HADVB: An Interpretable and Auditable Intrusion Detection Framework for Multimodal Industrial Communication

Gang Han, Xuan Chang*, Yaotian Wang, Zhongliang Zhang*

Xi'an University of Posts and Telecommunications, Xi'an 710121, China

Keywords: Multimodal; Intrusion detection; Machine learning; IoT; Data security

Citation: Han G., Chang X., Wang Y.T., et al.: HADVB: An Interpretable and Auditable Intrusion Detection Framework for Multimodal Industrial Communication. Data Intelligence, 2026, Vol. 8, Art. No.: 20260088. DOI: https://doi.org/10.3724/2096-7004.di.2026.0088

ABSTRACT

In the context of the continuous evolution of the Industrial Internet of Things (IIoT), network intrusion behaviors have become increasingly complex. These behaviors involve heterogeneous data sources, class imbalance, and attack variation. Such characteristics pose significant challenges to traditional intrusion detection methods. To address these challenges, this paper presents a multimodal intrusion detection framework named HADVB. The framework combines communication traffic data and node behavior logs. It applies a unified preprocessing pipeline to improve input consistency and stability. It also introduces a hybrid resampling strategy based on clustering, integrating HCBOU and ADASYN, to mitigate the impact of imbalanced class distribution. For feature learning, the system employs a denoising autoencoder to compress highdimensional features. The AdamW optimizer improves training robustness and generalization. In the classification phase, the framework employs a soft voting model that integrates Random Forest, XGBoost, and MLP to enhance detection performance across diverse attack scenarios. To improve model transparency and trustworthiness, the system incorporates SHAP and LIME for decision interpretability. It also employs structured logs and summary-based signatures to record and audit each inference in a chained manner. Experiments on three public datasets—WSN-DS, WSNBFSF, and InSDN—show that the proposed method achieves 99.03%, 99.69%, and 99.97% accuracy, respectively. It also demonstrates strong performance in detecting minority-class attacks such as Forwarding and Blackhole. Overall, HADVB establishes a systematic approach that integrates multimodal fusion, class imbalance handling, interpretability, and

*Corresponding authors: Zhongliang Zhang (E-mail: zlzhang2525@163.com; ORCID: 0000-0001-6555-7908); Xuan Chang (E-mail: 2923547941@qq.com; ORCID: 0009-0004-2775-3536).

© 2026 Chinese Academy of Sciences. Published under a Creative Commons Attribution 4.0 International (CC BY 4.0) license.

auditable inference. It provides practical support for building reliable intrusion detection systems in industrial environments.

1. INTRODUCTION

1.1 Research Background

The rapid rise of the Internet of Things (IoT) is driving digital transformation across smart manufacturing, connected vehicles, and remote healthcare. By linking physical devices with digital systems, IoT enhances sensing, autonomous decision-making, and coordination, improving efficiency and enabling intelligent infrastructure.

At the same time, widespread IoT adoption exposes serious security risks. Highly heterogeneous architectures—spanning device types, protocols, environments, and network topologies [1]—plus resource-constrained nodes and frequent real-time communications enlarge the attack surface, increasing susceptibility to DDoS, code injection, buffer overflows, privilege escalation, false routing, and data tampering. In critical sectors such as healthcare, energy, and transportation, compromises can disrupt services and even threaten national stability and strategic interests. IoT systems usually generate large volumes of multimodal and heterogeneous data. These include sensor readings, traffic flows, behavior logs, and control commands. Such data reflect system status and often contain early indicators of anomalies. They are essential for intrusion detection and security strategy formulation. However, traditional methods that rely on single-modality inputs and static rules show significant limitations when facing complex, dynamic, and cross-modal attacks.

These methods fail to meet the requirements of practical IoT security systems [2-3]. Relevant lessons from recent multimodal video understanding further motivate the proposed design: micro-action recognition shows that subtle, low-intensity behaviors demand fine-grained spatiotemporal modeling and robust shared representations [4], while partially relevant video retrieval demonstrates the need to reason over multi-scale temporal units and handle “partial relevance” via multiple-instance formulations when only segments carry discriminative evidence [5]. By analogy, IoT telemetry often exhibits weak, transient attack cues and segment-level relevance across modalities; therefore an effective IDS should align heterogeneous modalities, model fine-scale temporal dynamics, and focus attention on informative segments rather than entire streams.

1.2 Challenges and Problems

AI-based intrusion detection systems (IDS) are a major focus of IoT security, using deep learning, transfer learning, and graph neural networks to analyze device behavior, network traffic, and multimodal data for real-time detection of known and unknown attacks [6-7]. Yet key obstacles remain: severe class imbalance in IoT traffic (abundant normal vs. scarce malicious samples), challenging multimodal fusion due to structural differences, semantic gaps, and asynchronous timing, limited interpretability of deep

models, and weak generalization/adaptability to rapidly evolving attacks. These gaps motivate IDSs that are transferable, interpretable, and adaptable [8].

IDS continuously monitors networks to flag anomalies and threats. Rule-and signature-based baselines work well for known attacks—fast and with low false positives—but struggle with variants, zero-days, and distributed composite attacks. They depend on expert updates, respond slowly, and often rely on single-modality inputs lacking behavioral context, which undermines generalization and robustness in practical IoT deployments.

1.3 Research Techniques and Innovations

Multimodal learning [9] offers a promising path for AI-IDS. IoT behaviors are naturally multimodal: low-level traffic captures communication structure, while higher-level logs and control signals encode semantics and intent. Combining raw traffic, protocol content, and structured behavior logs yields more complete attack models. Within a unified feature space, multimodal representations improve anomaly detection and robustness to high-dimensional noise and stealthy attacks. Fusion methods—attention, contrastive learning, and shared embeddings—further capture cross-modal correlations and deviations, boosting accuracy and interpretability for traceable, adaptive defenses.

To address redundancy, dimensional mismatch, and noise in multimodal data, this paper introduces a denoising autoencoder (DAE) [10] as a feature encoder that learns shared structure self-supervisedly, compresses inputs while preserving semantics, and suppresses modality-specific noise. At the data level, this paper designs Hybrid Clustering-Based Over- and Under-sampling (HCBOU) [11]—cluster-aware over/under-sampling with intra-cluster synthesis for minority classes and downsampling for majority classes—augmented by Adaptive Synthetic Sampling Approach (ADASYN) to adapt distributions; this mitigates imbalance bias. For classification, a weighted soft-voting ensemble of Random Forest, XGBoost, and Multi-Layer Perceptron (MLP) leverages complementary inductive biases and reduces overfitting risk. To improve transparency, a blockchain-based audit mechanism [12] enforces compliance thresholds, autoevaluates after each inference, and immutably records results with visual summaries, supporting traceable and auditable lifecycle management in security-critical IoT settings.

In summary, this study proposes an integrated intrusion detection system named HADVB. The system addresses key challenges in IoT security, including noisy input data, feature redundancy, scarce attack samples, and lack of model trust. It offers a complete detection and decision pipeline from feature modeling to performance auditing. The system consists of the following technical modules:

- **Denoising Autoencoder**
- **HCBOU + ADASYN Sampling Mechanism**
- **Multimodal Feature Fusion**
- **Voting Classifier**
- **Consortium Blockchain Audit and Visualization**

With the coordinated design of the above modules, HADVB achieves an end-to-end intrusion detection workflow. This workflow integrates multimodal feature extraction, classification decisionmaking, data enhancement, and model verifiability. The system significantly improves detection of complex and low-frequency attacks. It also enhances robustness and generalization in typical IoT scenarios characterized by noisy data, dynamic topologies, and class imbalance. Furthermore, the blockchain-based audit mechanism strengthens the trustworthiness and traceability of the system in realworld deployments. This approach addresses the critical gap in current methods, which often lack verifiable outputs and external accountability.

The main contributions of this work are summarized as follows:

1) Intrusion Detection Framework with Multimodal Features: To address the heterogeneity of IoT modalities and the limited expressiveness of single features, this study proposes a multimodal input structure. It combines statistical traffic data and semantic node behavior descriptions to enhance the model's understanding and representation of complex attack behaviors.

2) HCBOU + ADASYN Data Augmentation Mechanism: This paper introduces a hybrid sampling strategy named HCBOU, which is based on cluster-guided intra-class synthesis and representative under-sampling. When combined with ADASYN, it effectively alleviates the scarcity of minority-class samples and addresses class imbalance in IoT intrusion detection.

3) Multimodal Denoising Autoencoder for Feature Extraction: A denoising autoencoder (DAE) is used to perform robust representation learning on fused multimodal inputs. It suppresses noise interference and enhances the model's ability to detect abnormal behaviors, especially under high-dimensional and complex data conditions.

4) Soft Voting Classifier with Multi-Model Cooperation: The detection module integrates Random Forest, XGBoost, and MLP in a weighted soft voting scheme. This ensemble leverages the diverse decision capabilities of individual models and improves overall classification performance while reducing the risk of overfitting.

5) Consortium Blockchain-Based Auditing Mechanism: A lightweight audit framework is proposed based on hash digest recording and threshold-based performance verification. The mechanism enables automatic validation of evaluation results and ensures their immutability through blockchain logging. This enhances the credibility and traceability of the deployed model.

2. RELATED WORK

In recent years, with the continuous evolution of cyber threats and the increasing complexity of IoT environments, researchers have paid growing attention to the use of machine learning techniques for building and improving intrusion detection systems (IDS). Many effective theories and methods have been proposed. These include advances in feature selection, model architectures, data augmentation, and anomaly detection. Such efforts have contributed to significant improvements in IDS performance. However, most existing approaches still rely heavily on single-modality inputs, such as network traffic

features or static protocol fields. These models often fall short when dealing with the practical challenges of device heterogeneity, dynamic behavior, and multimodal data fusion in IoT scenarios. Limitations remain in terms of detection accuracy, generalization ability, and responsiveness to unknown threats.

To address these challenges, recent studies have begun to explore the integration of multimodal information. This strategy aims to enhance the semantic modeling of complex attack behaviors. It typically involves the combination of structured data and unstructured information. These methods attempt to represent threats from multiple dimensions. Nevertheless, multimodal learning in IDS remains at an early stage. Open issues persist in modeling cross-modal consistency, designing effective fusion strategies, and ensuring robustness under real-world conditions. To clarify the technical foundation and innovation of this work, the following sections review representative studies from four key perspectives: multimodal representation learning, deep feature encoding, class imbalance handling, and the integration of verifiable trust mechanisms.

2.1 Multimodal Modeling

Recent studies on multimodal modeling have explored heterogeneous data integration, graph-based structures, transfer learning, and cross-domain fusion. These approaches aim to strengthen intrusion detection by improving accuracy, efficiency, and robustness in IoT security and related applications.

Sangameshwar B. et al [13] combined blockchain with multimodal analysis and proposed a malware detection framework for Android-based IoT environments. Their approach integrates structured traffic features with application-level behavioral information, optimized through clustering and dimensionality reduction. Smart contracts automate detection policies, while blockchain ensures immutability and traceability. The study demonstrates improved detection accuracy and system responsiveness, offering a trustworthy multimodal defense paradigm. Rozhin Yasaei et al [14] designed IoT-GRAF, a graph-based multimodal learning framework that fuses sensor-layer and communication-layer data in IoT environments. By constructing a heterogeneous graph and applying graph neural networks (GNNs), the system captures complex cross-layer semantics for anomaly detection and source tracing. Evaluations on real-world datasets confirm superior detection performance over single-modal methods and practical millisecond-level inference latency in fog computing scenarios.

To handle large-scale threats such as flooding and DDoS attacks, Ullah F et al [15] introduced an enhanced intrusion detection system that integrates multimodal big data representation with transfer learning. The model fuses semantic and visual features extracted from PCAP traffic, represented via word2vec embeddings and ResNet-based attention. These are classified using a CNN-LSTM model, with Spark preprocessing and game-theoretic validation improving robustness. The method achieves 98.2% accuracy on multiple datasets, highlighting the strength of combining multimodal fusion with transfer learning for high-intensity attack detection. Reis MJCS et al [16] emphasized multimodal fusion for intelligent decision-making in cooperative control systems. Their framework unifies heterogeneous streams from physical environments, vehicle behaviors, and real-time traffic feedback. This approach enhances

the understanding and prediction of urban dynamics, mitigates the weaknesses of single-source data, and improves model stability and generalization in real-world deployment scenarios.

2.2 Deep Feature Learning

As IoT technologies become deeply embedded in urban infrastructure, their systemic security risks have grown increasingly prominent. The high interconnectivity and multimodal heterogeneity of IoT systems offer intelligent capabilities but also expose them to frequent cyberattacks. Studies have shown that traditional security mechanisms, such as authentication, access control, and encryption, are often difficult to deploy effectively in resource-constrained, dynamic, and distributed environments. This situation calls for more intelligent and adaptable defense strategies.

In this context, machine learning (ML) and deep learning (DL) have been widely applied to IoT security. These data-driven techniques are now considered key enablers for improving intrusion detection system (IDS) performance. ML/DL methods based on behavioral modeling can automatically learn normal communication patterns between devices and identify potential anomalies or attacks. This allows for real-time system awareness and rapid threat response. Kiflay A et al [17] proposed a multimodal network intrusion detection system (NIDS) that combines traffic features with protocol payload information. The framework trains two separate Random Forest models and fuses their outputs via soft voting. On the UNSW-NB15 dataset, the method achieved 98% to 99% detection accuracy using only six flow features and the first 32 bytes of payload. This demonstrates strong generalization and deployment adaptability in IoT environments, and highlights the benefits of multimodal fusion in intrusion detection.

Wagan S A et al [18] addressed the security challenges of the Internet of Medical Things (IoMT) by introducing Duo-Secure, a multimodal intrusion detection framework. The system integrates multisource sensor data to distinguish between normal and malicious communication. It combines dynamic fuzzy C-means clustering with a custom Bi-LSTM model to perform behavioral modeling and anomaly detection on medical data. On a heart disease dataset with 36 attributes, the framework achieved a health prediction accuracy of 92.95% and a multimodal attack detection accuracy of 89.67%. These results validate its effectiveness in ensuring both safety and continuity in IoMT systems. Attar H et al [19] further explored key areas such as pretrained model transfer, abnormal traffic feature extraction, and semantic modeling of multimodal data. Their work supports the development of intelligent and adaptive security frameworks for IoT. However, despite these advances, practical deployments still face challenges such as limited model interpretability, data distribution shifts, and rapidly evolving attack variants. Future research must address robustness, scalability, and trust to enable a shift from passive defense to intelligent, coordinated, and proactive protection in IoT systems.

2.3 Data Imbalance

To address challenges in multimodal intrusion detection, researchers have proposed various innovative approaches from the perspectives of model transfer, feature reconstruction, and ensemble learning. These efforts continue to expand the technical landscape of IoT security.

Talukder et al [20] introduced a machine learning-based IDS model that integrates random oversampling, cluster-based feature embedding, and principal component analysis (PCA). The method aims to address large-scale and class-imbalanced intrusion detection scenarios. It achieved over 99.9% accuracy on UNSW-NB15, CIC-IDS-2017, and CIC-IDS-2018 datasets. Despite its strong performance, the model's scalability and computational efficiency in real-world settings still require further validation. Hossain et al [21] proposed an ensemble learning-based IDS framework. It combines multiple classifiers—such as Random Forest, XGBoost, and Bagging—with feature selection strategies. The system performs well across several evaluation metrics, including accuracy, F1-score, and Cohen's Kappa. It demonstrates particular strength in identifying complex and previously unseen threats.

Nabi et al [22] explored the role of dimensionality reduction in intrusion detection. Using the NSLKDD dataset, they evaluated several supervised learning algorithms. The results show that random projection outperforms PCA in terms of training speed and classification accuracy. The PART algorithm achieved 82.0% accuracy after dimensionality reduction, highlighting the potential of lightweight feature compression in practical deployments. Paya et al [23] developed Apollon, a robust defense framework against adversarial machine learning attacks in IDS systems. The framework employs multiple classifiers for intrusion recognition and dynamically selects the best classifier or ensemble using a multi-armed bandit (MAB) strategy with Thompson sampling. Experimental results show that Apollon can detect attacks effectively without compromising performance on standard traffic. It achieved over 99% accuracy on CIC-IDS-2017 and CIC-DDoS-2019 datasets.

Talukder et al [24] further proposed MLSTL-WSN, an intrusion detection method tailored for wireless sensor networks (WSN). The model integrates machine learning with the SMOTE-TomekLink algorithm to address class imbalance, a common issue in WSN datasets. By synthesizing minority class instances and removing Tomek links, the method balances the dataset and improves detection accuracy. Feature standardization ensures consistency and scalability. On the WSN-DS dataset, MLSTL-WSN achieved 99.78% accuracy in binary classification and 99.92% in multi-class settings. These results significantly outperform existing methods and demonstrate strong effectiveness in identifying and mitigating WSN-based intrusions.

2.4 Integration of Verifiable Trust Mechanisms

To strengthen verifiable trust mechanisms in diverse application domains, researchers have developed innovative solutions that incorporate risk assessment, in-network analysis, federated defenses, and decentralized auditability. Verifiable trust mechanisms ensure that system decisions are transparent, traceable, and tamper-resistant. In intrusion detection systems, they enhance not only the accuracy of threat detection but also the accountability and reliability of decision processes, thereby supporting the development of trustworthy and resilient defenses.

Renjith P N et al [25] proposed a multi-dimensional risk assessment framework that integrates realtime sensor data, V2V communication, and environmental cognition. By embedding an I-LSTM-based trust

mechanism, the framework continuously evaluates operational reliability and enhances the verifiable safety of autonomous vehicles. In the context of zero-trust networks, Zhao Z et al [26] designed FCPlane, a forwarding-computing integrated dataplane that reconciles packet forwarding and machine learning computation at the flowlet level. With the aid of a Markov chain model and multipath routing evaluations, FCPlane achieves efficient load balancing while ensuring reliable traffic analysis in practical deployments. Building on federated graph learning, Mu S et al [27] introduced TS-Defense, a Transformer-Statistical feature fusion framework. By applying balanced graph partitioning and a multimodal detection-forgetting module, it effectively isolates malicious clients and purifies poisoned gradients, thereby enabling robust and trustworthy collaboration across healthcare entities. Extending the focus to decentralized environments, Jyothi D et al [28] proposed a blockchain-enhanced intrusion detection framework that integrates multimodal CNN-LSTM analysis with adaptive federated learning and differential privacy. Blockchain-based audit logs and SHAP-based interpretability provide tamper proof accountability and transparent decision-making, which strengthen verifiable trust in modern distributed networks.

Overall, these studies demonstrate the trend of integrating verifiable trust mechanisms into diverse architectures—from autonomous systems and zero-trust networks to federated and blockchain-based IDS. Such integration not only embeds trust at the architectural level but also provides practical guarantees of accountability, transparency, and resilience, which are essential for next-generation intrusion detection systems.

2.5 Research Limitations and Gaps

Despite the notable progress achieved by many recent studies, several key limitations remain unresolved.

Sayegh H R et al [29] used LSTM with SMOTE to capture rare patterns, but robustness under noisy traffic is untested, per-class performance is underexplored, and reliance on a single traffic modality limits semantic/contextual awareness. Anthony et al [30] reported strong accuracy and F1 on CAN, CICIDS2017, and NSL-KDD with non-tree learners, yet performance degrades under extreme imbalance, noise robustness is limited, and no semantic alignment or cross-modal complementarity is modeled. Alqahtany et al [31] achieved up to 99.93% accuracy with EGWO-based feature selection plus RF, but the selection step is computationally heavy in noisy or resource-constrained settings and draws only on structured traffic, lacking redundancy compensation and contextual multimodal cues. Hazman et al [32] reached 99.9% with Boruta, AdaBoost/XGBoost on NSL-KDD and BoT-IoT, but complexity grows on imbalanced, high-dimensional data, and heterogeneous fusion or feature inconsistency is not systematically addressed.

In summary, while existing IDS methods have made considerable strides in improving classification accuracy and optimizing feature selection, they remain limited in areas such as multimodal information fusion, semantic collaboration across heterogeneous sources, adaptive modality selection, and verifiable auditing frameworks. Many studies fail to address core challenges such as inconsistent feature distributions across modalities, redundancy conflicts, and simplistic fusion strategies. These shortcomings restrict the potential to fully leverage complementary information inherent in multi-source IoT data. Therefore, there

is an urgent need to design multimodal intrusion detection systems that exhibit efficient fusion capability, robust perception, and trustworthy output—particularly for deployment in complex, dynamic, and low-trust network environments.

3. SYSTEM FRAMEWORK

This section presents a comprehensive description of the proposed multimodal intrusion detection framework, HADVB. The methodology includes the experimental workflow and implementation details. It covers the characteristics of the datasets used, the design principles of the core modules, and the specific processing techniques at each stage.

The overall framework comprises the following key stages: multimodal data acquisition and integration, data preprocessing and communication feature engineering, integrated data balancing, data standardization, feature enhancement using a denoising autoencoder, multi-model ensemble classification, and performance evaluation with a verifiable auditing mechanism. Additionally, SHAP and LIME are incorporated to enable interpretable analysis of model decisions. The system outputs include a confusion matrix, precision-recall (PR) curves, and other evaluation metrics across multiple dimensions. Figure 1 illustrates the modular architecture and experimental pipeline of the HADVB system.

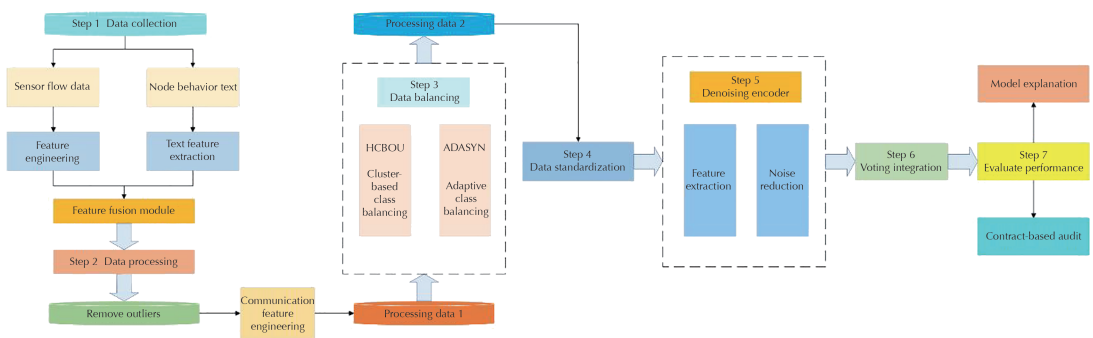


Figure 1. Workflow Architecture of the HADVB Intrusion Detection System for IoT Environments.

Step 1: Multimodal Data Acquisition and Fusion

Public IIoT intrusion-detection datasets are collected from Kaggle, including WSN-DS, WSNBFSF, and InSDN (2020). Each dataset provides network communication features (e.g., source/destination IPs, ports, traffic volume, protocol type, and transmission delay). Structured node-behavior descriptions (e.g., workload, communication status, and energy level) are included as textual inputs. Traffic features and text features are extracted separately and fused through an alignment module that maps both modalities into a shared feature space for downstream learning.

Step 2: Data Preprocessing and Communication Feature Engineering

To ensure data quality, the system performs a series of preprocessing tasks, including label encoding, missing value imputation, and the removal of blank or duplicate records. Outlier detection is also applied to eliminate anomalous entries. Given the critical role of communication behavior in intrusion detection, a communication feature engineering module is designed to extract derived metrics such as communication efficiency (comm_efficiency) and packet loss rate (packet_loss_rate) from raw traffic statistics. These features reflect actual communication quality between nodes and help increase the model's sensitivity to potential anomalies.

Step 3: Integrated Data Balancing Strategy

To address class imbalance in intrusion detection, the system implements a two-phase resampling mechanism. First, the HCBOU technique is applied. Minority class samples are clustered and used to generate synthetic samples via intra-cluster interpolation, while majority class samples are clustered and randomly downsampled within each cluster. This enhances both representativeness and diversity of the training data. Then, the ADASYN is used to adaptively oversample difficult minority instances, improving the model's generalization ability near decision boundaries. Experimental evaluations confirm that this combined sampling strategy significantly improves detection performance for minority attack types.

Step 4: Feature Standardization

To eliminate scale inconsistencies among features, the system applies Min-Max normalization to the resampled dataset. This linear transformation maps all numerical features into the [0, 1] interval. Such normalization improves the stability of gradient-based training, accelerates convergence, and prevents feature dominance. It is especially beneficial for ensemble models and neural networks, where input scale consistency is critical. The normalized features are then used as unified inputs for the autoencoder and classification models, contributing to the overall robustness and performance of the system.

Step 5: Feature Encoding with Denoising Autoencoder

After normalization, the system employs a denoising autoencoder (DAE) to perform deep representation learning on the normalized data. Specifically, noise is artificially injected into the input features during training. The denoising autoencoder learns to remove this noise and recover the underlying clean features. This process enables the model to capture essential data characteristics while filtering out noise. Consequently, the autoencoder produces more discriminative feature representations that improve anomaly detection accuracy. The learned features serve as enhanced inputs for subsequent classification tasks, boosting overall detection performance.

Step 6: Ensemble Classification via Soft Voting

The classification stage uses a soft voting ensemble that integrates Random Forest, XGBoost and MLP models. Each base learner is independently trained on the high-dimensional encoded feature space. Their

output probabilities are weighted and aggregated through the voting mechanism. This ensemble approach mitigates individual model biases and enhances stability and generalization, particularly in multi-class and minority class detection scenarios.

Step 7: Performance Evaluation and Trusted Auditing

Comprehensive evaluations are conducted on the three benchmark datasets using metrics such as accuracy, recall, F1-score, and Cohen's Kappa coefficient. To increase trustworthiness and verify model readiness for deployment, the system integrates a consortium blockchain-based auditing mechanism. This mechanism automatically assesses compliance against predefined performance thresholds. Audit logs and reports are immutably recorded on the blockchain. Visualizations of audit results provide transparency and support reliable deployment of the model in industrial IoT environments.

3.1 Data Preprocessing and Communication Feature Engineering

Data preprocessing is a critical step in building machine learning and deep learning models. It plays an important role in improving training stability, generalization ability, and feature representation effectiveness. After multimodal data fusion, this study explicitly separates input variables and target labels within the unified feature space. This separation ensures the correctness of variable dependencies during subsequent operations.

To prevent gradient explosion and learning anomalies during training, the system removes illegal values such as NaN, infinity (inf), and negative infinity (-inf) from the data. Duplicate records are also eliminated to reduce redundancy and interference in the learning process. Regarding labels, since most mainstream classifiers only support numeric target variables, label encoding is applied. Attack type labels are mapped to integers ranging from 0 to n−1, where n is the number of attack classes. This encoding ensures learnability while preserving the discrete nature of categorical variables. For missing values in numerical features, mean imputation is adopted to maintain consistent feature dimensionality and avoid modeling issues.

To further capture communication-layer behavior patterns, a communication feature engineering module is introduced. It derives two key metrics from inter-node traffic statistics: communication efficiency (comm_efficiency), defined as the amount of successfully transmitted data per unit time; and packet loss rate (packet_loss_rate), which measures transmission reliability. These communication features help detect network performance degradation caused by attacks, thereby enhancing the model's sensitivity to abnormal communication behaviors.

To validate communication feature engineering, this study analyzed the distributions of average forward packet length (fwd_pkt_len_avg) and packet rate difference (pkt_rate_diff) from the InSDN dataset. As shown in Figure 2, fwd_pkt_len_avg is highly skewed, with most flows consisting of short packets but a few containing extremely long ones. pkt_rate_diff reveals that while most flows have balanced rates, some exhibit significant asymmetry. These characteristics capture abnormal behaviors such as flooding and asymmetric traffic, providing discriminative inputs for subsequent feature extraction and classification.

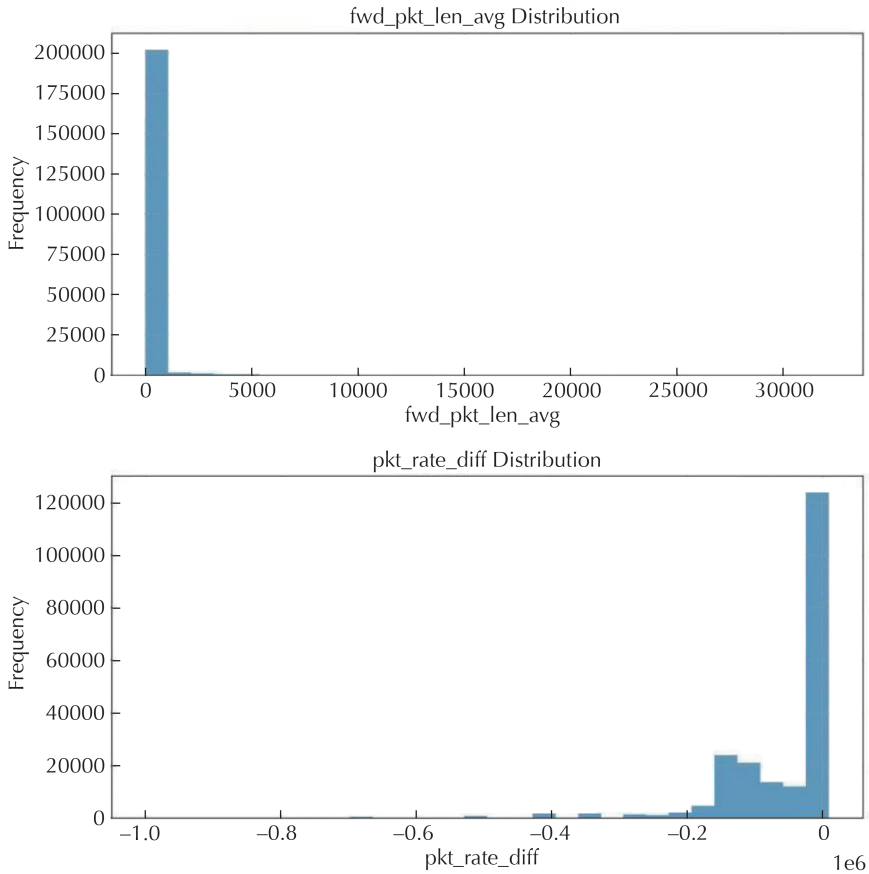


Figure 2. Distribution Histograms of Key Communication Features in the InSDN Dataset.

3.2 Integrated Data Balancing

After data preprocessing, severe class imbalance can still negatively impact model training. In IoT intrusion detection tasks, minority class attack samples often carry significant distinguishing information but are relatively scarce. This scarcity tends to bias models toward majority classes, reducing detection performance on critical attack types. To mitigate this issue, this study proposes an integrated data balancing strategy combining HCBOU and ADASYN. The goal is to enhance the representation and discrimination of minority classes.

HCBOU is a hybrid sampling method guided by clustering structures. It models intra-class cluster distributions to steer the oversampling and undersampling processes. In the oversampling phase, linear interpolation is applied within each minority class cluster. Specifically, for each pair of samples x_a and x_b within a minority cluster, new synthetic samples are generated using the following linear interpolation formula:

$$x_{new} = \alpha \cdot x_a + (1-\alpha) \cdot x_b \quad \alpha \sim \mu(0,\ 1) \tag{1}$$

Here, where x_a and x_b are two distinct sample vectors randomly chosen from the same minority cluster. Use $\alpha \sim \mu(0,\ 1)$ to denote that (\alpha) follows a uniform distribution. The newly generated synthetic sample x_{new} lies on the line segment connecting x_a and x_b.

In the undersampling phase of HCBOU, majority class samples x_{maj} are first clustered using a clustering algorithm. Then, representative samples are selected from each cluster to reduce redundancy and balance the class distribution.

$$x_{maj} \xrightarrow{KMeans(k)} \{C_1,\ C_2\ \ldots\ C_k\} \tag{2}$$

Here, x_{maj} denotes all majority class samples, and C_i represents the i-th cluster with k total clusters. Within each cluster, a fixed proportion $r \in (0,\ 1)$ of samples is randomly retained to form the preserved subset.

To further enhance the representation of boundary samples, this study integrates the ADASYN mechanism atop HCBOU to adaptively augment the oversampled data. ADASYN evaluates the classification difficulty of minority class samples based on their density in the majority class neighborhood and dynamically adjusts the number of synthetic samples accordingly. Specifically, fewer samples are generated for easily classified instances, while more synthetic samples are generated near decision boundaries and difficult-to-classify samples. This approach enables the model to focus more on the regions with the weakest discriminative power during training.

ADASYN generates synthetic samples adaptively, emphasizing minority samples close to the decision boundary, thereby improving the model's ability to learn minority classes. The main steps are as follows:

a. Equation 3 calculates the proportion of majority class samples among the neighbors of each minority class sample:

$$r_i = \frac{N_{more}}{N_{total}} \tag{3}$$

r_i measures the degree of imbalance for a minority class sample x_i. Here, N_{total} denotes the total number of neighbors of x_i, which is obtained by applying the k-nearest neighbors algorithm in the normalized feature space. In this study, $N_{total} = k$. The term N_{more} denotes the number of neighbors among these k samples that belong to the majority class. The latter can be formally expressed as Equation 4:

$$N_{\text{more}} = \sum_{x_j \in N_k(x_i)} 1\ [y_j \in C_{major}] \tag{4}$$

where $N_k(x_i)$ is the set of k nearest neighbors of x_i, y_j is the class label of sample x_j, and C_{major} represents the majority class. This calculation reflects how strongly a minority class instance is surrounded by majority class instances in its local neighborhood, which serves as the basis for adaptively generating synthetic samples in ADASYN.

b. The number of synthetic samples to be generated is calculated according to Equation 5:

$$G_i = r_i * G \tag{5}$$

Here, G is the total number of synthetic samples to be generated, and G_i represents the number of new samples generated for sample x_i.

c. New samples are generated using the linear interpolation method described in Equation 6:

$$x_{new} = x_i + \lambda \times (x_{nn} - x_i),\ \lambda \sim \bigcup(0,\ 1) \tag{6}$$

x_i denotes a minority class sample, and x_{nn} represents one of its nearest neighbors selected from the same minority class. Specifically, the k-nearest-neighbor set $N_k(x_i)$ of x_i is first obtained in the normalized feature space, excluding x_i itself. Then, the candidate minority neighbor set is constructed as $N_k^{min}(x_i) = \{x_j \in N_k(x_i) \mid y_j \in C_{min}\}$. One sample is uniformly randomly chosen from $N_k^{min}(x_i)$ as x_{nn}. By interpolating between x_i and its minority neighbor x_{nn} using Equation 6, new synthetic samples are generated near the decision boundary, which enhances the representation of difficult-to-classify regions.

This study conducts a systematic evaluation of the proposed HCBOU + ADASYN sampling strategy on three intrusion detection datasets. The experimental results, shown in Tables 1, 2 and 3, illustrate the changes in sample counts for each class before and after sampling across the three datasets. The results demonstrate that the integrated sampling strategy effectively reduces the disparity in sample

Table 1. Class Distribution of the Training Split of the WSN-DS Dataset Before and After HCBOU + ADASYN Processing.

Class	Imbalance	HCBOU + ADASYN
Blackhole	8039	136668
Flooding	2650	136295
Grayhole	11677	138294
Normal	272052	136025
TDMA	5310	136021

Table 2. Class Distribution of the Training Split of the WSNBFSF Dataset Before and After HCBOU + ADASYN Processing.

Class	Imbalance	HCBOU + ADASYN
Blackhole	9413	106227
Flooding	23875	104309
Forwarding	6116	105615
Normal	210280	105139

Table 3. Class Distribution of the Training Split of the InSDN Dataset Before and After HCBOU + ADASYN Processing.

Class	Imbalance	HCBOU + ADASYN
Normal	54739	98529
DDoS	58823	98488
Probe	49405	98529
User to Root (U2R)	14	98542
DoS	916	98526
Brute-Force Attacks (BFA)	236	98522

quantities between classes, achieves an overall balanced distribution, and establishes a more stable and representative training dataset foundation for subsequent model training.

3.3 Data Standardization

After the aforementioned data balancing, it is necessary to standardize the data for processing by the denoising autoencoder. Data standardization is a critical preprocessing step in machine learning that improves training stability, accelerates optimization, and mitigates the impact of differing feature scales on model learning. In this study, Min-Max normalization is applied to standardize the data, ensuring that all feature values fall within a consistent range. This enables the denoising autoencoder to learn information from different features more evenly. The mathematical expression for Min-Max normalization is shown in Equation 7:

$$X' = \frac{X - X_{\min}}{X_{\max} - X_{\min}} \tag{7}$$

Here, X' denotes the normalized data, while $X_{\min}$ and $X_{\max}$ represent the minimum and maximum values of the corresponding feature, respectively. Compared to Z-score normalization, Min-Max normalization is chosen in this study because it compresses data into a fixed range of [0, 1] or [–1, 1], making it particularly suitable for gradient-based optimization algorithms such as neural networks and logistic regression. This approach helps avoid the impact of feature scale on gradient updates. In this study, MinMaxScaler is applied to the data processed by HCBOU + ADASYN for normalization, as expressed by Equation 8:

$$X_{scaled} = MinMaxScaler(X_{resampled}) \tag{8}$$

This normalization method reduces the interference of outliers during subsequent training, accelerates model convergence, and improves the classifier's ability to recognize minority class samples, thereby providing a solid foundation for the subsequent denoising autoencoder (DAE) training.

3.4 Denoising and Feature Extraction

In modern Internet of Things environments, data generated by devices and sensors often contain noise caused by signal interference, transmission errors, and environmental factors. Such noise significantly degrades data quality and may lead to biased analysis results. Therefore, denoising and feature extraction are essential preprocessing steps. Denoising helps eliminate noise and irrelevant interference, enabling models to better identify meaningful information. Feature extraction reduces data dimensionality by selecting the most representative features from large sensor datasets, improving computational efficiency and optimizing model performance.

The multimodal data fusion used in this study combines network communication traffic features and semantic node behavior information. The resulting data exhibit complex characteristics such as high dimensionality, strong heterogeneity, and significant feature redundancy. Traditional methods often struggle to accurately model the underlying patterns of such complex data. To address these challenges, this study employs a deep learning-based denoising autoencoder (DAE) for simultaneous denoising and feature extraction.

A DAE is a deep neural network trained to reconstruct clean data from corrupted inputs by learning latent representations. Unlike traditional denoising techniques such as mean or Gaussian filtering, DAEs learn noise characteristics and distributions directly from the data. Through nonlinear transformations, they automatically optimize data representations, resulting in more accurate denoising.

The DAE architecture includes an input layer, encoder, latent space, decoder, and output layer, as illustrated in Figure 3. Its working principle is to inject noise into input data and train the model to remove this noise, recovering the original data structure. The encoder maps the noisy input to the latent space, while the decoder reconstructs the data from the latent representation. The loss function typically uses mean squared error (MSE) to measure the difference between the input and reconstructed data. The model is optimized by minimizing this loss during training.

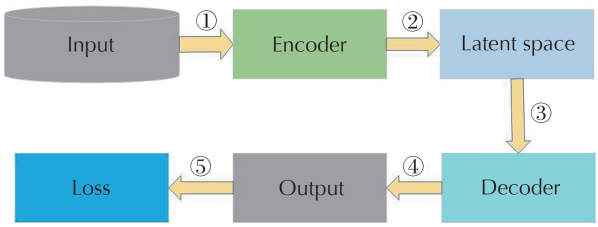


Figure 3. Architecture of the Denoising Autoencoder (DAE).

3.4.1 Input Layer: Introduction of Noisy Data

During the training of the denoising autoencoder (DAE), noise is first added to the input data. This noise can be generated in various ways, such as randomly setting some input elements to zero (masking noise)

or adding Gaussian noise. The input layer receives this corrupted data x_{noisy}. The noise injection process is described by Equation 9:

$$x_{\text{noisy}} = x + \varepsilon \tag{9}$$

Here, x denotes the original clean data, and ε denotes the injected noise (e.g., Gaussian noise) used to corrupt the input.

3.4.2 Encoder: Feature Extraction

The encoder maps the noisy input data x_{noisy} to a latent space, transforming high-dimensional input into a lower-dimensional feature representation. The encoder typically consists of multiple neural network layers that progressively abstract features and generate latent variables. This process can be expressed by Equation 10:

$$z = f_\theta(x_{\text{noisy}}) \tag{10}$$

Here, f_θ denotes the encoder function, where θ represents the network parameters. The latent representation z contains the key features of the input data.

3.4.3 Latent Space: Compressed Representation of Data

The latent space is a low-dimensional representation that preserves the most important features of the input data. Through the encoder's mapping, redundant information in the original data is removed, retaining the essential data characteristics. This low-dimensional latent representation provides critical cues for the decoder to subsequently reconstruct the original data.

3.4.4 Decoder: Data Reconstruction

The decoder transforms the low-dimensional latent representation z back to an output $\hat{x}$ with the same dimensions as the original data, representing the denoised reconstruction. The goal of the decoder is to recover a reconstruction as close as possible to the original clean input. This process is computed using Equation 11:

$$\hat{x} = g_\phi(z) \tag{11}$$

Here, g_ϕ represents the decoder function, where ϕ denotes the decoder parameters, and $\hat{x}$ is the reconstructed data.

3.4.5 Output Layer: Comparison Between Reconstructed and Original Data

The reconstructed data $\hat{x}$ produced by the output layer is compared to the original input data x to measure the reconstruction error. A commonly used loss function is the Mean Squared Error (MSE), which quantifies the difference between the original and reconstructed data. Equation 12 presents its calculation:

$$L = \frac{1}{n}\sum_{i=1}^{n}\left\|x_i - \hat{x}_i\right\|^2 \tag{12}$$

Here, n is the number of samples, and x_i and $\hat{x}_i$ denote the original and reconstructed data points of the i-th sample, respectively. The objective of the loss function is to minimize this error, enabling the model output to closely approximate the input data and effectively remove noise.

3.4.6 Optimization Process: Model Training

The network parameters are optimized through backpropagation. During training, the parameters of the encoder (θ) and decoder (ϕ) are iteratively updated to reduce the reconstruction error, ultimately achieving effective noise removal.

Figure 4 outlines the DAE workflow: noise v is injected into input x to form a corrupted sample from which the network reconstructs the original clean input. Unlike a standard autoencoder that may memorize inputs by minimizing reconstruction error, the DAE deliberately corrupts x and learns to recover it, yielding more robust, higher-level representations.

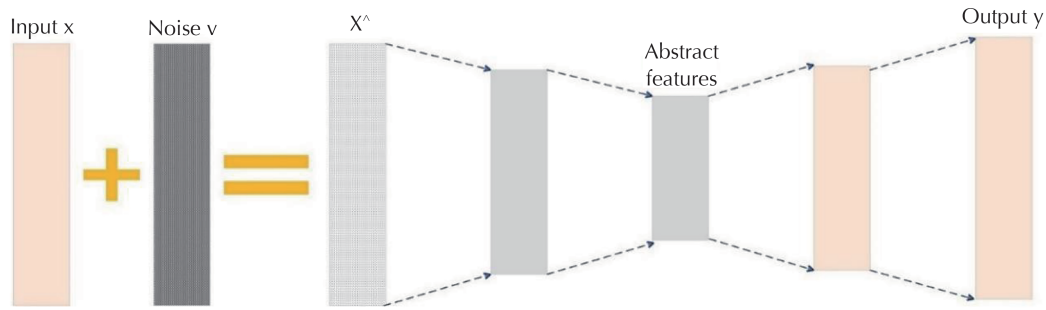


Figure 4. Noise Injection Process in the Denoising Autoencoder (DAE).

In IoT intrusion detection, noise arises from sensor faults, network latency, environmental interference, and transmission errors; given wide deployment and complex settings, such noise is unavoidable and degrades accuracy and robustness. To simulate these conditions, this paper adds Gaussian noise to the datasets. Gaussian noise—common with sensor inaccuracies and unstable links—follows a normal distribution, is generally independent of signal amplitude, and is unbiased and symmetric. Its effect is typically stable under strong signals but can fluctuate with changing conditions, impacting data quality to varying degrees. The injected noise ϵ is modeled as additive Gaussian noise sampled from a normal distribution; its probability density function is given in Equation 13:

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{13}$$

Here, μ is the mean and σ is the standard deviation, which determine the noise intensity and fluctuation range.

In this study, the DAE adopts a five-layer architecture comprising an input layer, three encoding hidden layers, and a decoding layer. Nonlinear transformations use the *tanh* activation, and batch normalization and dropout are applied after each layer to reduce overfitting. A sigmoid output layer is used to constrain reconstructions to the input range. During training, Gaussian noise is injected into the inputs to improve robustness and enhance malicious-attack detection. For optimization, AdamW is adopted to improve training efficiency and generalization; its decoupled weight decay suppresses unnecessary parameter growth and helps mitigate overfitting. Empirically, combining AdamW with batch normalization and dropout yields lower validation loss and smoother convergence across repeated runs.

As shown in Table 4, the features processed by the DAE exhibit a significant reduction in dimensionality across the three datasets—WSN-DS, WSNBFSF, and InSDN. The feature dimensions were compressed from 22, 23, and 83 after multimodal fusion to 12, 16, and 16, respectively. This effectively removes redundant information while preserving more discriminative representations. Based on these compressed features, the model demonstrates stable and low loss values on the validation sets: 0.006 for WSN-DS, 0.0016 for WSNBFSF, and 0.008 for InSDN. These results indicate that the selected AdamW optimization strategy and DAE architecture adapt well to multimodal feature compression and generalization. They maintain high training stability and representational quality across datasets of varying scales and structures, further validating their effectiveness and robustness in this study's task.

Table 4. Feature Dimensions and Validation Loss Pre- and Post-DAE on WSN-DS, WSNBFSF, and InSDN.

Dataset	Multimodal Features	Features After DAE Processing	Validation Loss
WSN-DS	22	12	0.006
WSNBFSF	23	16	0.0016
InSDN	83	16	0.008

3.5 Multi-Model Ensemble Classification

After adaptive feature learning through the Denoising Autoencoder (DAE), machine learning algorithms are applied for training on the processed data. Compared to single-model training, ensemble voting models can effectively improve classification accuracy, reduce overfitting, and enhance generalization ability. Therefore, this study adopts a voting ensemble model for training the datasets. A voting ensemble model is an ensemble learning approach based on multiple base learners, designed to improve the accuracy and stability of classification or regression tasks by combining the prediction results of multiple models. Compared with a single model, voting ensembles reduce overfitting, improve generalization, and leverage the complementary strengths of different types of learners. Voting ensembles are typically categorized into two types: soft voting and hard voting.

3.5.1 Hard Voting

Each base learner predicts the input samples independently, and the final class is decided by majority voting. This method is suitable for classification tasks, where the predicted class is the one receiving the most votes from all base learners:

$$y_{final} = mode(y_1, y_2, \ldots y_n) \tag{14}$$

Here, $y_1, y_2, \ldots, y_n$ denote the predicted classes from n base learners. The final predicted class is the one receiving the majority of votes.

3.5.2 Soft Voting

Soft voting is suitable for models that output class probabilities, such as logistic regression, random forest, and gradient boosting. Each base learner's predicted probabilities for each class are weighted and averaged. The final predicted class corresponds to the one with the highest averaged probability:

$$P_{final}(C_k) = \sum_{i=1}^{n} w_i P_i(C_k) \tag{15}$$

Here, $P_i(C_k)$ denotes the probability that the i-th base learner predicts the instance belongs to class C_k, and w_i represents the weight assigned to that model. The final predicted class is the one with the highest aggregated probability.

In this study, soft voting is chosen over hard voting because it reduces the impact of individual model errors, thereby enhancing the ensemble's robustness and stability in complex data environments. The ensemble combines both machine learning and deep learning algorithms, specifically Random Forest (RF), XGBoost, and MLP. The following provides a brief introduction to these three classifiers:

3.5.3 Random Forest, XGBoost, and MLP

XGBoost is a gradient-boosted tree method that builds trees sequentially with shrinkage, row/column subsampling, and explicit regularization plus early stopping, yielding strong performance on tabular data, robustness to missing values, and interpretable feature importances. A multilayer perceptron (MLP) is a feed-forward neural network that learns hierarchical, nonlinear representations via stacked hidden layers and modern training practices, providing powerful function approximation while being sensitive to input scaling and benefiting from careful regularization. Random Forest is a bagging-based ensemble of decision trees trained on bootstrapped samples with feature randomness; it aggregates predictions by majority vote, reducing variance and overfitting and providing a strong, stable baseline on high-dimensional or noisy data. Together, these models capture complementary inductive biases—trees are particularly effective for discrete interactions and missingness, while MLPs model smooth nonlinearities—making them well-suited for use in an ensemble.

3.6 Performance Evaluation and Smart Contract Auditing

In this study, multiple metrics are employed to evaluate the performance of the proposed model, including accuracy, precision, recall, confusion matrix, F1 score, Cohen's Kappa coefficient, and precision-recall (PR) curve. The following defines these performance metrics:

3.6.1 Accuracy, Precision, Recall and F1-Score

Accuracy is one of the most commonly used classification evaluation metrics. It represents the proportion of correctly predicted samples to the total samples. Accuracy is suitable for datasets with balanced class distributions. Its calculation is given by Equation 16:

$$Accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{16}$$

Precision measures the proportion of true positive samples among all samples predicted as positive. It assesses the reliability of the model's positive predictions. The calculation is given by Equation 17:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

Recall measures the proportion of true positive samples among all actual positive samples. It reflects the model's ability to correctly identify positive instances. The calculation is given by Equation 18:

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

The F1-Score is the harmonic mean of Precision and Recall, providing a balance between the two metrics. It is calculated as shown in Equation 19:

$$F1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

3.6.2 Cohen's Kappa Coefficient

Cohen's Kappa coefficient (Kappa) is a metric used to measure the agreement between a classification model's predictions and the true labels, especially in multi-class classification problems. Unlike traditional accuracy, Cohen's Kappa accounts for chance agreement, providing a more accurate reflection of a model's classification ability. This metric is particularly advantageous in imbalanced class scenarios.

In imbalanced datasets, accuracy is often dominated by the majority class, potentially leading to misleadingly high performance. Cohen's Kappa mitigates this bias by considering the expected agreement by chance, thus better reflecting model performance across all classes, especially for minority classes. A high Kappa value (close to 1) indicates strong agreement across multiple classes, while a low value suggests high randomness or bias in the classifier's predictions. Therefore, Cohen's Kappa offers a more

robust evaluation metric than accuracy, providing a realistic assessment of minority class recognition performance on imbalanced datasets. Its calculation is given by Equation 20:

$$k = \frac{P_o - P_e}{1 - P_e} \tag{20}$$

Here, p_o represents the observed agreement, i.e., the proportion of instances where both evaluators agree. p_e denotes the expected agreement by chance, which is the probability that evaluators independently classify samples identically.

3.6.3 Confusion Matrix and Precision-Recall Curve

The confusion matrix is a standard diagnostic that tabulates TP, FP, TN, and FN to reveal per-class errors, underpin metrics such as precision, recall, and F1, and guide optimization by exposing misclassification patterns for threshold tuning or feature engineering. Complementarily, because the datasets are imbalanced, the precision-recall curve (PR curve) is more informative than the ROC curve, as it emphasizes the precision-recall trade-off for the minority (positive) class. The area under the precision-recall curve (AUPRC), reported as average precision (AP), is used to quantify minority-class detection performance and support model selection.

3.6.4 Blockchain Auditing and Model Explainability

To enhance the trustworthiness and interpretability of the intrusion detection system, this study integrates a blockchain-based auditing mechanism with model explanation frameworks SHAP and LIME, improving both engineering practicality and theoretical verifiability.

For system trustworthiness, a lightweight credible logging and auditing mechanism based on blockchain principles is constructed from model evaluation logs. Specifically, after each model inference and evaluation round, a structured log record is automatically generated, including model name, timestamp, accuracy, F1 score, precision, recall, and Kappa coefficient, saved in JSON format. To prevent tampering, the system computes a hash digest of the JSON record, producing a unique fingerprint as evidence of model performance. These hash digests can be chained sequentially to form an immutable log structure, ensuring verifiability and traceability of performance results according to blockchain principles. This design addresses trust issues inherent in conventional log files during deployment, providing a lightweight, scalable mechanism for trustworthy record-keeping in intrusion detection systems.

For model decision interpretability, two mainstream explanation algorithms are employed: SHAP (Shapley Additive Explanations) and LIME (Local Interpretable Model-agnostic Explanations). These methods offer local and global explanations for the classification results of the voting ensemble model. SHAP leverages Shapley value theory from cooperative game theory to quantify each feature's marginal contribution to the model output. Its computation is formalized as:

$$\Phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f(S \cup \{i\}) - f(S) \right] \tag{21}$$

Here, ϕ_i represents the Shapley value of feature i, F denotes the full set of features, and S is a subset of features excluding i. This mechanism provides an additive explanation for each prediction decision, transforming the model output from a black box into an interpretable form.

In contrast, LIME approximates the original model's decision boundary locally by perturbing the original input samples and fitting a linear model within this local neighborhood. This approach reveals which features dominate a specific prediction. The core idea of LIME is to minimize a weighted local loss:

$$L(f,\ g,\ \pi_x) = \sum_{z \in Z} \pi_x(z)(f(z) - g(z))^2 + \Omega(g) \tag{22}$$

Here, f denotes the original model, g represents the local linear surrogate model, π_x is the locality weighting function, and $\Omega(g)$ is the complexity regularization term.

By integrating SHAP and LIME mechanisms, this study significantly improves the transparency and interpretability of the model during prediction. At the same time, it provides actionable guidance for operators and decision-making systems, thereby enhancing the practicality and trustworthiness of the intrusion detection system in industrial environments. Specifically, SHAP offers additive and modelagnostic feature attributions that can be aggregated into global and class-level importance analyses: (i) verifying that communication and topology features (e.g., DATA_R, Rank, send_code) play a key role in detecting Grayhole and TDMA attacks; (ii) uncovering potential spurious correlations to support feature refinement; and (iii) identifying features that increase false positives to help set more conservative alert thresholds. LIME, on the other hand, provides explanations at the instance level by fitting local surrogate models around each flagged flow, offering case-by-case decision logic. These results reveal which subtle feature changes may alter model predictions, helping analysts validate borderline samples, prioritize alerts, and fine-tune detection rules. Overall, these interpretability mechanisms not only uncover the internal logic of the model but also provide clear references for subsequent security policy design, alert prioritization, and defense rule optimization, thereby strengthening the practical value and decisionsupport capability of the detection system in real industrial scenarios.

4. EXPERIMENTAL ENVIRONMENT AND SCHEME

4.1 Experimental Environment

The experiments were conducted on a high-performance computing platform using a laptop equipped with an 8-core processor, supporting efficient concurrent task processing and enhanced multithreading. The system includes 16 GB of RAM, suitable for memory-intensive applications, and a 512 GB solid-state drive for data storage. To ensure efficient performance evaluation, this study utilized the Python programming language along with a suite of powerful libraries including Pandas, NumPy, Matplotlib,

Seaborn, Scikit-learn, Keras, TensorFlow, and Imbalanced-learn. These libraries facilitated data processing, model training, and experimental evaluation.

4.2 Experimental Design and Parameter Settings

This study develops the HADVB intrusion detection system framework, which integrates multimodal data, multi-stage feature processing, and a blockchain-based trust mechanism. The system leverages three publicly available IoT datasets supplemented by node behavior description texts, fusing structured and semantic information to form a multimodal input. In terms of data preprocessing, all numeric features with missing values were imputed using the mean strategy, which was found through preliminary trials to yield stable results without introducing significant distributional shifts. Subsequently, Min-Max normalization was applied across all datasets. This choice was validated by experiments showing that scaling inputs to the [0, 1] range improves the convergence of the denoising autoencoder, whose final layer employs a sigmoid activation. Using the same preprocessing pipeline across the three datasets ensured comparability and reproducibility.

For data balancing, a two-stage strategy combining HCBOU and ADASYN was employed. In HCBOU, the number of clusters was set to three, and the oversampling and undersampling ratios were fixed at 0.8 and 0.5, respectively. These parameters were verified through repeated experiments to provide a suitable trade-off between diversity of synthetic samples and preservation of representative majority data. Following this step, ADASYN with a "not majority" strategy was applied, and the results consistently demonstrated improved minority class recognition, particularly near the decision boundary. The stability of these settings was confirmed across all three datasets, which showed consistent improvements in macro-F1 and Cohen's Kappa compared with other ratios.

With regard to feature representation, a denoising autoencoder was constructed with hidden layer sizes of 64, 32, and 16 units. The architecture was designed to achieve effective compression while retaining critical information, and regularization was ensured by combining dropout with an L2 penalty of 1×10^{-3}. The network was optimized with AdamW at a learning rate of 1×10^{-4} under mean squared error loss, and training was performed with a maximum of 100 epochs. Early stopping with a patience of 5–8 epochs and adaptive learning rate scheduling were adopted to avoid overfitting. Extensive validation confirmed that this configuration consistently produced compact and discriminative embeddings, with experiments showing clear improvements in downstream classification accuracy relative to both shallower and deeper alternatives.

For classification, a soft voting ensemble comprising Random Forest, XGBoost, and MLP was implemented, with weights assigned as 3, 2, and 2, respectively. These weights were empirically validated, as preliminary results showed that Random Forest performed more robustly on tabular features and imbalanced conditions, justifying its higher contribution in the ensemble. XGBoost and MLP complemented this by providing non-linear decision boundaries and capturing residual patterns not covered by the Random Forest. The ensemble was validated across all three datasets, consistently outperforming individual classifiers in both macro-averaged and per-class F1-scores.

Finally, interpretability and reliability mechanisms were incorporated. SHAP and LIME were applied for local and global interpretability, with Kernel SHAP explanations stabilized by a background sample size of about 200, verified experimentally to balance computational cost and stability. For auditing, a blockchain-inspired mechanism was introduced to log predictions and evaluation outcomes, ensuring immutability and transparency. Experimental evaluation confirmed that the combination of interpretability and auditing not only improved trust in the system but also provided verifiable accountability, thereby supporting deployment in industrial IoT environments.

4.3 Dataset Description

This study utilizes three publicly available datasets to support multimodal intrusion detection research in IoT environments. The WSN-DS comprises 374,662 records managed by the LEACH routing protocol and includes four types of DoS attacks (blackhole, grayhole, flooding, and scheduling) alongside normal network behaviors. Generated via the NS-2 simulator and featuring 23 extracted attributes, WSN-DS offers detailed insights for detecting DoS attacks in wireless sensor networks. The WSNBFSF dataset contains over 3.1 million records spanning four traffic categories: blackhole, flooding, selective forwarding attacks, and normal traffic. With 18 features per record derived from real network environments, it serves as a rich resource for traffic pattern recognition and intrusion detection in wireless sensor networks. The InSDN dataset contains 205,168 samples each described by 84 features. It covers six classes including distributed denial-of-service (DDoS), normal traffic, probe, DoS, BFA, and unauthorized user access (User to Root). This dataset provides comprehensive coverage of benign and malicious activities across SDN components, facilitating robust evaluation of intrusion detection systems tailored for SDN platforms.

In addition to structured network traffic data, the study incorporates Node Behavior Description Texts—structured semantic data capturing context-aware node states in IoT environments. These texts document natural language descriptions of node behavior over time intervals, including relative position to cluster heads (“close,” “medium distance,” “far”), load levels (“high,” “moderate,” “low”), energy consumption, and communication quality (“normal,” “delayed,” “unstable”). Custom parsing functions translate these semantic descriptions into numerical features, which are aligned and fused with the corresponding network traffic data, creating a comprehensive multimodal feature space. During data fusion, node behavior texts are first standardized by mapping semantic attributes to discrete numeric variables according to predefined rules. Features are precisely aligned with network data via unique node identifiers and concatenated horizontally to form fused feature vectors. This procedure ensures that heterogeneous features from different sources are jointly modeled on the basis of semantic consistency, creating a high-dimensional input space that balances structural and semantic information. The fused features serve as unified inputs for subsequent denoising autoencoder encoding and classification model training.

5. RESULTS ANALYSIS

This section presents the evaluation results for each dataset and highlights the performance advantages of the proposed model based on these results.

5.1 Evaluation Results on the WSNBFSF Dataset

The HADVB framework was tested on the WSNBFSF dataset and compared with previous studies. As shown in Table 5, the proposed model outperforms those presented in [33-35] and [36] in terms of overall classification accuracy. In addition, HADVB achieves the highest recall among the compared methods and maintains competitive F1-score and precision, demonstrating balanced and robust classification performance across classes. Specifically, compared to the single SMOTE-based methods of Soni V et al [34], the multi-class accuracy improved by approximately 5%, demonstrating the effectiveness of the integrated HCBOU-ADASYN data balancing method proposed in this study. Unlike prior multimodal systems that rely on shallow early/late fusion, the denoising autoencoder learns a shared cross-modal representation that reduces redundancy and aligns semantics, supporting higher macro-F1 on minority classes. All results are obtained under a unified stratified five-fold protocol with fixed seeds and identical preprocessing, yielding low variance and indicating that the gains are consistent rather than foldspecific. In addition, HADVB couples accuracy with transparency via SHAP/LIME analyses and a blockchain-based audit that records decisions and enforces metric thresholds, offering stronger practical suitability than existing baselines.

Table 5. Comparative Performance Metrics of Different Studies on the WSNBFSF Dataset.

Study	Method	Accuracy	Recall	F1-Score	Precision	Cohen's Kappa
[33]	GRU	99.02%	97.73%	99.71%	99.80%	/
[34]	SMOTE-XGBoost	94.2%	97.4%	99.1%	99.6%	/
[34]	SMOTE-MLP	94.4%	94.3%	95.1%	97.3%	/
[35]	RF	99%	99%	98%	99%	/
[36]	DDoS-Net	98.12%	98.12%	98%	98.16%	/
This Study	HADVB	99.69%	99.69%	99.70%	99.69%	98.74%

Beyond the aforementioned metrics, the HADVB framework achieved a Cohen's Kappa value of 98.74% on the WSNBFSF dataset, demonstrating excellent class discrimination capability. This indicates that the model maintains a high degree of consistency across different classes. These results indicate not only strong classification performance but also consistent and balanced results across multiple data subsets, reflecting the model's robustness and generalization ability.

These outcomes can be further understood by examining the intrinsic characteristics of the WSNBFSF dataset and the principles of the HADVB framework. As shown in Table 2, the dataset exhibits a severe

imbalance, with minority classes such as Blackhole and Forwarding having only 9,413 and 6,116 samples, respectively, compared to the dominant class Flooding. By combining HCBOU with ADASYN, the framework expands minority classes to over 100,000 samples each, significantly alleviating imbalance and providing a more representative training set. This balancing process ensures that the ensemble classifiers are not biased toward the majority class and can better capture rare attack patterns.

These excellent results are also attributable to another advantage of HADVB: the integration of the DAE and the soft voting ensemble. The DAE enhances feature robustness by reconstructing clean signals from noisy inputs, thus emphasizing essential traffic patterns and improving separability of minority attack samples. On this basis, the soft voting ensemble integrates Random Forest, XGBoost, and MLP, leveraging their complementary strengths to reduce model bias and stabilize predictions. This joint design consistently improves macro-F1 and Kappa scores, confirming its effectiveness in handling imbalanced intrusion detection tasks.

Figure 5 illustrates a performance comparison between the proposed HADVB framework and several baseline models—including MLP, KNN, GRA, GRU, and CNN—on the minority class detection task of Blackhole attacks. It is evident that HADVB outperforms all comparison models in key metrics such as F1 score, precision, and recall. Specifically, HADVB achieved an F1 score of 97.71%, recall of 95.58%, and precision of 98.81%, significantly surpassing other models. The superior recall demonstrates the

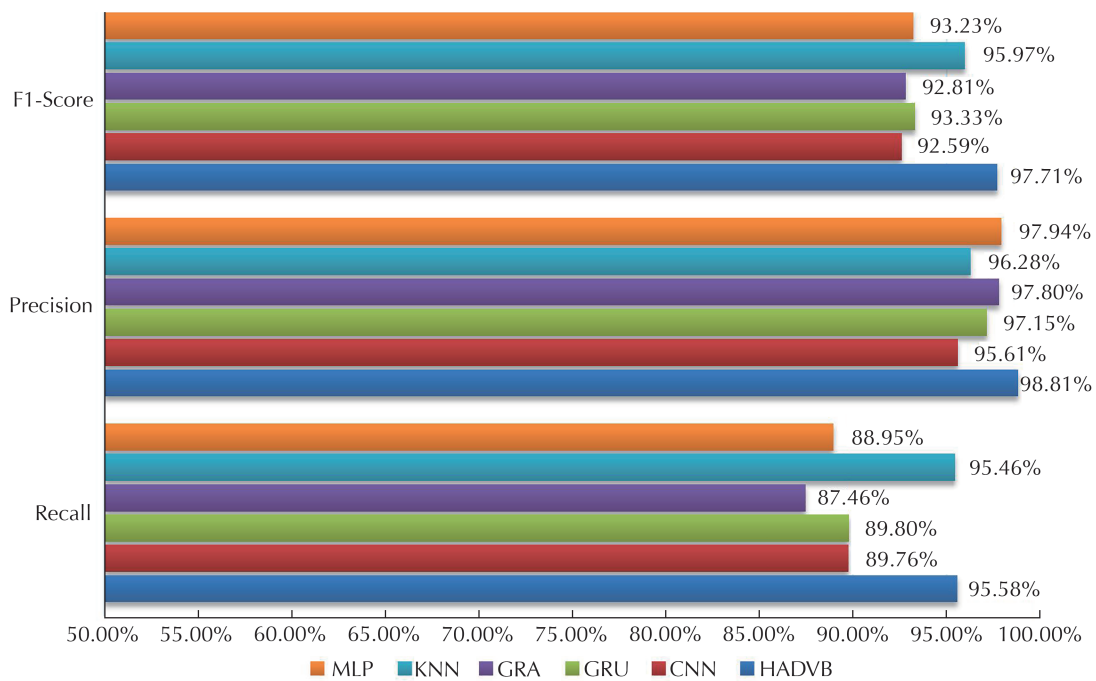


Figure 5. Comparative Classification Metrics for Blackhole Attack Detection.

effectiveness of ADASYN in generating boundary samples that make the denoising autoencoder more sensitive to subtle traffic anomalies, and also shows the advantage of the voting ensemble in balancing errors across different learners. Compared to GRU, HADVB improves the F1 score by approximately 4% and recall by nearly 6%. Compared to KNN, HADVB's precision and F1 score exceed by over 2%. Notably, in terms of recall, HADVB substantially reduces missed detections of minority class samples, validating its effectiveness in imbalanced data scenarios and demonstrating strong adaptability and robustness in detecting minority class attacks.

In the detection task for another minority class, Forwarding, within the WSNBFSF dataset, HADVB similarly demonstrates significant advantages. As shown in Figure 6, HADVB achieves Recall, Precision, and F1-Score values of 96.47%, 98.66%, and 97.55%, respectively, ranking first among all evaluated models. In comparison, the second-best model, KNN, attains Recall and Precision of 95.24% and 96.12%, respectively, with an F1-Score of 96.72%, which is still notably lower than HADVB. These improvements can be attributed to the denoising autoencoder's ability to extract compact and discriminative feature representations, which reduce noise in communication features and highlight abnormal traffic patterns.

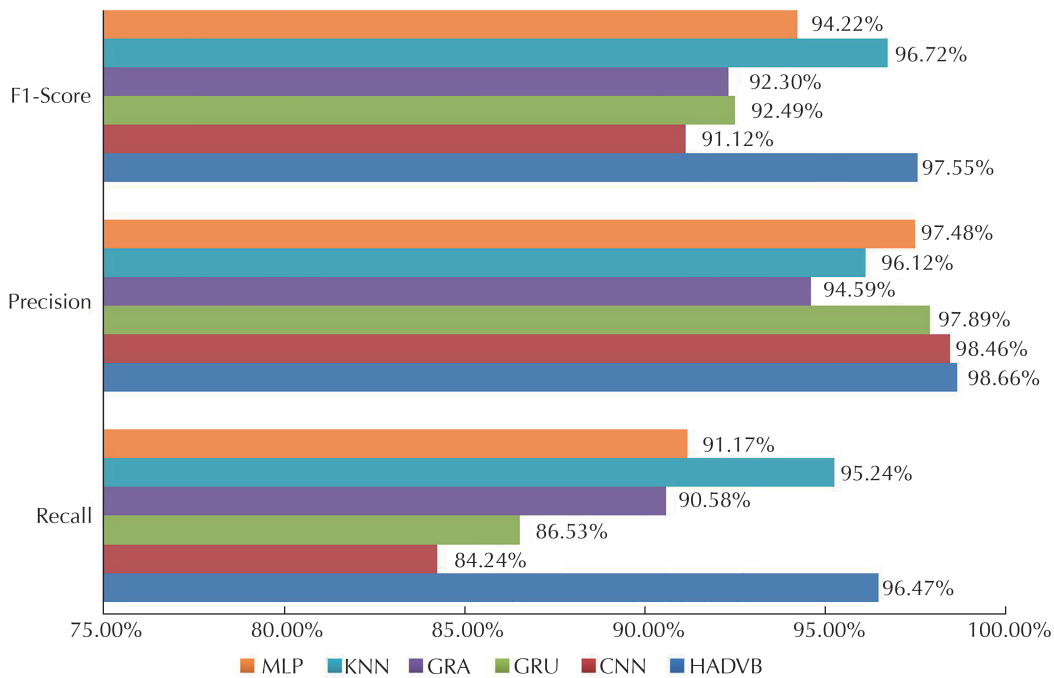


Figure 6. Comparative Classification Metrics for Forwarding Attack Detection.

Compared to deep learning models such as CNN and GRU, HADVB shows particularly substantial improvement in Recall, surpassing them by approximately 12% and 10%, respectively. This highlights its stronger robustness in reducing missed detections of Forwarding class samples. Additionally, HADVB

outperforms GRU and GRA in Precision, indicating superior performance in minimizing false positives. As a comprehensive metric balancing recall and precision, HADVB maintains a leading F1-Score, underscoring its comprehensive recognition capability for imbalanced attack categories like Forwarding. These results further validate the effectiveness of the proposed multimodal feature fusion strategy, the denoising autoencoder for robust feature learning, and the soft voting ensemble for balanced decisionmaking. Together, they demonstrate HADVB's stable detection ability against low-frequency threats in complex IoT environments.

5.2 Evaluation Results on the WSN-DS Dataset

The HADVB framework was subsequently evaluated on the WSN-DS dataset and compared with studies conducted by [37-39] and [40]. As summarized in Table 6, the proposed method achieves higher accuracy than these previous approaches, along with significant improvements in recall and F1 score. Additionally, Cohen's Kappa coefficient is included in the evaluation, with a score of 94.13%, demonstrating the model's excellent multi-class classification ability. A value close to 1 indicates a high agreement between the model's predictions and true labels, further confirming the consistency and strong classification capability of HADVB in multi-class tasks. Methods in [37-39] and [40] are singlemodal or rely on shallow early/late fusion and generally lack explicit imbalance handling, which constrains minority-class recall and stability.

Table 6. Comparative Experimental Results of Different Studies on the WSN-DS Dataset.

Study	Method	Accuracy	Recall	F1-Score	Precision	Cohen's Kappa
[37]	VG-IDS	96%	96%	96%	96%	/
[38]	SG-IDS	95%	95%	96%	98%	/
[39]	HMMs + GMMs	94.55%	75.77%	87.94%	98.41%	/
[40]	Autoencoder	91%	91%	91%	92%	/
This Study	HADVB	99.03%	98.90%	99.02%	98.84%	94.13%

To further validate the robustness and generalizability of the HADVB framework in multi-class scenarios, this study conducted a systematic evaluation on the WSN-DS dataset. As shown in Table 1, the dataset exhibits extreme imbalance, with the Normal class containing over 270,000 samples, while minority classes such as Flooding and TDMA have fewer than 6,000 samples. Such imbalance greatly challenges traditional classifiers, which tend to bias toward the majority class and misclassify rare attacks. By employing the HCBOU + ADASYN balancing strategy, the minority classes were expanded to over 130,000 samples each, effectively alleviating distribution skewness and ensuring sufficient representation of low-frequency attack patterns.

Figure 7 presents the Precision-Recall (PR) curve comparison from a macro-average perspective. The proposed HADVB model consistently maintains high precision across the recall range, achieving a mean average precision of 0.98. This stable curve indicates that the DAE plays a crucial role in enhancing feature robustness by filtering noise and extracting compact representations, which is especially beneficial for distinguishing subtle differences between attack categories such as Grayhole and TDMA. Compared with CNN and GRU, the superior PR performance reflects the advantage of the ensemble learning strategy, where soft voting integrates Random Forest, XGBoost, and MLP to reduce single-model bias and stabilize multi-class predictions.

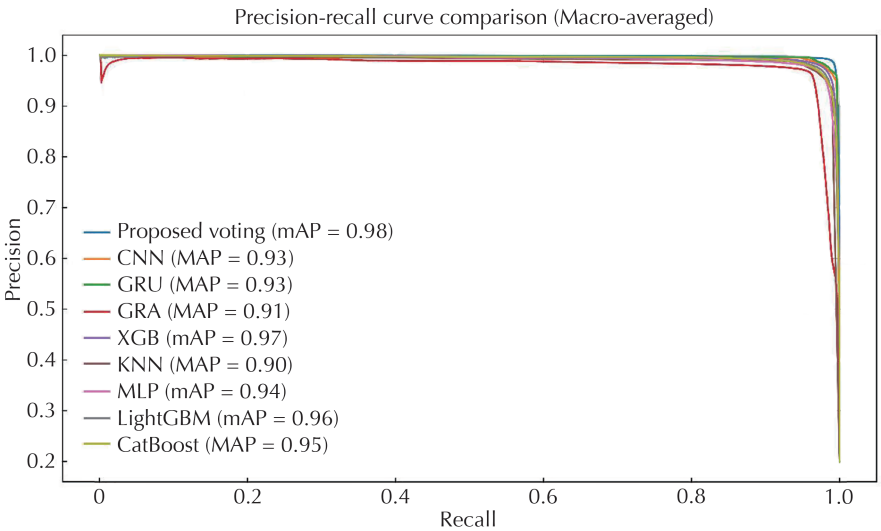


Figure 7. Macro-Averaged Precision-Recall Curve Comparison Across Models.

Figure 8 shows the confusion matrix heatmaps of different models. The results reveal that HADVB produces significantly fewer misclassifications in minority attack categories such as Blackhole, Flooding, Grayhole, and TDMA. Particularly, it achieves strong discrimination in the Grayhole class, effectively minimizing false positives that are common in other models. For the Normal category, despite its overwhelming sample size, the model maintains a very low false alarm rate, demonstrating that balancing strategies combined with DAE-based feature compression successfully prevent the majority class from dominating. In contrast, traditional methods such as KNN and GRA exhibit substantial confusion among attack categories, underscoring their limitations in handling imbalanced distributions and complex decision boundaries. These results collectively confirm that the HADVB framework not only achieves high accuracy but also leverages its algorithmic components to ensure robustness and fairness across all classes in highly imbalanced, multi-class intrusion detection tasks.

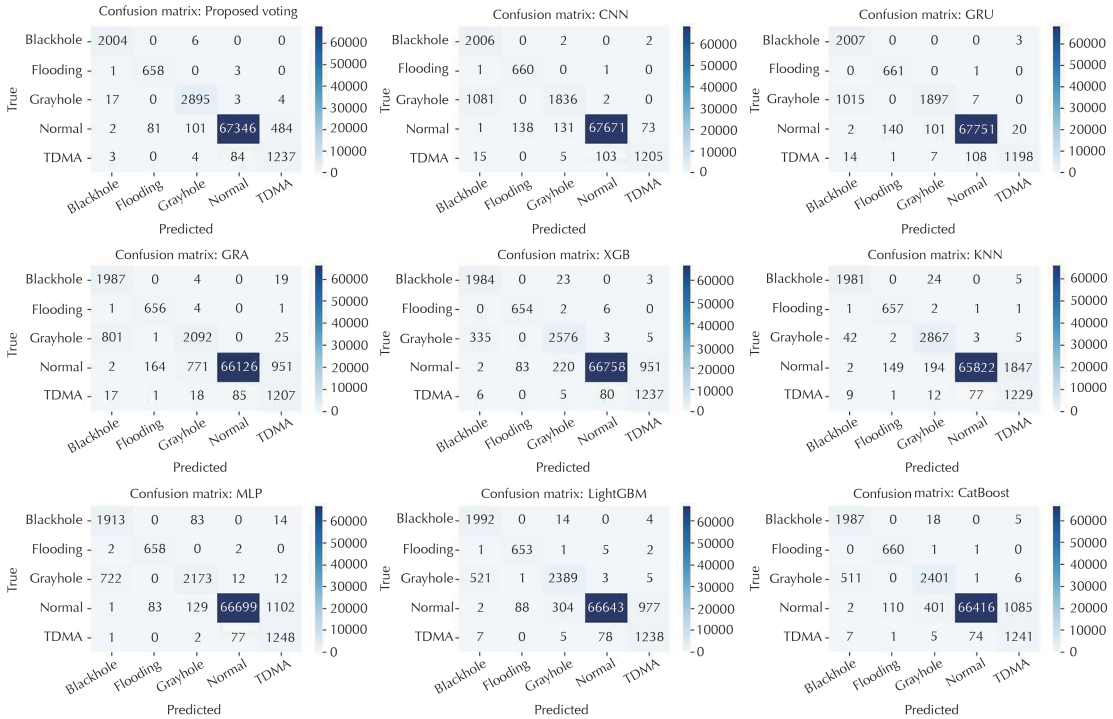


Figure 8. Confusion Matrix Heatmaps of Different Models.

To interpret the decision logic of the HADVB framework, this paper applied SHAP to the RF base learner within the soft-voting ensemble. Since the RF is trained on the DAE-encoded latent feature space, the SHAP beeswarm plot shows latent dimensions and therefore cannot directly display the original feature names (e.g., DATA_R, Rank, send_code). Nevertheless, from the plot it can be observed that certain encoded dimensions make significant positive SHAP contributions to attack predictions, indicating that the model effectively captures signals such as traffic bursts and abnormal variations in control/routing patterns, which are typically associated with selective forwarding (Grayhole) and routing/scheduling manipulation attacks. This demonstrates that the model leverages these critical communication-behavior cues to construct robust decision boundaries and reduce confusion among similar attack categories. For efficiency and clarity, the SHAP interpretability analysis was conducted only on a single base learner (RF); although the final predictions are generated by soft voting across multiple learners, the RF explanations are representative of the decision patterns captured by the ensemble.

For local interpretability, LIME was used to analyze the classification of Normal traffic. The results in Figure 9 reveal that features such as ADV_R, DATA_S, dist_CH_To_BS, and JOIN_R contribute positively and decisively to identifying legitimate communication patterns. At the same time, negative contributions from DATA_R and Rank highlight potential abnormal tendencies, showing how the model distinguishes benign traffic from latent threats at the instance level. This local explanation not only clarifies why

certain flows are classified as normal but also exposes early signs of suspicious behavior, providing analysts with actionable guidance for threat validation and response. Taken together, SHAP and LIME provide complementary perspectives that combine global consistency with local transparency, effectively demonstrating the interpretability of the framework in relation to the characteristics of intrusion methods.

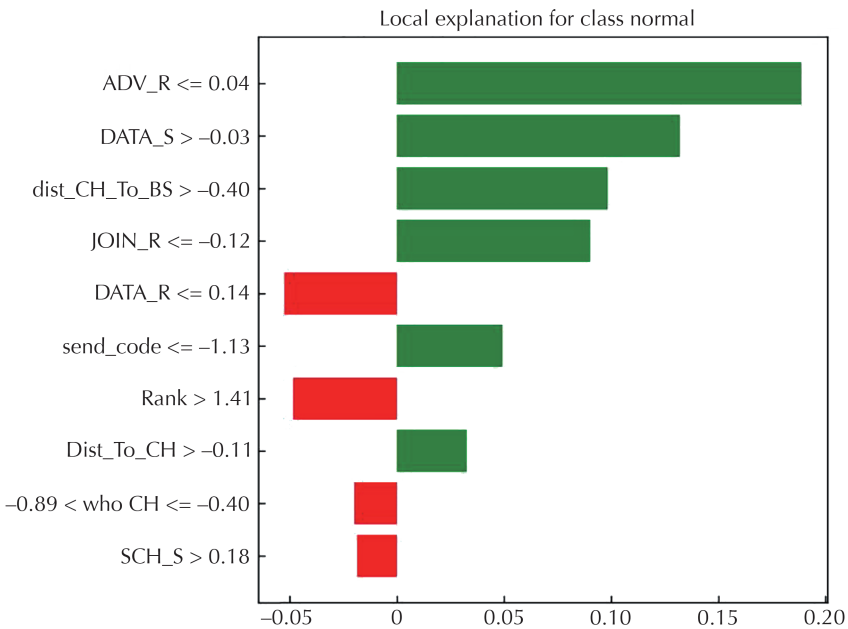


Figure 9. Local LIME Explanation for the Normal Class.

By combining SHAP and LIME, the decision logic of HADVB can be more clearly understood. The SHAP plot in Figure 10 shows that certain encoded dimensions strongly affect attack classification. These latent features reflect traffic surges and routing anomalies captured in the encoded space. LIME in Figure 9 complements this by grounding the explanation in the original feature space. It highlights that ADV_R, DATA_S, and dist_CH_To_BS support normal traffic, while high DATA_R and abnormal Rank contribute negatively. Together, the two methods show that latent dimensions correspond to variations in raw traffic and routing features, enhancing interpretability and reliability of the framework.

5.3 Evaluation Results on the InSDN Dataset

The proposed HADVB framework was further evaluated on the InSDN dataset to examine its generalizability under highly imbalanced conditions. As shown in Table 3, the dataset demonstrates an extremely skewed class distribution. While majority classes such as Normal, DDoS, and Probe contain more than 45,000 samples each, minority classes are severely underrepresented. In particular, U2R has only 14 samples, BFA contains 236, and DoS has just 916, making them highly susceptible to

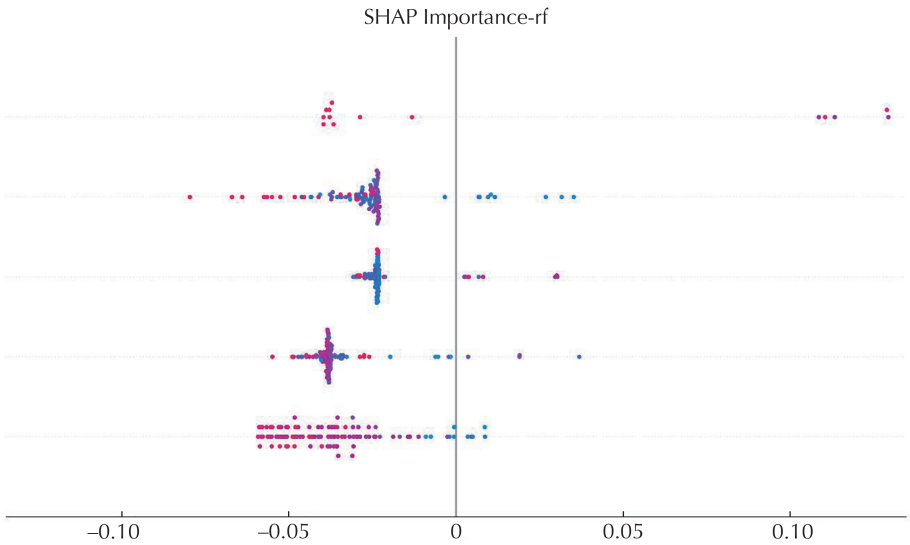


Figure 10. SHAP-based global feature importance for the Random Forest classifier.

misclassification or complete omission by traditional learning methods. Such extreme imbalance poses a substantial challenge for intrusion detection systems, as models trained without balancing techniques tend to overfit the dominant classes and fail to recognize rare but critical attack patterns.

To address this, the HADVB framework employs the HCBOU + ADASYN strategy, which expands minority classes to more than 98,000 samples each, effectively alleviating data skewness and providing sufficient representation for low-frequency categories. This resampling process establishes a balanced training foundation that enhances fairness and prevents the majority classes from dominating decision boundaries. During feature learning, the DAE was trained with the AdamW optimizer, yielding stable convergence. As reported in Table 7 and Figure 11, the validation loss reached as low as 0.008, while training and validation curves closely overlapped, indicating robust generalization without overfitting. The DAE effectively denoised traffic features and compressed them into compact, discriminative representations, which is especially critical for minority classes like U2R and BFA whose distinguishing features are subtle and easily obscured. In the classification stage, the soft voting ensemble combined Random Forest, XGBoost, and MLP. Each model contributed complementary advantages: Random Forest ensured robustness to noise and residual imbalance, XGBoost captured complex nonlinear traffic patterns, and MLP generalized well across the high-dimensional features produced by the DAE. The weighted probability averaging stabilized performance across classes and reduced bias from individual learners.

Table 7. Performance Evaluation Metrics on the InSDN Test Set.

Class	Accuracy	Recall	Precision	F1-Score	Accuracy	Loss	Cohen's Kappa
BFA	96.61%	96.61%	96.61%	96.61%	99.97%	0.008	99.95%
DDoS	99.98%	99.98%	99.99%	99.98%			
DoS	99.56%	99.56%	99.56%	99.56%			
Normal	99.97%	99.97%	99.96%	99.96%			
Probe	99.97%	99.97%	99.98%	99.97%			
U2R	99.99%	99.99%	99.99%	99.99%			

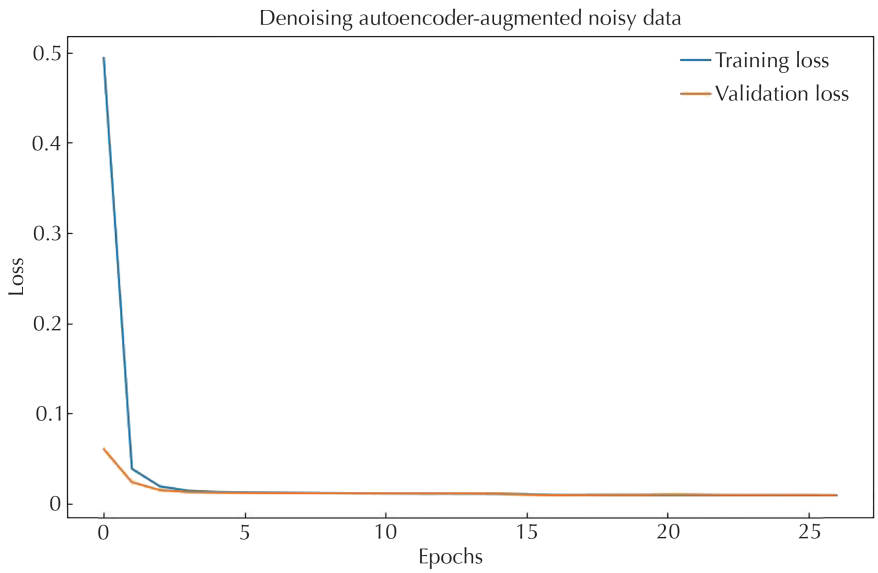


Figure 11. DAE Training and Validation Loss on InSDN Dataset.

The results summarized in Table 7 confirm that HADVB achieves excellent recognition across all classes. For minority categories, the U2R class attained 99.99% accuracy and recall with a 99.99% F1 score, while the DoS class achieved 99.56% recall and F1 score. This demonstrates that the combination of balancing and feature reconstruction effectively strengthened the model's focus on decision boundaries where minority classes are easily misclassified. Furthermore, the overall Cohen's Kappa of 99.95% validates both statistical reliability and strong class separability. These results collectively confirm that HADVB is highly effective in detecting rare intrusion types within severely imbalanced datasets, ensuring robustness and fairness across all categories.

This study tested the trusted auditing mechanism of the HADVB framework. Experimental results demonstrate that the mechanism can reliably verify and structurally archive the evaluation results of HADVB across various class detection tasks. Specifically, after each model evaluation round, the system automatically extracts key performance metrics including model name, timestamp, Accuracy, F1 score, Precision, Recall, and Cohen's Kappa coefficient, generating corresponding JSON audit log records (log files available in the Appendix). Based on preset auditing criteria, the system evaluates each model's status: if both F1 score and Kappa coefficient exceed the threshold of 0.9, the model is marked as "approved"; if either metric fails to meet the requirement, it is labeled as "violation." Figure 12 shows the audit status of all evaluated models, where all nine models passed without any violations. These results indicate that the integrated detection workflow maintains stable performance across multiple evaluation sets while complying with the auditing constraints. Although this auditing mechanism primarily serves as an auxiliary validation of the model evaluation process, the generated structured logs provide significant reference value for model quality management, deployment control, and result traceability. This mechanism lays a foundation for trusted model governance and compliance verification in subsequent deployments.

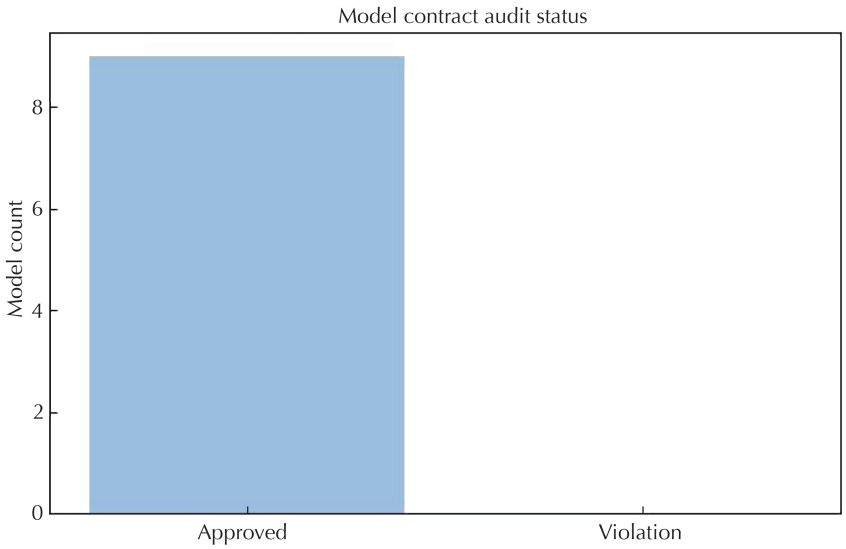


Figure 12. InSDN Model Contract Audit Status.

To mitigate randomness and prevent information leakage, stratified five-fold cross-validation was conducted on the InSDN, WSNBFSF, and WSN-DS datasets. In each fold, all preprocessing and model components were fit exclusively on the training split, and the held-out split was used only for evaluation. Random seeds were fixed, and hyperparameters and training procedures were kept identical across folds. The cross-validation results were as follows: InSDN achieved an accuracy of 0.9997 with a standard deviation of 0.0001 and a macro-F1 of 0.9775 with a standard deviation of 0.0169; WSNBFSF achieved an accuracy of 0.9969 with a standard deviation of 0.0003 and a macro-F1 of 0.9970 with a standard

deviation of 0.0016; WSN-DS achieved an accuracy of 0.9903 with a standard deviation of 0.0002 and a macro-F1 of 0.9902 with a standard deviation of 0.0012. The small variances indicate stable and reproducible performance across data splits, while the high macro-F1 values reflect strong minority-class detection capability.

5.4 Ablation Experiment

The paper isolates module effects with four ablations under identical preprocessing and the same voting classifier. A0 applies HCBOU-only balancing with the denoising autoencoder (DAE) kept and ADASYN disabled. A1 removes the DAE while retaining the two-stage balancing (HCBOU + ADASYN). A2 removes all balancing (no HCBOU, no ADASYN) while keeping the DAE. HADVB is the complete model with both DAE and HCBOU + ADASYN.

On the WSNBFSF dataset, as detailed in Table 8, the results show the expected pattern: A0 is the weakest with Accuracy 94.45%, Recall 92.58%, and F1-score 93.79%, reflecting residual imbalance even with HCBOU alone. Introducing full balancing in A1 raises Recall to 95.80% and F1-score to 96.33% because HCBOU trims redundant majority samples and ADASYN adds boundary-focused minority samples. Using only the DAE in A2 yields high Precision at 97.34% and F1-score 93.31%, while Recall drops to 89.31% since class skew is uncorrected; the cleaner, more compact features favor precision but cannot recover minority coverage. When both balancing and DAE are combined in HADVB, the model benefits from denoised features and well-covered minority regions, giving the strongest, most balanced multi-class performance.

Table 8. Ablation Study on WSNBFSF: Effect of HCBOU, ADASYN, and DAE.

Class	Accuracy	Recall	Precision	F1-Score
A0	94.45%	92.58%	96.57%	93.79%
A1	95.56%	95.80%	98.56%	96.33%
A2	93.32%	89.31%	97.34%	93.31%
HADVB	99.69%	99.69%	99.70%	99.69%

The complete HADVB model, which combines both modules, achieved the best overall results: Accuracy 99.69%, Precision 99.70%, Recall 99.69%, and F1-score 99.69%. This shows that balancing strategies mitigate distribution skew, while the autoencoder strengthens representation and noise resistance. Their complementary effects enable stable and reliable performance even under imbalance.

As shown in Figure 13, for minority-class attacks, HADVB also provided more balanced results than A1. In the Blackhole attack, A1 achieved Recall 94.15% and Precision 98.84%, while HADVB balanced Recall 95.03% and Precision 98.72%, producing a higher F1-score. A similar pattern occurred in the Forwarding attack, where A1 had Recall 96.76% and Precision 95.68%, but HADVB maintained Recall 96.08% and Precision 98.06%, again yielding a higher F1-score. These results confirm that the denoising autoencoder not only enhances generalization but also stabilizes the trade-offs among evaluation metrics, strengthening detection of minority-class attacks.

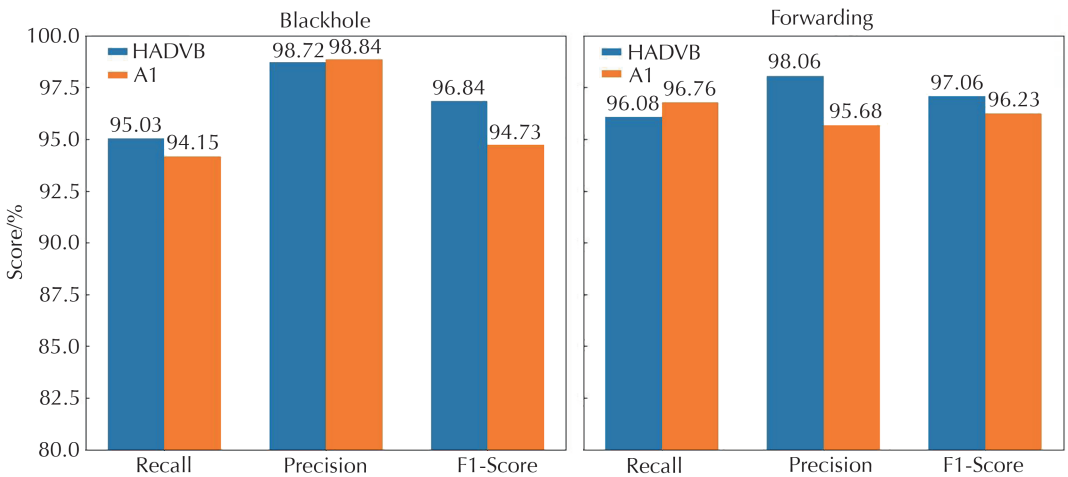


Figure 13. Ablation study results of the HADVB framework on the WSNBFSF dataset.

Figures 14 and 15 present the SHAP-based global explanations, which further substantiate this observation. In the case of A1, the classifier relies heavily on structural identifiers such as Node_id and temporal attributes, while semantic behavioral features like Event contribute less. Although data balancing mitigates class skew, the decision process still shows a bias toward superficial structural features, constraining generalization to unseen minority attacks. By contrast, with the denoising autoencoder integrated in HADVB, Event becomes the most influential factor, while Node_id and Time diminish in importance. This shift indicates that HADVB, through robust latent representations, emphasizes semantic behavioral patterns over spurious correlations, thereby improving both sensitivity and generalization to minority-class attacks and explaining its consistent superiority across all metrics.

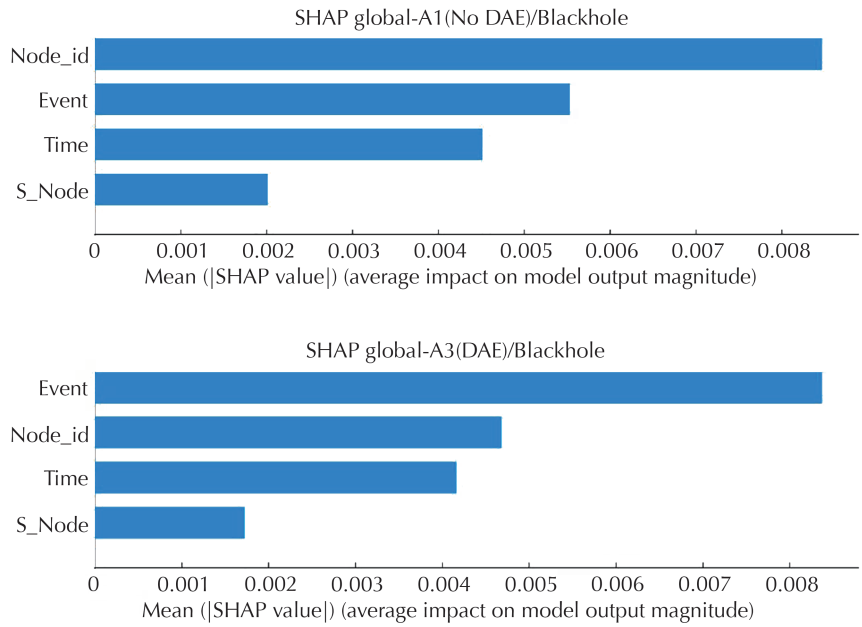


Figure 14. SHAP feature importance: A1 vs. HADVB on Blackhole attack.

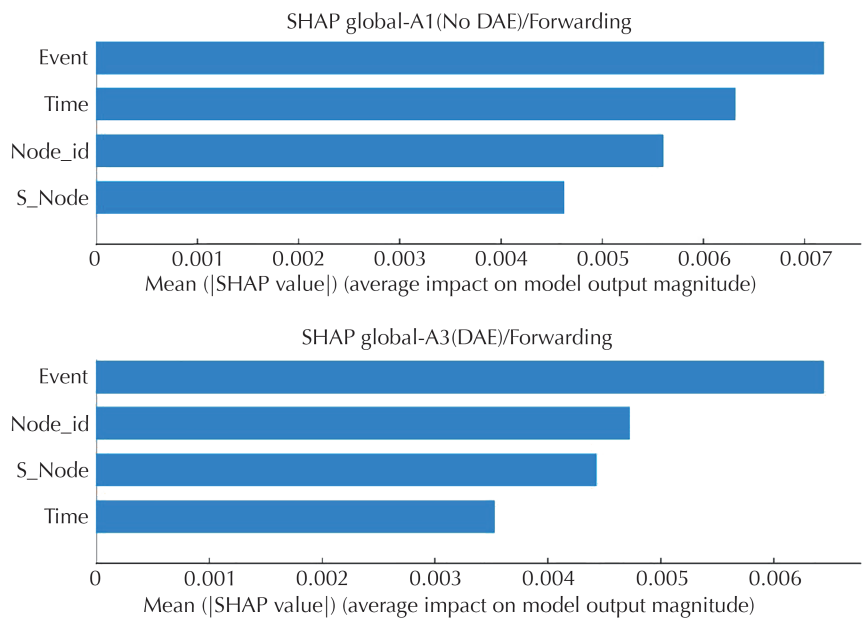


Figure 15. SHAP feature importance: A1 vs. HADVB on Forwarding attack.

6. CONCLUSION AND FUTURE WORK

The proposed HADVB framework addresses the needs of multimodal intrusion detection in industrial IoT environments by integrating communication traffic data and node behavior text. It improves data quality through unified standardization and anomaly removal processes, and enhances minority attack recognition via a combined HCBOU and ADASYN resampling strategy. The system employs a denoising autoencoder for deep feature extraction and constructs a soft voting ensemble model with Random Forest, XGBoost, and MLP. The framework achieves accuracies of 99.03%, 99.69%, and 99.97% on the WSN-DS, WSNBFSF, and InSDN datasets, respectively. To increase model trustworthiness and interpretability, SHAP and LIME are integrated for local and global decision explanations, and a blockchain-based logging mechanism ensures structured and tamper-proof inference records. However, as shown in Table 9, the HADVB framework's complexity incurs significant training time costs. The reported results reflect the computational overhead of a single end-to-end training run of HADVB. On the WSN-DS dataset, training takes 738.24 seconds, substantially exceeding the times of CNN (204.51 seconds), GRU (128.97 seconds), and lightweight models such as XGBoost (28.68 seconds). Similar trends are observed on the WSNBFSF (415.26 seconds) and InSDN (301.24 seconds) datasets, highlighting the relatively high computational cost of the proposed framework.

Table 9. Training Time Comparison of HADVB on WSN-DS, WSNBFSF, and InSDN Datasets (seconds).

Model	WSN-DS	WSNBFSF	InSDN
HADVB	738.24	415.26	301.24
CNN	204.51	68.57	106.89
GRU	128.97	51.42	48.64
KNN	82.91	23.81	33.46
MLP	496.93	289.12	214.08
XGBoost	28.68	5.72	16.53

Future work will focus on model compression strategies including pruning, knowledge distillation, and sparse activation, combined with mixed-precision training and other optimization techniques to improve performance. Additionally, exploration of online learning, modality attention mechanisms, graphbased modeling, on-chain permission governance frameworks, and incorporation of interpretability mechanisms into the training process will be conducted to enhance system adaptability, scalability, and trustworthiness in dynamic environments. These efforts aim to facilitate widespread practical deployment of HADVB in industrial intrusion detection scenarios.

AUTHOR CONTRIBUTIONS

Gang Han: Writing-Review and Editing.

Xuan Chang: Methodology; Writing-Original-Draft.

YaoTian Wang: Experiment.

YunLi Wang: Formal Analysis.

ACKNOWLEDGEMENT

This work was supported in part by the National Natural Science Foundation of China (Grant No. 62102312), Key Research and Development Program of Shaanxi (Program No. 2024GX-YBXM-079), the fund of the State Key Laboratory of Integrated Service Networks (Grant No. ISN24-13), the Young Talent fund of University Association for Science and Technology in Shaanxi, China (Grant No. 20210119), The Youth Innovation Team of Shaanxi Universities (Program No. 23JP160), and The Youth Innovation Team of Shaanxi Universities (Program No. 23JP162).

REFERENCES

[1] T. M. Booij, I. Chiscop, E. Meeuwissen, *et al*., "ToN_IoT: The role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets," *IEEE Internet Things J*., vol. 9, no. 1, pp. 485–496, 2021.

[2] Y. Zheng, Z. Li, X. Xu, *et al*., "Dynamic defenses in cyber security: Techniques, methods and challenges," *Digit. Commun. Netw*., vol. 8, no. 4, pp. 422–435, 2022.

[3] A. Heidari and M. A. Jabraeil Jamali, "Internet of things intrusion detection systems: A comprehensive review and future directions," *Cluster Comput*., vol. 26, no. 6, pp. 3753–3780, 2023.

[4] D. Guo, K. Li, B. Hu, *et al*., "Benchmarking micro-action recognition: Dataset, methods, and applications," *IEEE Trans. Circuits Syst. Video Technol*., vol. 34, no. 7, pp. 6238–6252, 2024.

[5] X. Chen, D. Liu, X. Yang, X. Li, J. Dong, M. Wang, and X. Wang, "PRVR: Partially relevant video retrieval," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 48, no. 2, pp. 1262–1277, F 2026.

[6] G. Corso, H. Stark, S. Jegelka, et al., "Graph neural networks," *Nat. Rev. Methods Primers*., vol. 4, no. 1, Art. no. 17, 2024.

[7] F. Alwahedi, A. Aldhaheri, M. A. Ferrag et al., "Machine learning techniques for IoT security: Current research and future vision with generative AI and large language models," *Internet Things Cyber-Phys. Syst*., vol. 5, pp. 1–16, 2024.

[8] V. Z. Mohale and I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Front. Artif. Intell*., vol. 8, Art. no. 1526221, 2025.

[9] P. Xu, X. Zhu, and D. A. Clifton, "Multimodal learning with transformers: A survey," *IEEE Trans. Pattern Anal. Mach. Intell*., vol. 45, no. 10, pp. 12113–12132, 2023.

[10] W. H. Lee, M. Ozger, U. Challita, et al., "Noise learning-based denoising autoencoder," *IEEE Commun. Lett*., vol. 25, no. 9, pp. 2983–2987, 2021.

[11] A. Salehi and M. Khedmati, "Hybrid clustering strategies for effective oversampling and undersampling in multiclass classification," *Sci. Rep.*, vol. 15, no. 1, Art. no. 3460, 2025.
[12] M. S. Islam, M. A. Rahman, M. A. Bin Ameedeen, et al., "Blockchain-enabled cybersecurity provision for scalable heterogeneous network: A comprehensive survey," *CMES-Comput. Model. Eng. Sci.*, vol. 138, no. 1, 2024.
[13] B. Sangameshwar, P. C. Mounika, P. Mandapati, et al., "A blockchain-based multimodal approach to malware detection in Android IoT ecosystems," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 7, 2025.
[14] R. Yasaei, Y. Moghaddas, and M. A. Al Faruque, "IoT-GRAF: IoT graph learning-based anomaly and intrusion detection through multi-modal data fusion," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, 2024, pp. 1–6.
[15] F. Ullah, A. Turab, S. Ullah, et al., "Enhanced network intrusion detection system for internet of things security using multimodal big data representation with transfer learning and game theory," *Sensors*, vol. 24, no. 13, Art. no. 4152, 2024.
[16] M. J. C. S. Reis, "Internet of things and artificial intelligence for secure and sustainable green mobility: A multimodal data fusion approach to enhance efficiency and security," *Multimodal Technol. Interact.*, vol. 9, no. 5, Art. no. 39, 2025.
[17] A. Kiflay, A. Tsokanos, M. Fazlali, et al., "Network intrusion detection leveraging multimodal features," *Array*, vol. 22, Art. no. 100349, 2024.
[18] S. A. Wagan, J. Koo, I. F. Siddiqui, et al., "A fuzzy-based duo-secure multi-modal framework for IoMT anomaly detection," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 35, no. 1, pp. 131–144, 2023.
[19] H. Attar, "Joint IoT/ML platforms for smart societies and environments: A review on multimodal information-based learning for safety and security," *ACM J. Data Inf. Qual.*, vol. 15, no. 3, pp. 1–26, 2023.
[20] M. A. Talukder, M. M. Islam, M. A. Uddin, et al., "Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction," *J. Big Data*, vol. 11, no. 1, Art. no. 33, 2024.
[21] M. A. Hossain and M. S. Islam, "Ensuring network security with a robust intrusion detection system using ensemble-based machine learning," *Array*, vol. 19, Art. no. 100306, 2023.
[22] F. Nabi and X. Zhou, "Enhancing intrusion detection systems through dimensionality reduction: A comparative study of machine learning techniques for cyber security," *Cyber Security and Applications*, Art. no. 100033, 2024.
[23] A. Paya, S. Arroni, V. García-Díaz, et al., "Apollon: A robust defense system against adversarial machine learning attacks in intrusion detection systems," *Comput. Secur.*, vol. 136, Art. no. 103546, 2024.
[24] M. A. Talukder, S. Sharmin, M. A. Uddin, et al., "MLSTL-WSN: Machine learning-based intrusion detection using SMOTE-Tomek in WSNs," *Int. J. Inf. Secur.*, vol. 23, no. 3, pp. 2139–2158, 2024.
[25] P. N. Renjith, S. Balasubramani, K. Ramesh, et al., "An initial risk assessment for multimodal with LSTM-based trust evaluation framework for autonomous vehicle security," *SN Comput. Sci.*, vol. 6, no. 2, pp. 1–15, 2025.
[26] Z. Zhao, Z. Li, X. Xie, Z. Liu, T. Li, J. Yu, F. Zhang, and B. Chen, "Verify all traffic: Towards zero-trust in-network intrusion detection against multipath routing," *IEEE J. Sel. Areas Commun.*, vol. 43, no. 6, pp. 2155–2171,2025.
[27] S. Mu, N. Lu, W. Shi, S. Kumari, K. Agarwal, and M. Alsabaan, "Multimodal feature fusion against poisoning attacks in federated graph learning," *IEEE Trans. Consum. Electron.*, vol. 71, no. 4, pp. 11350–11361, 2025.
[28] D. Jyothi, P. J. Vijay, M. K. Kumar, et al., "Design of an improved method for intrusion detection using CNN, LSTM, and block chain," *J. Theor. Appl. Inf. Technol.*, vol. 103, no. 1, 2025.

[29] H. R. Sayegh, W. Dong, and A. M. Al-madani, "Enhanced intrusion detection with LSTM-based model, feature selection, and SMOTE for imbalanced data," *Appl. Sci.*, vol. 14, no. 2, Art. no. 479, 2024.
[30] C. Anthony, W. Elgenaidi, and M. Rao, "Intrusion detection system for autonomous vehicles using non-tree based machine learning algorithms," *Electronics*, vol. 13, no. 5, Art. no. 809, 2024.
[31] S. S. Alqahtany, A. Shaikh, and A. Alqazzaz, "Enhanced Grey Wolf Optimization (EGWO) and random forest based mechanism for intrusion detection in IoT networks," *Sci. Rep.*, vol. 15, no. 1, Art. no. 1916, 2025.
[32] C. Hazman, A. Guezzaz, S. Benkirane, et al., "Toward an intrusion detection model for IoT-based smart environments," *Multimedia Tools Appl.*, vol. 83, no. 22, pp. 62159–62180, 2024.
[33] M. Dener, C. Okur, S. Al, et al., "WSN-BFSF: A new data set for attacks detection in wireless sensor networks," *IEEE Internet Things J.*, vol. 11, no. 2, pp. 2109–2125, 2023.
[34] V. Soni, D. Kukreja, and A. Malhotra, "Impact of class balancing on intrusion detection system for WSN-BFSF dataset," *SN Comput. Sci.*, vol. 5, no. 8, Art. no. 1105, 2024.
[35] T. Zhukabayeva, A. Pervez, Y. Mardenov, M. Othman, N. Karabayev, and Z. Ahmad, "A traffic analysis and node categorization-aware machine learning-integrated framework for cybersecurity intrusion detection and prevention of WSNs in smart grids," *IEEE Access*, vol. 12, pp. 91715–91733, 2024.
[36] F. Reza, "DDoS-Net: Classifying DDoS attacks in wireless sensor networks with hybrid deep learning," in *Proc. 6th Int. Conf. Electr. Eng. Inf. Commun. Technol. (ICEEICT)*, 2024, pp. 487–492.
[37] O. Ahmed, "Enhancing intrusion detection in wireless sensor networks through machine learning techniques and context awareness integration," *Int. J. Math., Stat., Comput. Sci.*, vol. 2, pp. 244–258, 2024.
[38] H. M. Saleh, H. Marouane, and A. Fakhfakh, "Stochastic gradient descent intrusions detection for wireless sensor network attack detection system using machine learning," *IEEE Access*, vol. 12, pp. 3825–3836, 2024.
[39] A. R. A. Moundounga and H. Satori, "Stochastic machine learning based attacks detection system in wireless sensor networks," *J. Netw. Syst. Manage.*, vol. 32, no. 1, Art. no. 17, 2024.
[40] F. Al-Quayed, Z. Ahmad, and M. Humayun, "A situation based predictive approach for cybersecurity intrusion detection and prevention using machine learning and deep learning algorithms in wireless sensor networks of industry 4.0," *IEEE Access*, vol. 12, pp. 34800–34819, 2024.

AUTHOR BIOGRAPHY



Gang Han is currently an associate professor at the School of Cyberspace Security, Xi'an University of Posts and Telecommunications, Xi'an, China. He is a senior member of CCF. His research interests include IOT ang Blockchain technology.
E-mail: hangang668866@163.com; ORCID: 0000-0002-2305-6870



Xuan Chang was born in Yingkou, China, in 2002. He is currently pursuing his on-the-job postgraduate studies in the School of Cyberspace Security at Xi'an University of Posts and Telecommunications. He has also participated in numerous competitions in the field of artificial intelligence and has received certain honors. With a strong passion for network security, He is dedicated to furthering his knowledge and skills in this field.
E-mail: 2923547941@qq.com; ORCID: 0009-0004-2775-3536



Yaotian Wang is currently a graduate student at the School of Cyberspace Security at Xi'an University of Posts and Telecommunications. He has been involved in a number of related research in the field of artificial intelligence. He also has a strong passion for cyber security and is committed to further developing his knowledge and skills in this field.
E-mail: 1767178707@qq.com; ORCID: 0009-0005-5248-1242



Zhongliang Zhang received his Ph.D. degree in system engineering from the Northeastern University, China. He is currently a full professor in the Department of Information Management and Information System, Hangzhou Dianzi University, China. His current research interests include federated learning, data mining, and evolutionary algorithms. He is the author of more than 50 papers on peer-reviewed journals including PR, INS, KBS, IPM, EAAI, PAAA, DSS, etc.
E-mail: zlzhang2525@163.com; ORCID: 0000-0001-6555-7908