Int. J. Sci. R. Tech., 2026 3(5)
A Multidisciplinary peer-reviewed Journal
www.ijsrtjournal.com [ISSN: 2394-7063]

Original Article

AI-Based Smart Firewalls: Intelligent Network Security Using Machine Learning and Behavioral Analysis

G. Eesa, P. Jai Sai Chandhan, K. Ch. Revanth Mahesh, Ch. Krishna Sri*

Dept. of Computer Science and Engineering, Jain Deemed-to-be University, Bangalore, India

ABSTRACT

With the rapid growth of digital communication and escalating cyber threats, traditional firewall systems have become fundamentally insufficient due to their exclusive reliance on static rules and signature-based detection methods. These conventional systems are incapable of adapting to novel, previously unseen attack patterns and require constant manual updating. This paper presents the design and comprehensive implementation of an AI-based smart firewall system that fundamentally enhances network security by integrating advanced machine learning algorithms, deep learning architectures, and real-time behavioral analysis. The proposed system is capable of analyzing large volumes of network traffic continuously, identifying complex patterns, and autonomously detecting both known and unknown threats including zero-day attacks, Advanced Persistent Threats (APTs), ransomware, and insider threats. By continuously learning from historical traffic data and live network sessions, the firewall dynamically adapts to evolving cyber threats, ensuring substantially improved detection accuracy and significantly reduced false positive rates. The working mechanism encompasses data collection from heterogeneous network sources, multi-stage preprocessing, AI model training using supervised and unsupervised paradigms, real-time traffic monitoring, automated threat classification, and intelligent decisionmaking with instant response. The system additionally incorporates a biometric face detection module implemented using OpenCV's Haar Cascade classifier, providing an additional layer of physical access control for critical network infrastructure. Experimental evaluation demonstrates classification accuracy of 90–98%, a reduction in false positives exceeding 60% compared to rule-based systems, and real-time threat response latency below one second. This approach provides a proactive, intelligent, and adaptive security solution suitable for modern network environments including cloud platforms, enterprise networks, Internet of Things (IoT) ecosystems, and large-scale web applications.

Keywords: artificial intelligence, smart firewall, machine learning, behavioral analysis, zero-day attack detection, anomaly detection, deep learning, face detection, OpenCV, network security, intrusion detection, reinforcement learning, LSTM, random forest, support vector machine.

INTRODUCTION

Artificial Intelligence (AI)-based smart firewalls represent the next generation of network security infrastructure, engineered to protect digital environments from a continuously expanding and increasingly sophisticated landscape of cyber threats. The growing complexity of modern cyberattacks demands security systems capable of learning, adapting, and responding autonomously—capabilities that fundamentally exceed the design parameters of conventional firewall technologies.

Traditional firewalls operate on the basis of administrator-defined static rule sets and pre-compiled signature databases. This architecture enables efficient blocking of recognized threats but is inherently reactive: a threat must be identified, analyzed, and documented before protection can be deployed. In an era where new malware variants, zero-day exploits, and advanced persistent threats emerge on a daily basis, this reactive paradigm leaves critical windows of vulnerability that adversaries actively exploit.

AI-based smart firewalls overcome these fundamental limitations by incorporating machine learning models that learn the statistical characteristics of both normal network behavior and known malicious activity. Once trained, these models can evaluate new network traffic in real time, comparing observed behavior against

Relevant conflicts of interest/financial disclosures: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

learned baselines and flagging anomalous patterns that deviate significantly from expected norms. This capability enables detection of threats that have never previously been encountered—including zero-day exploits for which no signature exists.

The security challenges addressed by this research are substantial and multi-dimensional. Zero-day attacks exploit previously unknown vulnerabilities and cannot be detected by signature-based systems. Advanced Persistent Threats (APTs) involve prolonged, stealthy intrusion campaigns that gradually exfiltrate sensitive data over extended periods. Ransomware encrypts organizational data and demands payment for decryption keys. Distributed Denial of Service (DDoS) attacks overwhelm network infrastructure by flooding it with traffic from thousands of compromised hosts. Insider threats originate from authorized users whose credentials have been compromised or who are acting maliciously. All of these threat categories require adaptive, behavioral detection capabilities that static rule-based systems cannot provide.

This paper makes the following primary contributions to the field of intelligent network security. First, it presents a comprehensive multi-model AI architecture that combines supervised learning for known threat classification, unsupervised anomaly detection for zero-day threat identification, and reinforcement learning for adaptive policy optimization. Second, it introduces an integrated biometric access control module using OpenCV's Haar Cascade classifier for face detection, providing an additional physical security layer. Third, it presents a continuous learning feedback mechanism that enables the system to improve its detection capabilities over time without manual intervention. Fourth, it provides experimental evaluation demonstrating substantial performance improvements over conventional firewall approaches.

The remainder of this paper is organized as follows. Section II surveys related literature in AI-based network security. Section III analyzes existing systems and articulates the limitations motivating the proposed work. Section IV describes the proposed system architecture in detail. Section V covers the implementation of all system modules. Section VI presents a detailed module-level description. Section VII reports experimental results and performance evaluation. Section VIII describes the testing and validation strategy. Section IX discusses limitations and known issues. Section X concludes the paper, and Section XI outlines future enhancement directions.

LITERATURE REVIEW

The literature on AI-based network security spans multiple disciplines including machine learning, deep learning, network analysis, behavioral analytics, and biometric authentication. This section reviews the most relevant and impactful contributions that form the foundation of the proposed system.

A. Machine Learning-Based Firewall Decision Systems

Research conducted in 2021 introduced a supervised machine learning framework for automating firewall packet classification decisions [1]. Prior to this work, firewall administrators were required to manually define and maintain extensive rule sets, a process that was both time-consuming and error-prone. The proposed approach trained classification models on labeled network traffic datasets, categorizing packets into allow, deny, or drop decision classes.

The study evaluated three classical machine learning algorithms: Decision Trees, Random Forest, and Support Vector Machines (SVM). Experimental results demonstrated that the Random Forest classifier achieved the highest accuracy, significantly outperforming traditional rule-based systems on benchmark datasets. The primary limitation identified was a dependency on the quality and representativeness of labeled training data, as models trained on limited datasets showed reduced generalization to traffic patterns from different network environments.

B. Firewall Log Analysis Using Deep Learning

A comprehensive 2022 study addressed the challenge of analyzing large-scale firewall log data using both classical machine learning and deep learning architectures [2]. The research motivated by the observation that firewall logs contain rich temporal and contextual information that shallow learning models fail to fully exploit. The dataset incorporated multiple classes of network activity spanning normal

operations, various attack categories, and anomalous behavior patterns.

Deep learning architectures including Artificial Neural Networks (ANN) and Long Short-Term Memory (LSTM) networks were applied to both raw feature vectors and temporal feature sequences extracted from log data. LSTM networks demonstrated particular strength in capturing sequential dependencies in traffic flows, enabling detection of multi-stage attack patterns that unfold across time. The approach handled large-scale data efficiently and successfully identified complex attack behaviors. Acknowledged limitations included high computational resource requirements and the need for substantial labeled training datasets.

C. Next-Generation AI Firewalls Comparative Study

A landmark 2023 comparative study systematically analyzed diverse AI-based firewall architectures and evaluated their relative performance using a comprehensive set of metrics including accuracy, precision, recall, F1-score, and false positive rate [3]. The study examined systems ranging from single-model classifiers to ensemble approaches and hybrid architectures combining multiple AI paradigms.

The comparative analysis conclusively demonstrated that AI-based firewalls outperform traditional systems across all evaluated metrics, particularly for detection of unknown and polymorphic threats. The study also identified that ensemble methods combining multiple base classifiers consistently achieved higher accuracy than single-model approaches. A key recommendation was that future systems should prioritize adaptive architectures capable of evolving alongside emerging attack methodologies.

D. High-Performance Computing and Cloud Security

Research published in 2024 investigated AI-based approaches to firewall rule refinement in high-performance computing (HPC) network environments [4]. HPC networks present unique security challenges due to their scale, performance requirements, and the sensitivity of the data they process. The proposed system employed machine learning to automatically identify and eliminate redundant or conflicting firewall rules, reducing rule base complexity while maintaining or improving security coverage.

A separate 2024 study specifically addressed machine learning-based adaptive firewalls for real-time cloud security [5]. Cloud environments introduce dynamic topology changes, elastic resource allocation, and multi-tenancy that make static rule-based security particularly inadequate. The proposed adaptive firewall updated its models continuously as new traffic data arrived, demonstrating that AI-based systems can maintain high detection accuracy even as the protected network environment evolves rapidly.

E. Reinforcement Learning and LLM-Enhanced Security

Advanced research from 2025 explored reinforcement learning (RL) based firewall architectures augmented by Large Language Models (LLMs) [7]. In this paradigm, the firewall is modeled as an RL agent that learns optimal allow/block policies by receiving reward signals based on the accuracy of its decisions. LLMs are used to interpret complex threat intelligence reports and translate them into actionable policy updates, enabling the firewall to incorporate qualitative threat knowledge alongside quantitative traffic analysis.

Complementary work on Dynamic AI-Augmented Firewalls for Real-Time Threat Mitigation demonstrated that combining reactive and proactive AI components produces superior results compared to either approach alone [6]. The reactive component responds immediately to detected threats while the proactive component continuously analyzes traffic trends to anticipate emerging attack patterns before they manifest.

F. Face Detection in Security Systems

Computer vision-based face detection has been widely studied as a component of physical security systems. The Viola-Jones algorithm implemented in OpenCV's Haar Cascade classifier [9] remains a foundational technique due to its computational efficiency and real-time performance. Modern deep learning approaches including MTCNN and RetinaFace achieve higher accuracy but at greater computational cost. For integration with network

security infrastructure where face detection serves as an access control layer, the Haar Cascade approach provides an appropriate balance of accuracy and efficiency.

G. Machine Learning Models for Intrusion Detection

Beyond firewall-specific research, the broader field of network intrusion detection has produced relevant insights. The NSL-KDD dataset [11], derived from the original DARPA KDD Cup 1999 dataset, has become a standard benchmark for evaluating intrusion detection systems. The CICIDS2017 dataset [12] provides a more modern benchmark incorporating contemporary attack types including web attacks, infiltration, and botnet traffic. The UNSW-NB15 dataset [13] offers a comprehensive collection spanning nine attack categories across 49 network features.

Research applying ensemble methods to these datasets has consistently demonstrated performance superior to singlemodel approaches. Random Forest [16] has emerged as a particularly effective base classifier due to its robustness to overfitting and ability to handle high-dimensional feature spaces. LSTM networks [15] provide complementary temporal modeling capabilities, capturing sequential attack patterns that unfold across multiple packets or sessions. Combining these approaches through ensemble voting, as proposed in this paper, leverages the complementary strengths of each model type.

H. Gaps in Existing Work

Despite substantial progress across these research areas, several critical gaps remain unaddressed. First, the majority of existing systems treat network-level security and physical access control as entirely separate concerns, missing opportunities for integrated multi-layer protection. Second, few systems provide a unified architecture that simultaneously addresses known threat classification, zero-day detection through anomaly analysis, and continuous adaptive learning within a single deployable system. Third, the combination of biometric authentication with behavioral network analysis represents a largely unexplored research direction that this paper addresses.

Fourth, existing literature rarely addresses the operational aspects of AI firewall deployment, including model retraining scheduling, performance monitoring, and analyst workflow integration. A deployed security system that achieves high benchmark accuracy but imposes unsustainable operational overhead provides limited practical value. The proposed system design explicitly addresses these operational concerns through automated continuous learning, analyst-focused alert management, and performance monitoring dashboards. Fifth, the reproducibility of reported results has been limited by inconsistent experimental methodologies and private datasets. This work employs publicly available benchmark datasets and reports comprehensive evaluation metrics to facilitate reproducibility.

EXISTING SYSTEM AND PROPOSED WORK

A. Existing System

Current network security deployments predominantly rely on traditional firewall architectures based on packet filtering, stateful inspection, and signature-based intrusion detection. These systems operate by maintaining rule tables that specify which packets should be allowed, denied, or logged based on criteria including source and destination IP addresses, port numbers, protocol types, and known malicious payload signatures.

While effective against known and catalogued threats, traditional firewalls exhibit fundamental architectural limitations that significantly reduce their effectiveness in modern threat environments. Signature databases require continuous manual updating to remain current, creating vulnerability windows between the emergence of new threats and the deployment of corresponding signatures. Static rule sets cannot adapt to behavioral changes in the protected network, resulting in both excessive false positives that block legitimate traffic and false negatives that allow malicious traffic to pass undetected.

Next-Generation Firewalls (NGFWs) represent an incremental improvement, incorporating application-layer inspection, SSL decryption, and limited anomaly detection capabilities. However, even NGFWs fundamentally depend on predefined

signatures and policies, lacking the ability to learn from network behavior and detect truly novel attack patterns without continuous administrator intervention.

B. Work Done in Existing Systems

Prior research has addressed specific aspects of intelligent firewall development with considerable success. Machine learning classifiers have been applied to packet-level and flow-level network data, demonstrating that automated classification can match or exceed human-configured rulebased approaches for known traffic categories. Deep learning models, particularly LSTM and convolutional neural networks, have shown strong performance on network intrusion detection benchmark datasets including NSL-KDD, CICIDS2017, and UNSW-NB15.

Anomaly detection research has demonstrated that statistical models of normal network behavior can be established and used to flag deviations that may indicate attacks. Techniques including isolation forests, autoencoders, and one-class SVM have been successfully applied to network traffic anomaly detection. Reinforcement learning has been applied to firewall policy optimization, with agents learning to balance security and performance objectives through interaction with network simulators.

C. Issues in the Existing System

Existing systems exhibit several critical limitations that motivate the proposed work. Poor generalization represents the most significant challenge: models trained on specific benchmark datasets often fail when deployed in production environments with different traffic characteristics. High computational requirements limit the deployment of sophisticated deep learning models in resource-constrained environments. Most existing systems analyze only a single modality of security data, missing the complementary information available from combining network-level and physical access control data.

The absence of continuous learning mechanisms means that deployed models gradually become outdated as network environments and attack methodologies evolve. Manual retraining is time-consuming and expensive, creating security gaps during update cycles. Additionally, most systems provide limited interpretability, producing classification results without explanations of the specific features that contributed to each decision, which complicates security analyst review and audit processes.

D. Proposed Solution

The proposed AI-based smart firewall addresses these limitations through a multi-paradigm architecture that combines supervised learning for known threat detection, unsupervised anomaly detection for zero-day threat identification, reinforcement learning for adaptive policy optimization, and computer vision-based face detection for physical access control. The system implements continuous online learning that automatically incorporates newly identified threats into updated models without requiring manual retraining cycles. A modular design ensures that individual components can be upgraded independently as improved algorithms become available.

PROPOSED SYSTEM

A. Introduction to the Proposed System

The proposed AI-based smart firewall is a comprehensive, multi-layered network security platform that integrates artificial intelligence, machine learning, deep learning, and computer vision technologies into a unified, deployable system. The architecture is specifically designed to overcome the fundamental limitations of conventional rulebased firewalls by providing automated, adaptive, and intelligent threat detection and response capabilities.

The system is built around a central AI inference engine that continuously evaluates network traffic using ensemble models trained on diverse datasets representing normal behavior and multiple attack categories. Surrounding this core are specialized modules for data collection, preprocessing, behavioral baseline establishment, threat classification, automated response, biometric access control, and continuous learning. All modules communicate through a standardized internal API, enabling independent development and upgrade of each component.

B. System Architecture Overview

The complete system architecture consists of eight primary components organized into three functional layers. The Data Layer encompasses the Data Collection Module and the Preprocessing Module, which together acquire, normalize, and prepare network traffic data for analysis. The Intelligence Layer contains the AI Model Training subsystem, the

Behavioral Baseline Engine, the Threat Detection and Classification Module, and the Face Detection Module. The Response Layer includes the Automated Response Module and the Continuous Learning Feedback System.

Network traffic enters the system through the Data Layer, where it is captured at the network interface level using packet capture libraries, normalized into standardized feature vectors, and forwarded to the Intelligence Layer for analysis. The AI inference engine evaluates each traffic flow and produces a classification label with an associated confidence score. If the confidence score exceeds the malicious classification threshold, the Automated Response Module is triggered to execute the appropriate countermeasure.

C. AI Learning Paradigms

The system employs three complementary machine learning paradigms that together provide comprehensive coverage of the threat detection problem. Supervised learning operates on labeled datasets containing examples of both normal network traffic and specific attack categories. Classification models trained using this approach learn discriminative features that distinguish malicious from benign traffic and can classify new traffic with high accuracy when it resembles previously seen patterns.

Unsupervised anomaly detection addresses the zero-day detection problem by learning a statistical model of normal network behavior without requiring labeled attack examples. The system continuously updates this model as new normal traffic is observed, maintaining an accurate representation of expected behavior. Traffic that deviates significantly from this learned normal profile is flagged as anomalous, regardless of whether it matches any known attack signature. This capability is critical for detecting novel attack variants that have never previously been observed.

Reinforcement learning provides the policy optimization layer, training an agent to make optimal allow/block decisions in the context of the current network environment. The RL agent receives reward signals based on the accuracy of its decisions—positive rewards for correct classifications and negative rewards for both false positives and false negatives— and gradually learns policies that maximize overall security while minimizing legitimate traffic disruption.

D. Behavioral Analysis Engine

The Behavioral Analysis Engine establishes and maintains baseline profiles of normal network behavior at multiple granularities: per-user, per-device, per-application, and per-subnet. These profiles capture statistical characteristics of normal traffic patterns including typical data transfer volumes, common communication partners, usual active hours, characteristic protocol distributions, and expected session durations. Deviations from established baselines trigger alerts that are forwarded to the classification engine for more detailed analysis.

The baseline engine employs exponentially weighted moving averages to maintain profiles that adapt gradually to legitimate changes in network behavior while remaining sensitive to sudden anomalous deviations. Separate thresholds are maintained for different types of behavioral metrics, with more sensitive thresholds for high-risk behaviors such as privileged account activity and access to sensitive data stores.

E. Face Detection and Biometric Access Control

The biometric access control layer employs OpenCV's Haar Cascade classifier to perform real-time face detection at physical access points to network infrastructure. This module interfaces with security cameras positioned at server room entrances, network operations center access points, and critical hardware locations. When personnel approach these areas, the face detection system identifies and logs the detected faces, cross-referencing them against an authorized personnel database.

The Haar Cascade approach is selected for this application because of its excellent balance of detection accuracy and computational efficiency. The classifier applies the Viola-Jones algorithm using a cascade of increasingly complex feature evaluators, enabling rapid rejection of nonface image regions and efficient focus on candidate face locations. This approach achieves real-time performance on standard hardware, a critical requirement for an access control system that must not impede authorized personnel.

IMPLEMENTATION

A. Technology Stack

The system implementation employs Python 3.10 as the primary programming language, leveraging its extensive ecosystem of machine learning, computer vision, and network analysis libraries. The machine learning pipeline uses scikitlearn for classical algorithms including Random Forest, Support Vector Machines, and Isolation Forest. Deep learning components are implemented using TensorFlow 2.x and Keras, providing access to LSTM networks and convolutional architectures. The reinforcement learning module uses Stable Baselines3 built on PyTorch.

Network traffic capture and analysis employ Scapy for packet-level operations and PyShark for higher-level flow analysis. Feature extraction from raw packets is performed using a custom pipeline that computes 41 statistical features per network flow, compatible with the NSL-KDD dataset format to facilitate benchmarking against published results. The face detection module uses OpenCV 4.8.x with the pretrained haarcascade_frontalface_default.xml model. The management dashboard is implemented as a Flask web application with a React.js frontend, providing real-time visualization of network traffic, threat alerts, and system performance metrics.

B. Data Collection and Preprocessing

The data collection module captures network traffic at the interface level using libpcap-based packet capture. Packets are processed in real time using a sliding window approach, with flow-level features computed for each bidirectional network session. The feature extraction pipeline computes 41 features per flow including duration, protocol type, service type, flag, source and destination bytes, land, wrong fragment, urgent, hot, failed logins, logged in status, number of compromised events, root shell access, su attempted, number of root accesses, number of file creations, number of shells, number of access files, number of outbound commands, is host login, is guest login, count, srv count, serror rate, srv serror rate, rerror rate, srv rerror rate, same srv rate, diff srv rate, srv diff host rate, destination host count, destination host srv count, destination host same srv rate, destination host diff srv rate, destination host same src port rate, destination host srv diff host rate, destination host serror rate, destination host srv serror rate, destination host rerror rate, and destination host srv rerror rate.

Preprocessing applies z-score normalization to continuous features and one-hot encoding to categorical features including protocol type, service, and flag. Class imbalance between normal and malicious traffic samples is addressed using SMOTE (Synthetic Minority Over-sampling Technique) for the training dataset. Features with near-zero variance across the training set are eliminated to reduce dimensionality and improve model training efficiency.

C. Face Detection Implementation

The face detection module initializes the Haar Cascade classifier by loading OpenCV's pre-compiled frontal face model: face_cascade = cv2.CascadeClassifier(cv2.data.haarcascades + 'haarcascade_frontalface_default.xml'). Input images are acquired from USB or IP security cameras at 30 frames per second. Each frame is first validated for successful capture, then converted from the BGR color space used by OpenCV to grayscale using cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY). Grayscale conversion reduces the input to a single channel, removing color information that is not needed for face detection and improving processing speed.

Face detection is performed using the detectMultiScale() function, which applies the classifier at multiple image scales to detect faces of varying sizes. The function is configured with a scale factor of 1.1, causing the detection window to increase in size by 10% at each scale step. The minNeighbors parameter is set to 5, requiring that five neighboring detection windows agree on the presence of a face

before a detection is confirmed, effectively eliminating false positives. A minimum detection size of 30x30 pixels excludes very small regions that are unlikely to represent actual faces.

For each face detected, the module draws a rectangular bounding box using cv2.rectangle() with purple color (255, 0, 255) and 2-pixel line thickness. Detected face regions are cropped, resized to a standard 128x128 pixel format, and passed to the face recognition module for identity verification against the authorized personnel database. Access control decisions are logged with timestamp, camera location, detected face image, and recognition result.

D. Machine Learning Model Training

The supervised classification model uses a Random Forest ensemble of 200 decision trees, each trained on a random subset of training samples and features. Random Forest is selected as the primary classifier because of its robustness to overfitting, ability to handle high-dimensional feature spaces, and native support for feature importance estimation. The model is trained on a combined dataset derived from NSL-KDD, CICIDS2017, and UNSW-NB15, incorporating diverse traffic types and attack categories.

The LSTM deep learning component processes sequential traffic data to capture temporal attack patterns. The network architecture consists of two LSTM layers with 128 and 64 units respectively, followed by two fully connected layers with ReLU activation and a softmax output layer for multiclass threat classification. Dropout layers with a rate of 0.3 are applied after each LSTM layer to prevent overfitting. The model is trained using the Adam optimizer with an initial learning rate of 0.001 and batch size of 256 for 50 epochs.

The anomaly detection component employs an Isolation Forest with 200 trees, trained exclusively on normal traffic samples. The contamination parameter is set to 0.05, indicating an expected anomaly rate of 5% in the test data. This model flags traffic samples that are isolated by the random forest in fewer splits than typical normal samples, indicating they occupy low-density regions of the feature space characteristic of anomalous behavior.

E. Ensemble Integration and Voting

The three component models—Random Forest classifier, LSTM classifier, and Isolation Forest anomaly detector—are integrated through a weighted voting mechanism. For each traffic sample, each model produces a threat classification label and a confidence score. The Random Forest and LSTM models each contribute a 40% weight to the final decision, while the Isolation Forest anomaly detector contributes 20%. This weighting reflects the higher classification accuracy of the supervised models while preserving the zero-day detection contribution of the anomaly detector.

The final classification decision is produced by computing the weighted sum of confidence scores for each class across all three models. If the weighted malicious confidence score exceeds 0.5, the traffic is classified as malicious; if it falls below 0.3, the traffic is classified as legitimate; values between 0.3 and 0.5 trigger a suspicious classification that routes the traffic to a honeypot environment for further analysis while allowing the original session to continue under enhanced monitoring.

A calibration post-processing step is applied to the raw model output scores using Platt scaling [10], transforming uncalibrated probability estimates into well-calibrated confidence values. Well-calibrated confidence scores are essential for the tiered response system, which depends on confidence thresholds to select appropriate response levels. Calibration is performed using a held-out validation dataset and updated during each retraining cycle.

F. Reinforcement Learning Policy Optimization

The reinforcement learning component models the firewall's allow/block decision as a Markov Decision Process (MDP). The state space represents the current network conditions including recent traffic volume, active connection counts, detected threat frequency, and time-of-day features. The action space includes four discrete actions: allow, block, rate-limit, and redirect-to-honeypot. The reward function assigns positive rewards for correctly classifying threats and correctly allowing legitimate traffic, and negative rewards for false positives that disrupt legitimate users and false negatives that allow attacks to proceed.

The RL agent is trained using Proximal Policy

Optimization (PPO), a policy gradient algorithm that provides stable training through constrained policy updates. Training is conducted in a network simulation environment using recorded traffic traces, enabling extensive exploration of the action space without impacting the production network. The trained policy is then deployed as an additional decision layer that can override or augment the ensemble classifier's recommendations when network conditions suggest that threshold adjustments are warranted.

G. Automated Response System

The automated response module implements a tiered response policy based on the threat severity level associated with each detected threat type. Level 1 responses, applied to suspicious traffic classifications, include enhanced logging, connection rate limiting, and notification of security analysts. Level 2 responses, applied to confirmed malicious classifications with moderate confidence, include immediate connection blocking, source IP address addition to a temporary blocklist valid for 24 hours, and real-time alert generation. Level 3 responses, applied to high-confidence malicious classifications involving critical attack types such as DDoS, privilege escalation, or data exfiltration, include complete source subnet blocking, isolation of affected endpoint devices, emergency notification to senior security personnel, and automatic incident report generation.

Response actions are implemented through integration with network infrastructure APIs. For managed switches and routers supporting NETCONF or REST APIs, the system directly programs access control list entries to block malicious traffic at the network edge. For environments without programmable network infrastructure, response actions are enforced through iptables rules applied at the firewall host. All response actions are logged with full context information and can be reviewed, modified, or reverted through the management dashboard.

MODULE DESCRIPTION

A. Data Input Module

The Data Input Module serves as the primary interface between the live network environment and the security system's analysis pipeline. The module operates at the network interface level, capturing all inbound, outbound, and internal traffic traversing monitored network segments. It supports multiple capture modes including promiscuous mode for comprehensive traffic visibility, port mirroring integration for deployment on managed network switches, and inline mode for active traffic inspection and blocking.

The module implements flow-based traffic aggregation, combining individual packets belonging to the same network session into unified flow records. Flow records are maintained in a hash table indexed by the five-tuple of source IP, destination IP, source port, destination port, and protocol, with idle timeout and active timeout parameters controlling flow expiry. Completed flow records are forwarded to the preprocessing pipeline for feature extraction.

B. Preprocessing Module

The Preprocessing Module transforms raw flow records into normalized feature vectors suitable for input to the AI classification models. The module applies the complete 41feature extraction pipeline described in Section V-B, computing statistical aggregates of packet-level measurements for each flow. Feature normalization using zscore standardization is applied using statistics computed from the training dataset, ensuring that the normalization parameters remain consistent between training and inference.

The preprocessing module also implements feature caching for flows that arrive in multiple batches, maintaining partial feature vectors for active flows and completing them when flows expire. This approach enables the system to make preliminary threat assessments based on partial flow data, supporting early detection of attacks that exhibit malicious characteristics in the initial packets of a session.

C. Traffic Monitoring Module

The Traffic Monitoring Module provides real-time visibility into network traffic patterns and system performance through a web-based dashboard

interface. The module aggregates traffic statistics at configurable time intervals, producing time-series data that is visualized using interactive charts showing traffic volume by protocol, top talkers by bandwidth consumption, geographic distribution of external connections, and threat detection event frequency.

Alert management functionality allows security analysts to review, acknowledge, escalate, and resolve detected threat events. Each alert includes comprehensive contextual information including the traffic flow details, the specific model features that contributed to the classification decision, the confidence score from each component model, and the automated response actions taken. Analysts can override automated response decisions and provide feedback that is incorporated into the continuous learning system.

D. Threat Classification Module

The Threat Classification Module implements the ensemble inference pipeline described in Section V-E, applying the trained Random Forest, LSTM, and Isolation Forest models to each processed flow record. The module is optimized for low-latency inference, targeting a classification latency below 50 milliseconds per flow to support real-time traffic analysis at line rate.

Multi-class classification supports 23 distinct threat categories including Normal, DoS variants (DoS slowhttptest, DoS Hulk, DoS GoldenEye, DoS Slowloris), DDoS variants

(DDoS HOIC, DDoS LOIC-HTTP, DDoS LOIC-UDP), Port Scan variants, Brute Force (FTP-Patator, SSH-Patator), Web Attacks (XSS, SQL Injection, Brute Force), Infiltration, Botnet, and Heartbleed. Fine-grained multi-class classification enables more targeted and appropriate automated responses compared to binary malicious/benign classification.

E. Continuous Learning Module

The Continuous Learning Module implements an online learning pipeline that periodically retrains the classification models using newly accumulated traffic data. The retraining pipeline runs on a scheduled basis, typically every 24 hours during low-traffic periods, incorporating confirmed threat classifications from analyst review and new normal traffic samples from the current operational environment.

Active learning techniques are employed to prioritize data samples for analyst review, focusing on samples near classification decision boundaries where analyst feedback provides the greatest improvement to model accuracy. Samples with high uncertainty—those where the ensemble models disagree substantially—are flagged as high priority for review. A model performance monitoring system continuously tracks classification accuracy, false positive rate, and false negative rate on a held-out validation set, triggering alerts if performance metrics degrade below defined thresholds.

F. Alert Management and Reporting Module

The Alert Management Module provides a structured workflow for security analyst review of detected threats. Alerts are prioritized using a multi-factor scoring system that considers threat severity, confidence score, asset criticality of the targeted network resource, and time since the last similar alert. High-priority alerts are surfaced immediately through push notifications, while lower-priority alerts are queued for periodic review.

Each alert record includes comprehensive contextual information: the complete five-tuple flow identifier, all 41 extracted flow features, individual model confidence scores from each ensemble component, the specific features that contributed most to the classification decision (computed using SHAP values for interpretability), the automated response actions taken, and links to relevant threat intelligence reports. This context enables analysts to make informed decisions about whether to accept, override, or escalate the system's response.

The reporting module generates scheduled and on-demand security reports including executive summaries of detected threat volumes and categories, trend analysis comparing current threat patterns to historical baselines, false positive and false negative rate tracking over time, and system performance metrics including classification latency and throughput. Reports are generated in PDF and HTML formats and can be automatically distributed to stakeholders via email.

G. Biometric Integration Module

The Biometric Integration Module manages the interface between the OpenCV face detection system and the broader security platform. When the face detection module identifies an individual at a monitored access point, the Biometric Integration Module performs identity lookup in the authorized personnel database, cross-references the detection timestamp against access schedules, evaluates whether the detected individual's access level authorizes entry to the specific location, and logs the access event with face image, recognition confidence, and access decision.

Integration with the network security components enables correlated analysis of physical and logical access events. Unusual combinations of physical and logical access, such as a user's network credentials being used while that user's physical presence cannot be confirmed at the corresponding workstation, trigger elevated security alerts that may indicate credential theft or account compromise. This cross-modal correlation represents a significant capability advantage over conventional single-modality security systems.

VII. RESULTS AND EVALUATION

A. Experimental Setup

The system was evaluated in a controlled laboratory environment using a dedicated network testbed comprising a monitored network segment, a traffic generation system, and the AI firewall deployment. Network traffic was generated using a combination of real captured traffic samples from publicly available datasets and synthetic attack traffic generated using established penetration testing tools including Metasploit, Hydra, and LOIC. The evaluation dataset comprised 2.8 million flow records across 23 traffic categories.

The AI classification models were trained on 70% of the available labeled data and evaluated on the remaining 30% held-out test set. All performance metrics reported are computed on the held-out test set to ensure unbiased evaluation. Comparative baseline results were obtained by configuring an equivalent network segment protected by a conventional Next-Generation Firewall with current signature databases and default behavioral analysis settings.

B. Classification Performance

The proposed AI ensemble achieved an overall classification accuracy of 97.3% on the held-out test set, with precision of 96.8%, recall of 97.1%, and F1-score of 96.9%. These results represent a substantial improvement over the baseline NGFW, which achieved 78.4% accuracy for known attack categories and 43.2% accuracy for zero-day attack simulation. Table I presents detailed performance metrics across individual attack categories.

Attack Category	Precision	Recall	F1-Score
Normal	98.2%	97.9%	98.0%
DoS Hulk	97.1%	96.8%	96.9%
DDoS HOIC	96.4%	97.2%	96.8%
Port Scan	98.5%	98.1%	98.3%
Brute Force	95.8%	96.3%	96.0%
Web Attack-XSS	94.9%	95.4%	95.1%
Botnet	96.7%	97.0%	96.8%
Infiltration	93.8%	94.1%	93.9%

TABLE I. Classification Performance By Attack Category

C. Comparison with Traditional Systems

The comparison with the conventional NGFW baseline across key performance dimensions reveals substantial advantages of the AI-based approach. The proposed system achieved a false positive rate of 2.3% compared to 14.7% for the baseline, representing an 84.4% reduction in legitimate traffic incorrectly classified as malicious. This improvement directly translates to reduced user disruption and lower analyst workload for false alert investigation.

Metric	AI Firewall	Trad. NGFW
Overall Accuracy	97.3%	78.4%
Zero-Day Accuracy	90.3%	43.2%
False Positive Rate	2.3%	14.7%
Avg. Latency (ms)	23 ms	8 ms
Throughput (1 Gbps)	100%	100%
Precision	96.8%	76.2%
Recall	97.1%	79.3%

TABLE II. AI Firewall vs. Traditional NGFW

D. Response Latency and Throughput

Response latency measurements demonstrated that the AI classification pipeline adds an average latency of 23 milliseconds per flow, well within the target of 50 milliseconds. At peak traffic loads of 1 Gbps, the system maintained classification throughput sufficient to evaluate 100% of network flows without packet dropping. The face detection module achieved real-time performance at 28 frames per second on a standard GPU-equipped server, meeting the real-time requirement for access control applications.

E. Continuous Learning Improvement

A 30-day longitudinal evaluation tracked system performance as the continuous learning module incorporated newly labeled samples. Overall classification accuracy improved from 95.8% to 97.3% over the evaluation period, a statistically significant improvement of 1.5 percentage points. The false positive rate decreased from 3.7% to 2.3% over the same period. These improvements demonstrate that the continuous learning mechanism provides meaningful ongoing performance gains in operational deployment scenarios.

F. Feature Importance Analysis

Feature importance analysis using SHAP (SHapley Additive exPlanations) values revealed the most discriminative features for threat classification. The top five features by importance were: destination host same service rate, service error rate, source bytes transferred, connection count to the same host, and service rerror rate. These features capture both volume-based anomalies characteristic of DoS and DDoS attacks, and behavioral anomalies characteristic of stealthy reconnaissance and infiltration attacks.

The Isolation Forest anomaly detector assigned highest anomaly scores to flows with unusual combinations of small packet sizes, high connection rates, and unusual port combinations—characteristics of port scanning and network reconnaissance activities. This pattern demonstrates that the unsupervised component successfully identifies structural anomalies in traffic that supervised models may miss when attack patterns differ from training examples.

G. Biometric Access Control Evaluation

The face detection module was evaluated on a separate dataset of 2,400 camera frames captured

under operational indoor lighting conditions across 40 authorized personnel. Face detection recall (proportion of frames containing faces where a face was detected) reached 96.3%, with a false positive rate (proportion of frames without faces where a face was incorrectly detected) of 1.1%. Identity verification accuracy against the authorized personnel database reached 94.7% at a 0.5 confidence threshold, rising to 98.2% when averaged across multiple consecutive frames from the same individual.

Cross-modal correlation analysis identified 3 anomalous access events during the 30-day evaluation period where network credential usage could not be correlated with physical access records. Security analyst review confirmed that two of these events represented legitimate remote access sessions from authorized personnel working from home, while one event triggered further investigation that identified a compromised credential being used for unauthorized network access. This result demonstrates the value of cross-modal security analysis.

VIII. TESTING AND VALIDATION STRATEGY

A. Unit Testing

Individual system components are tested in isolation using the pytest framework with mock network interfaces and synthetic flow data. Unit tests verify correct feature extraction from known packet sequences, accurate normalization of flow records, correct model inference on labeled test samples, and proper execution of automated response actions. Test coverage targets 90% of all code paths across all modules.

The face detection module is unit tested using a standardized test image dataset containing 500 images spanning various illumination conditions, face orientations, partial occlusions, and demographic variations. Detection rate, false positive rate, and processing latency are measured for each test condition to characterize the module's performance envelope.

B. Integration Testing

Integration tests verify the correct interaction between system components by injecting known traffic samples at the data capture interface and verifying that the expected response actions are triggered within specified latency bounds. Test scenarios include end-to-end DDoS attack detection and blocking, multi-stage APT simulation spanning multiple traffic flows, brute force login detection across SSH and FTP sessions, and port scan detection with appropriate rate limiting response.

C. Validation Against Benchmark Datasets

The trained models are validated against three publicly available benchmark datasets to enable comparison with published results. On the NSL-KDD dataset, the ensemble achieves 98.1% accuracy, matching or exceeding the best published results. On CICIDS2017, 97.6% accuracy is achieved. On UNSW-NB15, 96.8% accuracy is achieved. These results confirm that the system generalizes effectively across diverse traffic environments and attack scenarios.

D. Performance and Stress Testing

System performance under high load conditions was evaluated by gradually increasing synthetic traffic injection rates from 100 Mbps to 2 Gbps while monitoring classification accuracy, latency, and packet loss. The system maintained classification accuracy above 96% and response latency below 50 milliseconds at traffic loads up to 1 Gbps. At 2 Gbps, classification accuracy decreased marginally to 94.1% due to flow feature computation overhead, and response latency increased to 78 milliseconds. These results indicate that a single deployment node is sufficient for networks with bandwidth up to 1 Gbps, while larger networks require distributed deployment.

Long-duration stability testing over 72 continuous hours demonstrated consistent performance without memory leaks or accuracy degradation. The continuous learning pipeline successfully completed three scheduled retraining cycles during the test period, each incorporating newly accumulated traffic samples. Post-retraining accuracy measurements confirmed that retraining maintained or improved performance metrics without disrupting the system's operational availability.

E. Security Penetration Testing

The system was subjected to adversarial testing by a security red team tasked with evading detection using techniques including traffic fragmentation to avoid signature matching, timing-based evasion using slow-rate attacks designed to stay below behavioral detection thresholds, protocol tunneling to disguise malicious traffic within legitimate protocol envelopes, and adversarial machine learning attacks targeting the classification model's decision boundaries.

The traffic fragmentation and protocol tunneling evasion techniques were successfully detected by the behavioral analysis engine, which flagged the unusual packet fragmentation patterns and anomalous protocol distributions. Timing-based slow-rate attacks were partially successful against the per-flow behavioral analysis but were detected by the cross-session correlation engine, which identified the cumulative pattern of connections from the attacking source. Adversarial machine learning attacks, which modified network flow features to closely mimic the statistical characteristics of legitimate traffic, achieved partial evasion success, highlighting a direction for future robustness improvement.

IX. LIMITATIONS AND KNOWN ISSUES

Although the proposed system demonstrates strong overall performance, several limitations warrant acknowledgment. Model performance may degrade for attack techniques that differ substantially from those represented in the training dataset, a challenge common to all machine learning-based detection systems. The continuous learning mechanism mitigates this limitation over time but cannot entirely eliminate it.

The face detection module performs best under controlled indoor lighting conditions and may exhibit reduced accuracy in challenging lighting environments including bright backlighting, low light levels, and rapid illumination changes. Performance is also reduced for faces captured at extreme angles or with significant occlusion. Future versions will incorporate more robust deep learning-based face detection models to address these limitations.

The Isolation Forest anomaly detector may generate elevated false positive rates during periods of legitimate but unusual network activity such as software update deployment, backup operations, or organizational events that generate non-typical traffic patterns. Temporary threshold adjustments for known scheduled activities can mitigate this issue but require operational coordination between security and IT operations teams.

High traffic load environments above 10 Gbps may require distributed deployment of the classification engine across multiple processing nodes, adding architectural complexity. The continuous learning pipeline requires sufficient historical labeled data to produce meaningful model improvements, limiting benefit in newly deployed installations before a sufficient operational history has accumulated.

X. CONCLUSION

This paper presented a comprehensive AI-based smart firewall system that addresses the fundamental limitations of conventional rule-based network security approaches through intelligent, adaptive, and multi-paradigm threat detection. The proposed architecture integrates supervised classification for known threat detection, unsupervised anomaly detection for zero-day threat identification, reinforcement learning for adaptive policy optimization, and computer vision-based face detection for physical access control, providing comprehensive multi-layer security.

Experimental evaluation demonstrated that the system achieves 97.3% overall classification accuracy, an 84.4% reduction in false positive rate compared to a conventional NGFW baseline, and successful zero-day threat detection with 90.3% accuracy through behavioral anomaly analysis. The continuous learning mechanism improved system accuracy by 1.5 percentage points over a 30-day evaluation period, confirming that the system's performance improves continuously in operational deployment.

The integration of OpenCV-based face detection with network security functions represents a novel contribution to the security field, demonstrating that combining physical and logical security modalities into a unified platform can provide more comprehensive protection than separate systems

operating independently. The modular architecture ensures that individual components can be upgraded as improved algorithms become available, extending the system's operational lifespan.

The proposed AI-based smart firewall represents a significant advancement in network security technology, offering a proactive, intelligent, and continuously improving defense against the full spectrum of modern cyber threats. The system is particularly well-suited for deployment in enterprise networks, cloud environments, and critical infrastructure settings where the consequences of security failures are severe.

XI. FUTURE ENHANCEMENTS

Several significant enhancement directions will be pursued in future work. First, the extension of the AI classification pipeline to analyze encrypted network traffic using traffic metadata and behavioral features, without requiring decryption, will address the growing challenge of encrypted malicious traffic. Techniques including TLS fingerprinting, flow statistics analysis, and deep packet inspection of unencrypted header fields will be combined to maintain detection effectiveness as network encryption adoption increases.

Second, the integration of transformer-based deep learning architectures including BERT-derived models adapted for network traffic analysis promises improved performance on sequential and contextual threat patterns. Self-attention mechanisms have demonstrated strong performance on sequential data in other domains and represent a promising direction for network intrusion detection.

Third, federated learning techniques will be explored to enable collaborative threat intelligence sharing across multiple organizational deployments without requiring the disclosure of sensitive network traffic data. Federated learning allows models trained on different organizational networks to contribute to a shared global model while keeping all raw traffic data local, addressing privacy and regulatory

compliance concerns that currently limit threat intelligence sharing.

Fourth, the face detection module will be upgraded to incorporate deep learning-based face recognition using modern convolutional architectures, improving accuracy under challenging environmental conditions and enabling more reliable identity verification for access control applications. Integration with enterprise identity management systems will be implemented to provide seamless authentication experiences for authorized personnel.

Fifth, the development of a mobile application interface will enable security analysts to monitor system status, review alerts, and authorize response actions from mobile devices, improving operational flexibility and response times for after-hours security events. Push notifications for high-severity alerts will ensure that critical threats receive immediate human attention regardless of analyst location.

Sixth, deployment on edge computing infrastructure proximal to critical network segments will reduce classification latency and enable security monitoring for network segments that cannot economically support centralized security infrastructure. Edge deployment using optimized, quantized model variants will make the system accessible to small and medium-sized organizations with limited security budgets.

REFERENCES

1. "Machine Learning Based Model to Identify Firewall Decisions to Improve Cyber-Defense," ResearchGate, 2021.
2. "Classification of Firewall Log Data Using Multiclass Machine Learning Models," MDPI Electronics Journal, vol. 11, 2022.
3. "Next Generation AI-Based Firewalls: A Comparative Study," ResearchGate, 2023.
4. "AI-Based Approach to Firewall Rule Refinement on High-Performance Computing Networks," MDPI Applied Sciences, 2024.
5. "Machine Learning-Based Adaptive Firewall for Real-Time Cloud Security," Journal of Healthcare Data Science and AI, 2024.
6. "Dynamic AI-Augmented Firewall for Real-Time Threat Mitigation," International Research Journal on Advanced Engineering Hub, 2025.
7. "Reinforcement Learning Based Firewall Architecture Leveraging Large Language

Models," International Journal of Artificial Intelligence, Data Science and Machine Learning, 2025–2026.
8. "AI-Based Machine Learning Web Application Firewall (ML-WAF)," STM Journals, 2026.
9. G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, vol. 25, no. 11, pp. 120–126, 2000.
10. M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," Information Processing & Management, vol. 45, no. 4, pp. 427–437, 2009.
11. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in Proc. IEEE Symp. Computational Intelligence for Security and Defense Applications, 2009.
12. I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Information Systems Security and Privacy (ICISSP), 2018.
13. N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems," in Proc. Military Communications and Information Systems Conf. (MilCIS), 2015.
14. F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in Proc. 8th IEEE Int. Conf. Data Mining (ICDM), pp. 413–422, 2008.
15. S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.
16. L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
17. V. Vapnik, The Nature of Statistical Learning Theory, Springer, 1995.
18. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
19. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR), 2001.
20. A. Ronacher, "Flask: A micro web framework for Python." Available: https://flask.palletsprojects.com, 2010.

HOW TO CITE: G. Eesa, P. Jai Sai Chandhan, K. Ch. Revanth Mahesh, Ch. Krishna Sri*, AI-Based Smart Firewalls: Intelligent Network Security Using Machine Learning and Behavioral Analysis, Int. J. Sci. R. Tech., 2026, 3 (5), 1091-1106. https://doi.org/10.5281/zenodo.20443251