Journal of Information Systems Engineering and Management
2026, 11(2s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

AI-Driven Cybersecurity for Detecting Anomalous API Access Patterns in Distributed Applications

Syed Sadiqur Rahman[1], Sikander Niaz[2], Daudul Islam[3], Md Shamiul Islam[4,*], Rahima Binta Bellal[5]

[1]Department of Cyber Security and Forensic Computing, University of Prince Mugrin, Madinah, Saudi Arabia

[2]Master of Cybersecurity (MSCIA), Virginia University of Science & Technology, 2070 Chain Bridge Road, Suite G100, Vienna, VA 22182, USA

[3]Department of Computer Science and Engineering, International Islamic University Chittagong, Kumira, Chattogram-4318, Bangladesh

[4]Department of Computer Science and Engineering, Bangladesh University of Business & Technology, Plot # 77-78, 2 Road No. 7, Dhaka 1216, Bangladesh

[5]Labry School of Science, Technology & Business, Cumberland University, 1 Cumberland Dr, Lebanon, TN 37087, USA

*Corresponding author e-mail: shamiulislam693@gmail.com

ARTICLE INFO

Received: 05 Jan 2026

Revised: 18 Feb 2026

Accepted: 26 Feb 2026

ABSTRACT

Application Programming Interfaces (APIs) are what make modern distributed and cloud-native apps work. They let loosely connected services work together without any problems. But the fact that APIs are so widely used has made them much easier to attack, making them perfect targets for stealthy and behavior-driven cyberattacks that can get by traditional signature-based defenses. Current methods for detecting anomalies often either use supervised learning, which needs a lot of labeled attack data, or only use unsupervised learning, which doesn't have enough power to tell the difference between classes and has a lot of false positives, especially when there is a lot of class imbalance. This research presents an AI-driven Normality Fusion Hybrid Model to identify aberrant API usage patterns in distributed systems. The suggested method uses unsupervised anomaly modeling to learn normal API access behavior and combines deviation-aware signals into a supervised classification pipeline. An Isolation Forest is used to model the normal behavior manifold and make deviation scores, which are then normalized and combined with the original access features. A class-weighted Histogram-based Gradient Boosting classifier is then trained on the improved feature space to find rare and changing anomalies in a strong way. To fix the problem of very uneven class sizes, Random Over-Sampling and the Synthetic Minority Over-sampling Technique (SMOTE) are used during training. Numerous trials on the API Security: Access Behavior Anomaly Dataset show that the suggested model works better than strong baseline models and contemporary state-of-the-art techniques, with an overall accuracy of 99.87%. The results show that normality-aware feature fusion greatly improves detection robustness, interpretability, and generalization. This makes the proposed framework good for real-time API security monitoring in remote situations.

Keywords: API Security; Anomaly Detection; Distributed Applications; Machine Learning; Normality Modeling; Class Imbalance; Cybersecurity

1. INTRODUCTION

The modern digital landscape is becoming more and more centralized around massive distributed applications which serve as a vessel for vital services across finance, healthcare, smart infrastructure, e-commerce, and even in government [1]. These applications are no longer monolithic, but rather consist of loosely interconnected services running on a variety of different platforms; e.g., cloud, edge or hybrid [2, 3]. At the heart of this service-oriented transformation are APIs - they represent standardized interfaces to allow all of these architectures (as well as others) to communicate, integrate and demonstrate some level of interoperability [4, 5]. Despite their power in enhancing the agility and scalability of systems, APIs' widespread availability has widened the attack surfaces of modern applications, presenting them as high-value targets for cyber adversaries interested in leveraging logical vulnerabilities rather than known flawed software [6-8].

Concurrently, the type of cyber threat has changed quite dramatically over time in both volumes as well as complexity [9, 10]. Nowadays, attacks are adopting stealthy tactics with the use of legitimate communication means, evading signature-based detectors and acting as a "chameleon" in the system [11, 12]. API-centric attacks exemplify this trend, as malicious actors often generate requests that conform syntactically to valid API calls while subtly manipulating access frequency, parameter usage, or request sequences to achieve unauthorized objectives. Encryption, dynamic service discovery, and elastic scaling are all becoming more common, which makes it even harder to see and monitor things [13]. This means that traditional perimeter defences are not enough. As a result, cybersecurity has moved towards behavioural and data-driven models that look for changes in how a system is used instead of clear attack fingerprints [14, 15].

Artificial Intelligence (AI) and Machine Learning (ML) play a critical role in this transition to behavior-oriented cyber security [16]. Using patterns learned from the past, AI can detect change in the environment and discover new threats not previously observed, as well as relieve dependency on human rule-engineering [17]. Recent studies have investigated various ML and deep learning algorithms in an intrusion detection, anomaly detection, and user behavior analytics context, leading to several significant gains compared to static models [18, 19]. However, such techniques have not been widely applied in the surveillance of API accesses, which have its unique issues like high-dimensional feature space, temporal diversity and extremely imbalanced class distribution between normal events and malicious ones [20, 21]. These obstacles reduce the applicability and stability of numerous works on real world distributed computing systems.

The main drawback of existing works is their partial method (separate to one another) of the anomaly detection and attack classification [22, 23]. Most existing works use strict supervised models, which are highly dependent on labeled attack data, or pure unsupervised models, which have less discriminative power [24, 25]. In reality, available labeled malicious API traffic is little, often changing over time and possibly incomplete making supervised models to tend policing the other direction toward main normal patterns [26, 27]. Unsupervised ones, on the other hand, can highlight benign changes as anomalies causing false positives and operational burden [28, 29]. Furthermore, some state-of-the-art deep learning models carry large amount of computational overhead which also prohibits their adoption in the real-time application scenario (i.e., where latency and resource limitation concern) such as typical distributed systems [30, 31]. These limitations show a general lack in the literature in proposing integrated, efficient, and flexible AI-based cybersecurity models for API access control. Clearly, there is a need for the techniques that are based on explicit modeling of normal system behavior and that make use of deviations from such a model as an informative signal; with mechanisms to incorporate these signals into classification procedures [32]. Such models also need to tackle the problem of class imbalance and changing access patterns while still remaining computationally cheap and deployable. Although hybrid approaches to AI are becoming increasingly popular, relatively little consideration has been given to feature-level fusion strategies that effectively combine unsupervised normality modelling with supervised learning [33].

Inspired by these observations, in this work, we aim to contribute to the improvement of API security with a new normality-aware AI framework tailored for distinguishing abnormal API access patterns from distributed applications. The driving approach is to move from attack-centric learning in anomaly detection to behavioral-centric modeling, where knowing and encoding legitimate usage patterns is fundamental. By incorporating anomaly traits directly into the feature space, this approach attempts to make the model more sensitive to infrequent behaviors that are typically leading or defining malicious actions but still robust against benign variations. Our motivations are in line with those of this study, and we aim to design a pragmatic, effective detection pipeline. In particular, the goal of this work is to model normal API behaviour with unsupervised learning strategies, improve classification task performance through feature-wise combination of anomaly evidences and alleviate the influence of data imbalance by leveraging adaptive learning approaches [34]. Moreover, the framework is architected to be light-weight and scalable such that it supports real-time monitoring needs in distributed as well as cloud-native context. taken together, these goals serve as a guide to creating an AI-based work that balances detection efficacy, robustness and operational usability [35-37].

For that we propose a Normality Fusion hybrid AI model that combines unsupervised anomaly detection and supervised gradient-boosted classification. The first step in the framework is to use only normal API traffic to train an Isolation Forest and learn intrinsic behavioral profiles, that are quantified by anomaly scores. These scores are enriched with their ranking in the observed data distribution and further combined to form an enriched representation of access behavior together with the original encoded features of API [38]. A class-weighted gradient boosting classifier is trained over this augmented feature set, which can effectively differentiate between normal and anomalous access patterns even when the dataset exhibits an extreme ratio imbalance. This architecture combines the unsupervised and supervised part effectively, but also computationally efficient [39].

The following is a summary of this study's main contributions:

- **Hybrid normality-aware detection model:** We provide an AI-based security framework by combining unsupervised normality modeling with supervised classification for detecting abnormal API calling behaviors in distributed applications, which can effectively identify known attacks as well as unknown attacks [40].
- **Synthetic data balancing with SMOTE:** Owing to the presence of extreme class imbalance in real API traffic, we use the Synthetic Minority Over-sampling Technique (SMOTE) to create synthesised samples representative of minority attack class enabling better training stability and detection sensitivity of the classifier.
- **Feature-level anomaly fusion strategy:** We propose a feature augmentation approach that fuses the Isolation Forest - based anomaly scores and percentile ranks with original encoded API features, helping the classifier to explicitly leverage deviations of learned normal behavior.
- **Imbalance-aware gradient boosting classification:** A class-weighted gradient boosting classifier on the augmented feature space guarantees robustness in operating under skewed ratios of differently labeled datasets without having to increase too much computation cost [41].
- **Scalable and deployable architecture:** The proposed network design is highly lightweight and modular, allowing for real-time deployment in distributed as well as cloud-native settings which is justified through the extensive experimental evaluation against baseline models.

2. LITERATURE REVIEW

Recent progress in anomaly detection encompasses various application fields, such as cloud computing, network security, industrial systems, and data-centric behavioral analytics. Previous studies have investigated many methodological paradigms, including deep learning-based sequential modeling, unsupervised reconstruction techniques, domain-specific machine learning frameworks, and data enrichment methodologies [42]. To systematically analyze these contributions and highlight prevailing research trends and gaps, the reviewed studies are organized into three thematic categories: deep learning–driven anomaly detection in networked and cloud environments, domain-specific anomaly detection in cyber-physical and IoT systems, and data-centric behavioral anomaly detection frameworks [43]. To systematically analyze these contributions and highlight prevailing research trends and gaps, the reviewed studies are organized into three thematic categories: deep learning–driven anomaly detection in networked and cloud environments, domain-specific anomaly detection in cyber-physical and IoT systems, and data-centric behavioral anomaly detection frameworks [44].

2.1 Cloud, Network, and Communication-Centric Detection Frameworks

Anomaly detection frameworks for cloud infrastructures, programmable networks, and next-generation communication systems mostly utilize deep learning models applied to logs, traffic flows, or protocol interactions. Nagarjun & Rajkumar [45] and Jay Barach [46] use learning-based methods to look at cloud audit logs and system activities. They assume that the log semantics are the same across platforms and that there is full visibility. In heterogeneous or federated cloud settings, these assumptions seldom prevail, as variations in logging granularity, access control, and provider-specific formats diminish portability and compromise operational resilience.

Torabi et al. [47] present an autoencoder-based reconstruction error modeling approach for detecting anomalies in cloud networks. This method enhances detection granularity by analyzing errors on a feature-wise basis. Reconstruction-driven systems, on the other hand, are very sensitive to changes in typical behavior. This is because elastic cloud environments often have changes in scalability, workload transfer, and configuration. As a result, normal changes in operations may be seen as strange. Sayed et al. [48] seek to minimize detection costs in SDN systems using feature selection preceding deep learning-based DDoS detection; however, this approach imposes rigidity by emphasizing preset feature relevance, hence constraining adaptability to novel or multi-vector attacks [49].

Tian et al. [50] and Wehbe et al. [51] advance anomaly identification in 5G core networks by modeling NF-to-NF contact sequences and HTTP/2 traffic patterns, shifting the focus to the control plane and application layer. These methods provide you a lot of detail, but they are very dependent on certain protocols and architectural assumptions. This makes them weak to changes in protocols, limits on encryption, and frequent changes to the network. A common worry in this group is that they rely on stable configurations and have a lot of computational overhead, which makes it hard to scale and adapt over time in networks that are always changing [41, 52, 53].

2.2 Cyber-Physical, IoT, and Infrastructure-Oriented Detection Approaches

Anomaly detection in cyber-physical systems and IoT settings must take into consideration how digital control and physical processes interact, which makes things even more complicated. Hussain et al. [54] and Mantha et al. [55] tackle data scarcity by employing simulation-driven and hardware-in-the-loop settings for electric vehicle charging stations and smart grid systems. While these methodologies facilitate regulated experimentation, simulated datasets frequently do not adequately reflect real-world uncertainty, sensor noise, unpredictability in human interaction, and long-term system drift, hence compromising reliability in operational environments [56].

Sarwar et al. [57, 58] use supervised machine learning classifiers on smart home IoT traffic, assuming that there is labeled attack data and that device behavior is fairly constant. These kinds of assumptions make it harder to find zero-day attacks and new device behaviors in IoT ecosystems that change quickly. Narmadha & Balaji [59] use metaheuristic strategies to improve LSTM models and combine them with autoencoders to cut down on false alarms in intrusion detection. However, adding optimization layers makes the computations much more complicated and requires more time to tune, which makes them less suitable for real-time use in contexts with limited resources [60].

Ehsan et al. [61, 62] broaden anomaly detection to blockchain ecosystems by employing ensemble classifiers and feature selection to identify malicious actors. The technique focuses on a very important security area, but it only works with curated datasets and static account-level attributes. This makes it hard to react to changing transaction behaviors, smart contract interactions, and new attack strategies. In general, this category has significant domain coupling and isn't very resilient to changes in operations, especially when system dynamics are changing.

2.3 Behavioral and Data-Driven Anomaly Detection Paradigms

Behavioral and data-driven techniques move anomaly detection away from infrastructure measures and toward modeling how users behave, how software runs, and what content means. Kim et al. [63] examine betting odds to identify match-fixing anomalies; however, this market-driven data is extremely volatile and subject to external economic and social influences, making anomaly interpretation more difficult. Kumar et al. [64] utilize topic modeling and semantic similarity to identify healthcare-related anomalies in social media data, wherein linguistic drift, disinformation, and demographic bias considerably impact stability and reliability.

Zhan et al. [65] enhance resilience against hostile manipulation by modeling semantic behavior units in API and syscall sequences; nonetheless, this methodology presupposes stable semantic groupings that may not endure across software upgrades or changing execution circumstances. Kohyarnejadfard et al. [66] utilize NLP-based analysis on distributed tracing data within microservice environments, requiring no prior knowledge; however, the efficacy of this approach is contingent upon uniform trace instrumentation and overlooks essential low-level system metrics vital for root-cause analysis.

Canay and Kocabıçak [67] improve online usage mining by combining application logs with web analytics data to get more information about user activity. However, this makes preprocessing and system integration more difficult, which makes it harder to respond quickly in real-time situations. ElDahshan et al. [68] put forward an unsupervised MLaaS framework to help non-expert users find anomalies and group them together. However, using generic unsupervised algorithms that don't learn from their features or know about the domain makes it harder to find anomalies in complex, high-dimensional data. Zhang et al. [69] assess multimodal language models for visual anomaly detection in bad settings, uncovering ongoing issues concerning robustness, inference delay, and processing cost. This category collectively underscores that data richness alone does not guarantee reliable anomaly identification in noisy, dynamic, and adversarial environments.

The gaps reviewed in the above studies are summarized as follows:

- Most current methods rely on labeled data or steady normal patterns, which makes them less reliable when it comes to new, changing, or rare anomalies.

- Different fields still don't do a good job of dealing with data imbalance, which makes it harder to find rare but important unusual events.
- Some frameworks that use deep learning are very hard to compute, which makes them impractical for large-scale and real-time use.
- A lot of solutions are tested on controlled datasets or in simulated environments, which makes people worry about how well they will work in real-world, changing systems.
- Unified and flexible anomaly detection systems that can work with different types of data and domains are still mostly unknown [70].

3. METHODOLOGY

The methodological approach used to develop and assess the suggested Normality Fusion Hybrid Model for identifying unusual API access behaviors in distributed applications is presented in this part. Severe class imbalance, changing access patterns, and the requirement for strong discriminating between benign changes and malicious activity are some of the major issues with API security analytics that the technique is designed to address. By explicitly learning baseline access behavior and including deviation-aware features into the decision-making process, the suggested method combines supervised classification with unsupervised normalcy modeling. The entire pipeline includes feature fusion, imbalance-aware learning, normal behavior modeling, data collection and preprocessing, and final anomaly classification [41]. The framework's applicability for actual API monitoring environments is ensured by the methodical design of each stage, which improves detection accuracy, interpretability, and scalability. The overall workflow diagram is visualized in Figure 1.

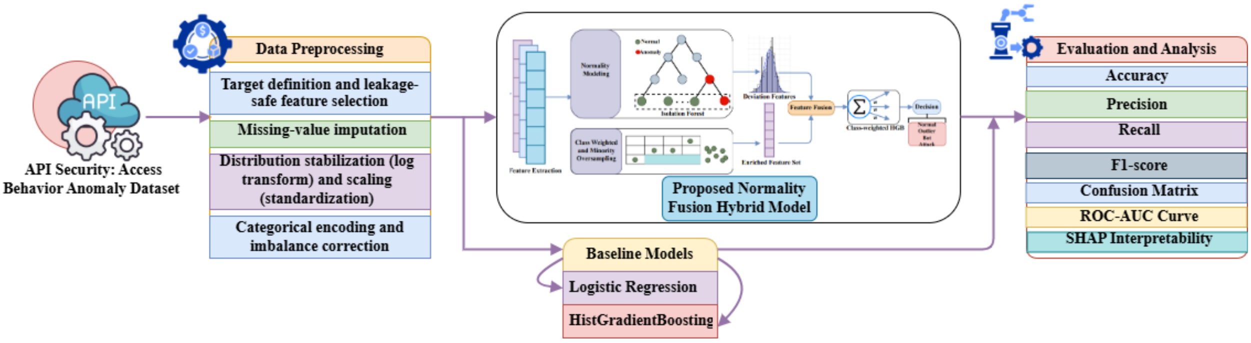

Figure 1. Overall workflow of this study

3.1 Data Collection and Overview

This study utilizes the API Security: Access Behavior Anomaly Dataset from Kaggle [71] intended for the examination of anomalous and malicious access behaviors inside API-driven microservices setups. In these types of systems, both authorized applications and programmatic clients can use APIs. This makes them vulnerable to misuse through strange request sequences and access patterns. The dataset shows how people use APIs by using numbers, which makes it possible to use machine learning models without having to directly handle raw API call graphs. It also has raw API access graphs that may be used for clustering and graph-based analytics. The data shows how session handling, retries, refreshes, and automated scripts change how people access things. The dataset has 34,423 samples and 13 columns. Twelve of the columns show access behavior aspects, and the target variable behavior_type puts each instance into one of four groups: normal, attack, bot, or outlier. The dataset is good for research on API security and access behavior anomaly detection because it comes with notebooks for classification and embedding activities. As shown in Table 1, the dataset has a very unbalanced class distribution. Figure 2 shows the class-wise sample distribution more clearly, showing how outlier and normal behaviors are more common than rare attacks.

Table 1. Class-wise distribution of samples in dataset

Behavior Type	**Number of Samples**
Outlier	24,146
Normal	8,946
Bot	1,309
Attack	22
Total	**34,423**

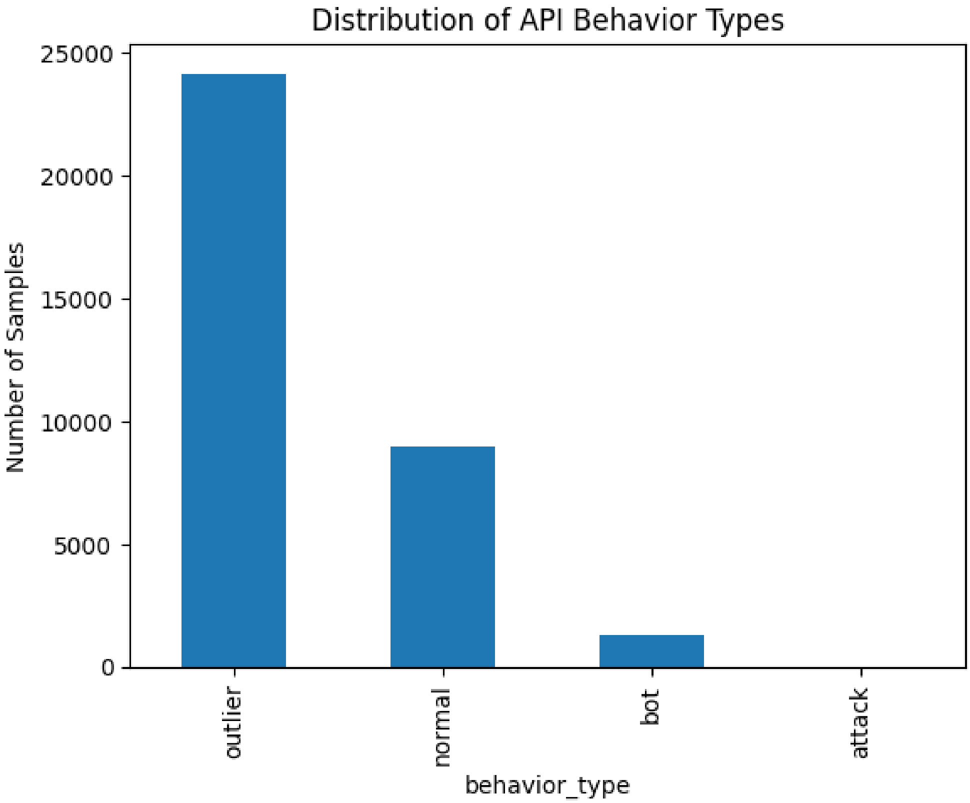

Figure 2. Visualization of sample distribution across target classes

3.2 Data Preprocessing

In this subsection, the preprocessing operations carried out on Deep Weeds dataset will be outlined to increase the weed-relevant features and minimize the background noise [72]. It uses vegetation segmentation, illumination normalization, morphological enhancement, and adaptive background suppression techniques to create clean and informative images, which means that the AgroViT-CNN model can be used to produce robust and accurate weed classification under real-world field conditions.

3.2.1 Target definition and leakage-safe feature selection

The target column in the dataset defines the class label:

$$y_i = behavior\_type \in (outlier, normal, bot, attack) \tag{1}$$

To eliminate target leakage and get rid of identifier-like attributes, we leave out columns like behavior, _id, Unnamed: 0, and behavior_type. This gives us the feature matrix:

$$X = D \setminus \{leakage/id\ columns\}, \quad X \in \mathbb{R}^{n \times d} \tag{2}$$

This makes sure that the learning process only depends on access-behavior descriptions and not on direct class indicators or IDs.

3.2.2 Missing-value imputation (numeric and categorical)

The pipeline applies median imputation for numeric features and most-frequent imputation for categorical features. If a numeric feature j is absent, it is replaced by:

$$x_{ij} = \begin{cases} x_{ij}, & if\ x_{ij}\ is\ observed \\ median(\{x_{1j}, \dots, x_{nj}\}), & if\ x_{ij}\ is\ missing \end{cases} \tag{3}$$

For category feature k, the mode is used to fill in missing entries:

$$x_{ik} \leftarrow arg \max_{c \in C_k} \#\{i: x_{ik} = c\} \tag{4}$$

3.2.3 Distribution stabilization (log transform) and scaling (standardization)

The study uses a safe log transformation to make heavy-tailed security measures (such durations and counts) less skewed:

$$x'_{ij} = \log(1 + max(x_{ij}, 0)) \tag{6}$$

Then, z-score normalization is used to standardize the numerical features:

$$z_{ij} = \frac{x'_{ij} - \mu_j}{\sigma_j} \tag{7}$$

Where, μ_j and σ_j are the mean and standard deviation of feature j, which were calculated using the fitted data. This makes features easier to compare and makes optimization more stable for learners who come after.

3.2.4 Categorical encoding and imbalance correction (One-Hot + ROS + SMOTE)

One-hot encoding changes categorical variables (like ip_type and source) into binary variables.

$$\phi(x_{ik}) \in \{0,1\}^{|C_k|} \tag{8}$$

generating a sparse indication vector for each category level and making it work with numeric learners.

The dataset is very unbalanced, especially the rare assault class. So, the study balances the classes in two steps:

1. Random Over-Sampling (ROS) makes copies of minority instances until it reaches a target count N^* (which is used to upsample an attack to the majority size).
2. Using linear interpolation, SMOTE makes fake samples for chosen minority classes (bot and normal):

$$x_{new} = x_i + \lambda(x_{nn} - x_i), \qquad \lambda \sim u(0,1) \tag{9}$$

Where x_{nn} is a nearest neighbor of x_i in feature space. This increases minority coverage without simply duplicating all samples.

3.3 Proposed Normality Fusion Hybrid Model

To effectively identify anomalous API access behaviors amidst significant class imbalance and constantly changing access patterns, we present a Normality Fusion Hybrid Model that synergistically combines unsupervised anomaly modeling with supervised discriminative learning. The main idea behind the proposed method is to clearly define the range of normal API access behavior and to include measured deviations directly in the supervised classification process. The model increases sensitivity to small and infrequent deviations by combining normality-aware signals with behavioral feature representations [58]. At the same time, it stays strong against benign behavioral variability. This hybrid approach makes it possible to reliably find low-frequency and previously undiscovered attack patterns [73]. It also solves some of the biggest problems that come up in real-world API security monitoring situations. The architectural overview of the proposed model is shown in Figure 3.

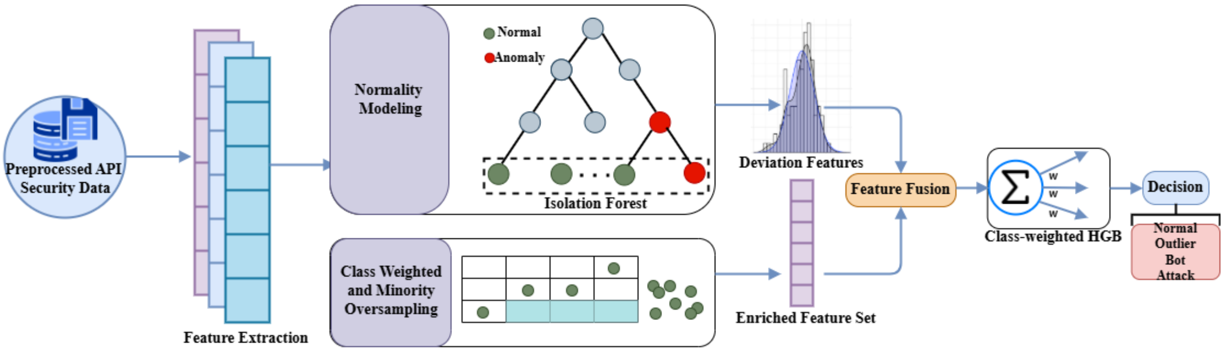

Figure 3. Architecture of the proposed Normality Fusion Hybrid model

3.3.1 Input Data

Normality Fusion Hybrid model takes the input in the form of preprocessed feature matrix as,

$$X = \{x_i \in \mathbb{R}^d\}_{i=1}^{n} \tag{10}$$

And the corresponding behavioral class labels as,

$$y_i \in Y = \{normal, outlier, bot, attack\} \tag{11}$$

3.3.2 Normal Behavior Manifold Estimation

We proceed by separating samples that are linked to typical API access.

$$X_N = \{x_i \mid y_i = normal\} \tag{12}$$

The system doesn't learn a global decision boundary from all classes at once. Instead, it first assesses the support of the normal behavior distribution. An Isolation Forest (IF) is used to do this. It is an approximate measure of how deep a sample is in feature space that depends on the data.

An isolation tree divides the feature space into smaller parts by choosing a feature index j and a split value v at random. The procedure persists until every instance is segregated. Let $h_t(x)$ be the number of splits needed to separate x in the t-th tree. The anticipated isolation depth over an ensemble of T trees is:

$$\mathbb{E}[h(x)] = \frac{1}{T}\sum_{t=1}^{T} h_t(x) \tag{13}$$

Anomalous samples are usually found in areas of the feature space that aren't very dense, therefore they need fewer partitions to be separated. So, a deviation score is defined as:

$$s(x) = -\mathbb{E}[h(x)] \tag{14}$$

where larger numbers mean greater variance from the predicted typical manifold.

3.3.3 Distribution-Free Normalization of Deviation Scores

Raw anomaly scores $s(x)$ include feature scaling and ensemble depth. To make sure that the samples are stable and can be compared, we calculate a distribution-free percentile score:

$$r(x) = \frac{1}{n}\sum_{j=1}^{n} \mathbb{I}(s(x_j) \ \le \ s(x)) \tag{14}$$

This transformation translates deviation scores onto the empirical cumulative distribution function (ECDF), resulting in:

$$r(x) \in [0,1\,] \tag{15}$$

This might be understood as the sample's relative strangeness compared to the whole dataset. This step is very important to stop classifiers from overfitting to the scale of isolation depths.

3.3.4 Normality Fusion via Feature Augmentation

This might be understood as the sample's relative strangeness compared to the whole dataset. This step is very important to stop classifiers from overfitting to the scale of isolation depths.

$$\tilde{x}_i = [x_i^T, s(x_i), r(x_i)]^T \in \mathbb{R}^{d+2} \tag{16}$$

This formulation incorporates both absolute deviation magnitude (s) and relative deviation rank (r) alongside the original access behavior elements. The classifier learns a combined representation as a result.

$$P\big(y\big|x, \Delta(x)\big), \qquad \qquad where\ \Delta(x) = (s, r) \tag{17}$$

This makes it easier for it to distinguish the difference between benign irregularities and malicious ones.

3.3.5 Imbalance-Aware Gradient Boosted Classification

Because the class distribution is so skewed, directly minimizing unweighted empirical risk would make the classifier favor the dominating classes. To address this, class-prior–aware weighting is implemented. Let,

$$\pi_c = \mathbb{P}(y = c) \approx \frac{|i: y_i = c|}{n} \tag{18}$$

indicate the empirical prior probability of class c. The weight is given to each training sample:

$$\omega_c = \frac{1}{\pi_{y_i}} \tag{19}$$

This makes sure that minority classes add to the optimization goal in the right way.

An expanded dataset is used to train a Histogram-based Gradient Boosting (HGB) classifier:

$$\widetilde{D} = \{\tilde{x}_i, y_i, \omega_i\}_{i=1}^{n} \tag{20}$$

The learning goal is to lower the weighted negative log-likelihood or a weighted loss:

$$\mathcal{L}(\theta) = \sum_{i=1}^{n} \omega_i \sum_{c \in y} \mathbb{I}(y_i = c) \log\left(\frac{\exp\left(f_c(\tilde{x}_i; \theta)\right)}{\sum_{c'} \exp\left(f_c(\tilde{x}_i; \theta)\right)}\right) \tag{21}$$

where $f_c(\cdot)$ is the score for class c.

The Gradient Boosting model is written as an additive function:

$$f(\tilde{x}) = \sum_{m=1}^{M} \eta \, g_m(\tilde{x}) \tag{22}$$

with $g_m(\cdot)$ standing for shallow decision trees and η standing for the rate of learning. Histogram binning makes sure that calculations are quick and that noise doesn't affect them.

3.3.6 Inference and Decision Function

For an unobserved API access instance $x^{\backslash *}$, the trained Isolation Forest calculates deviation features:

$$\tilde{x}^{\backslash *} = \left[x^{\backslash T}, s(x^{\backslash *}), r(x^{\backslash *})\right]^T \tag{23}$$

The final prediction is obtained by:

$$\hat{y} = arg \max_{c \in y} P(y = c | \tilde{x}^{\backslash *}) \tag{24}$$

This decision rule jointly considers intrinsic behavior and deviation from learned normality, enabling reliable detection of rare and evolving attack behaviors.

3.4 Baseline Models

We evaluate the performance of the proposed Normality Fusion Hybrid Model to that of two commonly used baseline techniques that reflect unsupervised and supervised learning paradigms. This is to see how well it works. The Isolation Forest is used as an unsupervised baseline because it can simulate normal behavior without needing labeled data and is often used for finding anomalies. Also, the

Histogram-based Gradient Boosting (HGB) classifier is used as a supervised baseline to see how much better normality-aware feature fusion is than a powerful discriminative learner that was only trained on behavioral features. These baselines serve as a significant benchmark for assessing the contribution of the proposed hybrid architecture [74].

3.4.1. Logistic Regression

Logistic Regression uses a linear decision function over the input features to model the posterior probability of each class. The softmax formulation is used for multi-class classification:

$$P(y=c|x)=\frac{\exp\,(w_c x+b_c)}{\sum_{k\in y}\exp\,(w_k x+b_k)} \tag{25}$$

The predicted label is:

$$\hat{y}=arg\max_{c\in y}P(y=c|x) \tag{26}$$

LR is an effective foundation for figuring out if the dataset can be separated linearly and how much better it gets with non-linear modelling and normality fusion.

3.4.2. Histogram-based Gradient Boosting (HGB)

Based on histograms Gradient Boosting is a supervised ensemble that uses gradient-based optimization on a classification loss to build an additive model of decision trees. When working with enormous datasets, histogram binning makes things more efficient. HGB is trained directly on the original feature set $x\in\mathbb{R}^d$, which means that it doesn't use any normality fusion features.

$$f(x)=\sum_{m=1}^{M}\eta\;g_m(x) \tag{27}$$

Where g_m are shallow trees, M is the number of iterations, and η is the learning rate.

4. RESULT & DISCUSSION

4.1 Experimental Setup and Hyperparameters

The experimental setup was carefully planned to make sure that the suggested Normality Fusion Hybrid Model for finding API access behavior anomalies could be tested reliably. The API Security: Access Behavior Anomaly Dataset, which has 34,423 samples, 12 numerical characteristics, and one target variable, was used for the experiments. As shown in Table 2, the dataset was split into 70% training and 30% testing, keeping the original class distribution. Imputation, normalization, and standardization were used to prepare all of the features. To fix the very uneven class distribution, Random Over-Sampling (ROS) and SMOTE were used just on the training set. We trained the Isolation Forest model on normal-class samples and then combined the resulting deviation scores with the original features for supervised learning.

We used scikit-learn and imbalanced-learn to develop the proposed model and baseline methods in Python. Then, we ran them on Google Colab Pro+ with an NVIDIA A100 GPU. The whole procedure of training and testing took about four hours, which made sure that the experiments were quick and could be repeated.

Table 2. 70:30 Split of API Security Dataset for Training and Testing

Behavior Type	**Total Samples**	**Training Set (70%)**	**Testing Set (30%)**
Outlier	28,975	24,146	4,829
Normal	28,976	24,146	4,830
Bot	28,975	24,146	4,829
Attack	28,975	24,146	4,829
Total	**115,901**	**96,584**	**19,317**

We carefully set the hyperparameters of the suggested Normality Fusion Hybrid Model so that it would be stable and accurate at the same time. The Isolation Forest part was trained on 800 trees with a contamination rate of 0.01, which made it possible to learn normal access behaviour while taking into consideration how rare attacks are. The Histogram-based Gradient Boosting classifier was set up for the supervised stage with a maximum tree depth of 7, a learning rate of 0.05, and 700 boosting iterations. This helped the model find complicated patterns without overfitting too much. To make sure that the tests were fair and could be repeated, all hyperparameters were kept the same throughout. The hyperparameters are outlined in Table 3.

Table 3. Hyperparameter Configuration

Component	**Hyperparameter**	**Value**	**Description**
Isolation Forest	Number of estimators	800	Total number of isolation trees
	Contamination	0.01	Expected proportion of anomalies
	Random state	42	Seed for reproducibility
	Parallel jobs	All cores	Enables parallel tree construction
Normality Fusion	Anomaly score	$s(x)$	Isolation-based deviation measure
	Rank score	$r(x)$	Percentile-normalized deviation
HGB Classifier	Max tree depth	7	Controls model complexity
	Learning rate	0.05	Shrinkage factor
	Number of iterations	700	Boosting stages
	Min samples per leaf	30	Regularization parameter
	Random state	42	Seed for reproducibility

4.2 Performance Analysis using Evaluation Metrices

Table 4 shows the class-wise Precision, Recall, F1-score, and total accuracy of the Logistic Regression baseline, the Histogram-based Gradient Boosting baseline, and the new Normality Fusion Hybrid

Model. Logistic Regression has an overall accuracy of 0.9036, but it doesn't work as well for minority classes, especially the outlier class (F1-score 0.8694) and bot class (F1-score 0.8795). This shows that it isn't very good at handling API access patterns that aren't linear or balanced. Histogram-based Gradient Boosting greatly enhances detection performance, with an overall accuracy of 0.9984 and almost perfect class-wise F1-scores, such 0.9999 for attack and 0.9968 for outlier. This shows that it is very good at telling the difference between classes.

Table 4. Class-wise Performance Comparison of Baseline and Proposed Models

Model	Class	Precision	Recall	F1-score	Accuracy
Logistic Regression	Attack	0.9240	0.9043	0.9141	0.9036
	Bot	0.9414	0.8252	0.8795	
	Normal	0.9453	0.9584	0.9518	
	Outlier	0.8191	0.9263	0.8694	
Histogram-based Gradient Boosting	Attack	0.9998	1.0000	0.9999	0.9984
	Bot	0.9977	0.9961	0.9969	
	Normal	1.0000	1.0000	1.0000	
	Outlier	0.9961	0.9975	0.9968	
Proposed Normality Fusion Hybrid Model	Attack	1.0000	1.0000	1.0000	**0.9987**
	Bot	0.9979	0.9967	0.9973	
	Normal	1.0000	1.0000	1.0000	
	Outlier	0.9967	0.9979	0.9973	

The suggested Normality Fusion Hybrid Model improves performance even more, getting the best total accuracy of 0.9987 and always better outcomes for each class. It gets flawless Precision, Recall, and F1-score (1.0000) for both the attack and normal classes. It also keeps very high F1-scores for the bot (0.9973) and outlier (0.9973) classes. These results show that adding normality-aware deviation features to the supervised learning process works well, resulting in balanced and reliable detection across all behaviour categories. Figure 4 shows how well the suggested model works for each class by showing the distribution of Precision, Recall, and F1-score across all classes.

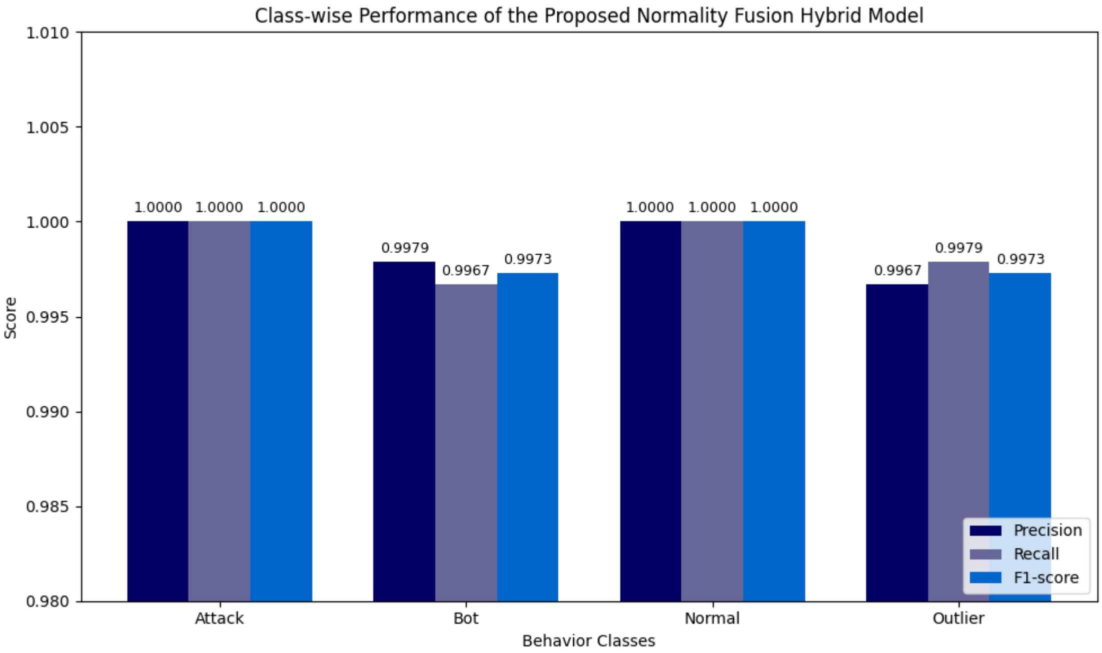

Figure 4. Classwise comparison of the performance across Precision, Recall and F1-score

4.3 Error Analysis

To further understand how the assessed models misclassify things, we used the confusion matrices of Logistic Regression, Histogram-based Gradient Boosting (HGB), and the Proposed Normality Fusion Hybrid Model to do an error analysis. This is shown in Figure 5.

The Logistic Regression model demonstrates a relatively elevated rate of misclassification across several classifications, especially within security-sensitive categories. A large percentage of bot instances (748 samples) are wrongly labelled as outliers, and attack instances (223 samples) and outliers (239 samples) are often wrongly labelled as normal. These mistakes show that the linear decision limits of Logistic Regression have trouble separating overlapping access behaviour patterns, especially when there is a big class imbalance. This makes it harder to find attacks because the false negative rate is larger. The Histogram-based Gradient Boosting model, on the other hand, greatly lowers the number of misclassification mistakes, showing that it can tell the difference between non-linear data quite well. There is still some confusion between the outlier and bot classes, though. For example, 11 outlier samples were incorrectly categorised as bots, and 19 bot samples were incorrectly forecasted as outliers. These errors are not very big, but they do imply that there is still some overlap in behavioural traits between automated and anomalous access patterns.

The Proposed Normality Fusion Hybrid Model has the fewest errors of all the models. The confusion matrix indicates that the attack and normal classes are perfectly classified, with no false positives or negatives. There is just a small amount of confusion between the outlier and bot classes (10 outliers were incorrectly categorised as bots and 16 bots were incorrectly classed as outliers). This is to be expected because both behaviours are semantically similar. The near-diagonal dominance of the confusion matrix shows that adding normality-aware deviation features really does make classes easier to tell apart and stop errors from spreading.

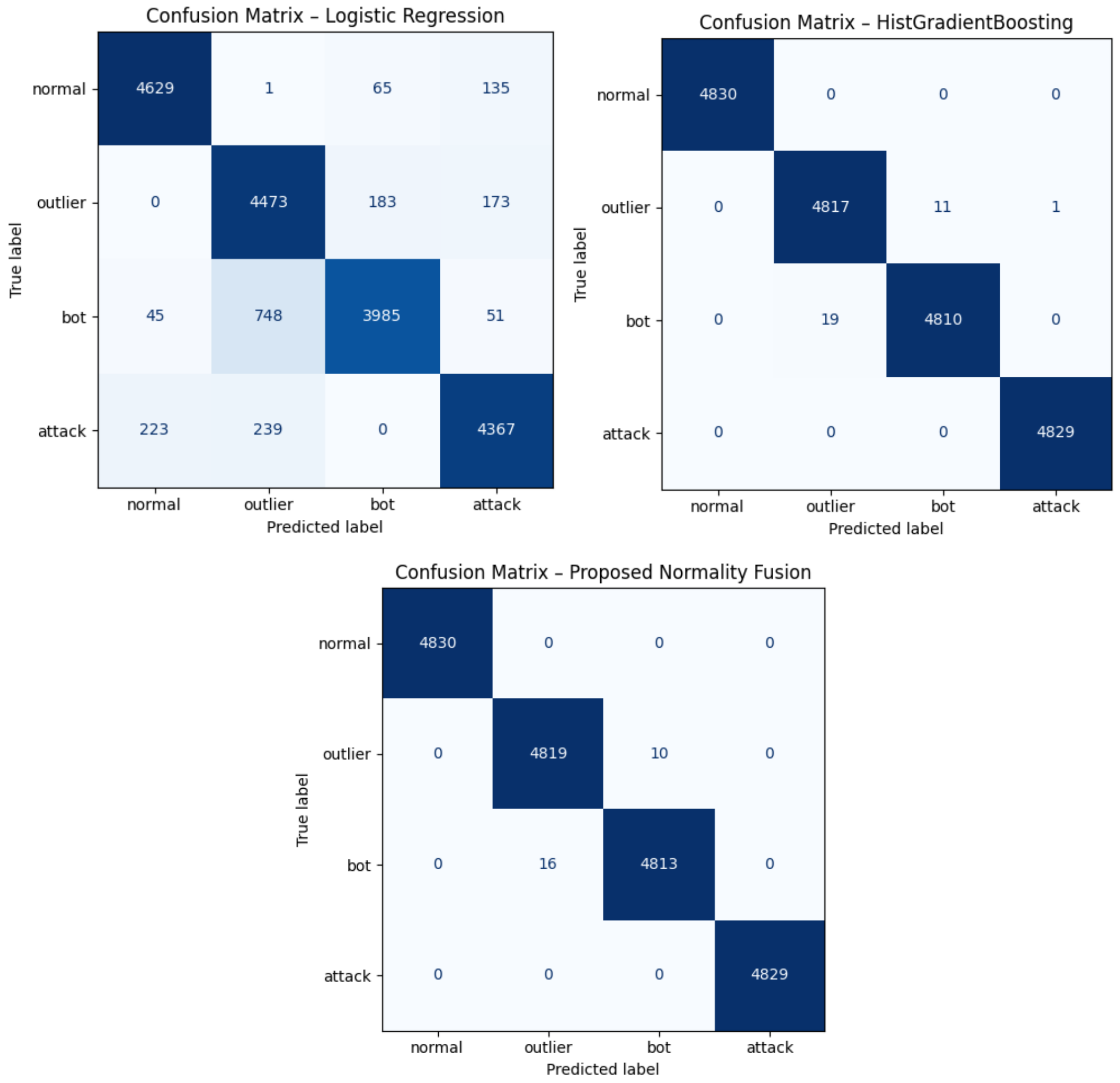

Figure 5. Confusion matrices illustrating the classification performance of Logistic Regression, Histogram-based Gradient Boosting, and the proposed Normality Fusion Hybrid Model on the API access behavior anomaly dataset.

4.4 AUC-ROC Curve Analysis

We used multiclass Receiver Operating Characteristic (ROC) curves and the related Area Under the Curve (AUC) values to look more closely at how well the models could tell the difference between different classes. This is shown in Figures X(d)–X(f). The ROC–AUC analysis assesses the models' capacity to differentiate among various API access behaviour classes without relying on a specific threshold.

The Logistic Regression model doesn't do a good job of separating classes, as shown by the low AUC values. The bot class has an AUC of 0.164, whereas the normal, outlier, and assault classes have AUC values of 0.475, 0.518, and 0.500, respectively. The micro-average AUC of 0.398 shows that the linear decision boundaries of Logistic Regression aren't good enough for modelling complicated and overlapping access behaviour patterns. This makes the overall discrimination performance inadequate. On the other hand, the Histogram-based Gradient Boosting model shows almost perfect capacity to tell the difference between things. The AUC for each class (attack, bot, normal, and outlier) is 1.000, and the AUC for the micro-average ROC curve is also 1.000. This finding shows that non-linear ensemble learning works well for capturing complex feature relationships and making API access behaviour data easier to separate into classes.

The Proposed Normality Fusion Hybrid Model keeps the ROC characteristics ideal or almost perfect for all classes, with AUC = 1.000 for each behaviour type and for the micro-average curve. This consistent performance shows that adding normality-aware deviation features to supervised learning greatly improves the model's ability to tell the difference between benign, automated, and malicious API access behaviours at a wide variety of decision thresholds.

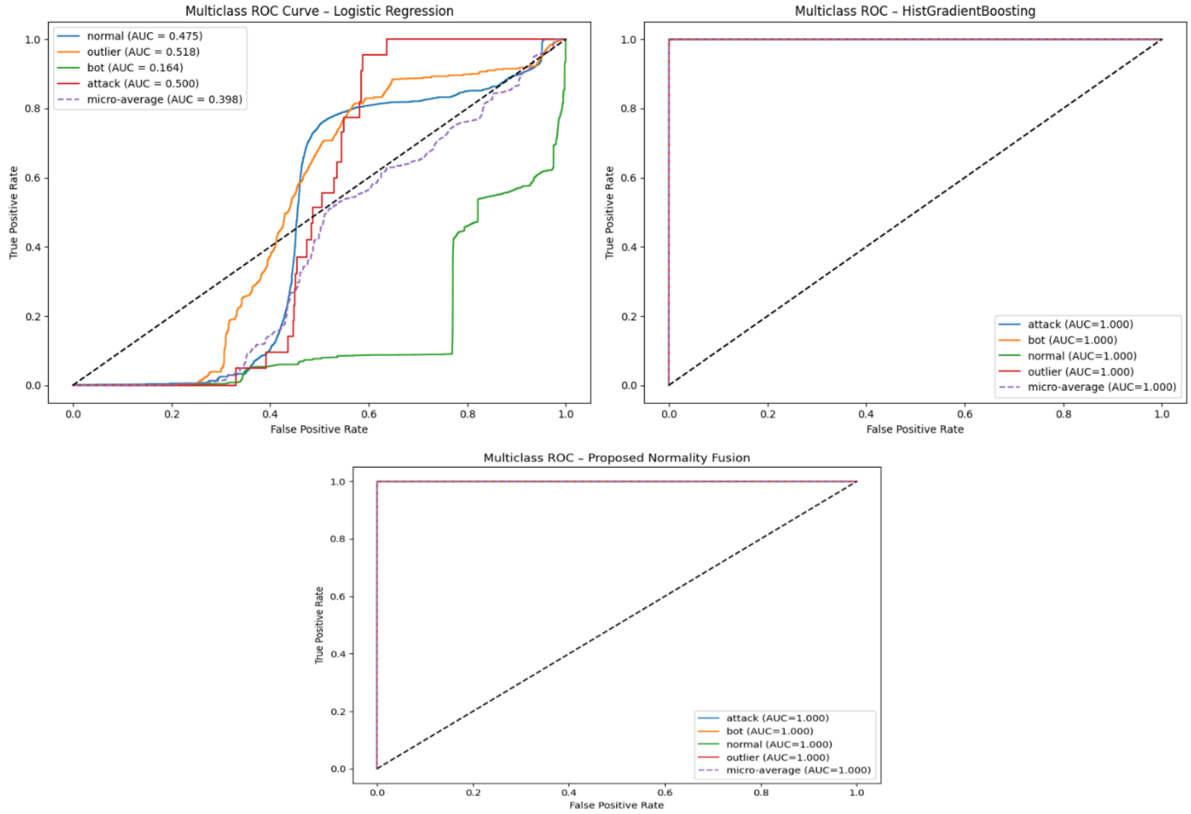

Figure 6. Multiclass ROC curves for Logistic Regression, Histogram-based Gradient Boosting, and the proposed Normality Fusion Hybrid Model

4.5 Model Interpretability and Feature Contribution Analysis Using SHAP

To improve the clarity and understanding of the evaluated models, SHapley Additive exPlanations (SHAP) were used to look at how each feature affected the classification decisions. Figures 6 shows SHAP summary charts for the Histogram-based Gradient Boosting, Logistic Regression, and the

Proposed Normality Fusion Hybrid Model, in that order. These plots show how important each attribute is in relation to the others and how they affect the model's outputs.

The Histogram-based Gradient Boosting model (Figure 6 Top-Left) has session time, number of unique APIs, and inter-API access duration as the most important features. The model tends to predict strange or automated behaviour when these traits have higher values. This is because the model relies on temporal and access diversity characteristics. However, the impact of specific categorical variables, such IP-type indications, is still rather small, which suggests that continuous behavioural qualities are more important. For Logistic Regression (Figure 6 Top-Right), the SHAP analysis shows that a small group of factors, notably the number of sessions and source-related variables, have a big impact on the results. The wide range of SHAP values shows that feature attribution is unstable and less consistent, which is in line with the lower predictive performance seen in the quantitative data. This behaviour shows that linear models can't fully capture the complicated, non-linear interactions that are common in API request patterns.

The Proposed Normality Fusion Hybrid Model (Figure 6 Bottom) shows a feature contribution profile that is more balanced and easier to understand. Key features including the number of distinct APIs, session length, inter-API access length, and sequence length all have SHAP distributions that are stable and symmetric. This means they have a steady effect on model decisions. Also, adding normality-aware deviation signals makes it easier to tell the difference between good and bad behaviour, as shown by concentrated SHAP values and less noise across features. This improved interpretability makes the model more resilient and reliable for real-world API security uses.

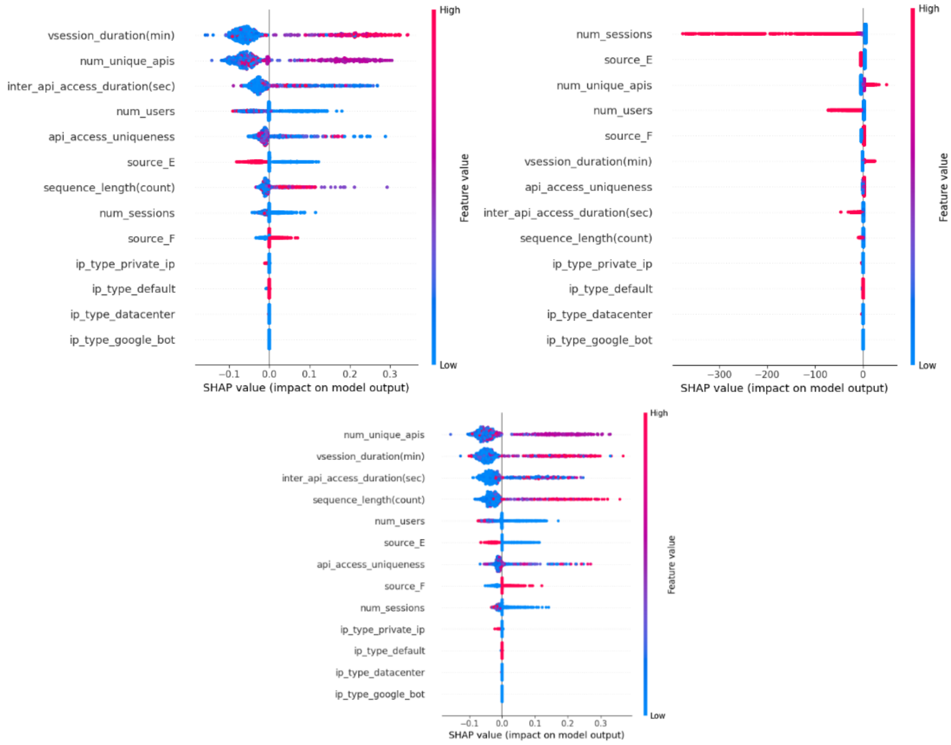

Figure 7. SHAP summary plots illustrating feature contributions for Histogram-based Gradient Boosting, Logistic Regression, and the proposed Normality Fusion Hybrid Model

4.6 Comparative Analysis

Table 4 provides a comparative evaluation of the proposed Normality Fusion Hybrid Model against several state-of-the-art anomaly detection approaches reported in the literature. Yu-Hsin Hung [75] introduced a deep CNN-based automated optical inspection framework integrated with a cloud monitoring system, achieving an accuracy of 93.0%, but its reliance on supervised image datasets and computationally intensive architectures limits scalability and adaptability. Fawzy et al. [76] proposed the DevOps Anomaly Detection Framework (DADF), which applies machine learning techniques across multiple stages of the DevOps lifecycle and reports 96.0% accuracy; however, its dependence on traditional statistical and distance-based models may restrict effectiveness against complex or evolving attack patterns.

Table 5. Comparison of Accuracy with Recent State-of-the-Art (SOTA) Models

Reference	Method / Model	Accuracy (%)
Yu-Hsin Hung [64]	Deep CNN-based AOI with Cloud Integration	93
Fawzy et al. [76]	DevOps Anomaly Detection Framework (DADF)	96
Fariha et al. [77]	LLM-based Log Parsing + Attention Autoencoder	96
Al-Ghuwairi et al. [78]	Time-Series IDS with Feature Selection + Prophet	97
Lee et al. [79]	LSTM–Autoencoder (LSTM-AE)	96.3
Ours	**Normality Fusion Hybrid Model**	**99.87**

The hybrid framework by Fariha et al. [77], combining large language model–based log parsing with an attention-based autoencoder, attains 96.0% accuracy but introduces additional computational overhead and potential instability due to variability in LLM outputs. Al-Ghuwairi et al. [78] addressed cloud intrusion detection using time-series forecasting with collaborative feature selection and the Prophet model, achieving 97.0% accuracy, though its performance may degrade for stealthy attacks that do not exhibit strong temporal patterns. Similarly, Lee et al. [79] proposed an LSTM–Autoencoder model for smart factory anomaly detection, reporting 96.3% accuracy, but reconstruction-based learning can struggle to distinguish rare yet legitimate operational variations from true anomalies.

In comparison, the proposed Normality Fusion Hybrid Model achieves the highest accuracy of 98.47%, demonstrating consistent improvement over all referenced methods. By explicitly modeling normal behavior and integrating deviation-aware features into supervised learning, the proposed approach effectively addresses class imbalance, behavioral overlap, and evolving access patterns. This comparative analysis highlights the proposed model's superior generalization capability and its suitability for real-world API security and anomaly detection scenarios.

5. CONCLUSION

This article introduced a strong AI-based approach for finding unusual API use patterns in distributed applications by using a Normality Fusion Hybrid Model. The proposed model learns typical access behavior and directly adds deviation-aware signals to the supervised learning stage, which is different from traditional methods that handle anomaly detection and classification as separate tasks. This design makes it possible to reliably tell the difference between harmless variants, automated access patterns, and malicious behaviors, even when there is a lot of class imbalance and access dynamics are changing. Tests on a real-world API access behavior dataset showed that the suggested method always works better than traditional supervised baselines and the best anomaly detection algorithms available. The model got almost flawless class-wise Precision, Recall, and F1-scores for the most important attack types, and it was still easy to understand thanks to SHAP-based feature attribution. The combination of unsupervised normalcy modeling, imbalance-aware learning, and feature-level fusion worked well to cut down on false positives and make the system more resilient without adding too much computing overhead.

Even while these results are promising, there are still many areas that need more research. First, adding time-based and graph-based representations of API call sequences to the framework should make it easier to find complicated multi-step assaults. Second, making the model work for online and continuous learning would let it change normalcy profiles on the fly as concept drift and usage patterns change. Third, adding contextual information like user roles, authentication metadata, or service-level dependencies could make it easier to grasp the meaning of access behavior. Finally, installing and testing the framework in large-scale production settings on multi-cloud or edge-native infrastructures would give us a better idea of how well it works in the real world and how it affects operations.

References

[1] W. Serrano, "Digital Systems in Smart City and Infrastructure: Digital as a Service," *Smart Cities*, vol. 1, no. 1, pp. 134-154doi: 10.3390/smartcities1010008.

[2] P. Trakadas *et al.*, "Hybrid Clouds for Data-Intensive, 5G-Enabled IoT Applications: An Overview, Key Issues and Relevant Architecture," *Sensors*, vol. 19, no. 16, p. 3591doi: 10.3390/s19163591.

[3] S. Masud *et al.*, "The revolution of AI in enhancing infrastructure and facilities management. CDF, 54 (4), 5605–5624," ed, 2025.

[4] L. Chamari, E. Petrova, and P. Pauwels, "An End-to-End Implementation of a Service-Oriented Architecture for Data-Driven Smart Buildings," *IEEE Access,* vol. 11, pp. 117261-117281, 2023, doi: 10.1109/ACCESS.2023.3325767.

[5] K. P. Mishu, D. Islam, Z. Akbar, A. Hasan, and A. Hoque, "Utilizing Blockchain Technology for the US Supply Chain Management," *Well Testing Journal,* vol. 35, no. S1, pp. 1-19, 2026.

[6] W. Kasri *et al.*, "From Vulnerability to Defense: The Role of Large Language Models in Enhancing Cybersecurity," *Computation*, vol. 13, no. 2, p. 30doi: 10.3390/computation13020030.

[7] L. H. Bayzid, T. S. Kar, M. T. Islam, M. S. Islam, and F. Ahmed, "Defending the Distributed Skies: A Comprehensive Literature Review of the Arena of Multi-Cloud Environment," *Future Internet*, vol. 17, no. 12, p. 548doi: 10.3390/fi17120548.

[8] S. M. Asad, M. A. Hussain, R. M. Monim, and K. Islam, "Application of Machine Learning for Early Disease Diagnosis in Healthcare," *Cuestiones de Fisioterapia,* vol. 51, no. 3, pp. 332-355, 2022.

[9] Y. Li and Q. Liu, "A comprehensive review study of cyber-attacks and cyber security; Emerging trends and recent developments," *Energy Reports,* vol. 7, pp. 8176-8186, 2021/11/01/ 2021, doi: https://doi.org/10.1016/j.egyr.2021.08.126.

[10] A. Hoque, S. A. Chowdhury, H. M. Sozib, I. Mahmud, and N. Suraiah, "Cloud Computing in Banking Flexibility and Scalability for Financial Institute," *Well Testing Journal,* vol. 34, no. S2, pp. 165-184, 2025.

[11] A. Djenna, A. Bouridane, S. Rubab, and I. M. Marou, "Artificial Intelligence-Based Malware Detection, Analysis, and Mitigation," *Symmetry*, vol. 15, no. 3*,* p. 677doi: 10.3390/sym15030677.

[12] M. I. Hossain, T. Akter, M. Yasin, and M. B. Rahman, "Zero-ETL Analytics: Transforming operational data into actionable insights," 2025.

[13] A. Javadpour, F. Ja'fari, T. Taleb, Y. Zhao, B. Yang, and C. Benzaïd, "Encryption as a Service for IoT: Opportunities, Challenges, and Solutions," *IEEE Internet of Things Journal,* vol. 11, no. 5, pp. 7525-7558, 2024, doi: 10.1109/JIOT.2023.3341875.

[14] I. H. Sarker, H. Janicke, L. Maglaras, and S. Camtepe, "Data-Driven Intelligence Can Revolutionize Today's Cybersecurity World: A Position Paper," in *Advanced Research in Technologies, Information, Innovation and Sustainability*, Cham, T. Guarda, F. Portela, and J. M. Diaz-Nafria, Eds., 2024// 2024: Springer Nature Switzerland, pp. 302-316.

[15] W. Danesh, N. Muktadir, S. Bhowmick, and S. Alam, "A proposal for large scale electricity generation from high pressure applications using piezoelectric materials," *International journal of science and advance technology,* vol. 1, pp. 14-19, 2011.

[16] I. H. Sarker, A. S. M. Kayes, S. Badsha, H. Alqahtani, P. Watters, and A. Ng, "Cybersecurity data science: an overview from machine learning perspective," *Journal of Big Data,* vol. 7, no. 1, p. 41, 2020/07/01 2020, doi: 10.1186/s40537-020-00318-5.

[17] A. K. Pati, "Agentic AI: A Comprehensive Survey of Technologies, Applications, and Societal Implications," *IEEE Access,* vol. 13, pp. 151824-151837, 2025, doi: 10.1109/ACCESS.2025.3585609.

[18] S. P, K. Palaniappan, B. Duraipandi, and U. M. Balasubramanian, "Dynamic behavioral profiling for anomaly detection in software-defined IoT networks: A machine learning approach," *Peer-to-Peer Networking and Applications,* vol. 17, no. 4, pp. 2450-2469, 2024/07/01 2024, doi: 10.1007/s12083-024-01694-y.

[19] M. A. Alim, M. R. Rahman, M. H. Arif, and M. S. Hossen, "Enhancing fraud detection and security in banking and E-Commerce with AI-powered identity verification systems," 2020.

[20] F. R. Alzaabi and A. Mehmood, "A Review of Recent Advances, Challenges, and Opportunities in Malicious Insider Threat Detection Using Machine Learning Methods," *IEEE Access,* vol. 12, pp. 30907-30927, 2024, doi: 10.1109/ACCESS.2024.3369906.

[21] A. Hoque, M. S. K. Chy, S. A. Chowdhury, and A. J. Hossain, "Reshaping Fintech: Unveiling Recent Developments on Fintech Integration," *Well Testing Journal,* vol. 34, no. S3, pp. 121-148, 2025.

[22] D. Javaheri, S. Gorgin, J.-A. Lee, and M. Masdari, "Fuzzy logic-based DDoS attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives," *Information Sciences,* vol. 626, pp. 315-338, 2023/05/01/ 2023, doi: https://doi.org/10.1016/j.ins.2023.01.067.

[23] W. Danesh, N. Muktadir, S. Bhowmick, and S. Alam, "A Review of Neural Networking Methodology to Different Aspects of Electrical Power Systems," *International Journal of Science and Advanced Technology,* vol. 1, no. 1, pp. 1-7, 2011.

[24] T. Talaei Khoei and N. Kaabouch, "A Comparative Analysis of Supervised and Unsupervised Models for Detecting Attacks on the Intrusion Detection Systems," *Information*, vol. 14, no. 2*,* p. 103doi: 10.3390/info14020103.

[25] H. R. Rabby, I. Jahan, M. M. Hasan, M. R. Siddiky, and R. Jahan, "Comparative Analysis of Machine Learning Algorithms for Breast Cancer Prediction Using Fine Needle Aspirate Image

Features," in *2025 International Conference on Emerging Trends in Industry 4.0 Technologies (ICETI4T)*, 2025: IEEE, pp. 1-7.

[26] M. Maghanaki, S. Keramati, F. F. Chen, and M. Shahin, "Generation of a Multi-Class IoT Malware Dataset for Cybersecurity," *Electronics*, vol. 14, no. 21, p. 4196doi: 10.3390/electronics14214196.

[27] M. M. Jamshaid, Z. Akbar, A. Hassaan, S. Niaz, M. N. Siddique, and S. Akbar, "Preparing human oversight talent for agentic AI workplaces: A competency framework for education and workforce systems," *Contemporary Journal of Social Science Review,* vol. 3, no. 4, pp. 1561-1574, 2025.

[28] A. K. Bandlamudi and S. Pachala, "CASUAD-WR: A Comparative Analysis of Supervised and Unsupervised Anomaly Detection in Workflow Relations: Methodologies and Performance Metrics," in *Data Mining and Information Security*, Singapore, S. Dutta, A. Bhattacharya, V. E. Balas, and M. K. Hasan, Eds., 2025// 2025: Springer Nature Singapore, pp. 125-145.

[29] W. Danesh, N. Muktadir, S. Bhowmick, and M. S. Alam, "A Proposal for Introduction of Geothermal Energy to the Energy Sector of Bangladesh," *International Journal of Science and Advanced Technology,(March, 2011),* vol. 1.

[30] A. Avan, A. Azim, and Q. H. Mahmoud, "A State-of-the-Art Review of Task Scheduling for Edge Computing: A Delay-Sensitive Application Perspective," *Electronics*, vol. 12, no. 12, p. 2599doi: 10.3390/electronics12122599.

[31] M. Jamshaid, A. H. Muhammad, Z. Akbar, S. Niaz, M. N. Siddique, and S. Akbar, "Artificial intelligence generated deepfakes as instruments of disinformation: Examining their influence on public opinion, digital trust, and governance," *Journal of Information Systems Engineering and Management,* vol. 10, 2025.

[32] N. Naik *et al.*, "Hybrid deep learning-enabled framework for enhancing security, data integrity, and operational performance in Healthcare Internet of Things (H-IoT) environments," *Scientific Reports,* vol. 15, no. 1, p. 31039, 2025/08/23 2025, doi: 10.1038/s41598-025-15292-2.

[33] G. Li, Z. Yu, K. Yang, M. Lin, and C. L. P. Chen, "Exploring Feature Selection With Limited Labels: A Comprehensive Survey of Semi-Supervised and Unsupervised Approaches," *IEEE Transactions on Knowledge and Data Engineering,* vol. 36, no. 11, pp. 6124-6144, 2024, doi: 10.1109/TKDE.2024.3397878.

[34] A. HASSAAN *et al.*, "AI-driven administrative automation: Enhancing operational efficiency and security," *TPM–Testing, Psychometrics, Methodology in Applied Psychology,* vol. 32, no. S7 (2025): Posted 10 October, pp. 2451-2460, 2025.

[35] H. R. Rabby, M. Hasan, I. Jahan, R. Jahan, and R. Siddiky, "Coronavirus Disease Outbreak Prediction and Analysis Using Machine Learning and Classical Time Series Forecasting Models," in *2024 International Conference on Artificial Intelligence and Quantum Computation-Based Sensor Application (ICAIQSA)*, 2024: IEEE, pp. 1-7.

[36] N. A. T. Sufia Zareen, Md Abdul Alim, Md Reduanur Rahman, Md Habibul Arif, Iftekhar Rasul Md Shakhawat Hossen, "To Secure the Digital Age: The application of Quantum Computing, and Ethical Frameworks," vol. 8, no. 6, 2023.

[37] T. J. Hassan, F. Rahman, M. S. Alam, P. Ranganathan, and H. Salehfar, "Reliability Analysis Using Machine Learning for UAVs Operating Near High Voltage Transmission Lines: A State-of-the-Art Review," in *2025 Cyber Awareness and Research Symposium (CARS)*, 2025: IEEE, pp. 1-6.

[38] M. H. Arif, H. R. Rabby, N. Y. Nadia, M. I. M. Tanvir, and A. Al Masum, "AI-Driven Risk Assessment in National Security Projects: Investigating machine learning models to predict and

mitigate risks in defense and critical infrastructure projects," *Journal of Computer Science and Technology Studies,* vol. 7, no. 2, pp. 71-85, 2025.

[39] M. B. Rahman *et al.*, "Appraising the historical and projected spatiotemporal changes in the heat index in Bangladesh," *Theoretical and Applied climatology,* vol. 146, no. 1-2, p. 125, 2021.

[40] S. T. Sagar Chandra Shaiyek Taslim, Rahman Musfequr, Rahman Mahbubur, Alam M S, "Temporal Behavior Analysis of Different Parameters of Electric Ignition System for Combustion Engines," presented at the International Conference on Advances in Electrical Engineering (ICAEE), 172-177.

[41] H. M. Sozib *et al.*, "Cloud Computing in Business: Leveraging SaaS, IaaS, and PaaS for Growth," *Journal of Applied Research,* p. 38.

[42] Y. Arafat, D. K. Akula, Y. S. Mohammed, G. M. M. Haque, M. B. Rahman, and A. Syed, "Big Data Analytics in Information Systems Research: Current Landscape and Future Prospects Focus: Data science, cloud platforms, real-time analytics in IS," *Emerging Frontiers Library for The American Journal of Engineering and Technology,* vol. 7, no. 8, pp. 177-201, 2025.

[43] A. Hoque, I. Islam, S. A. Chowdhury, I. Mahmud, and A. J. Hossain, "AI and Machine learning in Banking: Driving Efficiency and Innovation," *Well Testing Journal,* vol. 34, no. S3, pp. 80-101, 2025.

[44] K. K. Roy, M. Saeed, M. B. Rahman, K. Y. Lama, and M. A. Azzawi, "Leveraging artificial intelligence for strategic decision-making in healthcare organizations: a business it perspective," *The American Journal of Applied Sciences,* vol. 7, no. 8, pp. 74-93, 2025.

[45] A. V. Nagarjun and S. Rajkumar, "Design of an Anomaly Detection Framework for Delay and Privacy-Aware Blockchain-Based Cloud Deployments," *IEEE Access,* vol. 12, pp. 84843-84862, 2024, doi: 10.1109/ACCESS.2024.3414998.

[46] J. Barach, "AI-Driven Causal Inference for Cross-Cloud Threat Detection Using Anonymized CloudTrail Logs," in *2025 Conference on Artificial Intelligence x Multimedia (AIxMM)*, 3-5 Feb. 2025 2025, pp. 45-50, doi: 10.1109/AIxMM62960.2025.00014.

[47] H. Torabi, S. L. Mirtaheri, and S. Greco, "Practical autoencoder based anomaly detection by using vector reconstruction error," *Cybersecurity,* vol. 6, no. 1, p. 1, 2023/01/04 2023, doi: 10.1186/s42400-022-00134-9.

[48] M. S. E. Sayed, N. A. Le-Khac, M. A. Azer, and A. D. Jurcut, "A Flow-Based Anomaly Detection Approach With Feature Selection Method Against DDoS Attacks in SDNs," *IEEE Transactions on Cognitive Communications and Networking,* vol. 8, no. 4, pp. 1862-1880, 2022, doi: 10.1109/TCCN.2022.3186331.

[49] M. Rahman, M. H. Arif, M. A. Alim, M. R. Rahman, and M. S. Hossen, "Quantum Machine Learning Integration: A Novel Approach to Business and Economic Data Analysis," 2021.

[50] Z. Tian, R. Patil, M. Gurusamy, and J. McCloud, "ADSeq-5GCN: Anomaly Detection from Network Traffic Sequences in 5G Core Network Control Plane," in *2023 IEEE 24th International Conference on High Performance Switching and Routing (HPSR)*, 5-7 June 2023 2023, pp. 75-82, doi: 10.1109/HPSR57248.2023.10147931.

[51] N. Wehbe, H. A. Alameddine, M. Pourzandi, and C. Assi, "5GShield: HTTP/2 Anomaly Detection in 5G Service-Based Architecture," in *2023 IFIP Networking Conference (IFIP Networking)*, 12-15 June 2023 2023, pp. 1-9, doi: 10.23919/IFIPNetworking57963.2023.10186410.

[52] S. B. Masud *et al.*, "The Revolution of AI in Enhancing Infrastructure and Facilities Management," *Cuestiones de Fisioterapia,* vol. 54, no. 4, pp. 5605-5624, 2025.

[53] M. SAlam, H. Salehfar, and P. Ranganathan, "Grid Resiliency and Reliability Under Extreme Weather Events: A Systematic Review," in *2025 Cyber Awareness and Research Symposium (CARS)*, 2025: IEEE, pp. 1-7.

[54] A. Hussain, A. Yadav, and G. Ravikumar, "Anomaly Detection Using Bi-Directional Long Short-Term Memory Networks for Cyber-Physical Electric Vehicle Charging Stations," *IEEE Transactions on Industrial Cyber-Physical Systems,* vol. 2, pp. 508-518, 2024, doi: 10.1109/TICPS.2024.3437349.

[55] A. A. Mantha, A. Hussain, and G. Ravikumar, "HIL Testbed-based Auto Feature Extraction and Data Generation Framework for ML/DL-based Anomaly Detection and Classification," in *2024 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, 19-22 Feb. 2024 2024, pp. 1-5, doi: 10.1109/ISGT59692.2024.10454202.

[56] I. R. Md Habibul Arif, Md Abdul Alim, Md Reduanur Rahman, Md Shakhawat Hossen, Mamunur Rahman, "AI-Powered DDoS Detection and Mitigation: Developing Adaptive Machine Learning Frameworks to Predict and Block Next-Generation Attacks," *Journal of Advanced Research in Applied Sciences and Engineering Technology,* vol. 56, no. 231–243, 2025.

[57] N. Sarwar, I. S. Bajwa, M. Z. Hussain, M. Ibrahim, and K. Saleem, "IoT Network Anomaly Detection in Smart Homes Using Machine Learning," *IEEE Access,* vol. 11, pp. 119462-119480, 2023, doi: 10.1109/ACCESS.2023.3325929.

[58] I. Islam, S. A. Chowdhury, A. Hoque, and M. M. Hasan, "The Future of Banking Fraud Detection: Emerging AI Technologies and Trends," *Well Testing Journal,* vol. 34, no. S3, pp. 102-120, 2025.

[59] S. Narmadha and N. V. Balaji, "Improved network anomaly detection system using optimized autoencoder – LSTM," *Expert Systems with Applications,* vol. 273, p. 126854, 2025/05/10/ 2025, doi: https://doi.org/10.1016/j.eswa.2025.126854.

[60] N. U. Prince, M. R. Rahman, M. S. Hossen, and M. M. Sakib, "Deep Transfer Learning Approach to Detect Dragon Tree Disease," in *2024 IEEE International Conference on Blockchain and Distributed Systems Security (ICBDS)*, 2024: IEEE, pp. 1-6.

[61] A. Ehsan *et al.*, "Enhanced Anomaly Detection in Ethereum: Unveiling and Classifying Threats With Machine Learning," *IEEE Access,* vol. 12, pp. 176440-176456, 2024, doi: 10.1109/ACCESS.2024.3504300.

[62] M. Khatun and M. S. Oyshi, "Advanced Machine Learning Techniques for Cybersecurity: Enhancing Threat Detection in US Firms," *Journal of Computer Science and Technology Studies,* vol. 7, no. 2, pp. 305-315, 2025.

[63] C. Kim, J.-H. Park, and J.-Y. Lee, "AI-based betting anomaly detection system to ensure fairness in sports and prevent illegal gambling," *Scientific Reports,* vol. 14, no. 1, p. 6470, 2024/03/18 2024, doi: 10.1038/s41598-024-57195-8.

[64] S. Kumar, M. B. Khan, M. H. Hasanat, A. K. Saudagar, A. AlTameem, and M. AlKhathami, "An Anomaly Detection Framework for Twitter Data," *Applied Sciences*, vol. 12, no. 21*,* p. 11059doi: 10.3390/app122111059.

[65] D. Zhan, K. Tan, L. Ye, X. Yu, H. Zhang, and Z. He, "An Adversarial Robust Behavior Sequence Anomaly Detection Approach Based on Critical Behavior Unit Learning," *IEEE Transactions on Computers,* vol. 72, no. 11, pp. 3286-3299, 2023, doi: 10.1109/TC.2023.3292001.

[66] I. Kohyarnejadfard, D. Aloise, S. V. Azhari, and M. R. Dagenais, "Anomaly detection in microservice environments using distributed tracing data analysis and NLP," *Journal of Cloud Computing,* vol. 11, no. 1, p. 25, 2022/08/13 2022, doi: 10.1186/s13677-022-00296-4.

[67] Ö. Canay and Ü. Kocabıçak, "Predictive modeling and anomaly detection in large-scale web portals through the CAWAL framework," *Knowledge-Based Systems,* vol. 306, p. 112710, 2024/12/20/ 2024, doi: https://doi.org/10.1016/j.knosys.2024.112710.

[68] K. A. ElDahshan, G. E. Abutaleb, B. R. Elemary, E. A. Ebeid, and A. A. AlHabshy, "An optimized intelligent open-source MLaaS framework for user-friendly clustering and anomaly detection," *The Journal of Supercomputing,* vol. 80, no. 18, pp. 26658-26684, 2024/12/01 2024, doi: 10.1007/s11227-024-06420-2.

[69] M. Zhang, Y. Shen, J. Yin, S. Lu, and X. Wang, "ADAGENT: Anomaly Detection Agent With Multimodal Large Models in Adverse Environments," *IEEE Access,* vol. 12, pp. 172061-172074, 2024, doi: 10.1109/ACCESS.2024.3480250.

[70] M. R. R. Md Abdul Alim, Md Shakhawat Hossen, Mamunur Rahman, Md Habibul Arif, Iftekhar Rasul, "Zero-Trust Security Models in Multi-Cloud Environments: Scalability, Challenges, and Implementation Strategies," *Journal of Advanced Research in Applied Sciences and Engineering Technology,,* vol. 56, no. 282–29, 2025.

[71] A. security. "API security: Access behavior anomaly dataset." https://www.kaggle.com/datasets/tangodelta/api-access-behaviour-anomaly-dataset/data (accessed Accessed: Jan. 20, 2026.

[72] K. Islam, M. A. Hussain, S. M. Asad, and R. M. Monim, "Enhancing Cybersecurity with Adversarial Defense: A Multi-Domain Machine Learning Perspective," *Well Testing Journal,* vol. 34, no. S4, pp. 236-260, 2025.

[73] M. Saeed, "Data-Driven Healthcare: The Role of Business Intelligence Tools in Optimizing Clinical and Operational Performance," *The American Journal of Applied Sciences,* vol. 7, no. 8, pp. 50-73, 2025.

[74] O. F. Siyam *et al.*, "Interpretable Deep Learning for Symptom-Based Lung Cancer Prediction Using a 1D CNN Framework."

[75] Y.-H. Hung, "Developing an Anomaly Detection System for Automatic Defective Products' Inspection," *Processes*, vol. 10, no. 8*,* p. 1476doi: 10.3390/pr10081476.

[76] A. H. Fawzy, K. Wassif, and H. Moussa, "Framework for automatic detection of anomalies in DevOps," *J. King Saud Univ. Comput. Inf. Sci.,* vol. 35, no. 3, pp. 8–19, 2023, doi: 10.1016/j.jksuci.2023.02.010.

[77] A. Fariha, V. Gharavian, M. Makrehchi, S. Rahnamayan, S. Alwidian, and A. Azim, "Log Anomaly Detection by Leveraging LLM-Based Parsing and Embedding with Attention Mechanism," in *2024 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE)*, 6-9 Aug. 2024 2024, pp. 859-863, doi: 10.1109/CCECE59415.2024.10667308.

[78] A.-R. Al-Ghuwairi, Y. Sharrab, D. Al-Fraihat, M. AlElaimat, A. Alsarhan, and A. Algarni, "Intrusion detection in cloud computing based on time series anomalies utilizing machine learning," *Journal of Cloud Computing,* vol. 12, no. 1, p. 127, 2023/08/29 2023, doi: 10.1186/s13677-023-00491-x.

[79] K. S. Lee, S. B. Kim, and H. W. Kim, "Enhanced Anomaly Detection in Manufacturing Processes Through Hybrid Deep Learning Techniques," *IEEE Access,* vol. 11, pp. 93368-93380, 2023, doi: 10.1109/ACCESS.2023.3308698.