IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 March 2025, accepted 31 March 2025, date of publication 4 April 2025, date of current version 11 April 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3557959

RESEARCH ARTICLE

Multi-Stage Deep Learning for Intrusion Detection in Industrial Internet of Things

SEGUN I. POPOOLA[1], (Member, IEEE), YAKUBU TSADO[2], (Member, IEEE), ABIMBOLA A. OGUNJINMI[3], (Senior Member, IEEE), ERIKA SANCHEZ-VELAZQUEZ[1], YONGHONG PENG[1], (Member, IEEE), AND DANDA B. RAWAT[4], (Senior Member, IEEE)

[1]School of Computing and Information Science, Anglia Ruskin University, CB1 1PT Cambridge, U.K.
[2]Department of Computing and Mathematics, Manchester Metropolitan University, M1 5GD Manchester, U.K.
[3]School of Emerging Communication Technology, Ohio University, Athens, OH 45701, USA
[4]Department of Electrical Engineering and Computer Science, Howard University, Washington, DC 20059, USA

Corresponding author: Segun I. Popoola (segun.popoola@aru.ac.uk)

This work was supported in part by Innovate U.K. under the Cyber Security Academic Startup Accelerator Program 2024–2025 (CyberASAP) under Grant 10118855.

ABSTRACT The Industrial Internet of Things (IIoT) facilitates enhanced automation, predictive maintenance, real-time monitoring, and data analytics across various sectors, including manufacturing, energy, transportation, agriculture, and supply chain management, thereby improving productivity, efficiency, and operational safety. However, as IIoT networks continue to expand, it is imperative to secure them against increasingly sophisticated cyber threats. Deep Learning (DL) techniques have been extensively utilized for intrusion detection within IIoT systems. Nevertheless, addressing the class imbalance problem remains a significant challenge. The underrepresentation of certain attack types in training data frequently results in the development of DL models that struggle to accurately detect these categories of malicious activities. This limitation represents considerable risks to the security of IIoT networks, as undetected attacks and false alarms may lead to severe operational disruptions. In this paper, we propose a multi-stage deep learning (MSDL) method specifically designed to enhance intrusion detection within IIoT networks by addressing the class imbalance issue. We assessed the effectiveness of our approach utilizing two highly imbalanced datasets: X-IIoTID and WUSTL-IIoT. Our experimental findings indicate that the proposed MSDL method surpasses the baseline DL models as well as state-of-the-art oversampling and undersampling techniques. Specifically, the MSDL method exhibits significant improvements in recognizing minority-class attacks that are frequently misclassified. Consequently, the implementation of the MSDL for intrusion detection is anticipated to strengthen the overall security and resilience of IIoT systems, providing stronger protection against a diverse array of cyber threats in industrial applications.

INDEX TERMS Cyber attacks, intrusion detection, artificial intelligence, machine learning, deep learning, network security, Internet of Things, fifth industrial revolution.

I. INTRODUCTION

The Industrial Internet of Things (IIoT) integrates physical industrial assets with advanced information, communication, and intelligent electronics, thereby enhancing operational efficiency and fostering innovation within industrial environments. Using sensors, actuators, and data analytics, the IIoT facilitates real-time monitoring, predictive maintenance, and optimized processes, resulting in reduced downtime, improved safety, and increased resilience. These capabilities are indispensable to Industry 5.0, a paradigm prioritizing human-machine collaboration, sustainability, and adaptability. In this context, IIoT amplifies machine intelligence and promotes human-centric interactions, thereby driving innovation within industrial systems. The swift adoption of IIoT across various sectors, including manufacturing,

The associate editor coordinating the review of this manuscript and approving it for publication was Stefano Scanzio.

energy, healthcare, and logistics, exemplifies its transformative potential, leading to enhancements in diagnostics, cost reduction, and increased operational reliability. For instance, applications of IIoT in manufacturing have achieved a 70% decrease in diagnostic times, whereas the oil and gas sector has realized a 40% reduction in downtime. With the global IIoT market anticipated to reach $286.3 billion by 2029, its impact on industrial landscapes will continue to expand considerably.

The rapid growth of IIoT presents significant cybersecurity challenges by disrupting traditional industrial security models and exposing Industrial Control Systems (ICS) to vulnerabilities commonly found in Internet of Things (IoT) environments [1]. These interconnected systems have an elevated risk profile, making them more susceptible to cyberattacks that exploit the weaknesses inherent in networked environments [2]. Recent data reveal a sharp rise in security threats, with an 80% increase in ICS vulnerabilities reported in 2022 [3]. Additionally, the National Vulnerability Database (NVD) recorded 28,831 vulnerabilities in 2023, reflecting a 15% increase over 2022, with 28% and 47% categorized as high and medium risk, respectively [4]. These vulnerabilities illustrate the real dangers of cyberattacks, as evidenced by significant incidents such as the 2021 ransomware attack on the Colonial Pipeline, which disrupted fuel supply across The United States of America (USA) and caused substantial economic damage [5]. Similarly, CrashOverride and Industroyer2 malware attacks on Ukraine's electricity grid in 2016 and 2022 showcased how cyber threats can compromise ICS by exploiting communication protocols, leading to widespread blackouts affecting hundreds of thousands of people [6]. These incidents emphasize the urgent need for robust cybersecurity measures in IIoT systems to ensure operational continuity and safeguard the critical industrial infrastructure.

An Intrusion Detection System (IDS) is crucial in IIoT environments for maintaining security and privacy, given the vulnerabilities inherent in IIoT devices. Many IIoT devices have limited computing power and storage capacity, complicating standard encryption techniques. An IDS monitors network traffic and system activities for malicious actions or policy violations. Generally, IDS implementations use two techniques: signature-based detection and anomaly-based detection. Signature-based IDS compare observed activities against a database of known threat patterns, offering high accuracy and minimal false positives for recognized threats. However, it is less effective against zero-day exploits or novel attacks that lack the corresponding signatures. Conversely, an anomaly-based IDS establishes a baseline for normal network behavior and flag deviations as potential threats. This method effectively detects unknown threats but may generate more false positives, requiring fine-tuning to achieve an optimal balance between sensitivity and specificity.

Machine Learning (ML) and Deep Learning (DL) have significantly advanced intrusion detection in IIoT [7]. These techniques have proven highly effective in detecting and classifying complex network traffic patterns [8], [9]. For instance, recent work by Bocu et al. [10] demonstrated a real-time IDS designed specifically for software-defined 5G networks, highlighting the potential of combining SDN with advanced detection algorithms to improve security in high-speed, dynamic communication environments. Despite their success, current approaches face limitations, particularly owing to outdated datasets that do not adequately represent modern IIoT-based cyberattacks. In addition, multi-class classification methods used to identify specific attack types often struggle with class imbalance, where minority classes are underrepresented. This results in misclassification and incorrect labeling of the attack traffic as normal [11]. The issue of class imbalance is well-documented as a major problem that adversely affects model accuracy and leads to biased performance. DL models frequently have difficulty learning the features of minority classes, which results in poor performance in these areas [12], [13]. In the context of an IDS, misclassifying malicious traffic as benign presents a greater risk than blocking legitimate traffic because it can lead to undetected breaches and potentially cause significant damage or complete system failures.

To address the class imbalance problem in DL-based IDS, data sampling techniques such as oversampling and undersampling have been proposed to balance training data before classification [14]. These techniques adjust the size of the training data by adding or removing instances to achieve balanced learning. In this study, we demonstrate that the effectiveness of oversampling and undersampling techniques is limited, particularly when the data are highly imbalanced. Additionally, we propose a multi-stage deep learning (MSDL) method to improve intrusion detection in IIoT networks by addressing class imbalance. The main contributions of this study are as follows.

1) A novel DL-based method named MSDL enhances intrusion detection in IIoT networks by addressing the class imbalance issue.
2) A comprehensive taxonomy of recent related work, covering the X-IIoTID and WUSTL-IIoT datasets, as well as oversampling techniques (random oversampling (ROS), synthetic minority oversampling technique (SMOTE), adaptive synthetic sampling (ADASYN), and borderline-SMOTE (BLS)) and undersampling methods (random undersampling (RUS), Cluster Centroids (CC), Tomek Links (TL), and NearMiss (NM)).
3) Development of MSDL models for attack-specific detection using the X-IIoTID and WUSTL-IIoT datasets, with an evaluation based on classification performance (accuracy, precision, recall, F1 score) and computational efficiency (training and testing times).
4) Comparison of performance and efficiency of our proposed method with oversampling and undersampling techniques.

TABLE 1. Review of related work.

Ref	Year	Dataset		oversampling				undersampling				Multi-stage
		X-IIoTID	WUSTL-IIoT	SMOTE	ROS	ADASYN	BLS	RUS	CC	TL	NM	
[15]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[16]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[17]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[18]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[19]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[20]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗
[21]	2024	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
[22]	2024	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
[23]	2023	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗
[24]	2024	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
[25]	2024	✗	✓	✓	✗	✗	✗	✗	✗	✗	✗	✗
[26]	2024	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
[27]	2024	✗	✓	✓	✓	✗	✗	✓	✗	✗	✗	✗
[11]	2021	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[28]	2021	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[29]	2023	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[30]	2023	✗	✗	✓	✓	✓	✓	✓	✗	✓	✓	✗
[31]	2023	✗	✗	✓	✓	✗	✗	✓	✗	✗	✗	✗
[32]	2024	✗	✗	✓	✗	✗	✗	✗	✗	✗	✗	✗
[33]	2023	✗	✗	✓	✗	✗	✗	✗	✗	✓	✗	✗
[34]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
[35]	2023	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
[34]	2021	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
[36]	2022	✗	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
[37]	2024	✓	✗	✗	✗	✗	✗	✗	✗	✗	✗	✓
Ours	2024	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

The remainder of this paper is organized as follows. Section II reviews related work. Section III discusses IIoT in relation to critical infrastructure, addressing IIoT architecture, cyber threat landscape, and the cyber threat model. Section IV presents the multi-stage intrusion detection system for IIoT, detailing the datasets utilized and the experimental setup. Section V analyzes and discusses the results of our experiments and compares them with state-of-the-art methods from the recent related literature. Finally, Section VI summarizes our findings and outlines future research directions.

II. REVIEW OF RELATED WORK

Research on IDS for IIoT environments has advanced significantly, largely because of the availability of datasets such as X-IIoTID [38] and WUSTL-IIoT [2], which contain real-world IIoT network traffic and relevant attack types.

Numerous studies have used the X-IIoTID dataset to develop ML and DL models for detecting cyber threats. Al-Hawawreh et al. [38] proposed a methodology for generating a comprehensive intrusion dataset specifically for IIoT systems. This approach involved several stages: IIoT testbed setup, threat modeling, attack generation, data collection, feature extraction, and exploratory data analysis. The X-IIoTID dataset was then evaluated using a variety of ML and DL algorithms, including Decision Tree (DT), Naive Bayes (NB), K-Nearest Neighbor (KNN), Support Vector Machine (SVM), Logistic Regression (LR), Deep Neural Networks (DNN), and Gated Recurrent Units (GRU) for both binary and multi-class classification tasks.

Attique et al. [15] introduced an IDS for IIoT environments employing a Bidirectional Long Short-Term Memory (BiLSTM) model integrated with a self-adaptive attention mechanism. This mechanism dynamically prioritizes a dataset's critical elements, facilitating effective learning from limited data. The system incorporates the Shapley additive explanations (SHAP) technique from explainable artificial intelligence (XAI) to enhance transparency and user trust. Chaurasia et al. [16] proposed a Residual Network (ResNet)-based DL model trained using Federated Learning (FL) for intrusion detection in IIoT networks. ResNet's skip connections address the vanishing gradient problem, ensuring effective training on high-dimensional data. FL allows collaborative training while maintaining data privacy by preventing the uploading of network traffic to central servers.

Jayalaxmi et al. [17] developed a model to detect and analyze malware severity in encrypted traffic. The model utilizes a Cascading Forward Back Propagation Neural Network (CFBPNN) to detect malware and a J48 Decision Tree to evaluate malware severity. Dimensionality reduction was performed using an autoencoder, whereas the CFBPNN incorporated a cascade-correlation algorithm with backpropagation. Zidi et al. [18] proposed an IDS for the Internet of Agricultural Things (IoAT) that employed the Downsized Kernel Partial Least Squares (DKPLS) algorithm for feature extraction and dimensionality reduction, followed by classification using a Kernel Extreme Learning Machine (KELM).

Le et al. [19] addressed imbalanced multi-class data distributions in IIoT networks by proposing an IDS using extreme Gradient Boosting (XGBoost). The model effectively managed imbalanced data, fine-tuned hyperparameters, and processed missing values by sequentially adding trees to correct errors and iteratively optimizing predictions. Al-Hawawreh and Hossain [20] proposed a real-time security framework for edge private cloud systems in industrial environments, leveraging digital twin technology and online ensemble ML algorithms for real-time attack detection. The framework employs ensemble techniques such as bagging, boosting, stacking, and voting to identify malicious activities.

Several studies have leveraged the WUSTL-IIoT dataset to develop ML and DL models for cyber threat detection. Ahakonye et al. [21] integrated agnostic post-hoc explainability techniques, such as SHAP and Local Interpretable Model-Agnostic Explanations (LIME), with an ML-based IDS for IIoT. The IDS utilizes various algorithms, including DT, Random Forest (RF), Transformers, and DNN, for attack detection based on IIoT sensor and actuator data. Babayigit and Abubaker [22] proposed a hybrid DL model with optimized hyperparameters to identify malicious traffic in an IIoT. The model employed an autoencoder for dimensionality reduction and feature extraction, followed by a hybrid model of a Convolutional Neural Network (CNN) and GRU for classification. The hyperparameters were fine-tuned using Bayesian optimization. Popoola et al. [23] introduced a federated DL approach for intrusion detection in consumer-centric IoT environments. The method involved local DNN model training at the network edge and collaborative learning via a cloud server using the federated averaging (FedAvg) algorithm, ensuring privacy by retaining raw data on local devices.

A. CLASS IMBALANCE PROBLEM

Although several studies have utilized X-IIoTID and WUSTL-IIoT datasets to develop IDS models, class imbalance continues to pose a significant challenge. Although most models excel in multi-class classification tasks, they have difficulty detecting minority classes. For instance, in [23], backdoor and command injection attacks on the WUSTL-IoT dataset demonstrated limited performance, with recall values of 78.46% and 84.36%, and F1 scores of 87.93% and 89.17%, respectively. Similarly, in the X-IIoTID dataset, the exploitation attack class with fewer samples achieved a recall of 84.36% and F1 score of 90.96%. This inadequate performance in detecting minority classes raises concerns regarding the IIoT network security. Minority classes often represent low and high-impact attack types, such as backdoors, command injection, or exploitation attacks. In real-world IIoT systems, failure to detect these attacks can lead to significant vulnerabilities, allowing attackers to remain undetected and potentially cause severe disruptions. Given the critical nature of IIoT systems in fields such as manufacturing, energy, and infrastructure, undetected minority-class attacks can result in system failures, data breaches, or service interruptions. The performance gap in minority-class detection highlights the urgent need for models that can more effectively tackle class imbalance and ensure comprehensive detection across all attack types, particularly those that pose high risks to IIoT networks.

A taxonomy of recent work that has applied oversampling, undersampling, and multi-stage classification methods to address class imbalance in IDS for IIoT is presented in Table 1. This review encompasses the use of the X-IIoTID and WUSTL-IIoT datasets, along with four oversampling techniques (SMOTE, ROS, ADASYN, and BLS), four undersampling methods (RUS, CC, TL, and NM), and multi-stage classification approaches. The first column lists the reference for each study, followed by two columns indicating the datasets used, specifically X-IIoTID and WUSTL-IIoT. The subsequent columns show whether oversampling and undersampling techniques were employed. The final column indicates whether multi-stage classification methods were utilized. A check mark (✓) signifies that the technique or dataset was used, whereas a cross (✗) indicates that it was not. The "Ours" row at the bottom highlights the datasets and techniques applied in this study, distinguishing our contributions from previous work.

B. DATA SAMPLING METHODS

Data-sampling techniques are commonly used to address class imbalance by increasing the number of instances in minority classes. This paper focuses on popular oversampling methods, including SMOTE [39], ADASYN [40], BLS [40], and ROS [41].

Eid et al. [24] proposed an IDS for IIoT environments using an optimized CNN model, in which SMOTE was applied to balance the training data. A grid search was used for systematic hyperparameter tuning to improve model performance. Zukaib et al. [25] introduced a framework integrating federated learning (FL) and meta-learning to detect known and zero-day cyberattacks in Internet of Medical Things (IoMT) networks. SMOTE was employed to balance the dataset, whereas feature engineering utilized RF-based recursive feature elimination with cross-validation (RF-RFECV) and Principal Component Analysis (PCA). The three-phase meta-learning framework utilized various base learners, including DT, AdaBoost, Extra Trees (ET), and RF, followed by XGBoost as a meta-learner and FL for secure aggregation.

Eid et al. [27] evaluated six ML models (RF, DT, KNN, LR, SVM, and NB) for IDS in IIoT environments. They compared the impact of different balancing techniques, including cost-sensitive learning, RUS, ROS, SMOTE, and ensemble methods. SMOTE was found to be the most effective, which led to its adoption in subsequent experiments. Popoola et al. [28] proposed a memory-efficient deep learning method for botnet detection in IoT networks. SMOTE balanced the dataset, enabling a Deep Recurrent Neural Network (DRNN) to perform multi-class classification on low-dimensional balanced data.

Gunjal et al. [29] developed a network IDS to secure communication between Supervisory Control and Data Acquisition (SCADA) systems and IIoT devices. They used XGBoost for feature selection, SMOTE to address class imbalance, and deep learning models, such as multilayer perceptron (MLP), CNN, and Fully Connected Neural Networks (FCNN) to classify real-time network traffic. Melícias et al. [32] compared data augmentation techniques, including Generative Pre-trained Transformers (GPT) and SMOTE, to enhance multi-class intrusion detection in IIoT networks. The authors assessed the impact of these techniques on several detection algorithms, such as DT, RF, XGBoost, MLP, and TabNet.

Liu et al. [33] proposed an energy-efficient federated learning framework for intrusion detection in heterogeneous IIoT environments. The method uses an improved SMOTE based on K-Means++ to address data imbalance, generating minority class samples by interpolating between a sample and its cluster center. This approach, combined with TL for noise removal, improves classification accuracy. Thiyam and Dey [26] introduced a Causal Inference-Imbalanced Ratio (CIIR) method that combines ADASYN with instance hardness thresholds (IHT) for sampling and Boruta-ROC for feature selection. The CIIR metric evaluates the impact of sampling techniques on classification performance.

Undersampling techniques offer an alternative by reducing the number of majority class instances. This study discusses RUS, CC, TL, and NM. Cui et al. [31] proposed a multi-module intrusion detection system (GMM-WGAN-IDS) that combines a Gaussian Mixture Model (GMM)-based clustering algorithm for undersampling majority class samples with a GMM-based Wasserstein Generative Adversarial Network (WGAN) to generate minority-class samples. The system uses a CNN and Long Short-Term Memory (LSTM) networks for classification.

Some studies suggest that neither oversampling nor undersampling significantly improves the model performance. For example, Melícias et al. [32] reported no significant improvements from oversampling in their model. Huang et al. [30] found that models trained on the original dataset outperformed those trained with balanced datasets using either oversampling or undersampling methods. In addition, many existing models rely on outdated or non-specific datasets. Although Melícias et al. [32] used the recent Edge-IIoT dataset, its scope was limited to a single dataset. Other studies, such as Gunjal et al. [29], applied datasets such as CSE-CIC-IDS2018 and UNSW-NB15, which are not designed for IIoT environments. Similarly, Cui et al. [31] used the NSL-KDD dataset, which is outdated and non-IoT-specific. These limitations highlight the need for further research on developing IIoT-specific datasets and models to address the unique challenges of IIoT environments.

C. MULTI-STAGE CLASSIFICATION METHOD

Multi-class classification in intrusion detection systems (IDS) provides significant advantages over binary classification by accurately identifying specific attack types, thereby facilitating targeted threat mitigation strategies. Nonetheless, traditional multi-class classifiers commonly encounter challenges arising from severe class imbalance, frequently resulting in the misclassification of malicious traffic as benign. Multi-stage classification frameworks have emerged to mitigate this issue, decomposing the overall classification task into sequential stages. These methods typically initiate by distinguishing benign from malicious traffic, refining classification into more precise attack categories.

Multi-class classification within IDS offers notable benefits over binary classification by enabling more precise differentiation among specific attack types, thus supporting targeted mitigation strategies. Nevertheless, high class imbalance often leads to frequent misclassification of malicious traffic as benign. Hierarchical or multi-stage classification frameworks have emerged to address this challenge, dividing classification tasks into sequential layers or stages. Typically, these methods separate benign from malicious traffic, followed by progressively finer-grained differentiation among specific attack classes.

Mohd et al. [34] proposed a hierarchical multi-level intrusion detection structure that begins with a high-level classification (normal vs. malicious) and then further subdivides malicious traffic into Denial of Service (DoS), Root to Local (R2L), User to Root (U2R), and probing attacks. Each hierarchical level incorporates an ensemble of machine learning classifiers, such as SVM, probabilistic neural networks, DT, neural-fuzzy classifiers, smooth SVM, and kNN, with the best-performing models at each level combined into the final system. This explicit stage-by-stage setup ensures that a decision made at the first level (e.g., distinguishing benign from any attack) is refined at subsequent levels, where more specific attack types are identified. Although the hierarchical design in [34] yielded strong results on the KDD-99 dataset, later reproductions in this study on modern IIoT datasets revealed limitations in handling contemporary attack vectors and class imbalances characteristic of Industrial IoT networks.

Alzaqebah et al. [35] introduced a hierarchical classification framework leveraging Extreme Learning Machines (ELM) and an enhanced Harris Hawks optimization method. Their system also employs a multi-stage approach: at each stage, binary classifiers differentiate a targeted attack category from other classes, with feature selection and model weights optimized via meta-heuristic searches. Decomposing the problem into smaller binary tasks aims to reduce overall classification complexity and improve performance for multiple attack classes. However, their evaluation primarily used the UNSW-NB15 dataset, which is less reflective of the specialized protocols and class imbalance scenarios found in IIoT environments, constraining the real-world applicability of their hierarchical approach.

ElDahshan et al. [36] similarly employed a hierarchical strategy built upon meta-heuristic-optimized ELM. While details in their work indicate a pipeline of sub-classifiers, each focusing on key decision boundaries (e.g., normal vs. attack, followed by specialized sub-classification for various attack categories), their methodology relied heavily on dataset-level preprocessing (e.g., undersampling) rather than dynamic class isolation at each hierarchical layer. This pipeline-based structure demonstrated promising detection for common attacks but encountered challenges with minority-class detection, particularly because the chosen datasets (UNSW-NB15 and CICIDS-2017) do not incorporate the more nuanced IIoT-specific threats or heavily skewed distributions that are frequent in industrial environments.

Uddin et al. [37] proposed a three-level hierarchical IDS to reduce false negatives. Their method involves an initial binary split (normal vs. malicious), a broader attack-family classification, and ultimately a finer-grained distinction within each family. This tiered scheme is beneficial for delineating high-level and sub-level attacks. However, because their experiments were conducted on generic IDS datasets rather than specialized IIoT datasets, the hierarchical structure does not fully account for IIoT-centric attack behaviors, class imbalances, or specialized industrial protocols, factors essential to robust detection in IIoT networks.

In contrast to the above hierarchical methods, our proposed MSDL framework explicitly targets the severe class imbalance typical of modern IIoT datasets by recursively isolating low-performing (minority) classes at each stage. Specifically, if any class fails to meet a set performance threshold in the initial classification, MSDL generates a dedicated binary classifier to distinguish that minority class from the remaining attacks, an iterative strategy that ensures every underrepresented threat category receives specialized attention. Evaluations conducted on X-IIoTID and WUSTL-IIoT, both representative of genuine IIoT traffic and attack patterns, demonstrate that MSDL not only outperforms these earlier hierarchical approaches in terms of precision, recall, and F1-scores but also significantly reduces training time (by up to 50%), thus supporting real-time intrusion detection. Through this adaptive, stage-wise dataset partitioning, MSDL advances the state of the art in hierarchical intrusion detection by offering both high detection performance and practical scalability for IIoT-specific threats.

III. IIoT IN CRITICAL INFRASTRUCTURE

IIoT is extensively used in critical infrastructure sectors such as energy and utilities, transportation, and manufacturing, to improve efficiency, safety, and resilience by integrating legacy systems with advanced sensors, controls, communications, and data processing technologies [42]. This enables real-time monitoring, automation, and data-driven decision-making. In energy and utilities, the IIoT monitors and controls smart grids, facilitates advanced metering, and manages water treatment facilities, enhancing energy distribution and predictive maintenance [43]. In transportation, IIoT supports applications such as intelligent traffic management, real-time fleet tracking, and vehicle-to-infrastructure communications, improving the efficiency and safety on roads and in transit systems [44]. In manufacturing, IIoT forms the backbone of smart factories, integrating networked sensors, robotics, and automation systems for real-time production monitoring, quality control, and predictive maintenance, which collectively boost productivity and minimize downtime [45].

A. IIoT ARCHITECTURE

The Industrial Internet Reference Architecture (IIRA) is a comprehensive, standards-based framework designed to guide the development of interoperable IIoT systems [46]. IIRA's conceptual model is built around a three-tier architecture that provides a clear, modular framework for designing IIoT systems. At a high level, the model divides the IIoT system into three distinct layers: edge, platform, and enterprise.

1) EDGE LAYER

This is the foundational tier of IIoT systems that acts as a direct interface with the physical world. In this tier, physical devices, such as sensors, actuators, controllers, and other embedded systems, capture real-time data directly from industrial processes or environments. These devices communicate using various communication protocols, both wired and wireless.

2) PLATFORM LAYER

This is the intermediate tier of IIoT systems, positioned between the edge and higher-level enterprise layers. In this tier, aggregated data from edge gateways and devices are processed, stored, and analyzed to enable real-time decision-making and broader system optimization. The platform layer typically resides in data centers or private/hybrid clouds, offering more substantial computing, networking, and storage resources than the edge tier.

3) ENTERPRISE LAYER

This is the top-level tier of IIoT systems, where the critical strategic, financial, and administrative functions converge. This layer leverages the processed data and insights from the platform layer to guide high-level decision making, align operational objectives with corporate strategies, and ensure regulatory compliance across the organization. Enterprise-wide applications for data analytics, business intelligence, and centralized asset management typically reside in large data centers or corporate clouds, providing a unified interface for stakeholders such as executives, compliance officers, and IT administrators.

B. CYBER THREAT LANDSCAPE

IIoT has significantly expanded the cyber threat landscape across critical infrastructure sectors by increasing the potential entry points and vulnerabilities [43]. In the energy and utilities sector, IIoT integration exposes legacy SCADA systems and smart grid components to vulnerabilities such as weak authentication, insecure remote access interfaces, and inadequate network segmentation. These vulnerabilities can allow unauthorized control or manipulation of the vital infrastructure [47]. In transportation, connected sensors and control systems, essential for traffic management and autonomous vehicle operation, are at risk owing to insecure communication protocols, insufficient encryption, and flawed device authentication, making them susceptible to manipulation or remote hijacking [48]. In manufacturing, the adoption of IIoT for real-time monitoring and automation introduces weaknesses in industrial protocols (e.g., Modbus), outdated legacy systems, and insufficient network isolation, which can enable lateral movement by attackers and may lead to disruptions or sabotage of production [49].

Cyber attackers are actively exploiting vulnerabilities introduced by IIoT in critical sectors such as energy and utilities, transportation, and manufacturing. In energy and utilities, the 2016 Ukraine power grid attack employed Industroyer malware to exploit legacy IIoT protocols, leading to widespread outages [50]. In contrast, the 2021 Colonial Pipeline ransomware incident underscored risks in converged IT/IIoT networks [51]. Additional incidents include targeting a Florida water treatment plant in 2021, where hackers breached IIoT-connected SCADA systems and attempted to poison the water supply by altering the chemical levels [52]. Similarly, in 2020, Energias de Portugal (EDP) experienced a Ragnar Locker ransomware attack that disrupted renewable energy operations by compromising IIoT devices managing wind farms [53]. In 2019, a USA natural gas compressor facility faced emergency shutdown after attackers exploited phishing vulnerabilities to infiltrate IIoT-enabled ICS [54].

The transportation sector has experienced rising threats, such as the 2020 ransomware attack on the Toll Group, a major Australian logistics company, which paralyzed IIoT-connected freight management systems and halted global shipments [55]. The 2022 Danish State Railways attack disrupted the IIoT-enabled signaling systems [56]. Additionally, the 2017 NotPetya malware assault on the CSX Corporation hampered IIoT logistics networks through compromised software updates [57]. In 2023, a cyberattack on the UK's Royal Mail disrupted international parcel tracking systems reliant on IIoT sensors [58]. Meanwhile, in 2021, hackers infiltrated San Francisco's Muni Metro ticketing systems by exploiting unsecured IIoT payment kiosks to demand ransom [59]. Another incident includes the 2023 breach of the Port of Nagoya, Japan's largest maritime hub, where ransomware incapacitated IIoT cargo-handling equipment for days, resulting in $420 million delays [60].

In manufacturing, attackers persist in exploiting IIoT vulnerabilities. The 2017 Triton malware assault on a Saudi Arabian petrochemical plant manipulated IIoT safety systems [61], and the 2019 LockerGoga ransomware attack on Norsk Hydro spread from Information Technology (IT) systems to IIoT production lines [62]. The 2021 JBS Foods ransomware attack disrupted meat processing plants globally by targeting IIoT-enabled refrigeration and inventory systems [63]. In 2022, Toyota ceased production at 14 plants after a supplier's IIoT-based parts-ordering system was compromised [64]. Similarly, Honda encountered a cyberattack in 2020 that disrupted IIoT assembly line robots, leading to $1 billion in lost outputs [65]. The 2023 Clorox ransomware attack compelled the company to shut down IIoT-connected production lines for weeks, highlighting the risks in supply chain automation [66]. German wind turbine manufacturer Nordex also faced a 2022 attack that hijacked IIoT devices to monitor turbine performance, necessitating a global IT or Operational Technology (OT) shutdown [67].

C. CYBER THREAT MODEL

Cyberattacks on IIoT-enabled critical infrastructure follow a structured, multi-stage process to compromise systems and achieve malicious objectives. These attacks typically unfold in distinct phases, starting with reconnaissance to identify vulnerabilities in IIoT devices and networks, followed by weaponization to create tailored exploits. Attackers then deploy these tools during delivery by employing command injection or exploiting insecure protocols to gain initial access. Once inside, they establish persistence through backdoor installation, allowing for undetected re-entry, and perform lateral movements to penetrate deeper into interconnected IT/OT systems. Command and control (C&C) channels are established to orchestrate attacks remotely, culminating in the execution of malicious activities, including tampering with industrial processes, data exfiltration, deploying crypto-ransomware to paralyze operations, and launching DoS and ransom-driven DoS (RDoS) attacks to disrupt services. Each stage builds upon the previous stage, creating a cascading effect that exploits the interdependencies inherent in IIoT ecosystems. Notable frameworks that encapsulate these multi-stage attack patterns include the Lockheed Martin Cyber Kill Chain [68], which outlines stages from reconnaissance to actions on objectives; the MITRE ATT&CK framework [69], which provides a detailed matrix of adversarial tactics and techniques; and the Diamond Model [70], which offers a structured approach to correlate adversary, capability, infrastructure, and victim attributes during intrusions.

1) RECONNAISSANCE

This is the initial phase of a cyber attack, in which adversaries gather intelligence about a target to identify vulnerabilities and attack vectors. Attackers focus on mapping interconnected devices, protocols, and network architecture. This stage is critical because IIoT systems often rely on legacy technologies, insecure communication protocols, and poorly segmented networks, which makes them attractive targets. In the energy and utilities, reconnaissance centers on spotting vulnerabilities in IIoT systems such as smart grids, pipeline controls, and SCADA networks, often targeting legacy protocols (e.g., Modbus), unsecured remote access points, or misconfigured sensors. These weaknesses could enable attacks ranging from power grid sabotage (e.g., industroyer malware) to tampering with water treatment systems. In transportation, attackers map IIoT-enabled infrastructure, such as railway signaling, traffic management networks, and connected logistics tools, probing insecure Application Programming Interfaces (APIs), Global Positioning System (GPS) trackers, or exposed communication brokers (e.g., Message Queuing Telemetry Transport (MQTT)) to later disrupt operations. For manufacturing, reconnaissance prioritizes IIoT devices such as PLCs, robotic arms, and supply chain automation tools, identifying weaknesses in unpatched firmware, unencrypted sensor communications, or vendor-specific ICS software (e.g., Siemens SIMATIC), which adversaries exploit to halt production lines, steal intellectual property, or deploy ransomware.

2) WEAPONIZATION

This is the stage in a cyberattack where adversaries develop or adapt malicious tools (e.g., malware, ransomware, and exploit kits) to exploit vulnerabilities identified during reconnaissance. In IIoT environments, this often involves tailoring payloads to target specific industrial protocols, devices, or network architectures to ensure the attack can bypass defenses and achieve its intended disruption or compromise. In the energy and utilities, weaponization might involve crafting malware such as Industroyers to exploit legacy grid protocols (e.g., IEC 60870-5-104) or designing ransomware to shut down pipeline control systems, as seen in the Colonial Pipeline attack. For transportation, attackers weaponize threats by developing ransomware tailored to cripple IIoT logistics networks (e.g., the 2023 Port of Nagoya attack) or by creating spoofing tools to hijack GPS trackers in connected fleets. In manufacturing, weaponization often focuses on malware that targets PLCs (e.g., Triton, which manipulates safety systems) or ransomware designed to paralyze IIoT-driven production lines, such as the LockerGoga attack on Norsk Hydro.

3) DELIVERY

This stage of a cyberattack involves adversaries deploying their weaponized tools or exploiting them in the target environment to gain initial access. During this phase, malicious payloads (e.g., malware, ransomware, or exploit code) are transmitted through vectors that exploit vulnerabilities in industrial systems. Common delivery methods include phishing emails aimed at employees accessing IIoT management interfaces, compromised software updates from vendors, or exploiting unsecured communication protocols (e.g., MQTT and OPC Unified Architecture (OPC-UA)) to inject harmful commands. For instance, attackers may deliver ransomware through a phishing link to an engineer in the energy sector, hijack unpatched IIoT gateways in transportation systems to disrupt rail signaling, or exploit weak authentication in manufacturing PLCs to upload harmful firmware. Successful delivery allows attackers to breach defenses and set the stage for the subsequent phases. The effectiveness of this stage depends on the exploitation of human, procedural, or technical weaknesses in IIoT ecosystems.

a: COMMAND INJECTION

This technique involves cyberattacks, where adversaries exploit insecure interfaces to inject malicious commands into a system, often bypassing the intended controls. In IIoT environments, attackers manipulate inputs (e.g., user forms, API calls, or protocol messages) to execute unauthorized commands on devices such as PLCs, sensors, or gateways. This method poses significant risks to industrial systems owing to its direct interaction with physical processes. Command injection in the energy and utilities sector can target SCADA systems or smart meters through unsecured web interfaces or legacy protocols (e.g., Modbus), allowing attackers to manipulate grid operations or disrupt pipeline pressure controls. This can result in blackouts, equipment damage, or even environmental disasters (e.g., overriding gas leak detectors). In transportation, adversaries may inject malicious commands into IIoT-enabled traffic management systems (e.g., via GPS tracker APIs) to reroute vehicles, sabotage railway signaling, or disable port cargo equipment, potentially leading to derailments, supply chain delays, or safety crises. In manufacturing, exploiting insecure PLC interfaces or robotic arm control panels might enable attackers to halt production lines, tamper with safety protocols (e.g., disabling emergency shutdowns), or corrupt quality-control systems, resulting in costly downtime, risks to worker safety, or product recalls.

b: EXPLOITATION

This phase involves attackers leveraging identified vulnerabilities to perform malicious actions, such as deploying payloads or gaining unauthorized access. In IIoT systems, this often means exploiting weaknesses in software, protocols, or configurations (e.g., unpatched firmware, insecure APIs, or default credentials) to breach devices such as PLCs, sensors, or gateways. Unlike simple delivery, exploitation ensures an attacker's tools are activated within the target environment. Exploitation might involve exploiting vulnerabilities in legacy protocols (e.g., IEC 60870-5-104) or unpatched SCADA systems to hijack grid controls in the energy and utilities sector. For transportation, attackers can exploit insecure GPS tracker APIs or outdated firmware in railway signaling systems to disrupt logistics (e.g., rerouting shipments or halting trains). In manufacturing, exploitation often targets unsecured OPC-UA servers or vulnerable PLC firmware to halt production lines, steal intellectual property, or deploy ransomware.

4) BACKDOOR INSTALLATION

This involves creating covert and persistent access points in a system that enable attackers to bypass normal authentication and regain remote control. This often entails embedding malware (e.g., rootkits) or modifying firmware on devices such as PLCs, sensors, or gateways to maintain stealthy, long-term access. In the energy and utilities sector, backdoors might be implanted in grid control systems (e.g., Remote Terminal Units (RTUs) or SCADA servers) to facilitate future grid manipulation, as seen with the Industroyer, which allows attackers to trigger power outages remotely. For transportation, backdoors could target IIoT-enabled railway signaling systems or port cargo networks (e.g., via compromised GPS trackers), permitting adversaries to disrupt logistics or reroute shipments covertly. In manufacturing, attackers may install backdoors in PLCs or robotic arms, akin to the persistence of Triton malware in safety systems, to sabotage production lines, steal proprietary data, and stage ransomware attacks.

5) LATERAL MOVEMENT

This refers to attackers' technique to pivot from an initially compromised system to other devices or networks within an organization, thereby escalating access to critical assets. This often involves exploiting interconnected IT/OT networks, weak authentication between devices, or shared credentials to infiltrate deeper into the ICS, PLCs, or sensor networks. In the energy and utilities sector, lateral movement can involve attackers progressing from a breached corporate IT network to OT systems that control power grids or pipelines. They exploited unsegmented networks to tamper with SCADA systems or safety sensors. For transportation, adversaries may move from a compromised logistics management tool to IIoT-enabled railway signaling or port cargo systems. In manufacturing, lateral movement targets converged IT/OT environments, allowing attackers to jump from infected workstations to PLCs or robotic arms.

6) COMMAND AND CONTROL

This refers to the infrastructure and communication channels that attackers use to remotely control compromised systems, exfiltrate data, and orchestrate malicious activity. C&C frequently utilizes encrypted or protocol-mimicking communication (e.g., hiding within industrial protocols such as MQTT or OPC-UA) to evade detection while retaining persistent control over devices such as PLCs, sensors, and gateways. In the energy and utilities sector, C&C channels may intercept grid communication protocols (e.g., Modbus TCP) to covertly manipulate power distribution or pipeline valves, as demonstrated in attacks such as Industroyer, which employs C&C servers to coordinate grid disruptions. For transportation, attackers can exploit GPS tracking systems or railway signaling networks to reroute shipments or disrupt traffic management, akin to ransomware campaigns that remotely disable port cargo IIoT systems. In manufacturing, the C&C infrastructure might compromise PLCs or robotic arms through infiltrated OPC-UA servers, enabling adversaries to sabotage production lines or exfiltrate proprietary designs. C&C capitalizes on the connectivity of IIoT ecosystems to escalate attacks silently, posing risks of prolonged operational paralysis, safety breaches, and large-scale data theft.

This is the final stage of a cyberattack where adversaries activate their payloads to achieve objectives such as disrupting operations, stealing data, or causing physical harm. This phase leverages the access and control gained in earlier stages (e.g., backdoors and lateral movement) to interfere directly with industrial processes, devices, or data integrity. Attackers employ tactics tailored to exploit the interconnected real-time nature of IIoT systems, often targeting the convergence of IT and OT.

a: TAMPERING

This involves deliberately altering IIoT systems, data, or processes to disrupt operations, compromise safety, or inflict physical damage. This could mean manipulating sensor readings, overriding device commands, or corrupting firmware to destabilize industrial workflows. In the energy and utilities sector, tampering may involve falsifying pressure or temperature data in pipeline SCADA systems to trigger explosions or overloading smart grid sensors to cause cascading blackouts. For transportation, attackers can tamper with IIoT-enabled railway signaling systems to force incorrect track switches or manipulate traffic light controls to create a gridlock. In manufacturing, tampering can alter robotic arm programming to damage equipment, corrupt quality-control sensor data to release defective products, or override PLC safety protocols.

b: DATA EXFILTRATION

This refers to the unauthorized extraction of sensitive information from a system, often for espionage, sabotage, or financial gain. It involves stealing operational data (e.g., sensor readings, control logic, proprietary configurations) or sensitive intellectual property from interconnected industrial devices such as PLCs, SCADA systems, or cloud-based analytics platforms. Data exfiltration may target smart grid configurations, pipeline pressure logs, or customer usage patterns in the energy and utilities sector, enabling adversaries to replicate infrastructure for future attacks, manipulate energy markets, or sell proprietary data to competitors. For transportation, attackers can steal logistics schedules, GPS tracking data, port cargo manifests, disrupt supply chains, enable cargo theft, or expose vulnerabilities in traffic management systems (e.g., rerouting shipments based on stolen data). In manufacturing, the exfiltration of IIoT-driven production blueprints, quality-control metrics, or robotic programming codes could lead to IP theft, counterfeit production, or sabotage of product lines (e.g., leaking proprietary designs to competitors). Stolen IIoT data amplify the risks of financial loss, operational disruption, and competitive disadvantage, while also providing attackers with intelligence to stage more sophisticated follow-up attacks.

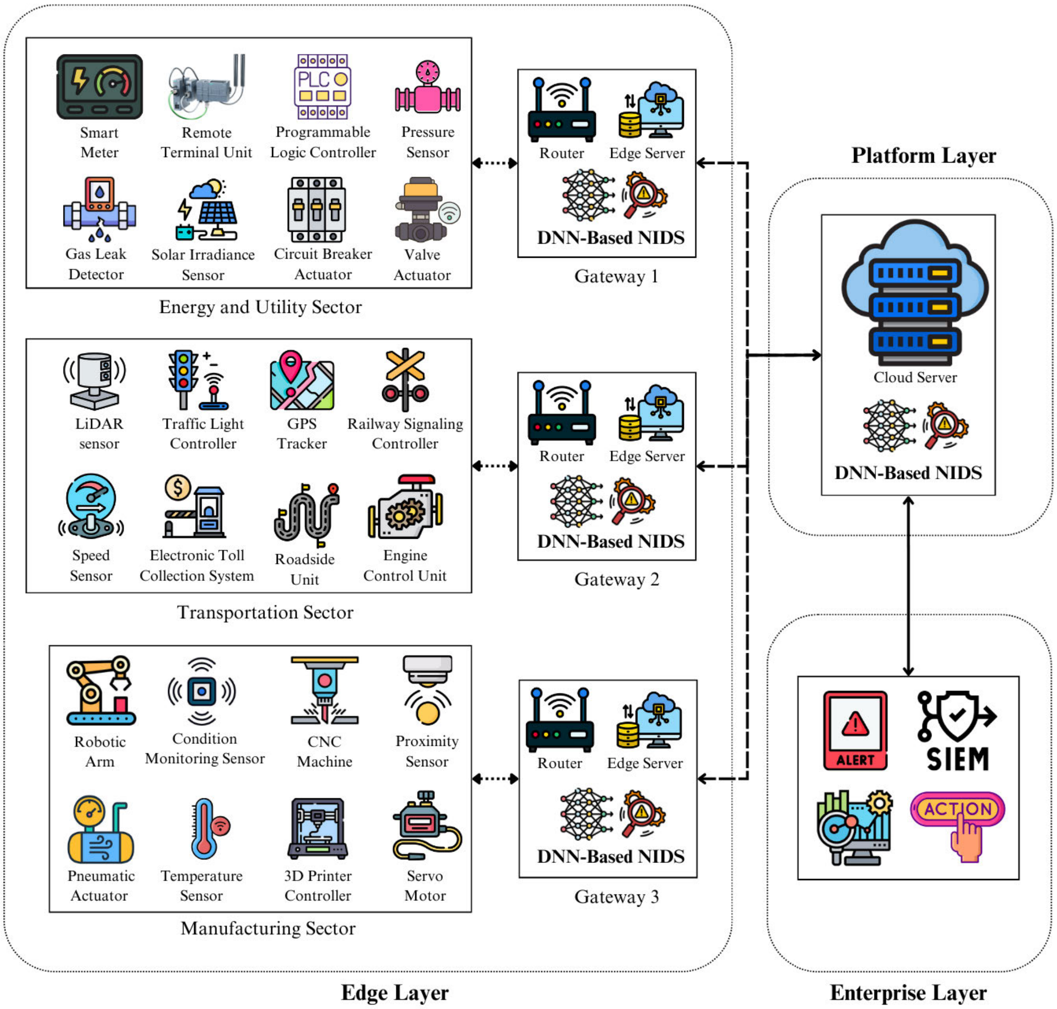


FIGURE 1. Proposed Multi-Layer Architecture for DNN-Based Intrusion Detection in IIoT-Enabled Critical Infrastructure.

c: CRYPTO RANSOMWARE

This malware encrypts critical data or systems, rendering them inaccessible until a ransom is paid. Attackers often target ICS, PLCs, or sensor networks to paralyze operations, leveraging the urgency of restoring physical processes to extort payments. In the energy and utilities sector, crypto ransomware can encrypt SCADA systems or pipeline control software, as seen in the 2021 Colonial Pipeline attack, halting fuel distribution and forcing emergency shutdowns. For transportation, ransomware may target IIoT logistics networks (e.g., port cargo systems or railway scheduling tools), as demonstrated in the 2023 Port of Nagoya attack, which froze cargo operations for days, costing millions in delays. In manufacturing, ransomware like LockerGoga has crippled IIoT-driven production lines and robotic systems, forcing companies like Norsk Hydro to revert to manual processes for weeks. Crypto ransomware exploits the reliance on real-time IIoT connectivity, causing cascading operational shutdowns, financial losses, and safety risks, such as disrupted energy grids, paralyzed supply chains, or unsafe factory conditions, while pressuring victims to pay ransoms to avoid prolonged downtime.

d: DOS AND RDOS

DoS attacks overwhelm IIoT systems with excessive traffic or requests, rendering them unavailable. RDoS combines DoS with extortion, threatening to sustain the attack unless a ransom is paid. These attacks target critical communication protocols (e.g., MQTT and OPC-UA) or device connectivity, exploiting the reliance on real-time data for industrial operations. In the energy and utilities sector, DoS/RDoS attacks can flood smart grid communication channels or SCADA systems, disrupting power distribution and triggering cascading outages. For transportation, adversaries might overwhelm IIoT-enabled traffic management networks or railway signaling systems with fake sensor data, causing gridlock or train delays. RDoS can target IIoT-driven assembly lines or inventory management systems in manufacturing, paralyzing production until ransoms are paid.

IV. MULTI-STAGE INTRUSION DETECTION IN IIoT-ENABLED CRITICAL INFRASTRUCTURE

Figure 1 shows the proposed multi-layer architecture for DNN-based intrusion detection in an IIoT-enabled critical infrastructure. Various IIoT devices and sensors across sectors, including energy and utilities, transportation, and manufacturing, continuously generate real-time network traffic at the edge layer. Local gateways capture these data streams and perform the necessary pre-processing operations to prepare raw data for further analysis. The pre-processed traffic is transmitted to the platform layer, where model development occurs. This layer utilizes aggregated data from multiple edge gateways to train and update the DNN models designed explicitly for intrusion detection. The platform layer was selected for this function because of its superior storage capacity and computational resources compared to the edge, which is constrained by power and hardware limitations. Once these models are finalized, they are distributed back to the local gateways, ensuring detection occurs at the edge and centrally at the platform layer. This dual-detection mechanism enhances threat identification and response capabilities. Finally, the enterprise layer consolidates the insights and alerts generated by these models into a Security Information and Event Management (SIEM) system, enabling rapid incident response and strategic decision-making.

A. DATA COLLECTION AT THE EDGE LAYER

In an IIoT-enabled critical infrastructure, the edge network typically comprises sensors, actuators, controllers, industrial switches, and routers for connectivity, along with local gateways for data aggregation and preliminary preprocessing. For instance, in the energy and utilities sector, substation gateways gather data from smart meters and RTUs to monitor grid performance; in transportation, roadside units collect and route traffic-flow sensor data; and in manufacturing, industrial IoT gateways oversee real-time status updates from robotic arms and condition-monitoring sensors on the production line.

Within this environment, devices fulfilling their core functions generate operational data such as sensor readings, control commands, and status updates. Because these devices exchange data with local controllers and transmit it to gateways, network traffic data are inherently produced as packets. These packets travel over wired or wireless connections, and gateways capture them by monitoring both inbound and outbound flows. Techniques such as port mirroring on managed switches, where a designated mirror port forwards copies of the traffic to the gateway, and running packet capture utilities directly on the gateway's network interfaces are commonly employed. Tools such as tcpdump,[1] Wireshark,[2] tshark,[3] and Suricata[4] facilitate real-time capture and logging of packets, which typically include source and destination IP addresses, ports, protocol headers (e.g., TCP, User Datagram Protocol (UDP)), payload data, sequence and acknowledgment numbers, and timestamps. Additional secondary data can be calculated from the raw traffic to improve intrusion detection capabilities, including flow duration, packet and byte counts, average packet size, inter-arrival times, and entropy measures. Multiple gateways within the edge network transmit the collected and preprocessed network traffic data to the platform tier, where they are further processed for the development of DL models for intrusion detection.

B. DL MODEL DEVELOPMENT AT THE PLATFORM LAYER

The platform layer acts as the central hub of an IIoT system, where network traffic data collected from multiple edge gateways are stored, processed, and analyzed. This layer has substantial computing, networking, and storage resources, typically housed in data centers or cloud infrastructures, to train DL models for intrusion detection in IIoT-enabled critical infrastructure. In our approach, the DL architecture used is a DNN.

1) DEEP NEURAL NETWORK

The proposed DNN architecture was designed to model complex patterns in IIoT network traffic data, which are typically structured in a tabular format. Whereas other deep learning architectures, such as CNNs, RNNs, and generative adversarial networks, excel in handling different types of data (e.g., images, sequential data, and generative tasks, respectively), DNNs are particularly effective for structured datasets. A DNN is a fully connected feedforward network consisting of an input layer, multiple hidden layers, and an output layer. In this architecture, each neuron in a given layer is connected to each neuron in the subsequent layer.

Let x represent the input vector, which contains the feature set of the IIoT network traffic data. The input is passed through the input layer as follows:

$$a^{(0)} = x, \tag{1}$$

where the number of neurons in the input layer corresponds to the feature count of the data. The DNN processed the input through a series of linear and non-linear transformations. Each hidden layer computes a weighted sum of its inputs, adds a bias, and applies the activation function σ_1 as follows:

$$a^{(1)} = \sigma_1 \left(W^{(1)} \cdot a^{(0)} + b^{(1)} \right), \tag{2}$$

$$a^{(2)} = \sigma_1 \left(W^{(2)} \cdot a^{(1)} + b^{(2)} \right), \tag{3}$$

$$a^{(k)} = \sigma_1 \left(W^{(k)} \cdot a^{(k-1)} + b^{(k)} \right), \tag{4}$$

where $W^{(k)}$ and $b^{(k)}$ are the weight matrix and bias vector for the kth hidden layer and $a^{(k)}$ is the output of that layer. The architecture, including the number of hidden layers, neurons, and activation functions, was determined experimentally.

[1] https://www.tcpdump.org/

[2] https://www.wireshark.org/

[3] https://tshark.dev/

[4] https://suricata.io/

The output layer operates similarly to the hidden layers but typically employs a different activation function, σ_2, suited for the classification task:

$$\hat{y} = \sigma_2\left(W^{(L)} \cdot a^{(L-1)} + b^{(L)}\right). \tag{5}$$

where $\hat{y}$ denotes the predicted output. The number of neurons in the output layer corresponded to the number of network traffic classes in the dataset.

2) MODEL TRAINING AND EVALUATION

DNN models were created to detect intrusions in IIoT networks by learning to differentiate between benign and malicious traffic patterns across various attack types. Let IIoT network traffic data be represented as D, where $X \in \mathbb{R}^{m\times n}$ is the feature set, and $Y \in \mathbb{R}^{m\times c}$ is the one-hot encoded label set. Here, m indicates the number of samples, n signifies the number of features, and c represents the number of traffic classes. Dataset D is split into a training set, D_{train}, for model development, and a test set, D_{test}, for evaluation.

Typically, IIoT datasets experience class imbalance, where majority classes are overrepresented while minority classes are underrepresented. This imbalance is measured by the Class Minority Ratio (CMR), which indicates the representation of each class i compared with the average class size:

$$\text{CMR}_i = \frac{n_i}{\frac{1}{c}\sum_{j=1}^{c} n_j}. \tag{6}$$

Minority classes have a CMR closer to zero, whereas majority classes have values close to one.

3) IIoT NETWORK TRAFFIC CLASSIFICATION

Classification tasks in IIoT can be either binary or multi-class tasks. In binary classification, a DNN model differentiates between classes such as "Normal" and "Attacks." The model employed the following sigmoid activation function:

$$\sigma_2(z) = \frac{1}{1 + e^{-z}} \tag{7}$$

which maps the outputs to a probability between 0 and 1. The performance of the model was evaluated using the binary cross-entropy loss function:

$$L(\hat{y}, y) = -\left[y\log(\hat{y}) + (1-y)\log(1-\hat{y})\right]. \tag{8}$$

In multi-class classification, the aim is to categorize the data samples into one of three or more classes. A softmax function is applied as follows:

$$\sigma_2(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{c} e^{z_j}} \quad \text{for } i = 1, \ldots, c, \tag{9}$$

which transforms the raw model outputs into probabilities. The categorical cross-entropy loss function assesses the difference between the true labels and predicted probabilities as follows:

$$L(\hat{y}, y) = -\sum_{i=1}^{c} y_i \log(\hat{y}_i). \tag{10}$$

In both the binary and multi-class settings, the weights and biases of the DNN model were iteratively adjusted using gradient descent to minimize the loss function:

$$W_{\text{new}} = W - \eta\frac{\partial L(W,b)}{\partial W}, \tag{11}$$

$$b_{\text{new}} = b - \eta\frac{\partial L(W,b)}{\partial b}, \tag{12}$$

where η denotes the learning rate. This process was repeated across multiple epochs, with each epoch representing a complete pass through of the training data.

4) CLASSIFICATION PERFORMANCE METRICS

The classification performance of the DNN models was assessed using four metrics: accuracy, precision, recall, and F1 score.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{13}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{14}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{15}$$

$$\text{F1 score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{16}$$

where True Positives (TP) and True Negatives (TN) represent the correctly predicted positive and negative samples, respectively, and False Positives (FP) and False Negatives (FN) denote incorrect predictions of positive and negative samples, respectively.

Accuracy reflects the model's overall correctness, whereas precision indicates the accuracy of the predicted positives. High precision minimizes false alarms, enhances confidence in alerts and improves the incident response efficiency. Recall evaluates a model's ability to identify true positives, indicating how effectively it recognizes actual threats. The F1 score, the harmonic mean of the precision and recall, balances these metrics. A high F1 score demonstrates the model's capability to consistently detect attacks with minimal false positives and false negatives, reducing missed threats and enhancing detection accuracy.

5) PROPOSED METHODOLOGY

Algorithm 1 describes the proposed methodology. This multi-stage intrusion detection process tackles the class imbalance in IIoT network traffic datasets to improve the classification performance of a DNN model across all classes. This approach seeks to enhance attack detection by breaking the classification task into multiple stages, starting with multi-class classification and then refining it with binary classifications using the MSDL method. end ifend if

The minimum F1 score, denoted as F_{min}, was predefined to assess the classifier's effectiveness. Initially, a multi-class DNN model (DNN_{multi}) is trained and tested on the complete IIoT network traffic dataset (D) to differentiate between benign traffic and various attack classes. This served as the

Algorithm 1 Multi-Stage Intrusion Detection

```
Input: $D$: complete IIoT network traffic dataset, $c$:
       number of classes in the dataset, $F_{min}$: minimum
       F1 score, $A$: attack data without benign samples;
       $A_{cmr}$: attack data of class with the highest CMR,
       $A_{others}$: remaining attack data without $A_{cmr}$
Output: DNN model(s)
Function multiClassifier($D$, $c$):
    Train and test DNN model using $D$ with $c$ classes;
    return F1 score for each class;
end
Function binaryClassifier($D$, $c_1$, $c_2$):
    Train and test DNN model using $D$ but $c_1$ and
    $c_2$ classes only;
    return F1 score for each class;
end
Execute multiClassifier($D$, $c$);
if F1 score for each class $\geq F_{min}$ then
    Save $DNN_{multi}$ model;
else
    Set $c_1$ = "Normal" and $c_2$ = "Attack";
    Execute binaryClassifier($D$, $c_1$, $c_2$);
    if F1 score for each class $\geq F_{min}$ then
        Save $DNN_{bin}$ model;
        Execute multiClassifier($A$);
        if F1 score for each class $\geq F_{min}$ then
            Save $DNN_{multi}$ model;
        else
            Compute the CMR for each class in $A$;
            Split $A$ into two classes: $A_{\mathrm{cmr}}$, and $A_{\mathrm{others}}$;
            Execute binaryClassifier($A$, $A_{cmr}$,
            $A_{others}$);
            if F1 score for each class $\geq F_{min}$ then
                Save $DNN_{bin}$ model;
                Update $A = A_{\mathrm{others}}$;
                Update $A_{\mathrm{cmr}}$ and $A_{\mathrm{others}}$;
                if $c = 1$ in $A$ then
                    Go to Step 23;
                else
                    Go to Step 17;
                end if
            end if
        end if
    end if
end if
```

baseline model. The F1 score for each class was compared with that of F_{min}. If all class F1 scores meet or exceed F_{min}, the model is considered effective, and the process concludes. If the multi-class model fails to achieve the necessary F1 score for all classes, the algorithm transitions to a binary classification approach. A binary DNN model (DNN_{bin}) was then trained and tested to classify data samples in D into two groups: "Normal" (benign traffic) and "Attack" (all malicious traffic combined). If DNN_{bin} meets the F1 score requirement, the algorithm conducts a new multi-class classification only on the attack data (A). If the updated DNN_{multi} does not meet the F1 score threshold, the algorithm calculates the CMR for each attack class in A. Attack data A is divided into two subsets: A_{cmr} (the attack class with the highest CMR) and A_{others} (the remaining attack data excluding A_{cmr}). The algorithm then recursively applies binary classification to A_{cmr} and A_{others}. If the new DNN_{bin} achieved a satisfactory F1 score, the model was saved, A was updated to include only A_{others}, and the algorithm returned to the multi-class classification step. If only two classes remain in updated A, the algorithm performs a final binary classification and concludes the process.

6) EXPERIMENTATION

The X-IIoTID [38] and WUSTL-IIoT [2] datasets were used to implement and assess the proposed methodology because they include real IIoT network traffic and relevant attack types. The X-IIoTID dataset comprises 59 features and 820,834 samples, whereas the WUSTL-IIoT dataset contains 41 features and 1,194,464 samples. To avoid model overfitting, irrelevant features such as temporal information (StartTime, LastTime), IP addresses (SrcAddr, DstAddr), and packet identifiers (sIpId and dIpId) were excluded from the WUSTL-IIoT dataset.

The X-IIoTID dataset consists of nine attack types: C&C, crypto ransomware, exfiltration, lateral movement, RDoS, reconnaissance, tampering, and weaponization. The WUSTL-IIoT dataset includes malicious traffic samples from backdoor, command injection, DoS, and reconnaissance attacks. Each dataset was divided into two subsets: 70% for training and 30% for testing. The distribution of IIoT network traffic samples across classes is presented in Table 2. Figure 2 shows the significant class imbalances in the datasets. In the X-IIoTID dataset, Crypto Ransomware, Exploitation, and C&C are considered minority classes because of their lower CMR, whereas Tampering, Exfiltration, Lateral Movement, Weaponization, Reconnaissance, RDoS, and Normal represent the majority classes with

TABLE 2. Distribution of IIoT network traffic data.

Dataset	Class	Code	Training	Testing
X-IIoTID	C&C	0	2029	834
	Crypto ransomware	1	299	159
	Exfiltration	2	15505	6629
	Exploitation	3	775	358
	Lateral movement	4	22195	9401
	Normal	5	294783	126634
	RDoS	6	99056	42205
	Reconnaissance	7	89297	38293
	Tampering	8	3553	1569
	Weaponization	9	47091	20169
WUSTL-IIoT	Backdoor	0	147	65
	Command injection	1	181	78
	DoS	2	54748	23557
	Reconnaissance	3	5781	2459
	Normal	4	775267	332181

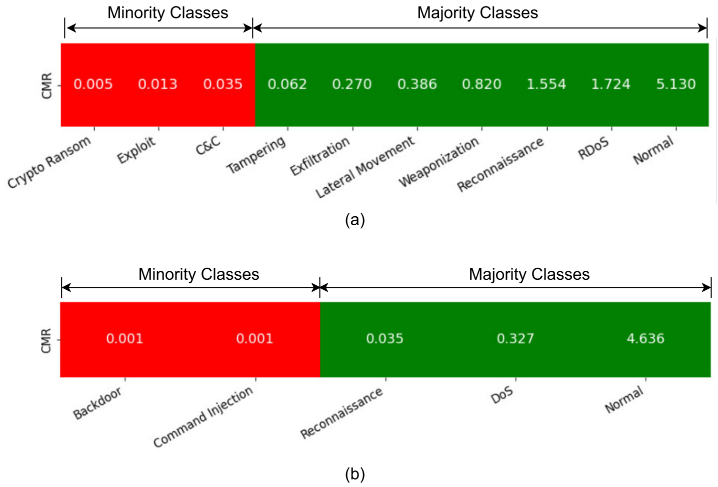


FIGURE 2. CMR and category of classes in (a) X-IIoTID and (b) WUSTL-IIoT datasets.

higher CMR. In the WUSTL-IIoT dataset, Backdoor and Command Injection were identified as minority classes with lower CMR, while Reconnaissance, DoS, and Normal were classified as majority classes with higher CMR.

The DNN model architecture consisted of three hidden layers, each containing 50 neurons. These layers are fully connected and utilize a Rectified Linear Unit (ReLU) activation function to capture complex patterns in the data. The model was compiled using the Adam optimizer, selected for its efficiency and adaptability. The learning rate (η) was set at 0.001, striking a balance between convergence speed and learning stability. The weights were updated after processing each batch of 250 samples, and the model was trained for 30 epochs. To monitor performance and reduce overfitting, 30% of the training data was reserved as a validation set, allowing for the evaluation of unseen data during training and supporting hyperparameter tuning. Simulations were conducted using Google Colaboratory,[5] which provides a cloud-based platform for running Python code. Numpy[6] was used for numerical computations and array operations. Pandas[7] managed data manipulation and analysis with its powerful data structures. Scikit-learn[8] facilitated data preprocessing. Imbalanced-learn[9] was employed to implement oversampling and undersampling techniques. TensorFlow Keras[10] enabled the development and training of deep learning models. The simulations were run on a Central Processing Unit (CPU) for computation.

V. RESULTS AND DISCUSSION

A. BASELINE DL MODELS

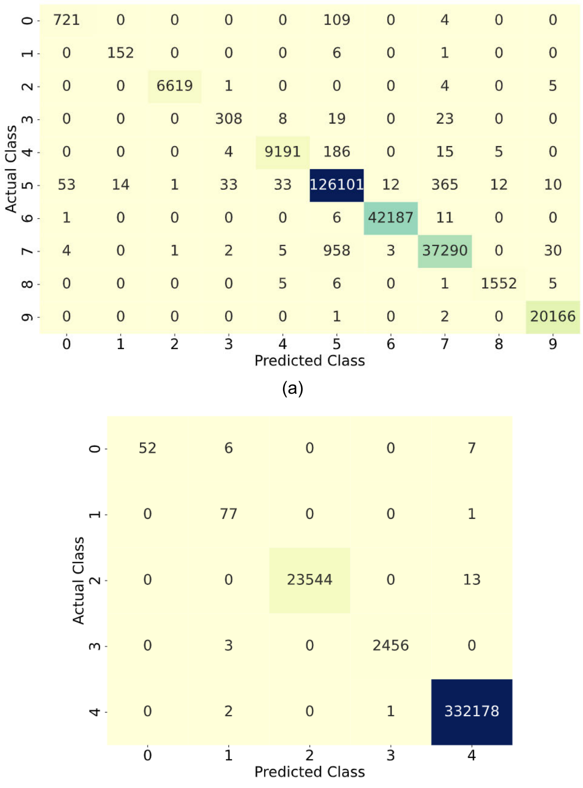
Two baseline models were trained and evaluated using the X-IIoTID and WUSTL-IIoT datasets, respectively. Table 3 presents their classification performance, with both models achieving nearly 100% accuracy across all the classes. However, this high accuracy can be misleading because of the significant class imbalance, as shown in Figure 3 and Table 2. In instances of severe class imbalance, accuracy alone is not a dependable metric for evaluating the model performance.

TABLE 3. Classification performance of the baseline DL models.

Dataset	Class	Accuracy	Precision	Recall	F1 score
X-IIoTID	**0**	99.93	**92.55**	**86.45**	**89.40**
	1	99.99	**91.57**	**95.60**	**93.54**
	2	100.00	99.97	99.85	99.91
	3	99.96	**88.51**	**86.03**	**87.25**
	4	99.89	99.45	97.77	98.60
	5	99.26	98.99	99.58	99.28
	6	99.99	99.96	99.96	99.96
	7	99.42	98.87	97.38	98.12
	8	99.99	98.92	98.92	98.92
	9	99.98	99.75	99.99	99.87
WUSTL-IIoT	**0**	100.00	100.00	**80.00**	**88.89**
	1	100.00	**87.50**	98.72	**92.77**
	2	100.00	100.00	99.94	99.97
	3	100.00	99.96	99.88	99.92
	4	99.99	99.99	100.00	100.00

The baseline models showed high precision, recall, and F1 scores for the majority classes, with very low false alarm rates ($\leq$ 1.13%) and fewer than 2.63% misclassified samples in each majority class. However, the performance of the minority classes was poor. In the X-IIoTID dataset, false positive rates were notably high: 7.45% of C&C attack samples were misclassified as RDoS attacks, Reconnaissance attacks, or benign traffic; 8.43% of Crypto Ransom attacks were misclassified as benign traffic and 11.49% of Exploitation attack samples were incorrectly identified as other types of attacks or benign traffic. Similarly, in the WUSTL-IIoT dataset, 12.5% of Command Injection attacks were misclassified as Backdoor attacks or benign traffic, whereas 20% of Backdoor attack samples were misclassified as Command Injection attacks or benign traffic. High false alarm rates

[5] https://colab.research.google.com/
[6] https://numpy.org/
[7] https://pandas.pydata.org/
[8] https://scikit-learn.org/stable/
[9] https://imbalanced-learn.org/stable/
[10] https://www.tensorflow.org/guide/keras



(a)

(b)

FIGURE 3. Confusion matrices for baseline DL models: (a) X-IIoTID dataset and (b) WUSTL-IIoT dataset.

can disrupt operations by generating frequent, unnecessary alerts, potentially desensitizing security analysts and leading to overlooked threats. Misclassifications consume valuable time and resources, diverting attention from genuine threats and potentially resulting in inadequate security measures. This highlights the need for improved deep learning methods to enhance the detection accuracy and reduce false positives for minority classes.

The training and testing times varied among the datasets. For the X-IIoTID dataset, training required 275.38 seconds, and testing took 17.39 seconds. In contrast, the WUSTL-IIoT dataset, which has 836,124 training samples and 358,340 testing samples, needed 363.73 seconds for training and 34.79 seconds for testing. The increased time is due to the larger dataset size, which requires more computational resources and time.

B. MULTI-STAGE DL MODELS

For the X-IIoTID dataset, the MSDL method produced five models: four binary models and one multi-class model, as illustrated in Figure 4. Table 4 presents the classification performance of these models. The binary models focused on benign traffic, RDoS attacks, lateral movement attacks, and reconnaissance attacks, whereas the multi-class model addresses C&C, crypto ransom, exfiltration, exploitation, tampering, and weaponization attacks. Both the binary and multi-class models achieved high precision, recall, and F1 scores, along with very low false alarm and misclassification rates (<1%). Compared to the baseline DL models, the misclassifications of C&C attacks decreased from 7.45% to 0%, and crypto-ransom attacks decreased from 8.43% to 0.75%. Similarly, the misclassification rates for exploitation attacks fell from 11.49% to 0.58%. The improved detection rates for C&C and crypto ransom attacks suggest that the MSDL method effectively reduces false positives. Table 5 compares the baseline models with the proposed models, demonstrating a significant enhancement in the detection accuracy for minority classes. These results emphasize the effectiveness of the multi-stage DL approach in boosting classification performance across diverse attack types in the X-IIoTID dataset.

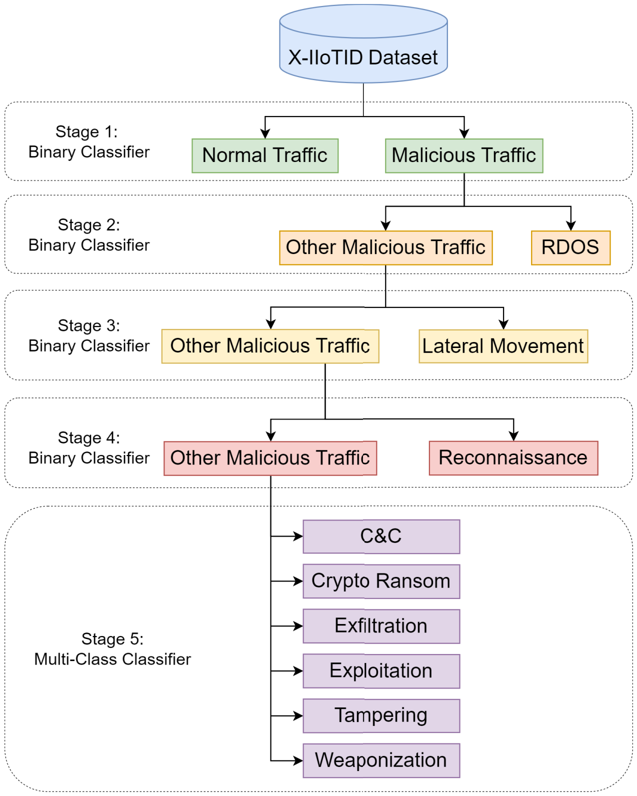


FIGURE 4. Multi-stage DL method for X-IIoTID dataset.

For the WUSTL-IIoT dataset, the MSDL method produces four binary models, as shown in Figure 5. Table 6 provides details of their performance. These models focus on identifying Benign Traffic, DoS, Reconnaissance, Backdoor, and Command Injection Attacks. The models achieved high precision, recall, and F1 scores, with false alarm rates below 1.3% and misclassification rates below 1.6%. The misclassification rate for Command Injection Attacks decreased from 12.50% to 1.28%, whereas for Backdoor Attacks, it decreased from 20% to 1.54%. Table 7 shows that

TABLE 4. Classification performance of MSDL model based on X-IIoTID dataset.

Stage	Type	Class	Precision	Recall	F1 score
1	Binary	Normal	**99.16**	**99.46**	**99.31**
		Generic Attack	99.42	99.10	99.26
2	Binary	Generic Attack	99.99	100.00	100.00
		RDoS	**100.00**	**99.99**	**99.99**
3	Binary	Lateral Movement	**99.74**	**99.66**	**99.70**
		Generic Attack	99.95	99.96	99.96
4	Binary	Generic Attack	99.90	99.78	99.84
		Reconnaissance	**99.83**	**99.92**	**99.88**
5	Multi	C&C	**100.00**	**100.00**	**100.00**
		Crypto Ransom	**99.25**	**100.00**	**99.62**
		Exfiltration	**99.98**	**100.00**	**99.99**
		Exploit	**99.42**	**99.42**	**99.42**
		Tampering	**99.87**	**99.93**	**99.90**
		Weaponization	**100.00**	**99.99**	**99.99**

TABLE 5. Classification performance of MSDL and baseline DL models based on X-IIoTID dataset.

Class	Precision		Recall		F1 score	
	Baseline	Proposed	Baseline	Proposed	Baseline	Proposed
0	92.55	100.00	86.45	100.00	89.40	100.00
1	91.57	99.25	95.60	100.00	93.54	99.62
2	99.97	99.98	99.85	100.00	99.91	99.99
3	88.51	99.42	86.03	99.42	87.25	99.42
4	99.45	99.74	97.77	99.66	98.60	99.70
5	98.99	99.16	99.58	99.46	99.28	99.31
6	99.96	100.00	99.96	99.99	99.96	99.99
7	98.87	99.83	97.38	99.92	98.12	99.88
8	98.92	99.87	98.92	99.93	98.92	99.90
9	99.75	100.00	99.99	99.99	99.87	99.99
Avg	**96.85**	**99.72**	**96.15**	**99.84**	**96.48**	**99.78**

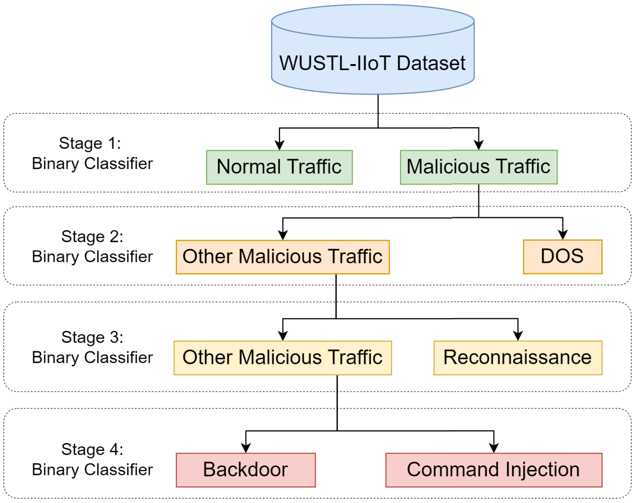


FIGURE 5. Multi-stage DL method for WUSTL-IIoT dataset.

TABLE 6. Classification performance of MSDL model based on WUSTL-IIoT-2021 dataset.

Stage	Type	Class	Precision	Recall	F1 score
1	Binary	Normal	**100.00**	**100.00**	**100.00**
		Generic Attack	99.98	99.95	99.97
2	Binary	DoS	**100.00**	**100.00**	**100.00**
		Generic Attack	100.00	99.96	99.98
3	Binary	Generic Attack	100.00	100.00	100.00
		Reconnaissance	**100.00**	**100.00**	**100.00**
4	Binary	Backdoor	**98.46**	**100.00**	**99.22**
		Command Injection	**100.00**	**98.72**	**99.35**

the proposed method significantly improved the detection accuracy for minority classes, underscoring its effectiveness on the WUSTL-IIoT dataset.

A notable reduction in false positive rates is critical for maintaining trust in IIoT systems and preventing operational disruptions caused by incorrectly flagged Benign Traffic. Improved detection accuracy for minority classes demonstrates the capability of the models to identify rare but potentially harmful attacks effectively. Lower false positive and misclassification rates allow security analysts to focus on genuine threats, leading to more efficient resource allocation and enhanced threat response. Accurate detection also reduces the risk of unnecessary network interruptions, minimizes downtime and ensures continuity of essential industrial processes.

TABLE 7. Classification performance of MSDL and baseline DL models based on WUSTL-IIoT dataset.

Class	Precision		Recall		F1 score	
	Baseline	Proposed	Baseline	Proposed	Baseline	Proposed
0	100.00	98.46	80.00	100.00	88.89	99.22
1	87.50	100.00	98.72	98.72	92.77	99.35
2	100.00	100.00	99.94	100.00	99.97	100.00
3	99.96	100.00	99.88	100.00	99.92	100.00
4	99.99	100.00	100.00	100.00	100.00	100.00
Avg	**97.49**	**99.69**	**95.71**	**99.74**	**96.31**	**99.71**

TABLE 8. Training and testing times for MSDL models.

Dataset	Stage	Train time (s)	Test time (s)
X-IIoTID	1	212.91	15.84
	2	108.96	8.30
	3	73.19	4.50
	4	75.68	4.65
	5	33.70	1.85
	Total	**504.44**	**35.13**
WUSTL-IIoT	1	319.08	31.00
	2	31.97	2.72
	3	6.85	0.77
	4	5.83	0.30
	Total	**363.73**	**34.79**

Table 8 shows that the MSDL method exhibited varying training and testing times across the X-IIoTID and WUSTL-IIoT datasets compared to the baseline models. For the X-IIoTID dataset, the MSDL method required 504.44 seconds for training and 35.13 seconds for testing, whereas the WUSTL-IIoT dataset had shorter durations of 363.73 seconds for training and 34.79 seconds for testing. The longer times observed for the X-IIoTID dataset can be attributed to its diverse class distributions. The WUSTL-IIoT dataset, where 92.72% of the samples belong to the Normal class, enabled the MSDL method to classify most samples accurately in the initial stage, thus reducing the computational burden in subsequent stages. In contrast, the more varied class distribution in the X-IIoTID dataset requires more extensive processing, resulting in longer training durations. While the MSDL method's multi-stage approach may require additional time, it ensures comprehensive and accurate classification across a broad range of classes.

TABLE 9. Classification performance of the proposed method, oversampling methods, and undersampling methods based on X-IIoTID dataset.

Metric	Code	oversampling Techniques				undersampling Techniques				Proposed
		ROS	SMOTE	ADASYN	BLS	RUS	CC	TL	NM	
Precision	0	59.02	75.94	64.20	75.10	12.33	4.98	87.21	2.56	100.00
	1	96.95	90.34	99.35	87.08	17.63	10.25	95.54	83.62	99.25
	2	100.00	99.94	99.94	99.97	91.77	86.55	99.98	94.99	99.98
	3	88.65	81.68	92.07	91.48	2.95	1.44	87.10	0.30	99.42
	4	94.69	95.28	93.44	93.69	41.38	33.41	99.62	6.51	99.74
	5	99.55	99.43	99.74	99.53	95.10	93.88	99.24	45.09	99.16
	6	99.95	99.89	99.91	99.95	97.92	94.15	99.94	45.02	100.00
	7	96.53	98.82	91.00	96.48	80.88	69.96	98.49	28.49	99.83
	8	98.36	97.26	97.32	97.37	32.16	9.65	97.87	9.74	99.87
	9	99.89	99.93	99.96	99.83	90.86	69.92	99.78	98.28	100.00
	Avg	**93.36**	**93.85**	**93.69**	**94.05**	**56.30**	**47.42**	**96.48**	**41.46**	**99.72**
Recall	0	93.41	92.33	92.45	92.57	93.65	96.64	89.93	47.60	100.00
	1	100.00	100.00	96.86	97.48	100.00	98.74	94.34	93.08	100.00
	2	99.95	99.94	99.89	99.91	99.22	98.37	99.89	28.59	100.00
	3	91.62	92.18	90.78	89.94	81.28	84.64	90.50	98.04	99.42
	4	99.11	98.80	99.30	99.06	86.83	86.07	97.46	7.71	99.66
	5	98.07	99.05	96.16	98.19	73.39	43.39	99.42	3.01	99.46
	6	99.99	99.99	99.99	99.97	99.68	99.87	99.98	86.11	99.99
	7	98.79	98.31	99.43	98.74	75.04	52.37	98.15	5.65	99.92
	8	99.55	99.55	99.36	99.04	96.88	99.17	99.62	23.58	99.93
	9	99.99	99.99	99.89	99.98	95.93	95.72	99.98	7.09	99.99
	Avg	**98.05**	**98.01**	**97.41**	**97.49**	**90.19**	**85.50**	**96.93**	**40.05**	**99.84**
F1 score	0	72.33	83.33	75.77	82.92	21.79	9.48	88.55	4.86	100.00
	1	98.45	94.93	98.09	91.99	29.97	18.58	94.94	88.10	99.62
	2	99.98	99.94	99.92	99.94	95.35	92.09	99.94	43.95	99.99
	3	90.11	86.61	91.42	90.70	5.69	2.82	88.77	0.61	99.42
	4	96.85	97.01	96.28	96.30	56.05	48.14	98.53	7.06	99.70
	5	98.80	99.23	97.92	98.85	82.84	59.35	99.33	5.65	99.31
	6	99.97	99.94	99.95	99.96	98.79	96.93	99.96	59.13	99.99
	7	97.65	98.56	95.03	97.60	77.85	59.90	98.32	9.43	99.88
	8	98.95	98.39	98.33	98.20	48.28	17.58	98.74	13.79	99.90
	9	99.94	99.96	99.93	99.91	93.33	80.81	99.88	13.23	99.99
	Avg	**95.30**	**95.79**	**95.26**	**95.64**	**60.99**	**48.57**	**96.69**	**24.58**	**99.78**

TABLE 10. Classification performance of the proposed method, oversampling methods, and undersampling methods based on WUSTL-IIoT dataset.

Metric	Code	oversampling Techniques				undersampling Techniques				Proposed
		ROS	SMOTE	ADASYN	BLS	RUS	CC	TL	NM	
Precision	0	19.17	18.24	41.61	28.64	3.28	7.01	96.30	1.01	98.46
	1	87.50	89.02	29.73	34.34	18.62	100.00	88.51	1.01	100.00
	2	100.00	100.00	99.97	99.97	82.76	95.53	99.99	72.82	100.00
	3	100.00	100.00	99.92	99.96	62.78	52.38	100.00	0.42	100.00
	4	100.00	100.00	100.00	100.00	99.91	98.84	99.99	97.64	100.00
	Avg	**81.33**	**81.45**	**74.24**	**72.58**	**53.47**	**70.75**	**96.96**	**34.58**	**99.69**
Recall	0	92.31	89.23	87.69	87.69	73.85	72.31	80.00	72.31	100.00
	1	98.72	93.59	98.72	87.18	79.49	79.49	98.72	80.77	98.72
	2	99.98	99.98	99.99	99.99	99.90	83.37	99.95	46.16	100.00
	3	100.00	99.88	100.00	99.92	89.39	99.88	99.88	56.45	100.00
	4	99.92	99.92	99.92	99.92	97.63	98.90	100.00	0.61	100.00
	Avg	**98.19**	**96.52**	**97.26**	**94.94**	**88.05**	**86.79**	**95.71**	**51.26**	**99.74**
F1 score	0	31.75	30.29	56.44	43.18	6.28	12.79	87.40	1.98	99.22
	1	92.77	91.25	45.70	49.28	30.17	88.57	93.33	1.99	99.35
	2	99.99	99.99	99.98	99.98	90.53	89.04	99.97	56.50	100.00
	3	100.00	99.94	99.96	99.94	73.76	68.72	99.94	0.83	100.00
	4	99.96	99.96	99.96	99.96	98.76	98.87	100.00	1.21	100.00
	Avg	**84.89**	**84.29**	**80.41**	**78.47**	**59.90**	**71.60**	**96.13**	**12.51**	**99.71**

TABLE 11. Training, testing, and sampling time for MSDL method and sampling techniques.

Dataset	Time (s)	Baseline	oversampling Techniques				undersampling Techniques				Proposed
			ROS	SMOTE	ADASYN	BLS	RUS	CC	TL	NM	
X-IIoTID	Training	275.38	1164.91	1158.61	1224.90	1255.83	6.45	5.68	260.89	5.71	504.44
	Testing	17.39	20.72	20.67	18.15	17.62	19.91	17.52	17.55	21.15	35.13
	Sampling	-	4.45	208.39	1751.50	1579.35	0.20	1862.70	3199.17	3.22	-
WUSTL-IIoT	Training	319.70	1444.41	1437.71	1301.77	1284.93	3.80	3.80	352.51	5.85	363.73
	Testing	25.73	25.57	25.43	25.46	24.80	25.23	24.82	25.61	26.96	34.79
	Sampling	-	1.16	22.42	406.83	381.63	0.34	349.27	6108.64	2.80	-

C. COMPARISON WITH OVERSAMPLING AND UNDERSAMPLING METHODS

The classification performances of four oversampling techniques (ROS, SMOTE, ADASYN, and BLS) and four undersampling techniques (RUS, CC, TL, and NM) were compared with the MSDL method to address class imbalance in the X-IIoTID and WUSTL-IIoT datasets. As shown in Tables 9 and 10, the MSDL method consistently outperformed these techniques across all metrics, achieving the highest average precision (>99.6%), recall (>99.7%), and F1 score (>99.7%) for both the datasets.

In the X-IIoTID dataset, BLS led to oversampling techniques with an average precision of 94.05%, whereas TL achieved the highest precision among undersampling methods at 96.48%. For recall, ROS excelled among oversampling techniques with 98.05%, and TL again led with 96.93%. SMOTE achieved the highest F1 score among the oversampling methods at 95.79%, with TL also leading to undersampling techniques at 96.69%. Although these sampling techniques performed well for the majority classes, the MSDL method demonstrated superior performance in minority class classification, outperforming the best sampling techniques by 7.98% to 14.67% in precision and F1 score for C&C and Exploitation attacks, highlighting its effectiveness in handling class imbalance.

In the WUSTL-IIoT dataset, SMOTE achieved the highest average precision of 81.45% among the oversampling techniques, while TL excelled among the undersampling methods with 96.96% precision. ROS led to recall with 98.19%, and TL performed best among undersampling techniques with 95.71%. SMOTE also had the highest F1 score among the oversampling methods at 84.89%, and TL was the top undersampling technique with an F1 score of 96.69%. Despite the strong performance of the sampling techniques for majority classes, the MSDL method surpassed them in minority class detection (Backdoor and Command Injection) by 2.25% to 13.54% across all metrics, confirming its effectiveness in highly imbalanced scenarios.

Table 11 shows that the MSDL method achieved more efficient training and testing times compared to the sampling techniques for both datasets. For the X-IIoTID dataset, the training time of the MSDL method was notably lower than that of oversampling techniques and also more efficient than undersampling methods like CC and TL when considering the entire process. Although the testing time of the MSDL method was slightly longer than most oversampling techniques, it was comparable to undersampling techniques. This increase is attributable to the multi-stage approach of the MSDL method, which, despite a longer testing time, provides better generalization and robust performance across various classes, as reflected in the improved metrics.

D. COMPARISON WITH RECENT RELATED WORK

Table 12 compares the classification performance of the MSDL method with that of recent related work using the X-IIoTID dataset. The methods from related work were reproduced with the same model parameters and computing environments to ensure a fair comparison.

TABLE 12. Comparison of the classification performance of related work based on X-IIoTID dataset.

Metric	Code	[35]	[36]	[34]	[37]	MSDL
Precision	0	91.58	0.76	99.52	96.48	100.00
	1	97.41	0.07	100.00	98.55	99.25
	2	100.00	4.62	99.98	99.93	99.98
	3	73.33	0.39	95.45	96.58	99.42
	4	99.74	5.91	99.82	99.35	99.74
	5	99.16	91.88	99.16	99.16	99.16
	6	99.96	25.18	100.00	99.99	100.00
	7	98.72	20.37	100.00	99.92	99.83
	8	99.16	0.93	99.22	99.55	99.87
	9	99.98	16.47	99.97	99.94	100.00
Recall	0	83.78	52.40	95.92	99.18	100.00
	1	94.96	32.08	94.12	100.00	100.00
	2	99.88	72.14	99.89	99.97	100.00
	3	78.81	61.73	90.00	85.98	99.42
	4	97.43	36.13	99.81	99.86	99.66
	5	99.46	50.90	99.46	99.46	99.46
	6	99.97	32.08	100.00	100.00	99.99
	7	98.37	33.86	100.00	99.77	99.92
	8	98.71	24.28	99.80	99.68	99.93
	9	99.97	85.32	99.99	99.99	99.99
F1 Score	0	87.51	1.50	97.69	97.81	100.00
	1	96.17	0.14	96.97	99.27	99.62
	2	99.94	8.69	99.94	99.95	99.99
	3	75.97	0.77	92.65	90.97	99.42
	4	98.57	10.15	99.82	99.61	99.70
	5	99.31	63.55	99.31	99.31	99.31
	6	99.96	28.22	100.00	99.99	99.99
	7	98.54	25.43	100.00	99.85	99.88
	8	98.93	1.80	99.51	99.61	99.90
	9	99.97	27.61	99.98	99.96	99.99

The method presented in [35] demonstrated strong performance for the majority classes but had limitations with minority classes such as Crypto Ransomware, Exploitation, and C&C. Regarding precision, [35] achieved high scores for majority classes such as Lateral Movement, Normal, RDoS, Reconnaissance, Tampering, and Weaponization, with values ranging from 97.18% to 99.99%. However, precision dropped significantly for the minority classes: 91.58% for C&C, 97.41% for Crypto Ransomware, and 73.33% for Exploitation. In comparison, the MSDL method achieved 100.00% precision across all classes, including minority classes. For recall, [35] performed poorly with the minority classes, achieving 83.78% for C&C, 52.40% for Crypto Ransomware, and 78.18% for Exploitation. In contrast, the MSDL method significantly improved the recall by achieving 100.00% across all classes. For the majority classes, recall ranged from 97.18% to 99.99%, but the MSDL method still outperformed it. The F1 scores further highlight the limitations of [35] for minority classes, with scores of 87.51%, 51.10%, and 75.97% for C&C, Crypto Ransomware, and Exploitation, respectively. In comparison, the MSDL method achieved perfect F1 scores of 100.00% across all classes, including the minority ones. These results demonstrate that while [35] performed well for the majority classes, it struggled significantly with the minority classes. The MSDL method showed superior and consistent performance, achieving 100.00% precision, recall, and F1 scores across all classes, making it particularly effective in handling imbalanced class distributions.

The method presented in [36] performed well for the majority classes but showed significant limitations for the minority classes, particularly Crypto Ransomware, Exploitation, and C&C. Regarding precision, [36] achieved high values for most majority classes, ranging from 99.29% for Reconnaissance to 99.99% for Normal. However, the precision was considerably lower for the minority classes:

0.76% for C&C, 1.07% for Crypto Ransomware, and 4.62% for Exploitation. In contrast, the MSDL method achieved 100.00% precision across all classes. Recall showed similar patterns in [36], with lower values for minority classes: 52.40% for C&C, 48.80% for Crypto Ransomware, and 46.99% for Exploitation. The MSDL method significantly improved the recall, achieving 100.00% for all classes. For the majority classes, recall ranged from 99.51% for Reconnaissance to 99.99% for Normal, but still fell short of the perfect score achieved by the MSDL method. The F1 scores for [36] were also very low for the minority classes: 0.77% for C&C, 2.12% for Crypto Ransomware, and 3.99% for Exploitation, reflecting a poor balance between precision and recall. In contrast, the MSDL method achieved perfect F1 scores of 100.00% across all classes. While [36] performed well for majority classes, its performance with minority classes was severely limited. The MSDL method demonstrated superior performance, achieving perfect results across all classes, including minority ones, highlighting its effectiveness in handling imbalanced class distributions.

The method in [34] performed well for majority classes but revealed some limitations for minority classes, particularly Crypto Ransomware, Exploitation, and C&C. Regarding precision, [34] achieved high scores for most majority classes, such as Lateral Movement, Normal, RDoS, Reconnaissance, and Weaponization, with values ranging from 99.16% to 100.00%. However, the precision for minority classes was lower: 92.52% for C&C, 90.86% for Crypto Ransomware, and 99.45% for Exploitation. In contrast, the MSDL method achieved 100.00% precision across all classes. For recall, [34] performed well for majority classes, achieving 100.00% for both Reconnaissance and Normal, but recall for the minority classes was lower: 95.92% for C&C, 91.96% for Crypto Ransomware, and 99.09% for Exploitation. The MSDL method outperformed [34], achieving 100.00% recall across all classes. The F1 scores followed a similar trend, with [34] achieving 94.17% for C&C, 91.90% for Crypto Ransomware, and 99.27% for Exploitation, while the MSDL method achieved 100.00% F1 scores for all classes. These results indicate that while [34] performed well for majority classes, it exhibited weaknesses in handling minority classes. By contrast, the MSDL method demonstrated superior and consistent performance across all classes, emphasizing its robustness in addressing imbalanced class distributions.

The method in [37] demonstrated strong performance for the majority classes but faced limitations with minority classes, particularly Crypto Ransomware, Exploitation, and C&C. For precision, [37] achieved high scores for the majority classes, including Lateral Movement, Normal, RDoS, Reconnaissance, and Weaponization, with values ranging from 98.33% to 99.92%. However, the precision for minority classes was lower: 96.48% for C&C, 98.35% for Crypto Ransomware, and 96.89% for Exploitation. In contrast, the MSDL method achieved 100.00% precision across all classes. For recall, [37] performed well for majority classes, with values ranging from 99.33% to 99.96%, but recall for minority classes was lower: 99.92% for C&C, 95.18% for Crypto Ransomware, and 95.10% for Exploitation. The MSDL method achieved 100.00% recall across all classes. The F1 scores for [37] were high for majority classes, ranging from 98.73% to 99.94%, but slightly lower for the minority classes: 98.19% for C&C, 96.74% for Crypto Ransomware, and 95.99% for Exploitation. In comparison, the MSDL method achieved perfect F1 scores of 100.00% across all classes. While [37] performed well for majority classes, it showed weaknesses in handling minority classes. However, the MSDL method demonstrated superior performance across all classes, reinforcing its effectiveness in managing imbalanced class distributions and ensuring consistent classification performance.

Table 13 compares the classification performance of the MSDL method with that of recent related works in the literature using the WUSTL-IIoT dataset. To ensure a fair comparison, methods from related studies were reproduced using the same model parameters and computing environments.

TABLE 13. Comparison of the classification performance of related work based on WUSTL-IIoT dataset.

Metric	Code	[35]	[36]	[34]	[37]	MSDL
Precision	0	94.92	1.10	100.00	95.83	98.46
	1	87.64	1.59	100.00	94.32	100.00
	2	99.98	99.96	100.00	100.00	100.00
	3	99.76	97.27	100.00	99.84	100.00
	4	100.00	99.99	100.00	100.00	100.00
Recall	0	82.35	56.92	93.24	90.20	100.00
	1	97.50	79.49	100.00	97.65	98.72
	2	99.98	99.82	100.00	99.98	100.00
	3	100.00	100.00	100.00	100.00	100.00
	4	100.00	99.46	100.00	100.00	100.00
F1 Score	0	88.19	2.17	96.50	92.93	99.22
	1	92.31	3.12	100.00	95.95	99.35
	2	99.98	99.89	100.00	99.99	100.00
	3	99.88	98.62	100.00	99.92	100.00
	4	100.00	99.72	100.00	100.00	100.00

The method proposed in [35] demonstrated high performance for majority classes (DoS, Reconnaissance, and Normal), achieving precision, recall, and F1 scores of nearly 100%. This is comparable to the MSDL method, which consistently achieved 100% across all metrics for these categories. However, [35] showed limitations in handling minority classes, particularly the Backdoor and Command Injection. For precision, it achieved 94.92% for Backdoor and 87.64% for Command Injection, whereas the MSDL method improved these values to 98.46% (+3.54%) and 100.00% (+12.36%), respectively. Similarly, for recall, [35] achieved 82.35% for Backdoor and 97.50% for Command Injection, while the MSDL method increased these to 100.00% (+17.65%) for Backdoor and 98.72% (+1.22%) for Command Injection. Regarding F1 scores, [35] reported 88.19% for Backdoor and 92.31% for Command Injection, whereas the MSDL method significantly improved these to 99.22% (+11.03%) and 99.35% (+7.04%), respectively. These results highlight the superior capability of the MSDL method to manage imbalanced class distributions, particularly by enhancing detection for minority classes without compromising the performance for majority classes.

The method in [36] demonstrated inconsistent performance across majority and minority classes. It achieved relatively high precision for majority classes: 99.96% for DoS, 97.27% for Reconnaissance, and 99.99% for Normal. However, these values are slightly lower than the MSDL method's, which attained 100% precision across all the majority classes. In contrast, [36] performed poorly for minority classes, securing precision values of only 1.10% for the Backdoor and 1.59% for Command Injection. The MSDL method surpassed these results, achieving 98.46% for the Backdoor and 100.00% for Command Injection. The recall for minority classes was also low in [36], with values of 56.92% for Backdoor and 79.49% for Command Injection. The MSDL method improved recall to 100.00% (+43.08%) for Backdoor and 98.72% (+19.23%) for Command Injection. For most classes, recall values in [36] were higher, showing 99.82% for DoS, 100.00% for Reconnaissance, and 99.46% for Normal, but still fell short of the perfect 100% recall achieved by the MSDL method. F1 scores for the minority classes in [36] were exceptionally low, at 2.17% for Backdoor and 3.12% for Command Injection. In contrast, the MSDL method attained 99.22% (+97.05%) for Backdoor and 99.35% (+96.23%) for Command Injection. For the majority classes, [36] achieved F1 scores of 99.89% for DoS, 98.62% for Reconnaissance, and 99.72% for Normal, still slightly below the perfect 100% F1 score of the MSDL method. These results underscore the significant shortcomings of [36] in handling imbalanced class distributions and their poor performance in minority classes.

The method in [34] performed well for the majority classes, achieving 100% precision, recall, and F1 scores. This is comparable to the MSDL method, which also achieved 100% for all metrics in these classes. However, [34] exhibited limitations in handling minority classes. For precision, it achieved 100% for both Backdoor and Command Injection, matching the MSDL method for Command Injection. However, the MSDL method achieved 98.46% for Backdoor. For recall, [34] reached 93.24% for Backdoor and 100% for Command Injection. The MSDL method improved recall for Backdoor by 6.76%, reaching 100%, while maintaining 98.72% for Command Injection. In terms of F1 scores, [34] secured 96.50% for Backdoor and 100% for Command Injection. The MSDL method enhanced the F1 score for Backdoor by 2.72%, achieving 99.22%, while obtaining 99.35% for Command Injection. These results indicate that while the method in [34] performed well for majority classes and showed high classification performance for minority classes, the MSDL method further improved recall and F1 scores for the Backdoor class.

The method in [37] also demonstrated high performance for the majority classes, achieving 100% precision, recall, and F1 scores. This is comparable to the MSDL method, which achieved 100% across all metrics for these classes. However, [37] demonstrated limitations in handling minority classes. For precision, it achieved 95.83% for Backdoor and 94.32% for Command Injection, while the MSDL method improved precision to 98.46% (+2.63%) for Backdoor and 100.00% (+5.68%) for Command Injection. For recall, [37] achieved 90.20% for the Backdoor and 97.65% for Command Injection. The MSDL method showed significant improvements, increasing recall for Backdoor by 9.80% to 100.00% and achieving 98.72% (+1.07%) for Command Injection. In terms of the F1 scores, [37] achieved 92.93% for Backdoor and 95.95% for Command Injection. The MSDL method improved these results, reaching 99.22% (+6.29%) for Backdoor and 99.35% (+3.40%) for Command Injection. These results demonstrate that although [37] performed well for majority classes, its performance for minority classes was lower. The MSDL method effectively addresses these limitations, showing significant improvements in precision, recall, and F1 scores, particularly for Backdoor and Command Injection. This highlights the robustness and superior capability of the MSDL method for managing imbalanced class distributions.

E. MULTI-STAGE DL MODEL DISTRIBUTED DEPLOYMENT

The training phase is centralized at the platform layer, utilizing high-performance servers, while the deployment of the multi-stage intrusion detection system is distributed across both the gateways and the platform. Deploying at gateways allows for rapid, low-latency detection of anomalous traffic directly at the network edge, ensuring an immediate local response to potential threats. This edge-level processing reduces the data volume sent upstream, conserving bandwidth and reducing the processing load on the central platform. Meanwhile, the platform layer, equipped with high-performance computing resources, conducts centralized analysis and correlates data across multiple gateways, further refining detection accuracy. This distributed deployment not only enhances scalability and load balancing but also improves system robustness and redundancy, ensuring continuous monitoring even if individual gateways encounter issues. This approach optimizes resource utilization across the entire IIoT-enabled critical infrastructure.

F. ACTIONABLE INTELLIGENCE AT THE ENTERPRISE LAYER

At the enterprise layer, outputs from the multi-stage intrusion detection system, originating from the gateways and the platform layer, are consolidated and transformed into actionable intelligence that supports strategic decision-making and risk management. This layer integrates intrusion detection alerts and threat analytics into centralized SIEM systems, providing real-time dashboards and comprehensive reports that guide executive actions and incident responses.

VI. CONCLUSION

In this paper, we propose an MSDL method to enhance intrusion detection in IIoT networks by addressing the class imbalance problem. This method divides the classification task into multiple stages. Initially, a multi-class DNN model was trained to distinguish between benign traffic and the various attack classes. If the initial multi-class model fails to meet a predefined F1 score threshold, a binary DNN model is trained to categorize data samples into ''Normal'' (benign traffic) and ''Attack'' (all malicious traffic combined). If this binary model satisfies the F1 score requirement, the algorithm performs a new multi-class classification focused solely on the attack data. The algorithm calculates the CMR for each attack class to identify the minority classes. The attack data are then divided into subsets based on the CMR, and binary classification is recursively applied to these subsets. The method evaluates the classification performance using precision, recall, and F1 scores. The MSDL method aims to achieve high precision, recall, and F1 scores across all classes, particularly enhancing the detection rates for minority-class attacks. This method was evaluated using two highly imbalanced datasets (X-IIoTID and WUSTL-IIoT), demonstrating its effectiveness in improving classification performance for diverse attack types. The MSDL method outperformed the baseline DL models and state-of-the-art oversampling and undersampling techniques, particularly in identifying minority-class attacks. Furthermore, the MSDL method exhibited more efficient training and testing times compared to existing sampling techniques, confirming its computational efficiency.

One of the critical challenges in the MSDL framework is the risk of error propagation, where errors in one stage can cascade and degrade the performance of the subsequent stages. To mitigate this risk, we set a stringent F1 score threshold of 99% at every stage of the MSDL process, ensuring that only highly reliable predictions passed through the stages. This approach maintained a high overall accuracy while minimizing error propagation across stages. In the future, we plan to explore lightweight MSDL architectures for deployment in resource-constrained IIoT devices and expand the applicability of the model to more specific IIoT domains, such as healthcare and autonomous transportation systems. This provides broader insights into the versatility and robustness of the MSDL method in diverse industrial contexts.

REFERENCES

[1] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, ''Challenges and opportunities in securing the industrial Internet of Things,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 2985–2996, May 2021.

[2] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, ''Machine learning-based network vulnerability analysis of industrial Internet of Things,'' *IEEE Internet Things J.*, vol. 6, no. 4, pp. 6822–6834, Aug. 2019.

[3] Claroty. (2022). *State of Xiot Security: 1h 2022*. Accessed: Sep. 12, 2024. [Online]. Available: https://claroty.com/resources/reports/state-of-xiot-security-1h-2022

[4] Nat. Inst. Standards Technol. (2024). *National Vulnerability Database*. Accessed: Sep. 12, 2024. [Online]. Available: https://nvd.nist.gov/

[5] J. T. Beerman, D. Berent, Z. Falter, and S. Bhunia, ''A review of colonial pipeline ransomware attack,'' in *Proc. IEEE/ACM 23rd Int. Symp. Cluster, Cloud Internet Comput. Workshops (CCGridW)*, May 2023, pp. 8–15.

[6] Cybersecurity and Infrastructure Security Agency. (2022) *Cisa cybersecurity advisory: Aa22-110a-ransomware activity targeting the healthcare and public health sector*. Accessed: Sep. 12, 2024. [Online]. Available: https://www.cisa.gov/news-events/cybersecurity-advisories/aa22-110a

[7] C. Ni and S. C. Li, ''Machine learning enabled industrial IoT security: Challenges, trends and solutions,'' *J. Ind. Inf. Integr.*, vol. 38, Mar. 2024, Art. no. 100549.

[8] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, ''Network intrusion detection for IoT security based on learning techniques,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2671–2701, 3rd Quart., 2019.

[9] S. I. Popoola, B. Adebisi, M. Hammoudeh, G. Gui, and H. Gacanin, ''Hybrid deep learning for botnet attack detection in the Internet-of-Things networks,'' *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4944–4956, Mar. 2021.

[10] R. Bocu, M. Iavich, and S. Tabirca, ''A real-time intrusion detection system for software defined 5G networks,'' in *Proc. Adv. Inf. Netw. Appl.*, vol. 227. Springer, 2021, doi: 10.1007/978-3-030-75078-7_44.

[11] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, K. Anoh, and A. A. Atayero, ''SMOTE-DRNN: A deep learning algorithm for botnet detection in the Internet-of-Things networks,'' *Sensors*, vol. 21, no. 9, p. 2985, Apr. 2021.

[12] S. M. H. Mirsadeghi, H. Bahsi, R. Vaarandi, and W. Inoubli, ''Learning from few cyber-attacks: Addressing the class imbalance problem in machine learning-based intrusion detection in software-defined networking,'' *IEEE Access*, vol. 11, pp. 140428–140442, 2023.

[13] J. M. Johnson and T. M. Khoshgoftaar, ''Survey on deep learning with class imbalance,'' *J. Big Data*, vol. 6, no. 1, pp. 1–54, Dec. 2019.

[14] A. Abdelkhalek and M. Mashaly, ''Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning,'' *J. Supercomput.*, vol. 79, no. 10, pp. 10611–10644, Jul. 2023.

[15] D. Attique, W. Hao, W. Ping, D. Javeed, and P. Kumar, ''Explainable and data-efficient deep learning for enhanced attack detection in IIoT ecosystem,'' *IEEE Internet Things J.*, vol. 11, no. 24, pp. 38976–38986, Dec. 2024.

[16] N. Chaurasia, M. Ram, P. Verma, N. Mehta, and N. Bharot, ''A federated learning approach to network intrusion detection using residual networks in industrial IoT networks,'' *J. Supercomput.*, vol. 80, no. 13, pp. 18325–18346, Sep. 2024.

[17] P. L. S. Jayalaxmi, M. Chakraborty, R. Saha, G. Kumar, and M. Conti, ''MADESANT: Malware detection and severity analysis in industrial environments,'' *Cluster Comput.*, vol. 27, no. 8, pp. 11347–11367, Nov. 2024.

[18] K. Zidi, K. Ben Abdellafou, A. Aljuhani, O. Taouali, and M. F. Harkat, ''Novel intrusion detection system based on a downsized kernel method for cybersecurity in smart agriculture,'' *Eng. Appl. Artif. Intell.*, vol. 133, Jul. 2024, Art. no. 108579.

[19] T.-T.-H. Le, Y. E. Oktian, and H. Kim, ''XGBoost for imbalanced multiclass classification-based industrial Internet of Things intrusion detection systems,'' *Sustainability*, vol. 14, no. 14, p. 8707, Jul. 2022.

[20] M. Al-Hawawreh and M. S. Hossain, ''Digital twin-driven secured edge-private cloud industrial Internet of Things (IIoT) framework,'' *J. Netw. Comput. Appl.*, vol. 226, Jun. 2024, Art. no. 103888.

[21] L. A. C. Ahakonye, C. I. Nwakanma, J. M. Lee, and D.-S. Kim, ''Machine learning explainability for intrusion detection in the industrial Internet of Things,'' *IEEE Internet Things Mag.*, vol. 7, no. 3, pp. 68–74, May 2024.

[22] B. Babayigit and M. Abubaker, ''Towards a generalized hybrid deep learning model with optimized hyperparameters for malicious traffic detection in the industrial Internet of Things,'' *Eng. Appl. Artif. Intell.*, vol. 128, Feb. 2024, Art. no. 107515.

[23] S. I. Popoola, A. L. Imoize, M. Hammoudeh, B. Adebisi, O. Jogunola, and A. M. Aibinu, ''Federated deep learning for intrusion detection in consumer-centric Internet of Things,'' *IEEE Trans. Consum. Electron.*, vol. 70, no. 1, pp. 1610–1622, Feb. 2024.

[24] A. M. Eid, B. Soudan, A. B. Nassif, and M. Injadat, ''Enhancing intrusion detection in IIoT: Optimized CNN model with multi-class SMOTE balancing,'' *Neural Comput. Appl.*, vol. 36, no. 24, pp. 14643–14659, Aug. 2024.

[25] U. Zukaib, X. Cui, C. Zheng, D. Liang, and S. U. Din, ''Meta-fed IDS: Meta-learning and federated learning based fog-cloud approach to detect known and zero-day cyber attacks in IoMT networks,'' *J. Parallel Distrib. Comput.*, vol. 192, Oct. 2024, Art. no. 104934.

[26] B. Thiyam and S. Dey, "CIIR: An approach to handle class imbalance using a novel feature selection technique," *Knowl. Inf. Syst.*, vol. 66, no. 9, pp. 5355–5388, Sep. 2024.

[27] A. M. Eid, B. Soudan, A. B. Nassif, and M. Injadat, "Comparative study of ML models for IIoT intrusion detection: Impact of data preprocessing and balancing," *Neural Comput. Appl.*, vol. 36, no. 13, pp. 6955–6972, May 2024.

[28] S. I. Popoola, B. Adebisi, R. Ande, M. Hammoudeh, and A. A. Atayero, "Memory-efficient deep learning for botnet attack detection in IoT networks," *Electronics*, vol. 10, no. 9, p. 1104, May 2021.

[29] F. Alzhouri, H. Gunjal, P. Patel, H. Ahmad, and D. Ebrahimi, "A smart network intrusion detection system for cyber security of industrial IoT," in *Proc. 4th Int. Conf. Intell. Data Sci. Technol. Appl. (IDSTA)*, Kuwai, Kuwait, 2023, pp. 67–75, doi: 10.1109/IDSTA58916.2023.10317861.

[30] M. Huang, T. Li, B. Li, N. Zhang, and H. Huang, "Fast attack detection method for imbalanced data in industrial cyber-physical systems," *J. Artif. Intell. Soft Comput. Res.*, vol. 13, no. 4, pp. 229–245, Oct. 2023.

[31] J. Cui, L. Zong, J. Xie, and M. Tang, "A novel multi-module integrated intrusion detection system for high-dimensional imbalanced data," *Int. J. Speech Technol.*, vol. 53, no. 1, pp. 272–288, Jan. 2023.

[32] F. S. Melícias, T. Ribeiro, C. Rabadão, L. Santos, and R. L. D. C. Costa, "GPT and interpolation-based data augmentation for multiclass intrusion detection in IIoT," *IEEE Access*, vol. 12, pp. 17945–17965, 2024.

[33] S. Liu, Y. Yu, Y. Zong, P. L. Yeoh, L. Guo, B. Vucetic, T. Q. Duong, and Y. Li, "Delay and energy-efficient asynchronous federated learning for intrusion detection in heterogeneous industrial Internet of Things," *IEEE Internet Things J.*, vol. 11, no. 8, pp. 14739–14754, Apr. 2024.

[34] N. Mohd, A. Singh, and H. S. Bhadauria, "Intrusion detection system based on hybrid hierarchical classifiers," *Wireless Pers. Commun.*, vol. 121, no. 1, pp. 659–686, Nov. 2021.

[35] A. Alzaqebah, I. Aljarah, and O. Al-Kadi, "A hierarchical intrusion detection system based on extreme learning machine and nature-inspired optimization," *Comput. Secur.*, vol. 124, Jan. 2023, Art. no. 102957.

[36] K. A. ElDahshan, A. A. AlHabshy, and B. I. Hameed, "Meta-heuristic optimization algorithm-based hierarchical intrusion detection system," *Computers*, vol. 11, no. 12, p. 170, Nov. 2022.

[37] M. Ashraf Uddin, S. Aryal, M. Reda Bouadjenek, M. Al-Hawawreh, and M. Alamin Talukder, "Hierarchical classification for intrusion detection system: Effective design and empirical analysis," 2024, *arXiv:2403.13013*.

[38] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3962–3977, Mar. 2022.

[39] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Jun. 2002.

[40] H. He, Y. Bai, E. A. Garcia, and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning," in *Proc. IEEE Int. Joint Conf. Neural Netw.*, Jun. 2008, pp. 1322–1328.

[41] K. Upadhyay, P. Kaur, S. Prasad, and L. Vidyapeeth, "State of the art on data level methods to address class imbalance problem in binary classification," *GIS Sci. J.*, vol. 8, no. 3, pp. 875–903, 2021.

[42] M. Alabadi, A. Habbal, and X. Wei, "Industrial Internet of Things: Requirements, architecture, challenges, and future research directions," *IEEE Access*, vol. 10, pp. 66374–66400, 2022.

[43] O. Aouedi, T.-H. Vu, A. Sacco, D. C. Nguyen, K. Piamrat, G. Marchetto, and Q.-V. Pham, "A survey on intelligent Internet of Things: Applications, security, privacy, and future directions," *IEEE Commun. Surveys Tuts.*, early access, Jul. 18, 2024, doi: 10.1109/COMST.2024.3430368.

[44] A. Hazarika, N. Choudhury, M. M. Nasralla, S. B. A. Khattak, and I. U. Rehman, "Edge ML technique for smart traffic management in intelligent transportation systems," *IEEE Access*, vol. 12, pp. 25443–25458, 2024.

[45] R. Rakholia, A. L. Suárez-Cetrulo, M. Singh, and R. Simón Carbajo, "Advancing manufacturing through artificial intelligence: Current landscape, perspectives, best practices, challenges, and future direction," *IEEE Access*, vol. 12, pp. 131621–131637, 2024.

[46] S.-W. Lin, E. Simmon, D. Young, B. Miller, J. Durand, G. Bleakley, A. Chigani, R. Martin, B. Murphy, and M. Crawford. (2022). *Industrial Internet Reference Architecture*. Industrial Internet Consortium. [Online]. Available: https://www.iiconsortium.org/wp-content/uploads/sites/2/2022/11/IIRA-v1.10.pdf

[47] H. Sarjan, A. Ameli, and M. Ghafouri, "Cyber-security of industrial Internet of Things in electric power systems," *IEEE Access*, vol. 10, pp. 92390–92409, 2022.

[48] U. Ahmad, M. Han, A. Jolfaei, S. Jabbar, M. Ibrar, A. Erbad, H. H. Song, and Y. Alkhrijah, "A comprehensive survey and tutorial on smart vehicles: Emerging technologies, security issues, and solutions using machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 25, no. 11, pp. 15314–15341, Nov. 2024.

[49] J. Leng, S. Ye, M. Zhou, J. L. Zhao, Q. Liu, W. Guo, W. Cao, and L. Fu, "Blockchain-secured smart manufacturing in industry 4.0: A survey," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 237–252, Jan. 2021.

[50] ESET Res. (2025). *Industroyer: A Cyber-weapon That Brought Down a Power Grid*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.welivesecurity.com/2022/06/13/industroyer-cyber-weapon-brought-down-power-grid/

[51] Cybersecurity Infrastructure Secur. Agency (CISA). (2025). *The Attack on Colonial Pipeline: What We've Learned & What We've Done Over the Past Two Years*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.cisa.gov/news-events/news/attack-colonial-pipeline-what-weve-learned-what-weve-done-over-past-two-years

[52] New York Times. (2025). *A Cyberattack in Florida Shows Vulnerability of U.S. Water Supply*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.nytimes.com/2021/02/08/us/oldsmar-florida-water-supply-hack.html

[53] ZDNet. (2025). *Energias De Portugal Hit By Ragnar Locker Ransomware*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.zdnet.com/article/energias-de-portugal-edp-hit-by-ragnar-locker-ransomware

[54] Wall Street J. (2025). *Cyberattack on U.S. Gas Pipeline Prompts Emergency Shutdown*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.wsj.com/articles/cyberattack-on-u-s-gas-pipeline-prompts-emergency-shutdown-11582587181

[55] Bitdefender. (2025). *Hackers Stole 220gb of Data in Toll Group Ransomware Attack*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.bitdefender.com/en-us/blog/hotforsecurity/hackers-stole-220gb-of-data-in-toll-group-ransomware-attack

[56] Reuters. (2025). *Danish Train Standstill on Saturday Caused By Cyber Attack*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.reuters.com/technology/danish-train-standstill-saturday-caused-by-cyber-attack-2022-11-03/

[57] A. Greenberg. (2025). *The Untold Story of Notpetya, the Most Devastating Cyberattack in History*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

[58] BBC News. (2025). *Royal Mail: Cyber Incident Hits Overseas Post Deliveries*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.bbc.co.uk/news/business-64244121

[59] Progressive Railroading. (2021). *San Francisco's Muni Hack: A Case Study in Prepping for Ransomware Attacks*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.progressiverailroading.com/security/article/San-Franciscos-Muni-hack-A-case-study-in-prepping-for-ransomware-attacks–50602

[60] J. Benjamin. (2025). *Ot Cybersecurity Breach Disrupts Operations At the Port of Nagoya, Japan*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.dragos.com/blog/ot-cybersecurity-breach-disrupts-operations-at-the-port-of-nagoya-japan/

[61] P. Marks. (2019). *Triton is the World's Most Murderous Malware, and It's Spreading*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.technologyreview.com/2019/03/05/103328/cybersecurity-critical-infrastructure-triton-malware/

[62] Microsoft. (2019). *Hackers Hit Norsk Hydro With Ransomware. The Company Responded With Transparency*. Accessed: Mar. 5, 2025. [Online]. Available: https://news.microsoft.com/2019/06/27/hackers-hit-norsk-hydro-with-ransomware-the-company-responded-with-transparency/

[63] Claroty. (2025). *Cyber Attack Overview: Jbs Foods Ransomware Incident*. Accessed: Mar. 5, 2025. [Online]. Available: https://claroty.com/blog/jbs-attack-puts-food-and-beverage-cybersecurity-to-the-test

[64] Reuters. (2025). *Toyota To Restart Japan Production After Cyberattack on Supplier*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.reuters.com/markets/stocks/toyota-shares-fall-after-domestic-factory-suspension-2022-03-01

[65] Control Eng. (2025). *Throwback Attack: Snake Ransomware Hits Honda Plants*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.controleng.com/throwback-attack-snake-ransomware-hits-honda-plants

[66] ThriveDX. (2025). *The Clorox Company's 2023 Cyberattack: Major Fallout, System Recovery, and Lessons Learned*. Accessed: Mar. 5, 2025. [Online]. Available: https://thrivedx.com/resources/article/clorox-companys-2023-cyberattack-fallout

[67] Can. Centre for Cyber Secur. (2022). *Cyber Threat Bulletin: Cyber Threat To Operational Technology*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.cyber.gc.ca/en/guidance/cyber-threat-bulletin-cyber-threat-operational-technology/

[68] Lockheed Martin Corp. (2011). *Cyber Kill Chain*. Accessed: Mar. 5, 2025. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[69] MITRE Corp. (2019). *Mitre Att&ck*. Accessed: Mar. 5, 2025. [Online]. Available: https://attack.mitre.org/

[70] Threat Intel Acad. (2020). *The Diamond Model of Intrusion Analysis: Summary*. Accessed: Mar. 4, 2025. [Online]. Available: https://www.threatintel.academy/wp-content/uploads/2020/07/diamond_summary.pdf

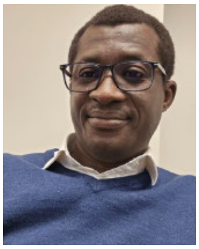
SEGUN I. POPOOLA (Member, IEEE) received the B.Tech. degree (Hons.) in electronic and electrical engineering from the Ladoke Akintola University of Technology, Nigeria, in 2014, the M.Eng. degree (Hons.) in information and communication engineering from Covenant University, Nigeria, in 2018, where he was recognized as the Overall Best Graduating Master's Student, and the Ph.D. degree in cyber security and artificial intelligence from Manchester Metropolitan University, U.K., in 2022, with a thesis on federated deep learning for botnet attack detection in IoT networks.

From 2018 to 2019, he was a Lecturer in communication engineering with Covenant University. From 2022 to 2024, he was a Lecturer in cyber security and artificial intelligence with Manchester Metropolitan University. Currently, he is a Senior Lecturer in cyber security and artificial intelligence with Anglia Ruskin University. He has authored and co-authored over 100 academic papers published in reputable journals and conference proceedings. He has contributed to various projects funded by Innovate U.K., focusing on AI, cybersecurity, and the IoT. His research interests include cybersecurity, machine learning, federated learning, the IoT, and wireless communications.

Dr. Popoola is a member of the editorial board of *Scientific Reports* journal and actively reviews for high-impact journals, such as IEEE Transactions on Industrial Informatics, IEEE Internet of Things Journal, and *Artificial Intelligence Review* (Springer). His work has been widely recognized, with citations placing him among the world's Top 2% Most-Cited Scientists, according to a study conducted at Stanford University, in 2024. He received the Global Exceptional Talent in Security and Privacy endorsement by The Royal Society, U.K., in 2022. He is a Registered Engineer with the Council for the Regulation of Engineering in Nigeria (COREN).

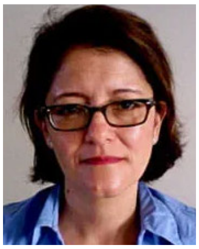
YAKUBU TSADO (Member, IEEE) received the B.Eng. degree in mechanical engineering from the Federal University of Technology, Minna, Nigeria, and the M.Sc. degree in communication systems and digital signal processing and the Ph.D. degree in electrical engineering (smart grid) from Lancaster University, U.K., where he was awarded the Centre for Global Eco-Innovation (CGE) Ph.D. Scholarship.

He is currently a Lecturer in cyber security and smart grid with the Department of Computing and Mathematics, Manchester Metropolitan University, U.K. He has previously held post-doctoral positions with Durham University and Manchester Metropolitan University, where he contributed as a Research Investigator on multiple projects, including the Horizon 2020-SmarterEMC2 Project, the Smart City (Triangulum) Project, and Nigerian Intelligent Clean Energy Marketplace (NICE) Project. He is actively engaged in advancing knowledge in these fields and has collaborated on various research initiatives aimed at enhancing smart grid technologies. His research interests include cyber security, artificial intelligence and machine learning, and digital twins in cyber-physical systems (CPS).



ABIMBOLA A. OGUNJINMI (Senior Member, IEEE) received the B.Tech. degree in electronic and electrical engineering from the Ladoke Akintola University of Technology, Nigeria, the M.B.A. degree from the University of Strathclyde, U.K., and the master's degree in information and telecommunication systems from the School of Emerging Communication Technologies, Ohio University, USA.

Since May 2012, he has been a Senior Project Deployment Manager with Nokia, where he has spearheaded digital transformation efforts, driven quality delivery, and ensured high levels of customer satisfaction. In addition to his role with Nokia, he is currently a Consulting Partner with e-Peak Systems Ltd., and the Founder and the Business Development Lead of Trioxpert Consulting. He has built a distinguished career in telecommunications and project management, with over a decade of experience leading complex, multimillion-dollar initiatives across Africa. His work spans a wide range of companies and projects, with a consistent emphasis on technology-driven solutions, stakeholder engagement, and efficient project execution within the telecom industry. He holds several professional certifications, including Project Management Professional (PMP), ITIL, and Cisco Certified Network Professional (CCNP), demonstrating a strong blend of technical expertise and managerial capability. His core competencies include project management, telecommunications infrastructure, and digital transformation. His contributions to the industry reflect his strategic insight, hands-on technical knowledge, and leadership in advancing technology project solutioning and deployment across the globe. He is a member of ISACA and ISC2 and a fellow of the Institute of Management Consultants.



ERIKA SANCHEZ-VELAZQUEZ received the M.Sc. degree in computer science from the Instituto Tecnológico de Estudios Superiores de Monterrey, Estado de México Campus, in 1999, the Ph.D. degree in electronic and electrical engineering from The University of Sheffield, U.K., in 2004, and the P.G.C.E. degree in higher education from Anglia Ruskin University, in 2014.

Currently, she is the Deputy Head of the School of Computing and Information Science, Anglia Ruskin University, Cambridge, where she also leads the Cisco Networking Academy. With over 20 years of experience in network protocols, penetration testing, software-defined networks (SDN), and the Internet of Things (IoT), she has been an active researcher and academic, focusing on computer networks and security. She began her research career developing intrusion detection systems by analyzing source code, later expanding her work to include secure protocols for mobile and wireless communication. She has led several collaborative projects funded by the British Council, CONACyT, and the Ministry of Defence, including research on AI for network automation and quantum computing in networking. Her publication record includes numerous papers in journals and conferences, covering topics, such as web-phishing detection, low-energy encryption for wireless devices, and QoS protocols for SDN. Some notable publications include works on OpenFlow communications security, botnet detection within cloud networks, and lightweight encryption schemes for wireless communication. She has also contributed to the community through invited talks, workshops, and serving on editorial boards. She has been recognized for her contributions to the field through various research grants and awards. She has been a member with the Visiting Professor Management Group, Royal Academy of Engineering, and has collaborated internationally on projects aimed at enhancing digital security. She is a member with the Cyber Security and Networking Research Group, where her research interests include SDN, the IoT security, and the development of new protocols for network management.

YONGHONG PENG (Member, IEEE) received the Ph.D. degree from South China University of Technology.

He is currently the Deputy Dean for Research and Innovation of the Faculty of Science and Engineering, Anglia Ruskin University (ARU), Cambridge, and he is a Professor of artificial intelligence. Prior to joining ARU, in June 2024, he was a Professor of artificial intelligence with Manchester Metropolitan University, where he was the Director of the University Centre for Advanced Computational Science. He holds various positions as a Visiting Professor, including with the Northern Care Alliance NHS Foundation Trust and Xiangya Hospital of Central South University. During his tenure with MMU, he provided strategic leadership in AI-powered interdisciplinary research, innovation, and partnerships, including leading MMU's membership in the Turing University Network. He also directed projects assessing cybersecurity risks to AI for U.K. Department for Science, Innovation and Technology (DSIT). He has authored numerous publications in high-impact journals, including recent works on AI frameworks for remote sensing, multimodal data analysis, and AI for health. His research has led to technological advancements and applications, particularly in enhancing AI safety and developing cooperative AI systems. His expertise spans artificial intelligence (AI), machine learning, and ethics, with a focus on algorithmic explainability, transparency, and model security.

Prof. Peng is a member of the UKRI Economic and Social Research Council (ESRC) Peer Review College and he has been recognized as an Academic Honorary Member of the Office of Health Improvement and Disparities (OHID). He is an Associate Editor of IEEE Access and an Academic Editor for *PeerJ* and *PeerJ Computer Science* and serves on the IEEE Computational Intelligence Society's Big Data Task Force.

DANDA B. RAWAT (Senior Member, IEEE) received the Ph.D. degree from Old Dominion University, Norfolk, VA, USA.

He is currently the Associate Dean of research and graduate studies, a Full Professor with the Department of Electrical Engineering and Computer Science (EECS), the Founding Director of the Howard University Data Science and Cybersecurity Center, and the Founding Director of the DoD Center of Excellence in Artificial Intelligence and Machine Learning (CoE-AIML), Howard University, Washington, DC, USA. He successfully led and established the Research Institute for Tactical Autonomy (RITA), the 15th University Affiliated Research Center (UARC), U.S. Department of Defense as the PI/Founding Executive Director of Howard University. He is engaged in research and teaching in the areas of cybersecurity, machine learning, big data analytics, and wireless networking for emerging networked systems, including cyber-physical systems (eHealth, energy, and transportation), the Internet of Things, multi-domain operations, smart cities, software-defined systems, and vehicular networks. He has secured over \$110 million as a PI and over \$18 million as a Co-PI in research funding from the U.S. National Science Foundation (NSF), the U.S. Department of Homeland Security (DHS), the U.S. National Security Agency (NSA), the U.S. Department of Energy, the National Nuclear Security Administration (NNSA), the National Institute of Health (NIH), the U.S. Department of Defense (DoD), DoD Research Labs, Industry (Microsoft, Intel, VMware, PayPal, Mastercard, Meta, BAE, and Raytheon), and private Foundations.

Dr. Rawat is a Lifetime Professional Senior Member of ACM, a Lifetime Member of the Association for the Advancement of Artificial Intelligence (AAAI), a Lifetime Member of SPIE, a member of ASEE and AAAS, and a fellow of the Institution of Engineering and Technology (IET). He was a recipient of the U.S. NSF CAREER Award, the U.S. Department of Homeland Security (DHS) Scientific Leadership Award, the President's Medal of Achievement Award (2023) at Howard University, and the Provost's Distinguished Service Award, in 2021. He has been serving as an Editor/Guest Editor for over 100 international journals, including an Associate Editor for IEEE Transactions on Information Forensics and Security, an Associate Editor for IEEE Transactions on Cognitive Communications and Networking, an Associate Editor for IEEE Transactions of Service Computing, an Editor for IEEE Internet of Things Journal, an Editor for IEEE Communications Letters, an Associate Editor for IEEE Transactions of Network Science and Engineering, and an Technical Editors for *IEEE Network*. He has been on the Organizing Committees for several IEEE flagship conferences, such as IEEE INFOCOM, IEEE CNS, IEEE ICC, IEEE GLOBECOM, and so on. He is an ACM Distinguished Speaker and an IEEE Distinguished Lecturer (FNTC and VTS).