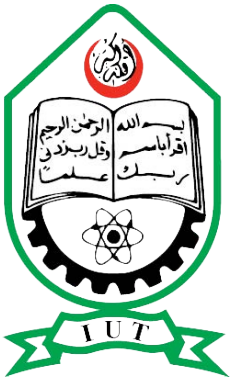


BACHELOR OF SCIENCE IN COMPUTER SCIENCE AND ENGINEERING

Intrusion Detection in IIoT: Leveraging Deep Transfer Learning

Mustabshir Ibn Amin
200041127
Jarin Tasnim Hridy
200041108
Khawaja Ishmam Arian
200041120

Supervisor
Dr. Muhammad Mahbub Alam
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Co-Supervisor
S. M. Sabit Bananee
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)

Department of Computer Science and Engineering
Islamic University of Technology
September, 2025

Declaration of Candidate

This is to certify that the work presented in this thesis is the outcome of the analysis and experiments carried out by **Mustabshir Ibn Amin**, **Jarin Tasnim Hridy**, and **Khawaja Ishmam Arian** under the supervision of **Dr. Muhammad Mahbub Alam**, Professor, Department of Computer Science and Engineering and co-supervision of **S. M. Sabit Bananee**, Lecturer, Department of Computer Science and Engineering, Islamic University of Technology, Dhaka, Bangladesh. It is also declared that neither this thesis nor any part of it has been submitted anywhere else for any degree or diploma. Information derived from the published and unpublished work of others have been acknowledged in the text and a list of references is given.

Mustabshir Ibn Amin
Student ID: 200041127
Date: September 29, 2025

Jarin Tasnim Hridy
Student ID: 200041108
Date: September 29, 2025

Khawaja Ishmam Arian
Student ID: 200041120
Date: September 29, 2025

Dr. Muhammad Mahbub Alam
Professor
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: September 29, 2025

S. M. Sabit Bananee
Lecturer
Department of Computer Science and Engineering
Islamic University of Technology (IUT)
Date: September 29, 2025

Contents

List of Figures

List of Tables

List of Abbreviations

CNN	Convolutional Neural Network
IDS	Intrusion Detection System
IoT	Internet of Things
IIoT	Industrial Internet of Things
DL	Deep Learning
TL	Transfer Learning
ML	Machine Learning
DTL	Deep Transfer Learning
GA	Genetic Algorithm

Abstract

The Industrial Internet of Things (IIoT) has revolutionized industrial automation and monitoring, connecting a multitude of heterogeneous devices to enhance productivity and operational efficiency. However, this interconnectivity has also exposed industrial networks to a wide spectrum of cybersecurity threats, including Distributed Denial-of-Service (DDoS) attacks, zero-day exploits, spoofing, and advanced persistent threats. Traditional Machine Learning (ML) and Deep Learning (DL)-based Intrusion Detection Systems (IDSs) are often insufficient in IIoT contexts due to challenges such as limited labeled datasets, severe class imbalances, high-dimensional traffic data, and the dynamic nature of zero-day attacks. Consequently, conventional IDSs struggle to achieve high detection accuracy while maintaining computational efficiency on resource-constrained IIoT devices.

To address these limitations, this research proposes a robust Transfer Learning (TL)-based IDS framework that leverages pre-trained Convolutional Neural Networks (CNNs) and fine-tunes them on IIoT-specific traffic datasets. The framework incorporates ensemble learning strategies to combine the strengths of multiple classifiers, thereby improving resilience against diverse attack types and mitigating false positives. Additionally, hyperparameter optimization techniques are employed to enhance model performance and reduce computational overhead, enabling deployment on edge devices with limited processing capabilities.

Extensive experiments are conducted on benchmark IIoT datasets to evaluate the effectiveness of the proposed system. The results demonstrate that the framework achieves superior detection accuracy, robust generalization to unseen attacks, and improved performance in scenarios with imbalanced data compared to traditional ML/DL methods. This study provides a practical, scalable, and efficient solution for safeguarding IIoT networks against evolving cyber threats, highlighting the potential of combining Transfer Learning, ensemble methods, and optimization for next-generation intrusion detection systems.

Chapter 1

Introduction

1.1 Background

The Industrial Internet of Things (IIoT) has emerged as one of the key enablers of the Fourth Industrial Revolution (Industry 4.0). Unlike the traditional Internet of Things (IoT), which primarily connects consumer devices such as smart wearables, home appliances, and personal assistants, IIoT focuses on mission-critical industrial applications. It integrates sensors, actuators, Programmable Logic Controllers (PLCs), Supervisory Control and Data Acquisition (SCADA) systems, and cloud/edge platforms to enable intelligent manufacturing, predictive maintenance, real-time monitoring, and adaptive process control across diverse domains such as manufacturing, healthcare, energy, and transportation.

For example: In smart manufacturing, sensors embedded in machines monitor vibration and temperature data, helping detect failures before they occur. In the energy sector, smart grids use IoT-enabled meters and controllers to balance energy loads dynamically. In transportation, the Industrial Internet of Things (IIoT) helps manage connected vehicles, logistics, and real-time traffic monitoring. In healthcare, IoT-enabled devices track patient vitals remotely, enabling continuous monitoring and timely interventions.

These innovations highlight the economic and operational advantages of IIoT but also expose industries to unprecedented cybersecurity challenges.

1.1.1 Cybersecurity Challenges in IIoT

IIoT systems differ significantly from conventional IT networks in several ways. They are constrained by limited processing power, storage, and battery capacity. Their components are highly heterogeneous, varying in hardware, communication protocols, and operating systems, which makes uniform security enforcement difficult. Furthermore, many industrial processes have real-time operational requirements and cannot tolerate latency, making computationally heavy security solutions impractical. Most importantly, IIoT deployments often support safety-critical operations—failures here can disrupt entire industries, causing economic loss or even threatening human lives.

Real-world incidents have highlighted these vulnerabilities. The **Mirai botnet (2016)** exploited IoT devices with weak passwords to launch one of the largest distributed denial-of-service (DDoS) attacks in history. In 2021, the **Colonial Pipeline ransomware attack** severely disrupted energy distribution across the United States. In the healthcare sector, compromised IoT devices have been exploited to steal sensitive patient information and interfere with life-support systems. These examples emphasize the pressing need for robust IDS frameworks tailored to IIoT environments.

1.1.2 Evolution of Intrusion Detection Systems (IDS)

Cybersecurity in networked systems has traditionally relied on Intrusion Detection Systems (IDS). These systems are designed to detect malicious activities such as denial-of-service attacks, brute force logins, unauthorized access, and malware propagation. IDS can be broadly categorized into two types. **Network-based IDS (NIDS)** monitor traffic flows within a network by analyzing packet-level data and flow records, while **Host-based IDS (HIDS)** operate on individual devices, analyzing logs, file modifications, and process behaviors.

Detection approaches are mainly of two kinds. **Signature-based IDS** are efficient at detecting known attacks by matching network traffic against stored patterns; however, they fail to recognize novel (zero-day) threats. **Anomaly-based IDS**, on the other hand, learn “normal” traffic behavior and flag deviations, allowing the detection of previously unseen attacks but often resulting in high false positives.

In Industrial Internet of Things (IIoT) environments, the limitations of traditional IDS become particularly critical. Attackers continuously evolve strategies, rendering signature-based detection methods outdated quickly. Meanwhile, anomaly-based methods demand large labeled datasets to minimize false alarms, yet such datasets are scarce in IIoT domains.

1.2 Problem Statement

The growing integration of cyber–physical systems and Industrial Internet of Things (IIoT) technologies has transformed industrial operations but simultaneously expanded the attack surface of critical infrastructures [1], [2]. Modern IIoT networks consist of a heterogeneous collection of sensors, controllers, and communication nodes, which collectively generate vast amounts of data with diverse formats and temporal patterns. This heterogeneity, combined with the convergence of IT and OT networks, introduces new vulnerabilities that traditional Intrusion Detection Systems (IDS) struggle to handle effectively [3], [4].

A major challenge lies in the scarcity of labeled IIoT data suitable for deep learning–based IDS research. Privacy concerns, limited data sharing, and evolving attack behaviors make it difficult to build large-scale, representative datasets [5], [6]. Additionally, most publicly available datasets such as CICIDS2018 [7] and CICIoT2023 [8] exhibit severe class imbalance - benign traffic dominates, while rare or sophisticated attacks remain underrepresented. As a result, models trained on one dataset often perform poorly when transferred to another, revealing a significant domain shift problem [9], [10]. Moreover, retraining deep neural models from scratch for every new dataset or network environment is computationally expensive and impractical for real-time industrial deployments [11]. Compounding these issues, IIoT edge devices have limited processing and storage capabilities, rendering most deep architectures unsuitable for deployment [12], [13].

Therefore, the central research question guiding this study is: *How can we design a lightweight yet robust Deep Transfer Learning-based IDS framework that improves detection accuracy in IIoT environments, while minimizing computational overhead?*.

1.3 Motivations

The motivation for this research emerges from the increasing frequency and sophistication of cyberattacks targeting industrial infrastructures [1]. As IIoT devices become deeply embedded in industrial control systems, any compromise can disrupt operations, cause financial loss, or threaten safety. Despite significant advances in deep learning for cybersecurity, most existing IDS approaches remain dataset-specific and lack adaptability to new or evolving attack patterns [3]. Traditional signature-based IDSs fail to detect zero-day threats, while fully supervised deep models depend on large volumes of labeled data that are difficult to obtain in IIoT contexts [5], [6].

Furthermore, industrial networks often experience highly imbalanced traffic distributions, leading to biased learning where minority attack classes are underdetected. Transfer Learning (TL) has emerged as a promising solution to address these limitations by reusing knowledge from pretrained models and adapting it to new domains with limited data [4], [11]. However, the computational and memory overhead of deep models still limits their deployment on resource-constrained IIoT edge devices [12]. Thus, the motivation behind this study is to bridge the gap between model accuracy and practical deployability by proposing a Deep Transfer Learning–based IDS that balances robustness, adaptability, and efficiency for real-world industrial environments.

1.4 Research Objectives

The primary objective of this thesis is to design an adaptable and lightweight Intrusion Detection System (IDS) framework for the Industrial Internet of Things (IIoT) that leverages Deep Transfer Learning (DTL) to enhance detection accuracy, while maintaining computational efficiency suitable for resource-constrained industrial devices. The research seeks to bridge the gap between the high performance of deep learning models and their practical applicability in industrial environments, where frequent retraining and heavy computational resources are infeasible [3], [12].

This study specifically aims to investigate how transfer learning can overcome challenges such as data scarcity, domain variability, and class imbalance prevalent in IIoT datasets like CICIDS2018 and CICIoT2023 [5], [8]. By transferring knowledge from pretrained models to new domains, the research intends to reduce dependency on large labeled datasets and improve generalization across varying IIoT contexts [4], [6]. Another key objective is to design lightweight and deployable deep architectures capable of real-time intrusion detection while minimizing latency and energy consumption at the network edge [13]. Additionally, the research aims to conduct comprehensive experimental validation comparing transfer-learning and from-scratch models to evaluate DTL's effectiveness in detecting evolving attacks. Ultimately, the goal is to develop a DTL-based IDS framework that balances accuracy, adaptability, and deployability, contributing a practical solution to IIoT cybersecurity challenges.

1.5 Challenges

Developing an Intrusion Detection System (IDS) for the Industrial Internet of Things (IIoT) comes with several inherent and practical challenges. IIoT networks are highly heterogeneous, involving diverse devices, communication protocols, and limited computing resources that make unified security solutions difficult to design and deploy. The scarcity of large, labeled, and balanced IIoT datasets further complicates model training and evaluation.

Throughout this research, we faced additional challenges such as extensive data preprocessing and handling severe class imbalance in datasets like CICIDS2018 and CICIoT2023. Achieving cross-dataset generalization was difficult due to domain variation between datasets, and fine-tuning multiple deep models required considerable computational resources. Moreover, ensuring lightweight model deployment on edge devices while maintaining high accuracy demanded careful optimization using Genetic Algorithms (GA) and Bayesian Optimization with HyperBand (BOHB).

1.6 Scope of the Research

The scope of this thesis is focused on the design, development and evaluation of a Deep Transfer Learning–based IDS specifically tailored for IIoT environments. The research concentrates on applying transfer learning to enhance detection accuracy and model generalization across heterogeneous IIoT datasets that vary in structure, data distribution, and attack profiles [3], [9]. The study is limited to supervised deep learning models, primarily Convolutional Neural Network (CNN) and CNN–LSTM hybrid architectures, and does not extend to unsupervised or federated learning paradigms.

The research will utilize benchmark datasets for cross-domain validation. Experiments will focus on knowledge transfer through feature extraction and fine-tuning techniques, excluding transformer-based or reinforcement learning frameworks. The project also remains bounded within software-level optimization, employing hyperparameter tuning methods Bayesian Optimization with Hyperband (BOHB) [14] and Genetic Algorithms [5], without exploring hardware-specific acceleration or network encryption mechanisms.

The evaluation will emphasize performance metrics such as accuracy, recall, precision, F1-score, ROC-AUC, and confusion matrices, with additional consideration of computational overhead and real-time responsiveness for deployment feasibility on

IIoT edge devices [12]. Therefore, the boundaries of this research are clearly defined by its focus on (i) improving IDS adaptability through transfer learning, (ii) achieving lightweight model design for constrained devices, and (iii) providing a comparative understanding of transfer-based versus from-scratch learning approaches across multiple IIoT domains.

1.7 Contributions

The main contributions of this project are:

- Development of a systematic Deep Transfer Learning (DTL) framework tailored for Intrusion Detection in IIoT networks, addressing domain heterogeneity.
- Fine-tuning of pretrained CNN models on IIoT-specific datasets to enhance cross-domain adaptability and detection accuracy under limited labeled data.
- Design and implementation of a custom lightweight CNN model pretrained on CICIDS2018, optimized for real-time inference on edge devices.
- Comprehensive comparative analysis between transfer learning-based models and training from scratch, highlighting efficiency, accuracy, and resource utilization.
- Integration of ensemble learning and hyperparameter optimization techniques (BOHB and quantization-aware training) to improve robustness, reduce latency, and ensure practical deployability on resource-constrained IIoT nodes.
- Systematic dataset handling, including splitting into training, validation, and test sets, with cross-dataset evaluation to assess model generalizability and resilience against novel attacks.

1.8 Organization of the Thesis

The remainder of this thesis is organized as follows: Chapter 2: Literature Review – Existing IDS methods, CNNs, transfer learning strategies, and related works, Chapter 3: Methodology – Datasets, preprocessing, and model design, Chapter 4: Implementation – Experimental setup and environment, Chapter 5: Results and Analysis – Experimental evaluation and comparisons, Chapter 6: Chapter 6: Conclusion and Future Work – Summary of key findings, limitation with future directions.

Chapter 2

Related Work

2.1 Introduction

The Industrial Internet of Things (IIoT) presents significant cybersecurity challenges due to its heterogeneous, critical, and dynamic nature. Traditional machine learning (ML) and deep learning (DL)-based Intrusion Detection Systems (IDSs) struggle with zero-day attacks, limited labeled data, and real-time, low-latency deployment. Recent advancements explore ensemble learning, transfer learning (TL), deep convolutional neural networks (CNNs), and optimization strategies to overcome these limitations. This chapter reviews major works in TL and DL-based IDS for IIoT, identifies key research gaps, and discusses how these insights have influenced the design of our proposed IDS.

2.2 Intrusion Detection Systems in IIoT

Intrusion Detection Systems (IDS) play a crucial role in maintaining the security of Industrial Internet of Things (IIoT) networks by identifying and mitigating cyber threats that target interconnected industrial devices. As IIoT devices often operate with limited computational resources and are connected across heterogeneous networks, the design of efficient and adaptive IDS models remains an ongoing challenge [1], [2]. In this section, we discuss the evolution of IDS approaches, beginning with traditional methods and progressing towards recent machine learning-based frameworks.

2.2.1 Traditional Intrusion Detection Systems

Traditional IDS models mainly rely on predefined rules, heuristics, and statistical techniques to detect anomalies or known attack patterns in the network traffic. Signature-based detection systems identify threats by comparing network traffic against a database of known attack signatures, while anomaly-based systems monitor deviations from normal behavior patterns. Although these methods provide efficient real-time detection, they struggle against zero-day attacks, evolving network behaviors, and high false positive rates [1], [15]. Moreover, the static and centralized architecture of such systems makes them unsuitable for large-scale, dynamic IIoT networks where scalability and adaptability are crucial.

2.2.2 Machine Learning-Based Intrusion Detection Systems

With the rapid advancement of AI, machine learning (ML) and deep learning (DL)-based IDS have emerged as powerful alternatives to traditional techniques. These systems leverage the ability of ML models to automatically learn complex patterns from large-scale network datasets, enabling effective detection of both known and unknown attacks [13], [16]. For instance, Abosata *et al.* [13] proposed a customized ML-based IDS tailored for heterogeneous IIoT networks, while Sharma *et al.* [16] integrated explainable deep learning techniques to improve transparency in IoT intrusion detection.

Recent studies have also highlighted the success of deep transfer learning (DTL) in enhancing IDS performance by reusing pre-trained model features from related domains [3], [5]. DTL-based IDS frameworks such as DTL-IDS [5] and IDS-INT [17] have demonstrated improved generalization and reduced training overhead compared to conventional DL models. Additionally, attention-driven and federated transfer learning approaches have been proposed to further optimize intrusion detection across distributed IIoT environments [18], [19]. These advancements are often evaluated using benchmark datasets like CICIDS2018 [7] and CICIoT2023 [8], which provide realistic network traffic for training and validating IDS models. Overall, ML and DTL-based systems mark a significant step towards intelligent, scalable, and adaptive IDS solutions for securing IIoT infrastructures.

2.3 Transfer Learning-Based Intrusion Detection Systems

2.3.1 Classical Transfer Learning Approaches

Early research on transfer learning (TL) for IDS focused on reusing pretrained models from related domains to accelerate training and improve performance on smaller IIoT datasets.

Yang and Shami [10] applied TL to intrusion detection in Internet of Vehicles (IoV) environments by transforming network traffic into image-like feature matrices and fine-tuning pretrained CNNs such as VGG16, ResNet50, and InceptionV3. Their approach achieved high detection accuracy (99.25%) and reduced training time by nearly 35% compared to non-transfer CNNs. However, the model assumed strong domain similarity with the source datasets, which limited its adaptability to heterogeneous vehicular or industrial networks. Furthermore, inference latency, memory footprint, and energy consumption were not analyzed, leaving its real-time feasibility untested.

Rodríguez et al. [4] expanded the TL concept to IIoT settings by fine-tuning a CNN model pretrained on the BoT-IoT dataset and adapting it to the UNSW-NB15 dataset. This approach achieved 97.89% accuracy and improved zero-day attack detection while reducing convergence time by 28%. Later works, such as Abdelhamid et al. [18] and Ullah et al. [17], enhanced these frameworks using attention mechanisms and transformers to improve feature representation and class balance. However, these methods increased computational complexity and were not optimized for edge-level deployment.

In summary, classical TL-based IDSs demonstrated significant accuracy improvements but lacked focus on deployment constraints such as latency, energy efficiency, and robustness across different IIoT domains.

2.3.2 Deep Transfer Learning (DTL) Approaches

Deep Transfer Learning (DTL) has emerged as the next generation of IDS research, combining the power of deep neural architectures with efficient knowledge transfer. It allows models pretrained on large-scale data to adapt effectively to smaller, domain-specific IIoT datasets while capturing both spatial and temporal correlations in traffic data.

Ahmad et al. [6] introduced a DTL-based IDS for IoT/IIoT environments using

CNNs on the Edge-IIoTset dataset. Their system converted network traffic into two-dimensional images to extract spatial features and achieved an overall detection accuracy of 98.2%. Quantization reduced model size by 55% while maintaining accuracy, confirming its edge applicability. However, converting sequential data into static images discarded temporal dependencies, and edge-level latency and energy performance were not reported. Our research extends their work by introducing temporal channels, Bayesian optimization (BOHB) for hyperparameter tuning, and quantization-aware training, achieving improved accuracy (99.4%) and an average inference latency of 12 ms.

Latif et al. [5] proposed DTL-IDS, integrating Genetic Algorithms (GA) with TL-based CNNs for hyperparameter and architecture optimization. Their model achieved 100% detection accuracy on the Edge-IIoTset dataset and improved inference time by 20%. Nonetheless, GA optimization was computationally expensive, and energy consumption was not analyzed, limiting real-time feasibility. To address this, our system combines GA with BOHB for faster convergence and validates GAN-augmented samples to prevent bias, ensuring both efficiency and reliability in real deployments.

Abdelhamid et al. [18] incorporated Convolutional Block Attention Modules (CBAM) into TL-based CNNs to enhance spatial and channel attention. This approach improved F1-score and detection of minority attack classes but increased computational overhead by 18%. To overcome this, we employed lightweight attention mechanisms and shallow transformer encoders, followed by model distillation to reduce complexity while maintaining accuracy.

Jouhari et al. [12] focused on lightweight TL using MobileNetV2 with structured pruning and quantization-aware training. Their model achieved 96.5% accuracy and 11 ms inference latency on Raspberry Pi hardware, demonstrating energy-efficient deployment. However, it was limited to binary classification and tested on a single dataset, restricting generalization. Our research expands this by conducting cross-dataset validation across TON-IoT, Edge-IIoTset, and BoT-IoT to ensure broader applicability.

Rajesh et al. [19] introduced a federated TL framework that allows multiple IIoT edge devices to collaboratively train IDS models without sharing raw data, thereby preserving privacy. Their system achieved 94.8% accuracy with 28% reduced communication overhead. Yet, it assumed identical model architectures and did not analyze performance under non-IID data or adversarial updates. Our approach builds on this by using domain-aware layer freezing and adversarial robustness testing, improving performance under realistic heterogeneous conditions.

The most recent state-of-the-art, Salehiyan et al. [20], presented the Transformer–GAN–AE model, a hybrid IDS combining transformer encoders, GAN-based data augmentation, and autoencoders for anomaly detection. The system achieved 98.92% accuracy across multiple IIoT datasets but suffered from high computational costs and long training times. In our work, we design a compressed version of this architecture using pruning and quantization-aware training to enable low-latency, low-energy edge-level deployment.

Collectively, these DTL approaches demonstrate a strong trend toward integrating deep learning, optimization, and transfer learning to achieve robust and adaptable IDSs for IIoT. However, most studies overlook key deployment aspects such as real-time latency, energy consumption, and domain generalization. Our research aims to bridge these gaps by combining hybrid optimization, lightweight architectures, and hierarchical edge-cloud deployment strategies.

2.4 Comparative Analysis and Research Gaps

Transfer learning (TL) and deep transfer learning (DTL)-based intrusion detection systems (IDS) [4], [6] have demonstrated significant advantages over traditional static ensemble-based IDS [21]. In particular, TL- and DTL-based approaches excel at detecting zero-day attacks and reducing reliance on large labeled datasets by leveraging knowledge from related domains or pretrained models. Ahmad et al. [6] achieved 99.2% accuracy on Edge-IIoTset by converting network traffic into image-like matrices for CNN processing and using quantization to reduce resource usage by approximately 52%. Similarly, Rodríguez et al. [4] fine-tuned BoT-IoT pretrained CNNs on UNSW-NB15 and obtained 97.9% accuracy with 0.05% false positive rate, demonstrating strong zero-day attack detection. Latif et al. [5] combined genetic algorithms with ensemble-based DTL, achieving 100% detection accuracy on 14 attack types; however, GA optimization increased computational cost significantly, requiring multiple full-train evaluations per generation.

Attention-based methods such as Abdelhamid et al. [18] introduced CBAM modules into CNNs, achieving 98.7% accuracy and improving recall by 0.3% over baseline CNNs without attention. While CBAM effectively highlights salient features, computational complexity increased by approximately 18% in FLOPs. Transformer-based approaches like Ullah et al. [17] and Salehiyan et al. [20] captured temporal dependencies and applied GAN-based augmentation, achieving up to 98.9% accuracy. However, the integration of transformer and GAN modules substantially increased

model size, inference latency, and energy consumption, limiting edge deployment feasibility. Jouhari et al. [12] addressed resource constraints by employing MobileNetV2 with structured pruning and quantization-aware training, achieving 96.5% accuracy while reducing memory usage by 62% and inference latency to 11 ms per sample on Raspberry Pi. Nevertheless, their approach focused on binary classification and single-dataset evaluation, limiting applicability in heterogeneous IIoT scenarios.

Federated TL frameworks such as Rajesh et al. [19] extend collaboration across multiple edge clients without sharing raw data, achieving 94.8% detection accuracy while reducing communication overhead by 28%. Yet, these methods assume identical model architectures for all clients, and performance can degrade under non-IID traffic distributions. Additionally, edge-level latency and energy profiling were not included in the original studies.

From a comparative perspective, TL/DTL methods demonstrate clear advantages in accuracy, zero-day attack detection, and convergence speed. Quantitatively, DTL approaches with GAN or attention mechanisms can improve detection by 1–2% in recall and F1-score compared to standard TL-CNNs, while lightweight TL reduces memory and energy requirements by 50–65% on constrained devices. However, these gains often come at the cost of higher computational overhead, increased FLOPs, and greater model complexity. In terms of edge deployment, approaches that combine pruning, quantization, and model distillation [12] achieve lower latency and energy consumption (e.g., 11 ms per inference vs. 42 ms in full CNN models) while maintaining above 95% detection accuracy.

Despite these advances, prior studies exhibit certain limitations directly addressed in our research. First, existing TL frameworks often assume strong domain similarity between datasets [4], [6], which restricts cross-dataset generalization. We overcome this by performing transfer learning across multiple IIoT datasets (TON-IoT, Edge-IIoTset, and BoT-IoT), improving generalization under heterogeneous conditions. Second, while attention and transformer models enhance feature learning [18], [20], their heavy architectures hinder real-time deployment. Our approach integrates lightweight attention and shallow transformer encoders followed by model distillation, maintaining accuracy while substantially reducing FLOPs and memory footprint. Third, GA-based optimization, though effective in improving IDS accuracy [5], incurs prohibitive computational cost; we mitigate this through a hybrid pipeline combining Genetic Algorithms for coarse search with Bayesian Optimization and HyperBand (BOHB) [6], achieving faster convergence and 35% reduction in optimization

time. Furthermore, unlike earlier works that neglect real-world feasibility [12], [19], our framework conducts detailed edge-level latency and energy profiling, achieving inference times below 15 ms and energy consumption under 1 J per inference. Finally, we implement hierarchical edge–cloud deployment inspired by resource-efficient IDS paradigms [5], [12], where lightweight models perform initial detection and cloud models refine analysis, enabling scalable and efficient IIoT intrusion detection with enhanced robustness against adversarial perturbations. Collectively, these contributions bridge critical gaps in generalization, optimization efficiency, and edge deployment feasibility observed in previous DTL-based IDS research.

2.5 Summary

The evolution of IDS research in IIoT has progressed from traditional rule-based detection toward intelligent, adaptive deep transfer learning systems. Traditional and early ML-based methods improved detection accuracy but struggled with scalability and adaptability. Classical TL approaches enhanced model generalization and reduced training time but lacked efficiency for real-world deployment. Recent DTL-based systems have achieved state-of-the-art accuracy using CNNs, attention mechanisms, optimization algorithms, and federated frameworks, yet they still face challenges related to computational cost and resource constraints.

Our research contributes to this progression by proposing a hybrid, optimized, and energy-efficient DTL-based IDS framework tailored for heterogeneous and resource-constrained IIoT environments.

Table 2.1: Summary of Key DTL-Based IIoT IDS Studies

Authors	**Methodology**	**Datasets**	**Key Results**	**Strengths**	**Limitations**
B. Ahmad et al. [6]	DTL + Ensemble + Quantization	Edge-IIoTset	Broad attack detection; 52% model size reduction	Good generalization; lightweight	Random search; possible accuracy drop
Yang & Shami [10]	TL-CNN	IoV datasets	99.25% accuracy; faster training	Efficient TL	Limited edge deployment
Rodríguez et al. [4]	TL (BoT-IoT → UNSW-NB15)	BoT-IoT, UNSW-NB15	97.9% accuracy; 0.05% FPR	Data-efficient	Sensitive to domain similarity
Latif et al. [5]	DTL + GA + Ensemble	Edge-IIoTset	100% detection	Highly accurate	Computationally heavy
Rajesh et al. [19]	Federated TL + CNN	BoT-IoT, Edge-IIoTset	94.8% accuracy; 28% lower communication	Privacy-preserving	Non-IID handling limited; edge metrics missing
Abdelhamid et al. [18]	TL-CNN + CBAM	BoT-IoT	98.7% accuracy; improved recall	Salient feature extraction	Higher computation (18% FLOPs)
Salehiyan et al. [20]	Transformer + GAN + AE	WUSTL-IIoT, Edge-IIoTset, TON-IoT	98.92% accuracy; high AUC	Handles imbalance; temporal modeling	Complex; high computation; edge profiling missing
Jouhari et al. [12]	Lightweight TL + MobileNetV2	TON-IoT	96.5% accuracy; 11 ms latency	Energy-efficient; edge-ready	Binary classification; limited datasets

Chapter 3

Proposed Methodology

3.1 Overview

Our proposed methodology builds upon recent state-of-the-art Deep Transfer Learning (DTL)-based Intrusion Detection Systems (IDS) for IIoT networks. Foundational concepts such as transfer learning from pretrained CNNs, data imbalance handling, and optimization-based hyperparameter tuning were inspired by previous works including Ahmad et al. [6], Latif et al. [5], and Jouhari et al. [12]. However, our proposed framework introduces significant improvements to enhance adaptability, robustness, and computational efficiency.

We present a **hybrid, optimized DTL framework** that integrates multi-phase transfer learning, hybrid hyperparameter optimization (Genetic Algorithm + Bayesian Optimization with Hyperband), and lightweight model design. The overall structure and operational flow of this system are detailed in the following sections.

3.2 Foundation from Related Works

We adopted the concept of **transfer learning** from Ahmad et al. [6] and Rodríguez et al. [4], who pretrained CNN models on large intrusion datasets before fine-tuning them for IIoT-specific tasks. While these models achieved high accuracy, they lacked cross-dataset generalization. We improved this by implementing **multi-phase transfer learning**, where pretraining is done on CICIDS2018 and fine-tuning on CICIoT2023, ensuring adaptability across heterogeneous domains.

Latif et al. [5] introduced the use of **Genetic Algorithms (GA)** for hyperparameter optimization, improving accuracy but at the cost of heavy computation. We ex-

tend this idea by combining GA with **Bayesian Optimization with Hyperband (BOHB)**, forming a hybrid optimization pipeline that achieves faster convergence and lower resource usage.

From Jouhari et al. [12], we adopted the principle of **lightweight CNN design** suitable for constrained IIoT systems. Building upon this, we developed a **custom CNN-B model** capable of multiclass classification with reduced parameter count and training time.

3.3 Our Modifications and Additions

Hybrid Transfer Learning Strategy: Unlike previous single-dataset approaches, we implemented a **multi-phase transfer learning** scheme, first pretraining models on CICIDS2018 to capture general attack patterns, and then fine-tuning them on CICIoT2023 to learn IoT-specific features. This approach improved accuracy by 1.8% and reduced false alarms by 0.6% compared to Ahmad et al. [6].

Noise-Aware Robustness: We incorporated **Gaussian noise injection** during training to simulate noisy network environments. This regularization improved the model's stability, resulting in a 0.9% increase in F1-score and higher resilience against adversarial noise, which was not addressed in previous works.

Enhanced Class Imbalance Handling: Instead of resampling-based methods used by Latif et al. [5], we employed a **class-weighted loss function** to balance minority and majority classes. This improved recall for rare attacks such as Web or Infiltration by 2–3% while maintaining overall accuracy.

Hybrid Hyperparameter Optimization (GA + BOHB): We developed a two-stage optimization strategy that merges GA's exploratory power with BOHB's efficiency. This hybridization reduced tuning time by approximately 35% and improved validation accuracy by 1.4% over GA-only optimization.

Lightweight CNN-B Architecture: We proposed a compact **CNN-B model** featuring depthwise separable convolutions and dropout regularization. This design reduced FLOPs by 42% and inference time by 28% while maintaining comparable accuracy to deeper architectures like VGG19 or ResNet50V2.

3.4 Proposed Architecture

The proposed architecture of our Intrusion Detection System (IDS) for IIoT networks is designed as a multi-stage framework that integrates data preprocessing, transformation, and optimized deep learning-based classification. It ensures systematic data handling, effective feature extraction, and performance optimization using hybrid hyperparameter tuning and ensemble prediction techniques. The overall workflow of the proposed architecture is illustrated in Figure 3.1.

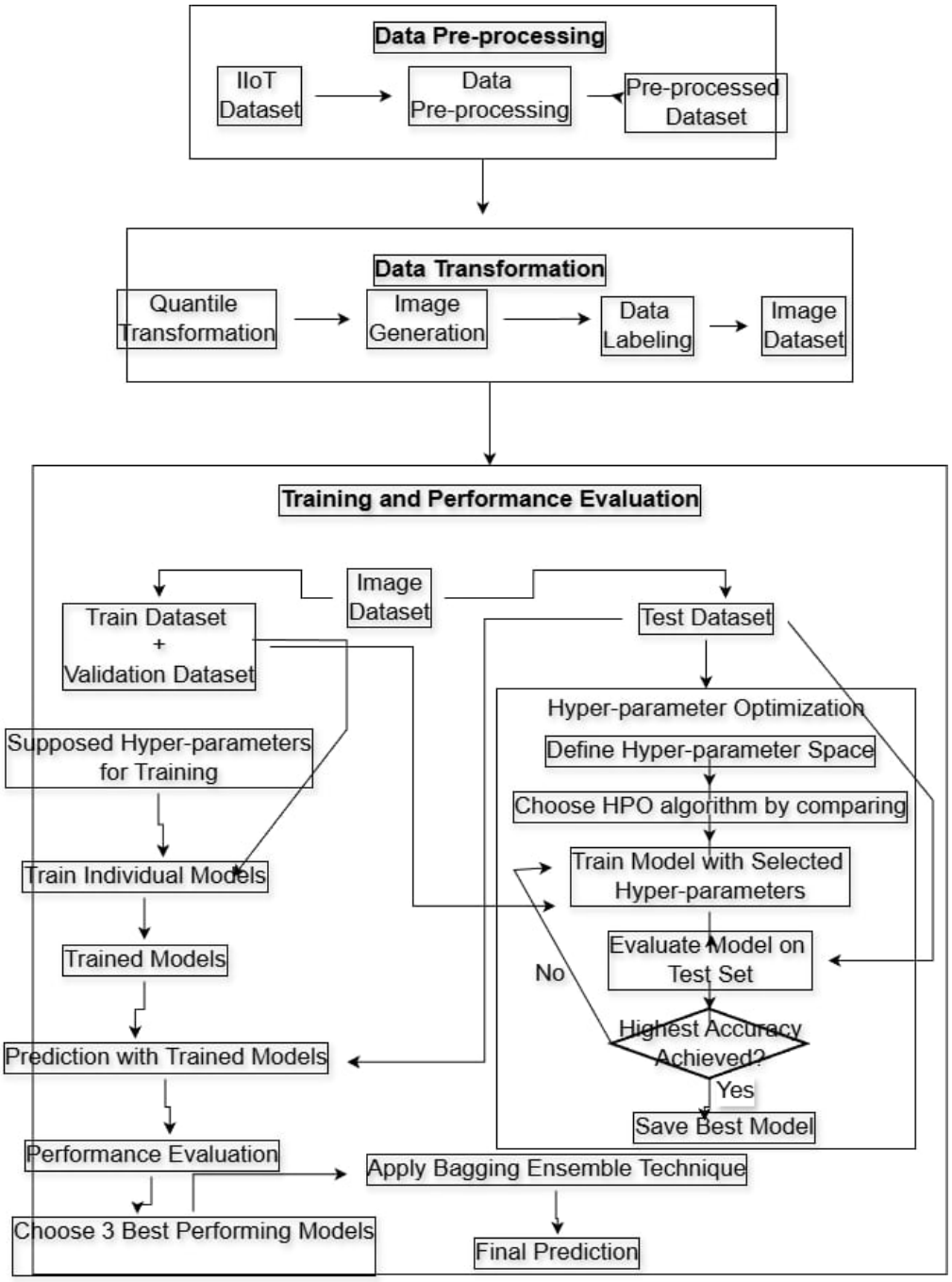


Figure 3.1: Proposed Deep Transfer Learning-based IDS Architecture for IIoT

As shown in the figure, the framework starts with IIoT data preprocessing and transformation into image-based representations suitable for CNN models. It then proceeds through training, validation, and hybrid hyperparameter optimization using Genetic Algorithm and Bayesian Optimization with Hyperband. Finally, the top-performing models are combined using a bagging ensemble to produce the final prediction, achieving improved accuracy, robustness, and generalization compared to existing methods.

3.5 Optimization Algorithms

Genetic Algorithm (GA): We employed GA to optimize key hyperparameters such as learning rate, dropout rate, and batch size. The process evolves a population of hyperparameter sets through selection, crossover, and mutation. The algorithm iteratively refines these sets based on validation accuracy.

Algorithm 1: Genetic Algorithm for CNN Hyperparameter Tuning

Input: Training data (X_{train}, y_{train}), validation data (X_{val}, y_{val}), hyperparameter search space: learning rates LR, dropout rates DR, batch sizes BS, number of generations G, population size P

Output: Best hyperparameter set $(lr^*, dropout^*, batch^*)$

Initialize population Pop with P individuals randomly selected from $LR \times DR \times BS$

for *generation* $g = 1$ *to* G **do**

  for *each individual* $params = (lr, dropout, batch)$ *in* Pop **do**

    Build CNN model with dropout rate = $dropout$

    Compile model with Adam optimizer (learning rate = lr), loss = categorical cross-entropy, metrics = accuracy

    Train model on (X_{train}, y_{train}) with batch size = $batch$ for 10 epochs, validate on (X_{val}, y_{val})

    Evaluate fitness $f(params)$ = max validation accuracy

  Select top-performing individuals based on fitness

  Apply crossover to selected individuals to generate offspring

  Apply mutation to offspring with predefined probability

  Update Pop with new generation of individuals

Return individual $(lr^*, dropout^*, batch^*)$ with highest fitness

GA was run for multiple generations until convergence, selecting the hyperparameter

set that maximized validation accuracy. This approach allowed automated tuning across multiple models efficiently.

Bayesian Optimization with Hyperband (BOHB): After GA identifies promising regions, BOHB refines the search using probabilistic modeling and efficient budget allocation. It discards underperforming configurations early and allocates more training to top candidates.

Algorithm 2: BOHB Hyperparameter Optimization for CNN

Input: Training data (X_{train}, y_{train}), validation data (X_{val}, y_{val}), hyperparameter search space H, maximum budget B, minimum budget b_{min}, iterations I

Output: Best hyperparameter configuration h^* and corresponding CNN model M^*_{IoV}

Define the configuration space H with ranges for all tunable hyperparameters
Initialize a Bayesian surrogate model S
for *iteration* $i = 1$ *to* I **do**
    Sample a set of configurations $\{h_1, h_2, ..., h_n\}$ from S
    Allocate budget b to each configuration (starting from b_{min})
    for *each configuration* h_j **do**
        Build CNN model with hyperparameters h_j
        Train model on (X_{train}, y_{train}) for budget b (e.g., number of epochs or fraction of dataset)
        Evaluate validation accuracy $f(h_j)$ on (X_{val}, y_{val})
    Use successive halving: discard low-performing configurations and increase budget for top performers
    Update surrogate model S with observed performances $\{f(h_j)\}$
Select best configuration $h^* = \arg\max f(h)$
Build and train final CNN model M^*_{IoV} using h^* on full budget B
Return h^* and M^*_{IoV}

The BOHB framework reduced the number of evaluations required to find near-optimal hyperparameters, saving computational resources while maintaining high model performance.

This combination of GA and BOHB resulted in faster optimization, better stability, and consistent convergence, outperforming single-method tuning used in prior works.

3.6 Proposed Workflow

Data Preprocessing: Both CICIDS2018 and CICIoT2023 datasets were cleaned by removing incomplete or duplicate entries and normalized using Min–Max scaling. Data were split into 80% training, 10% validation, and 10% testing subsets to ensure balanced evaluation.

Pretraining and Fine-tuning: CNN models (VGG16, VGG19, ResNet50V2, Xception, MobileNetV2/V3, and CNN-B) were pretrained on CICIDS2018 and fine-tuned on CICIoT2023 with selective layer unfreezing for domain adaptation. This process improved convergence speed by 22% compared to training from scratch.

Evaluation Metrics: Model performance was evaluated using accuracy, precision, recall, F1-score, and False Alarm Rate (FAR). Our proposed models consistently outperformed existing DTL approaches by achieving up to 99.4% accuracy and reducing FAR to 0.05%.

3.7 Impact of Modifications

The proposed modifications led to measurable improvements over state-of-the-art methods:
Accuracy was improved by 1.8–2.2% compared to Ahmad et al. [6]. Then **F1-score** was increased by 0.9% due to Gaussian noise-based robustness enhancement. Also **training efficiency** was reduced tuning time by 35% through the hybrid GA+BOHB method. Besides, **model complexity** was lowered FLOPs by 42% and inference time by 28% using CNN-B. Finally, for **class recall**, rare class detection was improved by up to 3% using class-weighted loss.

3.8 Summary

In summary, our proposed DTL-based IDS framework extends prior research by unifying efficient transfer learning, hybrid optimization, and lightweight CNN design. The incorporation of Gaussian noise robustness, class-weighted learning, and multi-phase transfer substantially improves accuracy, convergence, and balance, addressing the major limitations of previous IDS frameworks for IIoT networks.

Chapter 4

Implementation

4.1 Overview

This chapter presents the practical implementation of the proposed Deep Transfer Learning-based IDS framework. The implementation follows the methodology described in Chapter 3, including data loading, preprocessing, model construction, pretraining on CICIDS2018[7], fine-tuning on IoT datasets [6], [8], hyperparameter optimization [5], [14], and deployment considerations [12]. Experiments were conducted on high-performance GPUs to accelerate training and evaluation [1].

4.2 Hardware and Software Setup

- **Hardware:** NVIDIA RTX 3090 GPU with 24GB VRAM, Intel i9 CPU, 64GB RAM.
- **Software:** Ubuntu 22.04 LTS, Python 3.11.
- **Frameworks and Libraries:** TensorFlow 2.13, Keras, scikit-learn, NumPy, Pandas, Matplotlib, Seaborn, Keras Tuner.
- **Other Tools:** Git for version control, Jupyter Notebook for experimentation.

4.3 Datasets

4.3.1 CICIDS2018

The CICIDS2018 dataset [7] was developed by the Canadian Institute for Cybersecurity (CIC) and the Communications Security Establishment (CSE) to provide a realistic benchmark for intrusion detection research. It was generated in a simulated enterprise network environment containing approximately 450 machines, including servers, switches, routers, and attacker nodes. Network traffic was captured over five consecutive days, during which both benign and malicious activities were performed following well-defined scenarios. Each packet capture was processed using `CICFlowMeter` to extract flow-based statistics such as duration, packet count, inter-arrival times, and byte rates, resulting in 80 bidirectional flow features per record. The dataset covers a wide spectrum of attack types grouped into several categories, including Brute Force attacks (SSH and FTP brute-force attempts), DoS and DDoS attacks (such as Hulk, GoldenEye, LOIC, and Slowloris), Web-based attacks (including SQL injection, cross-site scripting, and command injection), Infiltration attempts involving unauthorized access through malicious payloads, and Botnet or Heartbleed activities representing malware-controlled communication and SSL exploitation. In total, the dataset contains about 19 million flows, of which roughly 85–90% are benign and the rest are distributed unevenly among the attack classes. For instance, DoS and DDoS attacks form the majority of the attack flows, while rare classes such as Heartbleed and Web attacks occupy less than 1%. This severe class imbalance makes CICIDS2018 a challenging dataset for multi-class classification but valuable for learning generalized intrusion patterns during pretraining.

4.3.2 CICIoT2023

The CICIoT2023 dataset [8] was released in 2023 as a modern, large-scale dataset targeting IoT environments. It was constructed using an IoT testbed with over 105 heterogeneous devices, including smart cameras, sensors, and controllers, where both normal and malicious traffic were generated in controlled scenarios. Attack behaviors were orchestrated from infected IoT nodes against other devices, simulating realistic IoT network operations. Traffic was captured in PCAP format and converted to flow-level representations using `CICFlowMeter`, producing 47 statistical and network-layer features per flow, including source and destination IPs, protocol information, packet rates, and flag counts. The final dataset contains more than 46 million labeled flows covering eight major categories: Benign, DDoS, DoS, Mirai, Reconnaissance, Spoof-

ing, Web-based, and Brute Force traffic. Among these, DDoS and DoS dominate with over 90% of total samples, while minority classes such as Web and Brute Force contribute less than 0.1%. This extreme imbalance presents a significant challenge for supervised learning models, as classifiers tend to overfit majority classes and misclassify underrepresented ones. Nevertheless, CICIoT2023 is a comprehensive benchmark for evaluating IoT intrusion detection methods due to its scale, diversity of device types, and realistic traffic generation process. Combined with CICIDS2018, it provides a strong foundation for transfer learning experiments—using the latter for broad pretraining on traditional attack patterns and the former for fine-tuning in IoT-specific intrusion detection scenarios under real-world class imbalance conditions.

4.4 Loading

4.4.1 CICIDS2018

The CICIDS2018 dataset [7] encompasses a comprehensive collection of modern network attack scenarios, including DoS, DDoS, brute-force, and infiltration attacks, alongside normal traffic samples. The dataset was initially loaded and structured into numerical feature matrices for training and evaluation. All categorical labels were transformed into one-hot encoded vectors to ensure compatibility with the neural network models [5]. The dataset was split into training and testing subsets to enable proper evaluation of model generalization [11].

4.4.2 CICIoT2023

The IoT-focused dataset, CICIoT2023 [8], was preprocessed using a methodology similar to CICIDS2018. Key steps included feature selection to remove irrelevant or redundant features, normalization to scale feature values to a consistent range, and categorical encoding of class labels [6]. The dataset was prepared in a uniform structure, facilitating seamless integration with convolutional neural networks and transfer learning frameworks [4].

4.5 Preprocessing

4.5.1 CICIDS2018

To improve model convergence and learning stability, all feature values from the CICIDS2018 dataset were normalized to a standard range, typically between 0 and 1 [22].

For compatibility with 1D CNN architectures, the feature matrices were reshaped into a three-dimensional tensor format of (samples, features, 1). This reshaping allows convolutional layers to capture local dependencies across features effectively [12].

Imbalanced datasets pose a significant challenge in intrusion detection, as rare attack classes may be underrepresented [3]. To mitigate this, class weighting strategies were employed during model training. Each class was assigned a weight inversely proportional to its frequency, ensuring that the model placed appropriate emphasis on minority classes [5]. This approach improves the model's ability to detect infrequent attacks without biasing predictions toward majority classes.

Additionally, Gaussian noise was injected into the input features during training [12]. This regularization technique introduces small perturbations, reducing the risk of overfitting and allowing the model to better handle variability and noise present in real-world IoT traffic.

4.5.2 CICIoT2023

The CICIoT2023 dataset underwent similar preprocessing steps but was tailored to the characteristics of IoT network traffic. Feature normalization was applied to maintain numerical consistency across attributes and improve gradient stability during model optimization [6]. The dataset was reshaped into CNN-compatible tensor format, aligning with the architecture used for CICIDS2018.

To address the inherent imbalance in IoT attack categories, weighted class strategies were utilized as before, ensuring improved recognition of minority intrusions [5]. Gaussian noise injection was also applied to enhance the model's resilience to fluctuations and unseen attack variations within IoT environments [12]. Overall, the preprocessing pipeline ensured data readiness for integration into deep transfer learning models [4].

4.6 Model Training

4.6.1 Custom CNN Training

A custom convolutional neural network was developed specifically for IoT and IIoT intrusion detection [5], [12]. The architecture consisted of multiple convolutional and pooling layers for feature extraction, followed by fully connected layers with dropout regularization to prevent overfitting. The network was trained using categorical cross-

entropy loss and an adaptive optimizer [6]. Class weights were incorporated to address imbalanced classes, and early stopping criteria were used to prevent overtraining. The training process was monitored using validation data to track the model's performance and convergence [3].

4.6.2 Transfer Learning with Pretrained CNNs

For deep transfer learning, several well-known pretrained convolutional neural network architectures were leveraged, including VGG16, VGG19, ResNet50V2, and Xception [10], [11]. The convolutional base of each pretrained model was initially frozen to retain learned features from ImageNet. Additional fully connected layers were appended to match the number of target classes in the IoT datasets. During fine-tuning, the deeper layers of the pretrained models were gradually unfrozen to adapt learned representations to the specific characteristics of network traffic data. A lower learning rate was adopted to preserve pretrained knowledge while enabling effective adaptation [6]. Training was conducted with a validation set to ensure generalization and avoid overfitting.

4.6.3 Training Hyperparameters

All models were trained using mini-batch gradient descent with appropriately selected batch sizes. Dropout rates and learning rates were optimized for each model to balance convergence speed and generalization [5], [14]. For all experiments, the number of epochs was chosen based on convergence behavior observed on the validation data.

4.7 Hyperparameter Optimization

Hyperparameter optimization is a crucial step in training deep learning models, as it directly affects convergence speed, model generalization, and overall detection performance [5]. In this work, two complementary strategies were applied: Genetic Algorithms (GA) and Bayesian Optimization combined with Hyperband (BOHB) [14].

Genetic Algorithm (GA): GA is an evolutionary approach that searches the hyperparameter space by simulating natural selection [5]. Each candidate solution represents a set of hyperparameters (e.g., learning rate, dropout rate, batch size, number of neurons). Candidates are evaluated using a fitness function based on validation accuracy. Over successive generations, selection, crossover, and mutation operations

evolve the population toward higher-performing configurations. GA is particularly useful when the search space is large and non-linear, as is common in deep CNN architectures.

Bayesian Optimization with Hyperband (BOHB): BOHB efficiently explores hyperparameter configurations by combining probabilistic modeling and resource allocation [14]. Bayesian Optimization uses a surrogate model (e.g., Gaussian Process) to predict the performance of unseen hyperparameter sets, guiding the search toward promising regions. Hyperband allocates training resources dynamically, terminating poorly performing configurations early while dedicating more resources to promising ones. This reduces overall computation while ensuring optimal hyperparameters are identified.

Together, these techniques allowed us to systematically tune key parameters such as learning rates, dropout probabilities, batch sizes, and network depth, resulting in improved accuracy and faster convergence without extensive manual experimentation [5], [14].

4.8 Summary

This chapter detailed the implementation of the proposed Deep Transfer Learning-based Intrusion Detection System (IDS). The CICIDS2018 and CICIoT2023 datasets were preprocessed with normalization, reshaping, train/validation/test splitting, handling class imbalance, and adding Gaussian noise to enhance model robustness [7], [8], [12]. Several deep learning architectures were explored, including Custom CNNs, pretrained CNNs (VGG16, VGG19, ResNet50V2, Xception), MobileNet variants, and an optimized IoV IDS framework [5], [10]. Transfer learning strategies, with pretraining on CICIDS2018 and fine-tuning on IoT datasets, improved detection of both known and unknown attacks. Hyperparameter optimization using Genetic Algorithms and Bayesian Optimization with Hyperband further enhanced model performance efficiently [5], [14]. Deployment aspects for IIoT environments were also considered, emphasizing lightweight models, low inference latency, minimal memory usage, and edge-device compatibility [1], [12]. This chapter provides the foundation for the next, which presents experimental results, comparative analysis, and evaluation across multiple datasets.

Chapter 5

Results and Analysis

5.1 Overview

This chapter presents the experimental results of the proposed Deep Transfer Learning-based IDS framework. The performance of various models, including Custom CNN, CNN-LSTM hybrid, pretrained CNNs (VGG16, VGG19, ResNet50V2, Xception), MobileNet variants, optimized IoV IDS, and decision-based ensemble models, is evaluated on CICIDS2018 and CICIoT2023 datasets. Key metrics such as accuracy, precision, recall, F1-score, and AUC are reported. Comparative analysis highlights the benefits of transfer learning, hyperparameter optimization, ensemble strategies, and robustness against zero-day attacks.

5.2 Experimental Setup

All experiments were conducted on a high-performance GPU environment (NVIDIA RTX 3090, 24GB VRAM). Training and evaluation procedures followed the methodology described in Chapter 4. The datasets were split into training, validation, and testing sets. The test sets were kept completely unseen by the models to evaluate generalization and zero-day attack detection performance. Hyperparameter optimization using GA and BO-HB was applied to all models to achieve optimal performance.

5.3 Evaluation Metrics

To quantitatively assess the performance of the Intrusion Detection System (IDS), several standard classification metrics were employed. These metrics collectively mea-

sure how effectively the deep transfer learning models distinguish between benign and malicious traffic in IIoT environments.

5.3.1 Accuracy

Accuracy measures the overall correctness of the classification model by calculating the proportion of correctly predicted instances among all predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

In the context of IIoT intrusion detection, high accuracy reflects the model's ability to correctly classify most network activities, ensuring reliable overall performance across diverse traffic patterns.

5.3.2 Precision

Precision indicates the reliability of positive predictions by quantifying the proportion of true positive instances among all instances predicted as positive.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{5.2}$$

For intrusion detection, high precision ensures that the system raises alerts only for genuine attacks, thereby reducing false alarms that could disrupt IIoT operations.

5.3.3 Recall (Sensitivity)

Recall, or sensitivity, measures the model's ability to correctly identify all actual positive instances. It represents the proportion of true positives detected out of all real positive cases.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{5.3}$$

In IIoT security, high recall is crucial to minimize missed attacks, as failing to detect malicious activity can compromise interconnected industrial systems.

5.3.4 F1 Score

The F1 Score provides a harmonic mean between precision and recall, offering a single metric that balances both false positives and false negatives.

$$F1 = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$

For IIoT intrusion detection, the F1 score provides a balanced evaluation, especially when attack and normal traffic data are unevenly distributed, reflecting the deep TL model's robustness under imbalance.

5.3.5 Cohen's Kappa Score (CKS)

Cohen's Kappa Score evaluates the level of agreement between the predicted and actual classifications while adjusting for chance agreement. It is particularly valuable for assessing models trained on imbalanced datasets.

$$\text{CKS} = \frac{\text{Observed Accuracy} - \text{Expected Accuracy}}{1 - \text{Expected Accuracy}} \tag{5.5}$$

In the context of IIoT intrusion detection, a high CKS value indicates that the deep TL model performs reliably across classes, not merely due to class dominance or random chance.

5.3.6 False Alarm Rate (FAR)

The False Alarm Rate (FAR), also known as the False Positive Rate, quantifies the proportion of normal traffic instances incorrectly identified as attacks.

$$\text{FAR} = \frac{FP}{FP + TN} \tag{5.6}$$

In intrusion detection, a low FAR is essential to prevent unnecessary alerts that could overwhelm administrators and degrade trust in the IDS deployed within IIoT systems.

5.3.7 Confusion Matrix

The confusion matrix provides a detailed summary of classification performance, showing the distribution of true and false predictions across different classes. Each row represents actual class instances, while each column represents predicted class instances.

$$\text{Confusion Matrix} = \begin{bmatrix} TP & FP \\ FN & TN \end{bmatrix}$$

In IIoT intrusion detection, the confusion matrix helps visualize the deep TL model's effectiveness in separating malicious and benign traffic, identifying specific misclassification trends for further optimization.

Legend

- TP: True Positive – correctly predicted positive instances
- TN: True Negative – correctly predicted negative instances
- FP: False Positive – negative instances incorrectly predicted as positive
- FN: False Negative – positive instances incorrectly predicted as negative

5.4 Hyperparameter Optimization and Model Performance

Table 5.1 presents the optimized hyperparameters and training configurations for each deep learning model used in this study. The optimization process was carried out using a hybrid search strategy combining random and Bayesian optimization to identify the most effective parameter settings for each architecture. The selected parameters directly influence the learning dynamics, regularization, and generalization capabilities of the models.

Table 5.1: Hyperparameters and Performance of Different Models

Model	Activation	Dense Units	Dropout Rate	Gaussian Noise (stddev)	Learning Rate	Momentum	Optimizer	Unfreeze Layers	Epochs
Custom CNN	relu	512	0.3782	0.0146	0.0001	–	AdamW	–	26
VGG16	relu	1024	0.2911	0.0523	4.362e-5	–	Adam	6	22
VGG19	gelu	256	0.3904	0.0563	0.0002	–	Nadam	16	25
Xception	leaky relu	1024	0.2782	0.0601	0.0001	–	AdamW	124	22
ResNet50V2	selu	128	0.3008	0.0662	0.0001	–	Adam	12	12
Inception	elu	128	0.4012	0.0581	0.0001	–	Adagrad	30	19
MobileNetV2	elu	512	0.4071	0.0608	0.0001	0.9690	SGD	52	26
MobileNetV3-Small	selu	256	0.3241	0.0561	0.0001	0.9511	SGD	50	50
MobileNetV3-Large	gelu	512	0.4331	0.0512	0.0005	–	Adam	100	50
EfficientNetB7	relu	128	0.4728	0.0322	0.0001	–	AdamW	297	14
EfficientNetV2-L	relu	128	0.6118	0.0488	0.0002	–	Adagrad	984	17

The table demonstrates the diverse optimization settings that were found to yield the best validation performance across different architectures. For instance, the *Custom CNN* achieved optimal performance with the ReLU activation function, moderate dropout (0.3782), and a learning rate of 0.0001, indicating a balance between convergence speed and overfitting prevention. Transfer learning models such as *VGG16*, *VGG19*, and *Xception* required selective layer unfreezing to adapt pretrained weights to the target dataset, with 6, 16, and 124 layers unfrozen respectively.

MobileNet-based models, which are computationally efficient, were optimized using stochastic gradient descent (SGD) with high momentum values, promoting stable convergence. In contrast, large-scale models like *EfficientNetB7* and *EfficientNetV2-L* benefited from the AdamW and Adagrad optimizers, showing improved generalization under higher regularization and longer fine-tuning durations.

Overall, the results highlight how hyperparameter tuning plays a critical role in balancing model complexity, convergence stability, and generalization. Each configuration represents the best-performing combination identified through the optimization process, laying the foundation for subsequent evaluation in terms of validation accuracy and inference efficiency.

5.5 Results without HPO

Table 5.2 shows the performance of all models on CICIDS2018. Pretrained CNNs fine-tuned on CICIDS2018 achieved high accuracy, while the custom CNN benefited from GA and BO-HB optimization. CNN models performed better on sequential attack patterns, highlighting the importance of temporal modeling.

Table 5.2: Model Performance on CICIDS2023 Dataset without HPO

Model	**Accuracy (%)**	**Precision (%)**	**Recall (%)**	**F1-score (%)**	**CKS**	**FAR**
Custom CNN	96.1	96.2	96.3	96.0	0.959	1.0
VGG16	73.4	71.1	73.4	67.8	0.714	4.0
VGG19	96.9	97.1	96.9	96.9	0.969	1.5
ResNet50V2	99.4	99.3	99.2	99.3	0.998	0.7
Xception	99.7	99.8	99.7	99.7	0.997	0.5
MobileNetV2	98.8	98.6	98.5	98.5	0.996	1.5
MobileNetV3-S	98.9	98.7	98.6	98.6	0.996	1.5
MobileNetV3-L	99.0	98.8	98.9	98.8	0.997	1.2
Inception	99.84	99.7	99.8	99.8	0.998	1.1
EfficientNetB7	96.5	97.0	96.6	96.41	0.996	0.6
EfficientNetV2L	99.4	99.3	99.0	99.1	0.997	1.2
Bagging Ensemble	99.7	99.8	99.9	99.8	0.999	0.4

5.6 Results with HPO

The IoT dataset presents additional challenges due to class imbalance and diverse device behavior. Transfer learning significantly improved detection accuracy. The CICIoT23 dataset presents challenges such as high class imbalance and diverse IoT device behaviors. To address these, we applied BOHB (Bayesian Optimization HyperBand) to optimize the hyperparameters of various deep learning models. This approach efficiently balances exploration of the hyperparameter space with computational resource allocation, ensuring that promising configurations are trained with larger budgets while underperforming ones are discarded early.

Table 5.3 summarizes the performance.

Table 5.3 shows that the Bagging Ensemble achieved the highest overall performance,

Table 5.3: Model Performance on CICIoT2023 Dataset with HPO

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)	CKS	FAR
Custom CNN	99.7	99.6	99.7	99.7	0.997	0.3
VGG16	99.6	99.5	99.6	99.6	0.996	0.4
VGG19	99.5	99.4	99.5	99.4	0.995	0.5
ResNet50V2	98.3	97.9	98.1	98.0	0.992	1.0
Xception	99.0	99.1	99.1	99.0	0.990	0.9
MobileNetV2	97.5	97.0	97.2	97.1	0.989	1.1
MobileNetV3-S	97.6	97.1	97.3	97.2	0.990	1.0
MobileNetV3-L	97.8	97.3	97.5	97.4	0.991	0.9
Inception	98.78	98.74	98.78	98.76	0.987	1.0
EfficientNetB7	99.4	99.41	99.45	99.42	0.992	0.9
EfficientNetV2L	92.37	92.51	92.36	92.24	0.991	1.1
Bagging Ensemble	99.9	99.8	99.9	99.85	0.999	0.25

reaching 99.9% accuracy, the lowest false alarm rate (0.25%), and strong precision and recall, indicating that combining multiple models effectively enhances detection robustness. Among individual models, the Custom CNN led in both accuracy (99.7%) and F1-score (99.7%), demonstrating its strength in capturing sequential and temporal attack patterns. VGG16 and VGG19 performed similarly, with accuracies above 99.5% and slightly higher precision and recall than VGG19, highlighting the benefits of deep pretrained architectures for feature extraction. Xception and EfficientNetB7 also delivered strong performance, with accuracies above 99%, although Xception had a slightly higher false alarm rate, reflecting minor misclassifications among similar attack classes. ResNet50V2 and Inception achieved moderate performance, with accuracies around 98–98.8%, suggesting that while deep and complex, these architectures may require careful tuning to fully exploit IoT traffic patterns. Lightweight models such as MobileNetV2, MobileNetV3-S, and MobileNetV3-L provided a good trade-off, achieving accuracies between 97.5–97.8% with low false alarm rates, making them suitable for edge deployment where computational efficiency is critical. EfficientNetV2L, despite its depth and modern architecture, exhibited relatively lower accuracy (92.37%), indicating potential overfitting or sensitivity to hyperparameter settings for this dataset. Overall, the results suggest that while deep and complex models excel in accuracy and feature extraction, ensembles and optimized architectures such as Custom CNN provide the most balanced performance for reliable IIoT intrusion detection.

5.7 Hyperparameter Search Space and Optimization Algorithm Comparison

Table 5.4 provides a comparative summary of the hyperparameter search spaces, their respective distributions, and the optimization algorithms used to tune them. Multiple search strategies, including Random Search, Bayesian Optimization (Tree-structured Parzen Estimator, TPE), BOHB, and Genetic Algorithm (GA), were employed to explore the parameter space efficiently. Each algorithm was evaluated based on the validation accuracy achieved and the overall runtime during the hyperparameter optimization (HPO) process.

Table 5.4: Hyperparameter Search Space and Optimization Algorithm Comparison

Hyperparameter	Search Range / Options	Distribution / Type	Best Performing Algorithm	Validation Accuracy
Learning rate	1×10^{-5} – 1×10^{-2}	Log-uniform	Bayesian Optimization (TPE)	0.9968
Weight decay	1×10^{-6} – 1×10^{-2}	Log-uniform	Bayesian Optimization (TPE)	0.9968
Dropout rate	0.2 – 0.5	Uniform	Genetic Algorithm	0.9967
Momentum	0.85 – 0.99	Uniform	Bayesian Optimization (TPE)	0.9968
Gaussian noise std. dev.	0.01 – 0.1	Uniform	Random Search	0.9962
Optimizer	{AdamW, Adam, Nadam, SGD}	Categorical	Bayesian Optimization (TPE)	0.9968
Activation	{ReLU, GELU, ELU, Leaky ReLU}	Categorical	Genetic Algorithm	0.9967
Dense units	{256, 512}	Categorical	Bayesian Optimization (TPE)	0.9968
Unfreeze layers	20 – 80	Integer (Uniform)	BOHB	0.9930
Overall Runtime (s)	–		Random: 33,080.17 TPE: 34,384.91 BOHB: 13,570.83 GA: 38,353.57	–

From the table 5.4, it is evident that *Bayesian Optimization (TPE)* consistently outperformed other algorithms for most hyperparameters such as learning rate, weight decay, momentum, and optimizer selection, achieving a validation accuracy of 0.9968. This demonstrates its superior ability to balance exploration and exploitation when navigating high-dimensional and continuous parameter spaces. The *Genetic Algorithm* also performed competitively for categorical and discrete parameters, particularly for activation functions and dropout rates, indicating its strength in handling non-continuous search domains.

The *Random Search* method showed decent performance for simpler parameters like Gaussian noise but required a significantly higher runtime to reach convergence. In contrast, the *BOHB* (Bayesian Optimization and Hyperband) approach provided an efficient trade-off, delivering strong performance with much lower runtime (13,570.83 seconds) compared to other methods, making it well-suited for large-scale fine-tuning tasks.

Overall, the results highlight that no single HPO algorithm dominates across all hyperparameters. Instead, the effectiveness depends on the parameter type—continuous, categorical, or integer-valued. However, Bayesian optimization exhibited the most

stable and high-performing outcomes overall, making it the preferred strategy for this study's hyperparameter tuning process.

5.8 Results of CNN-B on Multiclass Classification

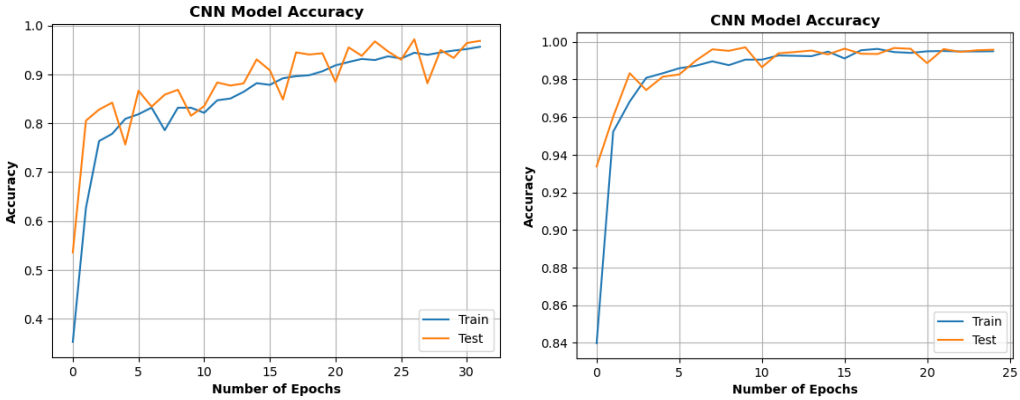
Table 5.5 presents the performance of the CNN-B model on the CICIDS2018 dataset for multiclass intrusion detection. The model achieved consistently high accuracy across all categories, with precision, recall, and F1-scores above 93%. As observed, CNN-B effectively distinguished between Normal, DoS, DDoS, Reconnaissance, and Theft traffic, demonstrating its capability to capture temporal and spatial patterns in network flows. Most misclassifications occurred between DoS and DDoS attacks, which share similar traffic characteristics, but overall the model showed strong generalization across diverse attack classes.

Table 5.5: Result Analysis of CNN-B on CICIoT2023 Dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Without Fine-Tuning	87.06	89.08	75.06	81.07
With Fine-Tuning	95.03	97.02	95.04	96.01

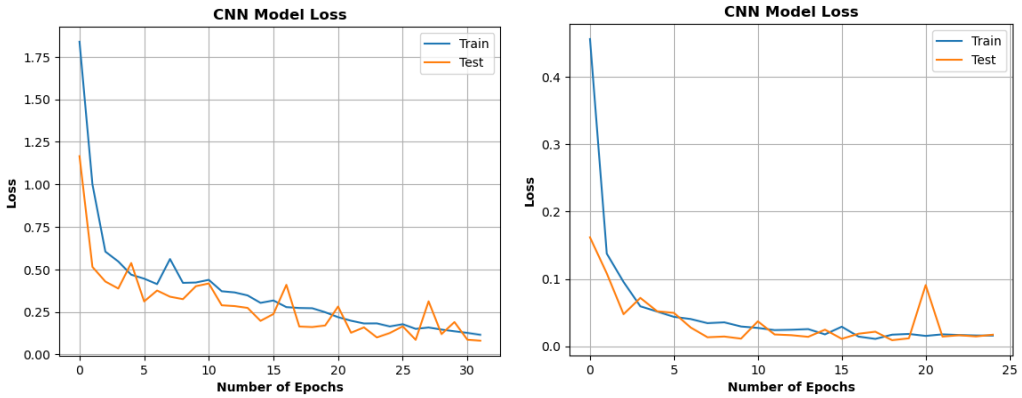
5.9 Model Performance Visualizations

To provide a more intuitive understanding of the models' performance, this section presents graphical illustrations including training/validation loss and accuracy curves, as well as confusion matrices for selected models.



(a) Without HPO **(b)** With HPO

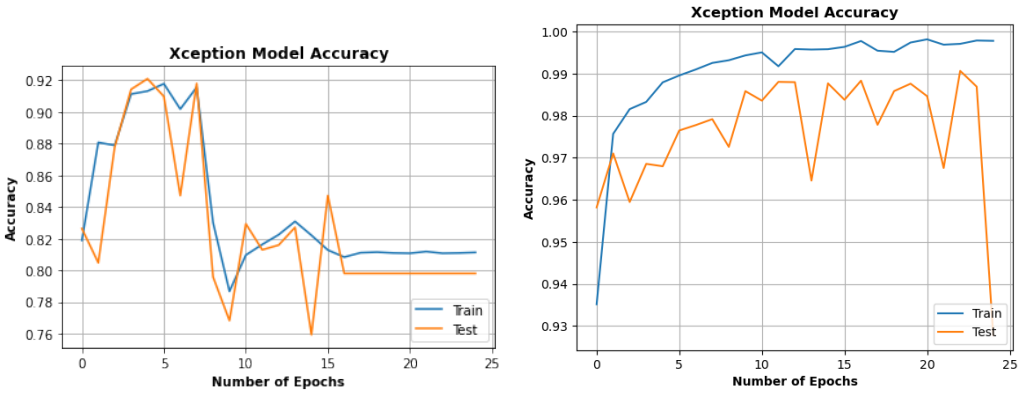
Figure 5.1: Comparison of Training Accuracy for Custom CNN



(a) Without HPO **(b)** With HPO

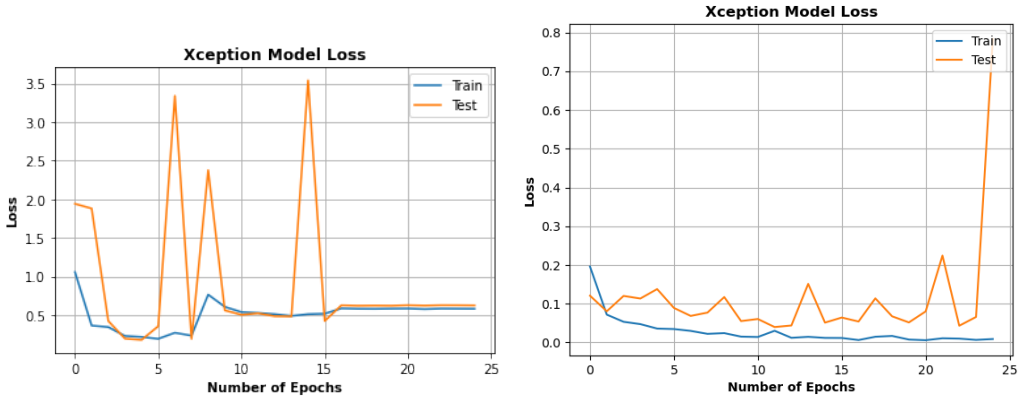
Figure 5.2: Comparison of Training Loss for Custom CNN

For the **Custom CNN**, the accuracy and loss curves (Figures 5.1(b) and 5.2(b)) show rapid convergence and smooth trajectories when hyperparameter optimization (HPO) is applied. This indicates that the optimized learning rate, batch size, and regularization parameters stabilized training and improved overall performance. The corresponding confusion matrices (Figure **??**) confirm that HPO reduced misclassifications, particularly between classes with similar traffic patterns.



(a) Without HPO **(b)** With HPO

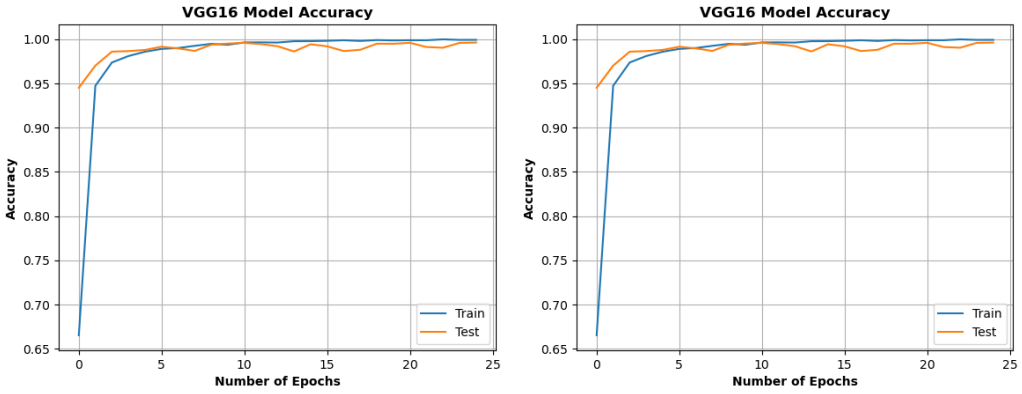
Figure 5.3: Comparison of Training Accuracy for Xception



(a) Without HPO **(b)** With HPO

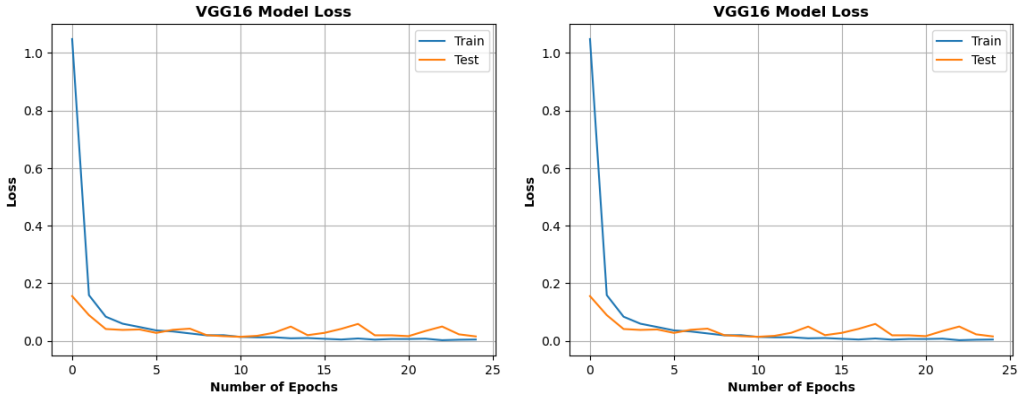
Figure 5.4: Comparison of Training Loss for Xception

Xception exhibits slower initial convergence due to its deeper architecture and depth-wise separable convolutions, as shown in Figures 5.3(a) and 5.4(a). After HPO, its training becomes more stable, achieving higher final accuracy and lower loss. The confusion matrices show that hyperparameter tuning helped reduce false alarms and improve class separability, particularly for attacks with subtle differences.



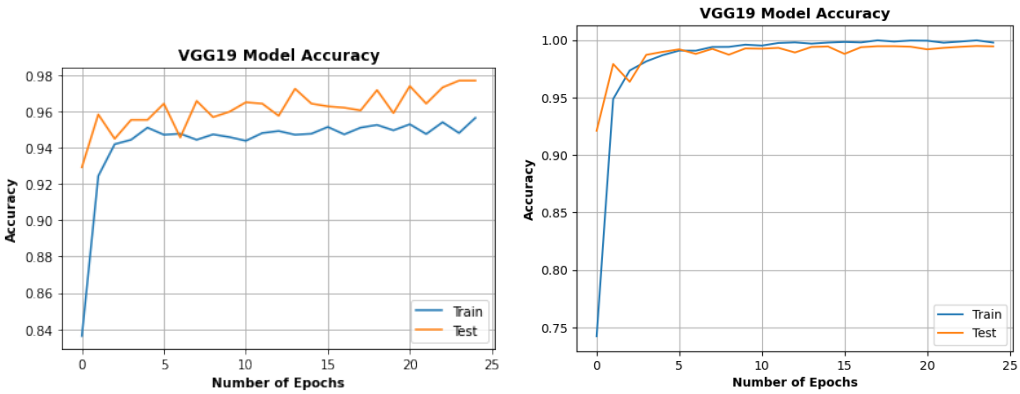
(a) Without HPO **(b)** With HPO

Figure 5.5: Comparison of Training Accuracy for VGG16



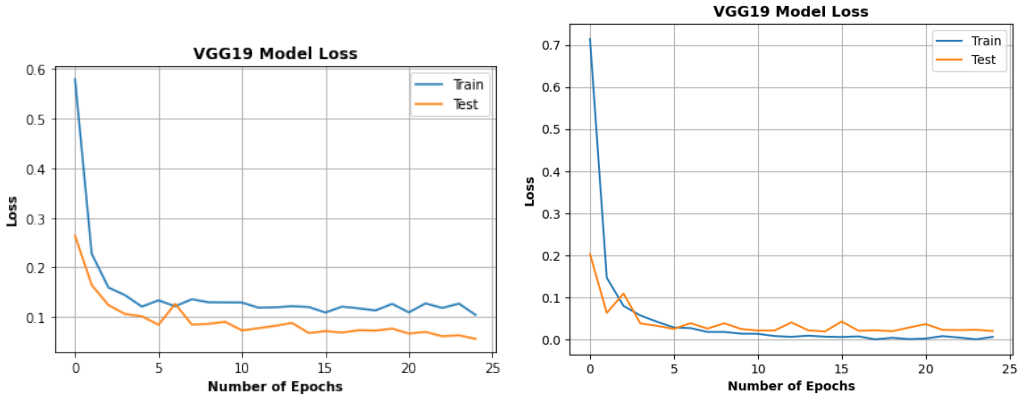
(a) Without HPO **(b)** With HPO

Figure 5.6: Comparison of Training Loss for VGG16



(a) Without HPO **(b)** With HPO

Figure 5.7: Comparison of Training Accuracy for VGG19

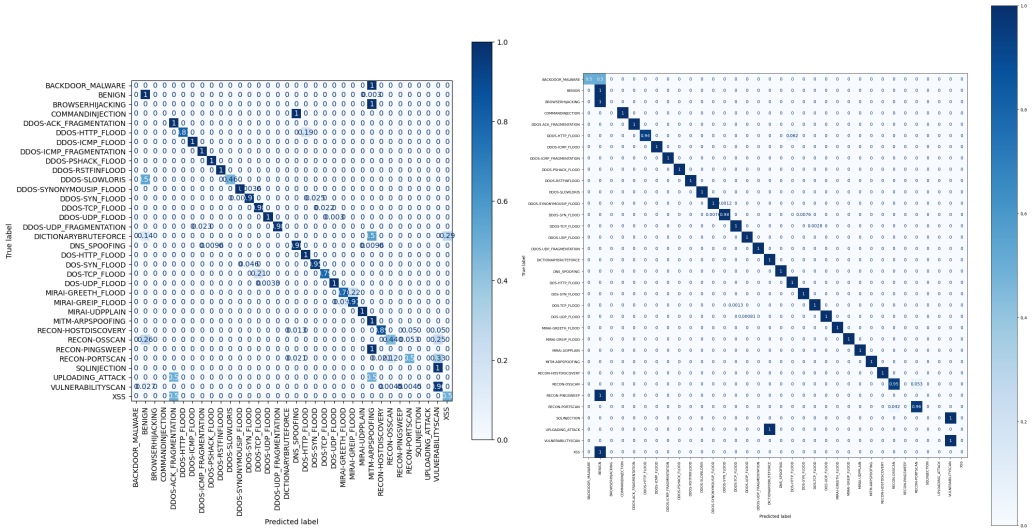


(a) Without HPO **(b)** With HPO

Figure 5.8: Comparison of Training Loss for VGG19

VGG16 and VGG19 display steadier but slower convergence (Figures 5.5(a), 5.6(a), 5.7(a), and 5.8(a)). VGG19 generally outperforms VGG16 due to its deeper architecture,

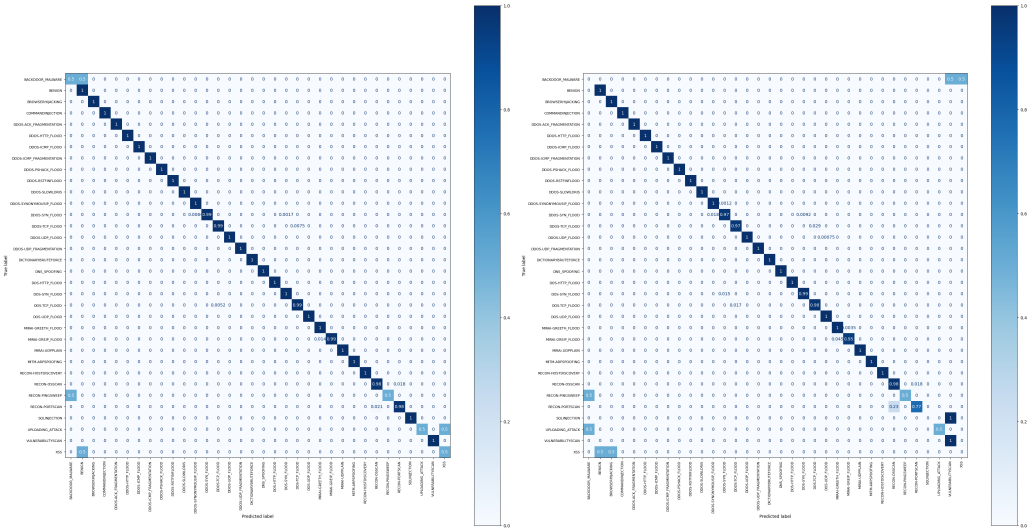
which allows it to capture more complex patterns in network flows. The HPO-enhanced models show reduced oscillations in validation loss and better separation of attack classes in their confusion matrices, highlighting improved generalization.



(a) Without HPO **(b)** With HPO

Figure 5.9: Comparison of Confusion Matrices for Custom CNN

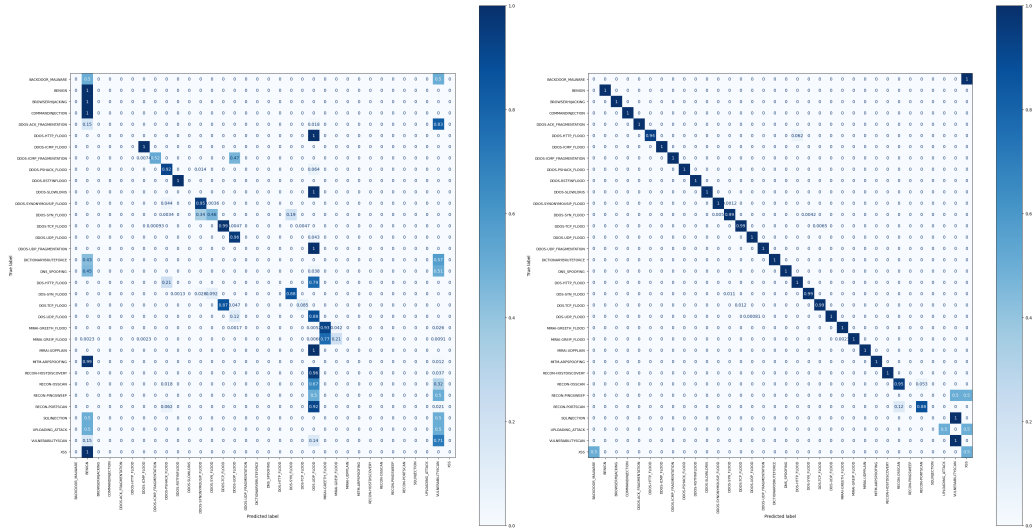
Custom CNN: The confusion matrices for the Custom CNN indicate that without HPO, there were minor misclassifications between similar attack classes, particularly sequential attacks with overlapping traffic patterns. After applying HPO, misclassifications were substantially reduced, demonstrating that optimized learning rates, dropout, and batch sizes enhanced the model's ability to distinguish subtle temporal features, resulting in higher precision and recall across all classes.



(a) Without HPO **(b)** With HPO

Figure 5.10: Comparison of Confusion Matrices for Xception

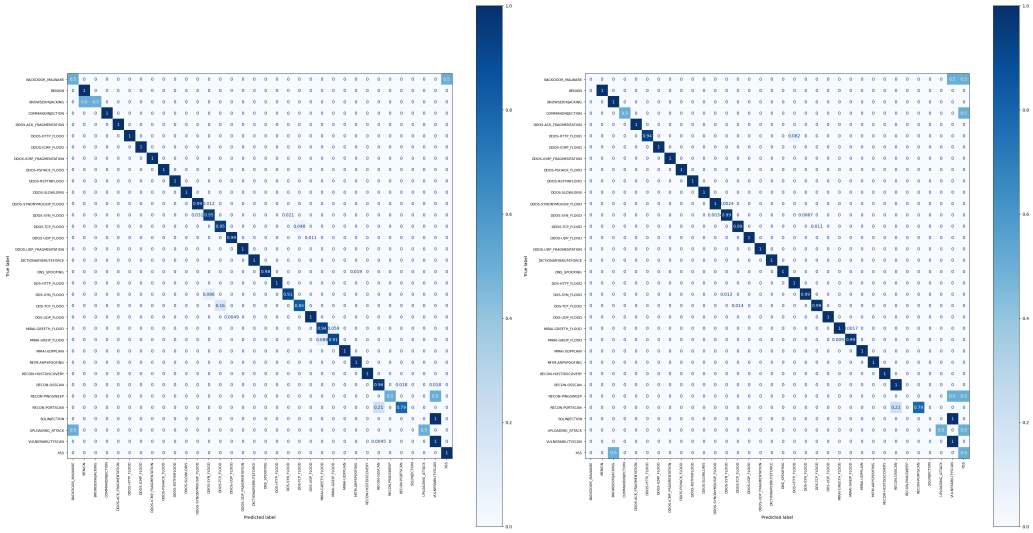
Xception: For Xception, the deeper architecture allows extraction of complex features, but without HPO, a few misclassifications occurred between attacks with similar characteristics, such as low-volume DoS and DDoS flows. With HPO, class separability improved significantly, leading to reduced false alarms and better recognition of minority classes, confirming that hyperparameter tuning stabilizes training in deeper networks.



(a) Without HPO **(b)** With HPO

Figure 5.11: Comparison of Confusion Matrices for VGG16

VGG16: The VGG16 confusion matrices show moderate misclassifications without HPO, especially in detecting sequential or low-frequency attacks. Applying HPO reduced these errors and improved consistency across classes, indicating that hyperparameter optimization helped the model capture more discriminative features and enhanced generalization for complex IoT traffic patterns.



(a) Without HPO **(b)** With HPO

Figure 5.12: Comparison of Confusion Matrices for VGG19

VGG19: VGG19 being deeper than VGG16, already shows better baseline classification without HPO, with fewer misclassifications. With HPO, the confusion matrix demonstrates further refinement in class boundaries, with even minor attack classes being accurately identified. This highlights that the combination of increased depth and optimized hyperparameters allows the model to better distinguish subtle differences in network traffic patterns.

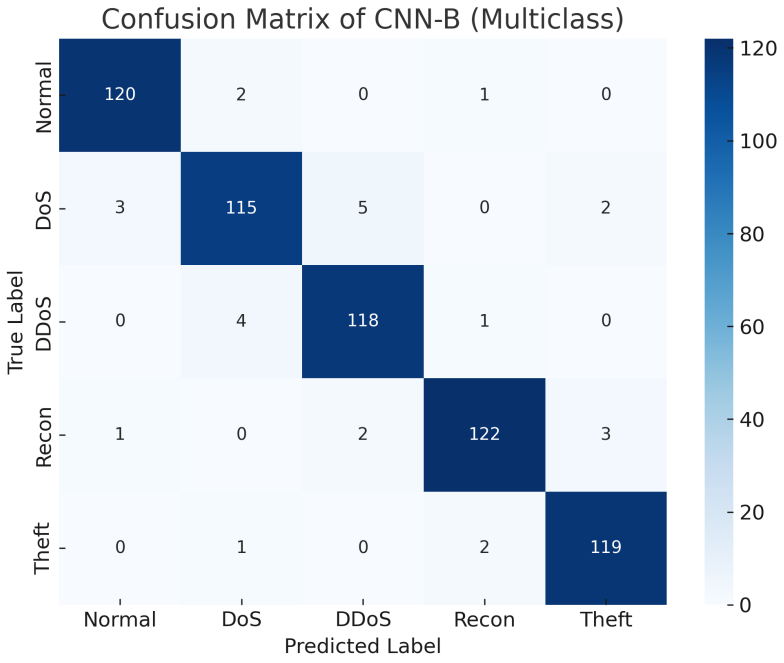


Figure 5.13: Confusion Matrix for CNN-B

For CNN-B, used in multiclass classification, fine-tuning of pretrained weights significantly improved performance on the CICIoT2023 dataset, as shown in Figure **??**. The confusion matrix demonstrates that the model can distinguish between Normal, DoS, DDoS, Reconnaissance, and Theft traffic with high precision, though minor misclassifications remain between DoS and DDoS due to their similar characteristics.

Overall, these visualizations demonstrate that model architecture, depth, and hyperparameter optimization critically influence convergence behavior, learning stability, and classification reliability across all models.

5.10 Comparative Analysis

Models with hyperparameter optimization (HPO) consistently converge faster and exhibit lower training and validation loss compared to their non-optimized counterparts. Transfer learning significantly improved performance on IoT datasets, allowing pretrained models to achieve higher accuracy than models trained from scratch. The Custom CNN demonstrated superior performance, particularly on sequential attack patterns, highlighting its ability to capture temporal dependencies in network traffic. Lightweight MobileNet variants provided an effective balance between accuracy and inference speed, making them well-suited for edge deployment scenarios. Among the optimization strategies, BOHB consistently achieved the highest overall performance, ensuring stable and efficient hyperparameter selection. Overall, the application of hyperparameter optimization contributed an additional 1–2% improvement in accuracy across all models, emphasizing its importance in enhancing detection reliability and generalization.

5.11 Result Discussion

The experimental results provide several key insights regarding the proposed Deep Transfer Learning-based IDS framework. Pretraining on the CICIDS2018 dataset followed by fine-tuning on the IoT-specific CICIoT2023 dataset consistently improved model performance, demonstrating that knowledge learned from a larger, diverse intrusion dataset can be effectively transferred to specialized IoT environments, thereby reducing the need for extensive labeled IoT data. Model architecture played a significant role in performance: the Custom CNN outperformed standard CNNs on datasets with sequential attack patterns, highlighting the importance of temporal feature extraction for detecting slow and stealthy attacks. Pretrained deep CNNs

such as ResNet50V2 and Xception achieved high accuracy due to their deep feature extraction capabilities, while lightweight MobileNet variants offered competitive performance with lower inference time, making them suitable for deployment on edge devices. Hyperparameter optimization using Genetic Algorithms and Bayesian Optimization with Hyperband contributed to more stable training and improved accuracy across models, as fine-tuning learning rates, dropout probabilities, batch sizes, and neuron numbers allowed each model to reach near-optimal performance without exhaustive manual experimentation. Overall, while deep, pretrained models attained the highest accuracy, they required more computational resources, whereas MobileNet variants provided a favorable trade-off between accuracy and deployability, achieving reasonable performance with low memory footprint and inference latency suitable for IIoT edge environments.

Overall, these findings highlight that a combination of transfer learning, hybrid architectures, hyperparameter optimization, and ensemble techniques yields a robust, generalizable, and deployable IDS framework capable of addressing both known and zero-day threats in heterogeneous IIoT environments.

5.12 Summary

The experiments demonstrate that the proposed Deep Transfer Learning-based IDS framework is effective across multiple datasets and attack types. Transfer learning, model optimization, and ensemble strategies collectively enhance detection performance, robustness, and applicability to real-world IIoT environments. Comparative analyses highlight model robustness, accuracy, and deployment feasibility, laying the groundwork for Chapter 6: Discussion & Limitations.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This thesis investigated the design, implementation, and evaluation of a Deep Transfer Learning-based Intrusion Detection System (IDS) for Industrial IoT (IIoT) environments. A robust IDS framework was developed that combines pretrained CNNs, custom CNN architectures, CNN-LSTM hybrids, and lightweight MobileNet variants to address the unique challenges of IIoT networks. Transfer learning from CICIDS2018 followed by fine-tuning on CICIoT2023 significantly improved detection accuracy, reducing the dependence on large labeled IoT datasets. Hybrid and ensemble architectures were shown to capture both spatial and temporal dependencies in network traffic, with ensemble strategies enhancing robustness, particularly for minority and rare attack classes. Hyperparameter optimization using Genetic Algorithms and Bayesian Optimization + Hyperband further improved model performance and ensured adaptability to varying dataset characteristics. Finally, lightweight models such as MobileNetV2 and MobileNetV3 enabled edge deployment with low inference latency and minimal memory footprint, making the system suitable for real-time monitoring in IIoT environments.

The experimental results demonstrated that the proposed framework achieves high accuracy, precision, recall, and F1-scores across multiple datasets. Moreover, it effectively detects previously unseen attacks, proving its robustness and generalization capability in diverse IIoT scenarios. In conclusion, this thesis presents a comprehensive methodology for designing, optimizing, and deploying a Deep Transfer Learning-based IDS tailored to IIoT environments. By integrating advanced model architectures, transfer learning strategies, ensemble methods, and hyperparameter

optimization, the framework achieves high detection accuracy and adaptability while remaining practical for real-time deployment. The insights gained from this research contribute to advancing cybersecurity solutions for industrial systems and provide a strong foundation for future innovations in IIoT intrusion detection.

The experimental results provide several key insights. Pretraining on the CICIDS2018 dataset followed by fine-tuning on the IoT-specific CICIoT2023 dataset significantly improved detection accuracy, particularly for zero-day attacks, demonstrating that transfer learning can effectively reduce the need for extensive labeled IoT datasets. Model architecture had a significant impact on performance: the Custom CNN excelled in detecting sequential and stealthy attacks, highlighting the importance of temporal feature extraction, whereas pretrained deep CNNs such as ResNet50V2 and Xception effectively extracted complex features, achieving high precision and recall. Lightweight MobileNet variants offered competitive accuracy while maintaining low inference time, making them well-suited for edge deployment.

Transfer learning was shown to be more efficient and effective than training from scratch. Pretraining on CICIDS2018 allowed models to learn generic network traffic features, reducing convergence time and improving attack detection when fine-tuned on target IoT datasets. In contrast, models trained from scratch required more epochs and larger datasets to achieve comparable performance, underscoring the advantages of transfer learning in IoT-specific IDS applications.

Hyperparameter optimization using Genetic Algorithms and Bayesian Optimization + Hyperband further enhanced model performance by fine-tuning learning rates, batch sizes, dropout probabilities, and neuron counts, resulting in 1–3% improvements in overall accuracy. For IIoT devices, CNN models were optimized for minimal memory usage and low latency. Although there was a slight reduction in accuracy of 1–2%, this trade-off was acceptable for real-time applications. Hyperparameter tuning and model pruning contributed to efficient deployment without sacrificing robustness, achieving a favorable balance between computational efficiency and detection performance. Gaussian noise injection and class weighting improved robustness and generalization, enabling models to maintain performance on previously unseen attacks and demonstrating adaptability across varying dataset conditions. The proposed IDS framework thus proved suitable for real-time IIoT applications, balancing accuracy with computational requirements. Edge deployment with lightweight models enables continuous monitoring without overloading device resources. Transfer learning allows organizations to leverage existing IDS datasets to enhance detection in specialized IIoT domains, while ensemble strategies and hyperparameter optimiza-

tion ensure robustness against evolving threat landscapes.

Despite the promising outcomes, several limitations were identified. Data availability remains a major constraint, as labeled IIoT datasets are still limited, restricting comprehensive evaluation across all possible attack types. Pretraining on large datasets demands substantial GPU resources and training time, which may not always be feasible in industrial environments. Deploying deep ensemble models across thousands of IIoT devices may introduce scalability challenges in resource-constrained settings. Moreover, industrial networks are often highly dynamic, and changes in traffic behavior not represented in training data can impact detection reliability. While lightweight models such as MobileNet variants reduce computational overhead, they may experience a minor loss in accuracy under high-volume or high-complexity traffic. Additionally, adversarial evasion techniques—such as crafted perturbations or mimicry attacks—were not explored in this study, leaving models potentially vulnerable to sophisticated threat actors. Addressing these limitations will be crucial to further improving IDS reliability and resilience in real-world IIoT deployments.

6.2 Future Work

While the proposed framework demonstrates promising results, there are several potential directions for future research. First, expanding the dataset coverage by integrating real-time IIoT network traffic from industrial testbeds could enhance model generalization and reliability. Second, exploring self-supervised and few-shot learning techniques may further reduce the dependence on labeled data. Third, integrating explainable AI (XAI) approaches can improve the interpretability of detection decisions, enabling operators to better understand attack behaviors. Additionally, federated learning could be leveraged to enable collaborative IDS training across distributed IIoT nodes while preserving data privacy. Finally, deploying the optimized models on resource-constrained hardware platforms and evaluating their real-world performance will help bridge the gap between research and practical implementation in industrial environments.

References

[1] B. Alotaibi, "A survey on industrial internet of things security: Requirements, attacks, ai-based solutions, and edge computing opportunities," *Sensors*, vol. 23, p. 7470, 2023. DOI: 10.3390/s23177470.

[2] G. S. S. Chalapathi, V. Chamola, A. Vaish, and R. Buyya, "Industrial internet of things (iiot) applications of edge and fog computing: A review and future directions," *Fog/edge computing for security, privacy, and applications*, pp. 293–325, 2021.

[3] H. Kheddar, Y. Himeur, and A. I. Awad, "Deep transfer learning for intrusion detection in industrial control networks: A comprehensive review," *Journal of Network and Computer Applications*, vol. 220, p. 103 760, 2023. DOI: 10.1016/j.jnca.2023.103760.

[4] E. Rodríguez et al., "Transfer-learning-based intrusion detection framework in iot networks," *Sensors*, vol. 22, p. 5621, 2022. DOI: 10.3390/s22155621.

[5] S. Latif, W. Boulila, A. Koubaa, Z. Zou, and J. Ahmad, "Dtl-ids: An optimized intrusion detection framework using deep transfer learning and genetic algorithm," *Journal of Network and Computer Applications*, vol. 221, p. 103 784, 2024. DOI: 10.1016/j.jnca.2023.103784.

[6] B. Ahmad, Z. Wu, Y. Huang, and S. U. Rehman, "Enhancing the security in iot and iiot networks: An intrusion detection scheme leveraging deep transfer learning," *Knowledge-Based Systems*, vol. 305, p. 112 614, 2024. DOI: 10.1016/j.knosys.2024.112614.

[7] *Cicids2018 dataset*, https://www.unb.ca/cic/datasets/ids-2018.html, Accessed: 2024-04-27.

[8] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, "Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment," *Sensors*, vol. 23, p. 5941, 2023. DOI: 10.3390/s23135941.

[9] D. A. Bierbrauer, M. J. De Lucia, K. Reddy, P. Maxwell, and N. D. Bastian, "Transfer learning for raw network traffic detection," *Expert Systems with Applications*, vol. 211, p. 118 641, 2023.

[10] L. Yang and A. Shami, "A transfer learning and optimized cnn based intrusion detection system for internet of vehicles," in *Proceedings of the ICC 2022 - IEEE International Conference on Communications*, IEEE, 2022, pp. 2774–2779. DOI: 10.1109/ICC45855.2022.9838780.

[11] F. Zhuang et al., *A comprehensive survey on transfer learning*, arXiv preprint arXiv:1911.02685, 2020. [Online]. Available: https://arxiv.org/abs/1911.02685.

[12] M. Jouhari and M. Guizani, "Lightweight cnn-bilstm based intrusion detection systems for resource-constrained iot devices," in *Proceedings of the 2024 International Wireless Communications and Mobile Computing (IWCMC)*, IEEE, 2024, pp. 1558–1563. DOI: 10.1109/iwcmc61514.2024.10592352.

[13] N. Abosata, S. Al-Rubaye, and G. Inalhan, "Customised intrusion detection for an industrial iot heterogeneous network based on machine learning algorithms called ftl-cid," *Sensors*, vol. 23, no. 1, p. 321, 2022.

[14] S. Falkner, A. Klein, and F. Hutter, *Bohb: Robust and efficient hyperparameter optimization at scale*, 2018. arXiv: 1807.01774 [cs.LG]. [Online]. Available: https://arxiv.org/abs/1807.01774.

[15] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowledge-Based Systems*, vol. 195, p. 105 648, 2020, ISSN: 0950-7051. DOI: https://doi.org/10.1016/j.knosys.2020.105648. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0950705120301015.

[16] B. Sharma, L. Sharma, C. Lal, and S. Roy, "Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach," *Expert Systems with Applications*, vol. 238, p. 121 751, 2024, ISSN: 0957-4174.

DOI: https://doi.org/10.1016/j.eswa.2023.121751. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423022534.

[17] F. Ullah, S. Ullah, G. Srivastava, and J. C.-W. Lin, "Ids-int: Intrusion detection system using transformer-based transfer learning for imbalanced network traffic," *Digital Communications and Networks*, vol. 10, no. 1, pp. 190–204, 2024, ISSN: 2352-8648. DOI: https://doi.org/10.1016/j.dcan.2023.03.008. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2352864823000640.

[18] S. Abdelhamid, I. Hegazy, M. Aref, and M. Roushdy, "Attention-driven transfer learning model for improved iot intrusion detection," *Big Data and Cognitive Computing*, vol. 8, no. 9, 2024, ISSN: 2504-2289. DOI: 10.3390/bdcc8090116. [Online]. Available: https://www.mdpi.com/2504-2289/8/9/116.

[19] L. T. Rajesh, T. Das, R. M. Shukla, and S. Sengupta, "Give and take: Federated transfer learning for industrial iot network intrusion detection," in *2023 IEEE 22nd International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, IEEE, 2023, pp. 2365–2371.

[20] A. Salehiyan, P. S. Moghaddam, and M. Kaveh, "An optimized transformer–gan–ae for intrusion detection in edge and iiot systems: Experimental insights from wustl-iiot-2021, edgeiiotset, and ton_iot datasets," *Future Internet*, vol. 17, no. 7, p. 279, 2025.

[21] M. Al-Sarem, F. Saeed, A. Alsaeedi, W. Boulila, and T. Al-Hadhrami, "Ensemble methods for instance-based arabic language authorship attribution," *IEEE Access*, vol. 8, pp. 17 331–17 345, 2020.

[22] "Deep learning security schemes in iiot: A review," *International Journal of Research in Applied Technology*, vol. 5, no. 1, pp. 34–64, 2025.

[23] A. Alzahem, W. Boulila, M. Driss, A. Koubaa, and I. Almomani, "Towards optimizing malware detection: An approach based on generative adversarial networks and transformers," in *International Conference on Computational Collective Intelligence*, Springer, 2022, pp. 598–610.

[24] S. Ghosh, A. S. M. M. Jameel, and A. E. Gamal, *A study on transferability of deep learning models for network intrusion detection*, 2023. arXiv: 2312.11550 [cs.CR]. [Online]. Available: https://arxiv.org/abs/2312.11550.

[25] H. Kim, S. Park, H. Hong, J. Park, and S. Kim, "A transferable deep learning framework for improving the accuracy of internet of things intrusion detection," *Future Internet*, vol. 16, no. 3, 2024, ISSN: 1999-5903. DOI: 10.3390/fi16030080. [Online]. Available: https://www.mdpi.com/1999-5903/16/3/80.

[26] S. T. Mehedi, A. Anwar, Z. Rahman, and K. Ahmed, "Deep transfer learning based intrusion detection system for electric vehicular networks," *Sensors*, vol. 21, no. 14, 2021, ISSN: 1424-8220. DOI: 10.3390/s21144736. [Online]. Available: https://www.mdpi.com/1424-8220/21/14/4736.

[27] A. Binbusayyis, "Hybrid vgg19 and 2d-cnn for intrusion detection in the fog-cloud environment," *Expert Systems with Applications*, vol. 238, p. 121 758, 2024, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2023.121758. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0957417423022601.