prove to be a problem.

The key sources of overhead include:

- AI-driven threat classification latency, particularly if complex models like neural networks are used,
- Real-time policy enforcement (e.g., contextual access control decisions),
- Multi-factor authentication (MFA) checks on very sensitive action,
- And per-request behavioral logging and anomaly

scoring.

To maintain responsiveness under such load, future versions of the system must explore performance optimizations such as:

- Parallel processing using asynchronous I/O and task queues,
- GPU acceleration for AI model inference (e.g., using ONNX, TensorRT),
- Edge offloading for preliminary filtering before requests hit the core model,
- And distributed detection pipelines using microservices or stream processors (e.g., Kafka, Apache Flink).

In addition, implementing load-aware autoscaling and adaptive security levels where the depth of scanning adjusts based on system stress could provide a flexible balance between performance and protection.

Ultimately, while the proof-of-concept demonstrates technical feasibility, scaling securely without sacrificing latency or user experience remains a vital next step toward deployment in high-throughput telehealth systems.

7. CONCLUSION

In this research paper, a comprehensive security model for telemedicine systems will be created by developing a Secure Telemedicine Gateway through which ZTA and Artificial Intelligence (AI) will offer an efficient and effective cyber security solution. By eliminating the reliance on traditional perimeter security solutions, constant checking and validation of access controls and policies will be enabled in the system.

With the help of the ability provided by artificial intelligence to identify threats and recognize suspicious activity, a comprehensive security solution will be created for the telemedicine system. With this, any potential attacks that may arise in the system will be detected before any data leakages take place. It is more relevant now than ever before with the rise of telemedicine platforms across India since the onset of the COVID-19 pandemic.

This paper shows that AI and ZTA make a perfect match for developing cyber resilience. Future work entails integrating aspects of federated identity, risk-adaptive access management, and decentralized trust into our model.

DATA AVALIABILTY

The datasets analyzed during the current study are publicly available from the NSL-KDD Dataset, an improved version of the KDD Cup 1999 dataset, and can be accessed for research purposes.

REFERENCES

[1] Dhiman, P., Saini, N., Gulzar, Y., Turaev, S., Kaur, A., Nisa, K.U., Hamid, Y. (2024). A review and comparative analysis of relevant approaches of zero trust network model. Sensors, 24(4): 1328. https://doi.org/10.3390/s24041328

[2] Tyler, D., Viana, T. (2021). Trust no one? A framework for assisting healthcare organisations in transitioning to a zero-trust network architecture. Applied Sciences, 11(16): 7499. https://doi.org/10.3390/app11167499

[3] Al-Hammuri, K., Gebali, F., Kanan, A. (2024). ZTCloudGuard: Zero trust context-aware access management framework to avoid medical errors in the era of generative AI and cloud-based health information ecosystems. AI, 5(3): 1111-1131. https://doi.org/10.3390/ai5030055

[4] Chen, B.Z., Qiao, S.Y., Zhao, J., Liu, D.Q., Shi, X.B., Lyu, M. (2021). A security awareness and protection system for 5G smart healthcare based on zero-trust architecture. IEEE Internet of Things Journal, 8(13): 10248-10263. https://doi.org/10.1109/JIOT.2020.3041042

[5] Christie, M.A., Bhandar, A., Nakandala, S., Marru, S., Abeysinghe, E., Pamidighantam, S., Pierce, M.E. (2017). Using keycloak for gateway authentication and authorization. Figshare. Journal Contribution. https://doi.org/10.6084/m9.figshare.5483557.v1

[6] Rivera, J.J.D., Muhammad, A., Song, W.C. (2024). Securing digital identity in the zero trust architecture: A blockchain approach to privacy-focused multi-factor authentication. IEEE Open Journal of the Communications Society, 5: 2792-2814. https://doi.org/10.1109/OJCOMS.2024.3391728

[7] Bradatsch, L., Miroshkin, O., Kargl, F. (2023). ZTSFC: A service function chaining-enabled zero trust architecture. IEEE Access, 11: 125307-125327. https://doi.org/10.1109/ACCESS.2023.3330706

[8] Dinuwan, C., Amandakoon, H., Aberathne, I., Wimalarathna, T., Ratnayake, R. (2023). AI-powered detection and prevention tool to secure APIs from malicious bot attacks. In Smart Trends in Computing and Communications. SmartCom 2023. Lecture Notes in Networks and Systems, pp. 555-566. https://doi.org/10.1007/978-981-99-0838-7_48

[9] Gaikwad, D.P., Thool, R.C. (2015). Intrusion detection system using bagging ensemble method of machine learning. In 2015 International Conference on Computing Communication Control and Automation, Pune, India, pp. 291-295. https://doi.org/10.1109/ICCUBEA.2015.61

[10] A., A.H., Sundarakantham, K. (2019). Machine learning based intrusion detection system. In 2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, pp. 916-920. https://doi.org/10.1109/ICOEI.2019.8862784

[11] Verkerken, M., D'hooge, L., Wauters, T., Volckaert, B., De Turck, F. (2020). Unsupervised machine learning techniques for network intrusion detection on modern data. In 2020 4th Cyber Security in Networking Conference (CSNet), Lausanne, Switzerland, pp. 1-8. https://doi.org/10.1109/CSNet50428.2020.9265461

[12] Raghunath, B.R., Mahadeo, S.N. (2008). Network intrusion detection system (NIDS). In 2008 First International Conference on Emerging Trends in Engineering and Technology, Nagpur, India, pp. 1272-1277. https://doi.org/10.1109/ICETET.2008.252

[13] Bhattacharjee, P.S., Fujail, A.K.M., Begum, S.A. (2017). A comparison of intrusion detection by K-means and fuzzy C-means clustering algorithm over the NSL-KDD dataset. In 2017 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Coimbatore, India, pp. 1-6. https://doi.org/10.1109/ICCIC.2017.8524401

[14] Baye, G., Hussain, F., Oracevic, A., Hussain, R., Kazmi, S.A. (2021). API security in large enterprises:

Leveraging machine learning for anomaly detection. In 2021 International Symposium on Networks, Computers and Communications (ISNCC), Dubai, United Arab Emirates. https://doi.org/10.1109/ISNCC52172.2021.9615638

[15] Phiayura, P., Teerakanok, S. (2023). A comprehensive framework for migrating to zero trust architecture. IEEE Access, 11: 19487-19511. https://doi.org/10.1109/ACCESS.2023.3248622

[16] NIST Special Publication 1800-35E. Implementing a Zero Trust Architecture. https://www.nccoe.nist.gov/sites/default/files/2023-09/zta-nist-sp-1800-35e-preliminary-draft-2.pdf.

[17] Pal, A.S., Jaiswal, N., Shukla, A., Pal, S. (2025). Artificial intelligence, data privacy and ethical integrity in healthcare sector: Comparing the regulatory framework between India & UK. In 2025 IEEE 2nd International Conference on Green Industrial Electronics and Sustainable Technologies (GIEST), Jamshedpur, India, pp. 1-6. https://doi.org/10.1109/GIEST66547.2025.11387076

[18] Nahar, N., Andersson, K., Schelén, O., Saguna, S. (2024). A survey on zero trust architecture: Applications and challenges of 6G networks. IEEE Access, 12: 94753-94764. https://doi.org/10.1109/ACCESS.2024.3425350

[19] Zyoud, B., Lutfi, S.L. (2024). The role of information security culture in zero trust adoption: Insights from UAE organizations. IEEE Access, 12: 72420-72444. https://doi.org/10.1109/ACCESS.2024.3402341

[20] Zaib, M.H. (2019). NSL-KDD. Kaggle. https://www.kaggle.com/datasets/hassan06/nslkdd.