

UNIVERSITÀ DEGLI STUDI DELL'AQUILA

DIPARTIMENTO DI INGEGNERIA E SCIENZE DELL'INFORMAZIONE E MATEMATICA (DISIM)

Dottorato di Ricerca in **Ingegneria e Scienze Dell'Informazione**
Curriculum **Systems Engineering, Telecommunications and HW/SW platforms**
XXXVIII° **Ciclo**

Titolo della tesi

A Blockchain-Secured Transfer Learning and Transformer-Based Framework for Next-Generation Intrusion Detection Systems

SSD: IINF-03/A

Supervisor
Prof. Dajana CASSIOLI

Co-Supervisor(s)
Dr. Andrea MAROTTA
Dr. Walter TIBERTI

Dottorando
Zeeshan ALI
Matriculation No: 291378

Anno Accademico 2023-2026

This thesis is dedicated to my beloved parents.

Acknowledgements

I would like to express my sincere gratitude to all those who have supported and guided me throughout the completion of this thesis.

First and foremost, I would like to thank my supervisor for their invaluable guidance, constructive feedback, and continuous encouragement during the course of this research. Their insights, patience, and expertise greatly contributed to the development of this work and to my growth as a researcher.

I am also deeply thankful to the faculty members and staff of the department for providing a stimulating academic environment and the resources necessary to carry out this research. Their support and assistance throughout my academic journey have been truly appreciated.

My heartfelt appreciation goes to my friends and colleagues who provided motivation, thoughtful discussions, and encouragement during challenging moments. Their companionship made this journey more meaningful and enjoyable.

Finally, I would like to express my deepest gratitude to my family for their unwavering love, support, and belief in me. Their encouragement and sacrifices have been the foundation that made this achievement possible.

Zeeshan ALI
L'Aquila, Italy
September 2026

Declaration

This thesis is the original work of the author: Zeeshan Ali, Graduate Student Member IEEE, composed with the collaboration of Dajana Cassioli, Senior Member IEEE, Walter Tiberti, Member IEEE, Andrea Marotta, Member IEEE, Carlo Centofanti, Member IEEE, and Piergiuseppe Di Marco, Member IEEE.

Abstract

The rapid evolution of Internet of Things (IoT), Industrial Internet of Things (IIoT), and edge-enabled systems has substantially expanded the cyber-attack surface of modern digital infrastructures. In such settings, conventional Intrusion Detection Systems (IDS) frequently underperform because of severe class imbalance, evolving attack behavior, heterogeneous network conditions, un-secure and malicious Artificial Intelligence (AI) models, and the computational constraints of edge and industrial platforms. This thesis addresses these limitations through a unified research that combines transformer-based learning, Deep Transfer Learning (DTL), confidential model sharing and screening, and lightweight telemetry driven detection architectures.

The first research stream develops transformer-enhanced intrusion detection model based on feature extraction through Bidirectional Encoder Representations from Transformers (BERT) and classification through Multi-Layer Perceptron (MLP) for unbalanced network traffic. The results demonstrate that contextual representation learning improves the detection of both frequent and rare attack classes while reducing dependence on manual feature engineering. This methodology is subsequently extended to IIoT environments, where robust and accurate attack detection is critical for operational continuity. Building on these findings, the thesis proposes DTL-based edge architectures and compact IDS designs for IoT devices, improving cross-domain generalization and deployment feasibility in resource-constrained settings.

The second line of work investigates secure model exchange for collaborative learning across distributed nodes. To enable Transfer Learning (TL) without exposing Intellectual Property (IP), the thesis introduces a confidential model-sharing protocol that integrates local encryption and digital signatures with decentralized storage via InterPlanetary File System (IPFS) and blockchain-governed access control on Ethereum. An extended framework further incorporates artifact screening, requester-specific key recovery, and immutable provenance recording, enabling secure, selective, and auditable reuse of model artifacts in mutually untrusted environments.

The third line of work addresses real-time Distributed Denial of Service

(DDoS) detection in virtualized infrastructures through kernel-level observability. In this framework, Extended Berkeley Packet Filter (eBPF) telemetry captures fine-grained network and application events with low overhead, while Machine Learning (ML)-based analytics deployed within a Security Information and Event Management (SIEM) platform provide near real-time attack detection. The resulting architecture offers low-latency monitoring without requiring modifications to protected applications and supports modular deployment in cloud and multi-Virtual Machines (VM) environments.

Comprehensive experimental evaluation on representative cybersecurity benchmark datasets, including Network Security Laboratory Knowledge Discovery and Data Mining 2009 (NSL-KDD 2009), Canadian Institute for Cybersecurity Intrusion Detection System 2017 (CIC-IDS 2017), Washington University in St. Louis Industrial Internet of Things (WUSTL-IIoT) 2018, WUSTL-IIoT 2021, and Canadian Institute for Cybersecurity Advanced Persistent Threats Dataset for Industrial Internet of Things (CIC-APT-IIoT) 2024, together with system-level validation of confidential model sharing and eBPF-driven DDoS detection, confirms that the proposed approaches provide high detection capability, strong adaptability across heterogeneous environments, secure and efficient model dissemination, and suitability for near real-time operation. Overall, this thesis delivers a scalable and practically grounded framework for next-generation cyber defense across IoT, IIoT, edge, and distributed collaborative learning ecosystems.

Keywords: IDS, Cybersecurity, AI, ML, Deep Learning (DL), TL, IoT, Edge Computing (EC), Blockchain, eBPF.

List of Publications

J: Journal publication; **C:** Conference publication.

Appended Published Papers:

J1 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Empowering Network Security: BERT Transformer Learning approach and MLP for Intrusion Detection in Imbalanced Network Traffic. *IEEE Access*, 2024

C1 Z. Ali, A. Marotta, W. Tiberti, O. Odoardi, D. Cassioli, and P. Di Marco. Enhancing IIoT Security: BERT-Driven Intrusion Detection with MLP in Industrial Networks. In *2025 IEEE 11th World Forum on Internet of Things (WF-IoT)*, pages 1–7, 2025

C2 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge. In *2025 IEEE 9th Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–6, 2025

J2 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios. *IEEE Internet of Things Journal*, 12(21):44882–44896, 2025

Appended Papers Under-review/Accepted:

C3 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. Confidential Model Sharing Protocol for Distributed Transfer Learning and Blockchain-Governed Access. Submitted to *10th IEEE Research and Technologies for Society and Industry (RTSI)* conference (Accepted).

J3 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. A Blockchain-Enabled Framework for Confidential Model Sharing in Distributed Transfer Learning with Artifact Screening. Submitted to *IEEE Transactions on Information Forensics and Security* (Under Review).

C4 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. Real-Time DDoS Detection by Integrated eBPF Telemetry and Machine Learning-enhanced SIEM. Submitted

to *26th International Conference on Intelligent Multimedia, Networking, and Security (IMNS)* conference (Accepted).

Papers not appended in the thesis

J4 Z. Ali, N. Aslam, A. Marotta, W. Tiberti, and D. Cassioli. A Systematic Review of Kernel-Level Security Mechanisms, Vulnerability Detection and Mitigation in Modern Operating Systems. *Sensors*, 26(8), 2026

C5 C. Aouiche, E. M. Aouiche, A. Aouiche, Z. Ali, Y. Alkhrjiah, S. Hamdi, A. Gaidi, A. Marotta, and D. Cassioli. Improving 6G Network Performance with Linear Regression and Stacking GBM. In *2025 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–6, 2025

J5 Z. Ali, A. Akram, N. Aslam, and M. S. Khurram. Supervised Learning Approach for Intrusion Detection in Unbalanced Network Traffic. *VFAST Transactions on Software Engineering*, 13(2):01–12, 2025

Contents

I Thesis Background **1**

1 Introduction **2**
1.1 Research Challenges and Thesis Contributions 3
1.2 Basic Concepts 5
1.2.1 IoT and IIoT 5
1.2.2 Cybersecurity and Attack Surface 5
1.2.3 IDS 5
1.2.4 ML and DL 7
1.2.5 Transformers 9
1.2.6 TL 10
1.2.7 Class Imbalance 10
1.2.8 EC 11

2 State of the art **13**
2.1 Intrusion Detection in IoT/IIoT 13
2.1.1 Traditional Feature Selection and Classification 13
2.1.2 Generalization and TL 14
2.2 Secure and Collaborative Model Management in Distributed TL . . . 16
2.3 Telemetry-Driven Threat Detection in Virtualized Infrastructures . . 18

3 Thesis Contribution **20**
3.1 Publications appended to the thesis 21
3.2 Publications not appended to the thesis 26

II Conclusion **29**

Bibliography **40**

III Appended Published Papers **41**

J1 Empowering Network Security BERT Transformer Learning Approach and MLP for Intrusion Detection in Imbalanced Network Traffic **42**

C1 Enhancing IIoT Security BERT-Driven Intrusion Detection with MLP in Industrial Networks **59**

C2 A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge **67**

J2 Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios **74**

IV Appended Papers Under-review/Accepted **90**

C3 Confidential Model Sharing Protocol for Distributed Transfer Learning and Blockchain-Governed Access **91**

J3 A Blockchain-Enabled Framework for Confidential Model Sharing in Distributed Transfer Learning with Artifact Screening **97**

C4 Real-Time DDoS Detection by Integrated eBPF Telemetry and ML-enhanced SIEM **111**

List of Figures

1.1 Architecture of the Host-based Intrusion Detection System (HIDS) . . 6
1.2 Architecture of the Network-based Intrusion Detection System (NIDS) 6
1.3 Architecture of the Signature-based IDS 7
1.4 Architecture of the Anomaly-based IDS 8
1.5 Taxonomy of ML and DL Algorithms. 8
1.6 A comparison between the traditional ML workflow and the TL paradigm. 10
1.7 Class distributions between a Balanced Dataset and an Imbalanced Dataset. 11
1.8 Architecture of EC in an IoT ecosystem 11

List of Tables

1.1 Mapping of thesis papers to the core research problems addressed in this dissertation. 5

List of Abbreviations

AES-256-GCM Advanced Encryption Standard with a 256-bit key in Galois/Counter Mode

AI Artificial Intelligence

ANN Artificial Neural Network

APT Advanced Persistent Threats

ASLR Address Space Layout Randomization

BERT Bidirectional Encoder Representations from Transformers

CIC-APT-IIoT Canadian Institute for Cybersecurity Advanced Persistent Threats Dataset for Industrial Internet of Things

CIC-IDS 2017 Canadian Institute for Cybersecurity Intrusion Detection System 2017

CNN Convolutional Neural Network

CPS Cyber Physical Systems

DDoS Distributed Denial of Service

DEP Data Execution Prevention

DL Deep Learning

DNN Deep Neural Network

DT Decision Tree

DTL Deep Transfer Learning

eBPF Extended Berkeley Packet Filter

EC Edge Computing

ECDSA Elliptic Curve Digital Signature Algorithm

FL Federated Learning

FN False Negative

FP False Positive

FTL Federated Transfer Learning

GA Genetic Algorithm

GBM Gradient Boosting Machine

GCM Galois Counter Mode

H5 Hierarchical Data Format 5

HIDS Host-based Intrusion Detection System

IDS Intrusion Detection Systems

IID Independent and Identically Distributed

IIoT Industrial Internet of Things

IoT Internet of Things

IP Intellectual Property

IPFS InterPlanetary File System

IT Information Technology

LR Logistic Regression

LSTM Long Short-Term Memory

MAC Mandatory Access Control

MCP Model Context Protocol

MEC Mobile Edge Computing

MGA Modified Genetic Algorithm

ML Machine Learning

MLP Multi-Layer Perceptron

NIDS Network-based Intrusion Detection System

NLP Natural Language Processing

NSL-KDD 2009 Network Security Laboratory Knowledge Discovery and Data Mining 2009

ONNX Open Neural Network Exchange

OS Operating System

OT Operational Technology

P2P Peer-to-Peer

PIO Pigeon Inspired Optimizer

PRE Proxy Re-Encryption

PRISMA Preferred Reporting Items for Systematic Reviews and Meta-Analyses

RAG Retrieval-Augmented Generation

ResNet Residual Network

RF Random Forest

RNN Recurrent Neural Network

SHAP SHapley Additive exPlanations

SIEM Security Information and Event Management

SL Supervised Learning

SMOTE Synthetic Minority Over-sampling Technique

SVM Support Vector Machine

TL Transfer Learning

UNSW-NB 2015 University of New South Wales Network-Based 2015

VGG Visual Geometry Group

VM Virtual Machines

WUSTL-IIoT Washington University in St. Louis Industrial Internet of Things

XDP eXpress Data Path

Part I

Thesis Background

Chapter 1

Introduction

Society's rapid digitalization has fueled interconnected systems where sensing, communication, and automation converge. IoT and IIoT technologies now support applications ranging from smart homes and healthcare to industrial automation and critical infrastructure. At the same time, this pervasive connectivity has significantly expanded the cyber-attack surface, making security a first-order system requirement. In this context, IDS plays a central role by continuously monitoring network and system activity to identify suspicious or malicious behavior.

Recent advances in ML and DL have enabled more adaptive and robust IDS architectures [32]. These systems leverage supervised, unsupervised, and deep representation learning paradigms to analyze network traffic and system logs, improving the detection of both known and unknown threats. Nevertheless, key limitations remain, including extensive feature-engineering requirements, sensitivity to class imbalance and weak generalization across diverse domains and operating contexts [4].

In response, the research community has explored the use of advanced DL architectures, such as transformers and TL, to enhance IDS performance. Transformer-based models, particularly BERT, have demonstrated strong capabilities in learning contextual representations [38]. By converting network traffic features into textual representations, these models can capture complex dependencies and subtle attack patterns that traditional methods may overlook. TL further enables the reuse of knowledge acquired from one dataset or environment in another, improving adaptability and reducing the need for retraining from scratch. For deployment on resource-constrained edge and IoT devices, however, lightweight model architectures and model compression techniques remain essential for achieving efficient inference [44].

Despite these advances, several fundamental research gaps remain open. First, severe class imbalance in cybersecurity datasets biases models toward majority classes and weakens sensitivity to minority and emerging attacks. Second, models trained on a specific dataset often exhibit limited transferability across domains because of shifts in traffic composition, protocols, and attack behavior. Third, high-capacity deep models frequently exceed the memory, latency, and computational budgets of

edge and industrial devices. Fourth, collaborative learning introduces a distinct security problem: even when training is effective, model artifacts must be shared with guarantees of confidentiality, integrity, provenance, and controlled access in trustless environments. Finally, operational defense in cloud and virtualized systems requires low-overhead, near real-time telemetry and analytics pipelines that can scale without intrusive instrumentation.

The novelty of this thesis lies in addressing these challenges as successive stages of a unified intrusion detection lifecycle rather than as isolated research problems. Instead of focusing solely on improving detection accuracy, the proposed research considers the complete pipeline required for practical and trustworthy IDS deployment. The work begins by improving detection performance under class imbalance and domain shift using transformer-based and TL techniques. It then investigates resource-aware DTL strategies that enable efficient deployment on edge and IoT devices with limited computational resources. Building on these models, the thesis addresses the secure sharing and governance of trained model artifacts through mechanisms that ensure confidentiality, integrity, provenance, and controlled access in decentralized environments. Finally, the work extends to operational cloud and virtualized infrastructures by developing low-overhead, telemetry-driven intrusion detection methods capable of near real-time monitoring. Collectively, these contributions establish an end-to-end framework that bridges accurate learning, efficient deployment, secure collaboration, and scalable operational monitoring for modern IDS applications.

Although published and submitted papers are appended in the later parts of this dissertation, this work is not presented as a compilation thesis. The dissertation is a unified research thesis in which the appended publications serve as evidence of, and detailed accounts of, individual studies within a common research programme. The studies are connected by a single overarching objective, namely the development of accurate, adaptable, secure, and deployable intrusion-detection solutions.

1.1 Research Challenges and Thesis Contributions

This thesis addresses the identified challenges through a coherent sequence of studies that progressively expand from IDS modeling to secure collaborative learning and operational detection pipelines. Each paper contributes to one stage of that lifecycle while preserving a common methodological thread based on contextual representation learning, transferability, and deployability:

- **Class imbalance and contextual feature learning in network IDS (J1):** We propose an innovative approach for imbalanced intrusion detection that exploits BERT, Synthetic Minority Over-sampling Technique (SMOTE), and MLP, and demonstrate that transformer-based contextual embeddings, combined with data rebalancing, substantially improve minority-class detection while

maintaining high overall performance.

- **Industrial-domain robustness for IIoT security (C1):** We extend the BERT-SMOTE-MLP paradigm to IIoT/Operational Technology (OT) traffic, validating multiclass and binary attack detection on industrial benchmarks datasets.
- **Lightweight edge deployment through DTL (C2):** We introduce a lightweight *supermodel* strategy that transfers knowledge across heterogeneous edge datasets, reducing training burden on resource-constrained IoT devices while preserving high detection accuracy.
- **Scalable deep TL for decentralized edge environments (J2):** We formalize and evaluate a DTL-enabled *supermodel* architecture, including BERT-Base and BERT-Large analysis, to improve cross-domain generalization and practical readiness for dynamic edge scenarios.
- **Confidential and governed model sharing for collaborative TL (C3):** We propose a secure model publication and access protocol that combines local cryptographic protection, IPFS-based artifact distribution, and blockchain-governed access management.
- **Screened and provenance-aware secure model lifecycle (J3):** We extend the model-sharing pipeline with pre-publication artifact screening, delegated key access, and lineage-aware synchronization, supporting secure reuse and update of models in distributed trustless settings.
- **Real-time DDoS detection via kernel telemetry and SIEM analytics (C4):** We design a decoupled eBPF-to-SIEM pipeline for cloud/multi-VM environments, enabling low-overhead telemetry collection and near real-time ML-based detection of volumetric attacks.

Collectively, these contributions establish an end-to-end perspective on modern cyber defense: from accurate intrusion detection under data imbalance and domain shift, to secure collaborative model dissemination, and finally to deployable real-time monitoring pipelines for operational environments.

The remainder of this chapter is organized as follows. Section 1.2 introduces the background concepts that are necessary to frame the thesis contributions, namely IDS paradigms, transformer-based learning, TL, class imbalance, and edge constraints. Chapter 2 reviews the state of the art and positions the thesis with respect to existing work in intrusion detection, secure collaborative model management, and telemetry-driven operational defense.

Research Problem / Contribution Axis	J1	C1	C2	J2	C3	J3	C4
Class imbalance-aware intrusion detection	✓	✓	✓	✓			
Contextual transformer feature learning (BERT-based)	✓	✓	✓	✓			
IIoT/OT-specific threat detection validation		✓					
Distributed TL and lightweight edge *supermodel* deployment			✓	✓			
Confidential blockchain-governed model sharing					✓	✓	
Artifact screening and lineage-aware model lifecycle						✓	
Kernel telemetry + SIEM integrated real-time DDoS detection							✓

Table 1.1: Mapping of thesis papers to the core research problems addressed in this dissertation.

1.2 Basic Concepts

1.2.1 IoT and IIoT

IoT denotes interconnected devices that exchange data and support automated decision making across consumer and cyber-physical environments. In this thesis, the term matters because the underlying data are heterogeneous, distributed, and often resource constrained, which directly affects how detection models are trained and deployed.

IIoT extends the same connectivity paradigm to industrial and critical-infrastructure settings by combining sensing, control, and analytics across OT and Information Technology (IT). For security research, the key distinction is not the definition itself but the stronger operational constraints: stricter reliability requirements, more structured traffic, and higher sensitivity to latency, availability, and domain shift.

1.2.2 Cybersecurity and Attack Surface

The increasing interconnectivity of digital systems, cloud platforms, and IoT/IIoT devices enlarges the attack surface available to adversaries. In this thesis, that expansion is important because it motivates both data-driven intrusion detection and lightweight operational monitoring.

Cybersecurity, in this context, is the set of mechanisms used to preserve confidentiality, integrity, availability, and traceability under active attack. The thesis focuses on detection and model governance aspects of cybersecurity rather than on generic prevention mechanisms, which is why monitoring, provenance, access control, and low-overhead analytics recur throughout the chapter.

1.2.3 IDS

IDS are systems that continuously monitor network traffic, host activity, or telemetry streams to identify malicious behavior and policy violations. For this thesis, they are the primary analytical layer through which imbalance handling, TL, and telemetry-driven inference are evaluated.

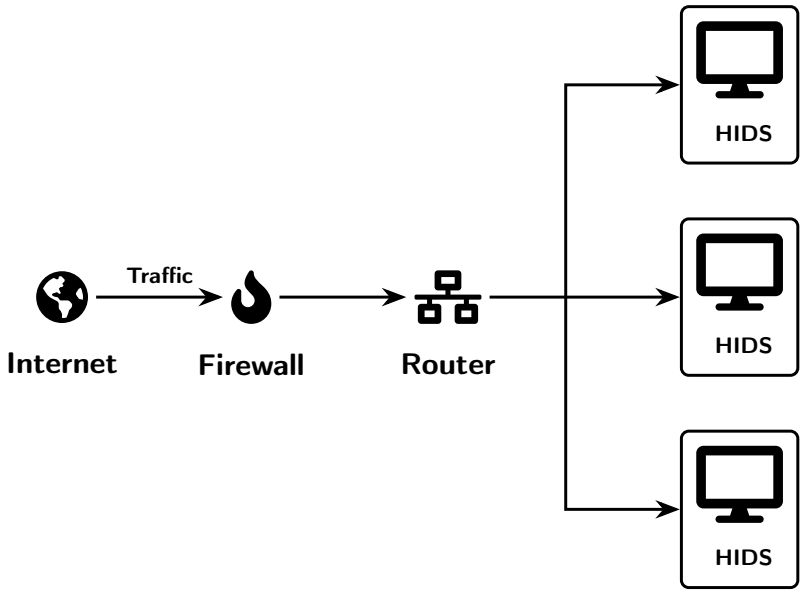


Figure 1.1: Architecture of the HIDS

Depending on the monitoring scope and deployment location, IDS can be broadly classified into the following categories:

- *HIDS:* A HIDS monitors activity on individual devices as shown in the Figure 1.1, including logs, file integrity, process behavior, and Operating System (OS) events. It is relevant where endpoint-level visibility is needed to complement network-based monitoring [55].
- *NIDS:* A NIDS analyzes traffic flows to identify suspicious patterns and abnormal behavior as shown in Figure 1.2. This thesis primarily builds on the NIDS setting because the appended studies operate on network-flow data, industrial traffic, and telemetry-derived events [47].

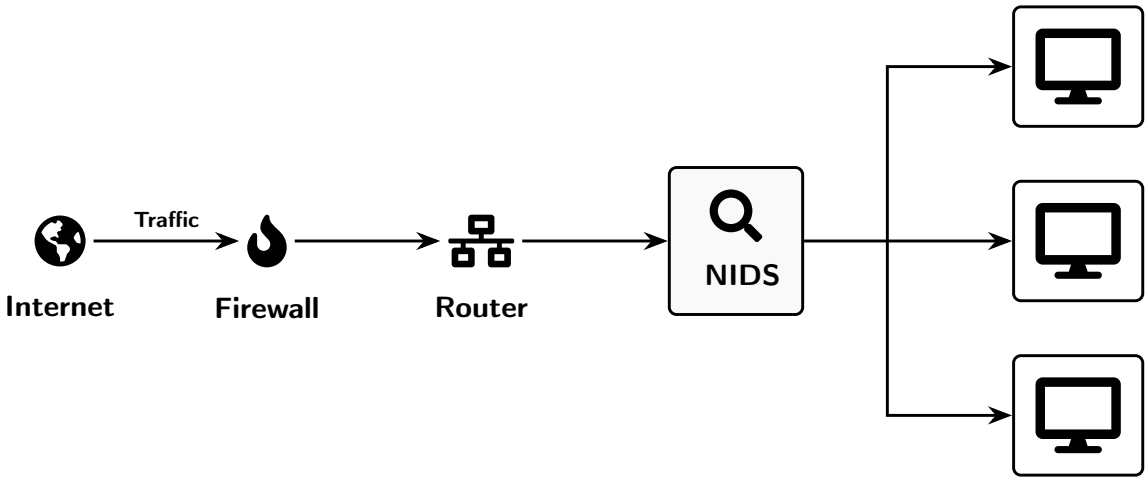


Figure 1.2: Architecture of the NIDS

IDS Approaches

IDS employ different detection methodologies to identify malicious activities and security threats within networks or host systems. The two primary detection approaches are signature-based detection and anomaly-based detection, each offering distinct advantages and limitations.

- *Signature-based IDS:* Signature-based IDS identify attacks by comparing observed events, traffic patterns, or system activities against a database of known attack signatures and predefined threat patterns as shown in the Figure 1.3. This approach is highly effective for detecting previously identified attacks with high accuracy and low False Positive (FP) rates. However, its effectiveness depends on continuously updated signature databases, making it less capable of detecting zero-day attacks, unknown threats, or rapidly evolving attack techniques [49].

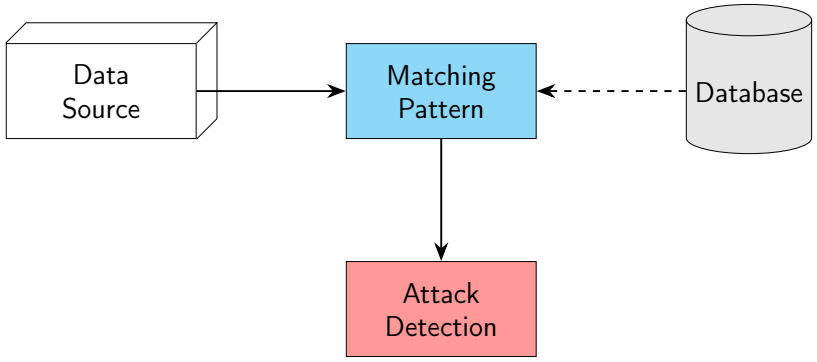


Figure 1.3: Architecture of the Signature-based IDS

- *Anomaly-based IDS:* Anomaly-based IDS detect threats by identifying deviations from established normal behavior profiles of network traffic or system activities as shown in the Figure 1.4. They are more aligned with the thesis because they can detect unseen attacks, by learning patterns associated with legitimate behavior, these systems can recognize previously unseen or novel attacks that do not match known signatures. Although anomaly-based approaches provide greater adaptability and broader threat coverage, they often produce higher FP rates due to the difficulty of accurately modeling normal behavior in dynamic environments [30].

1.2.4 ML and DL

ML is a branch of AI that enables systems to learn patterns, relationships, and knowledge from data in order to make predictions, classifications, or decisions without being explicitly programmed for every task. Traditional ML algorithms, such as Decision Tree (DT), Support Vector Machine (SVM), and clustering methods, often

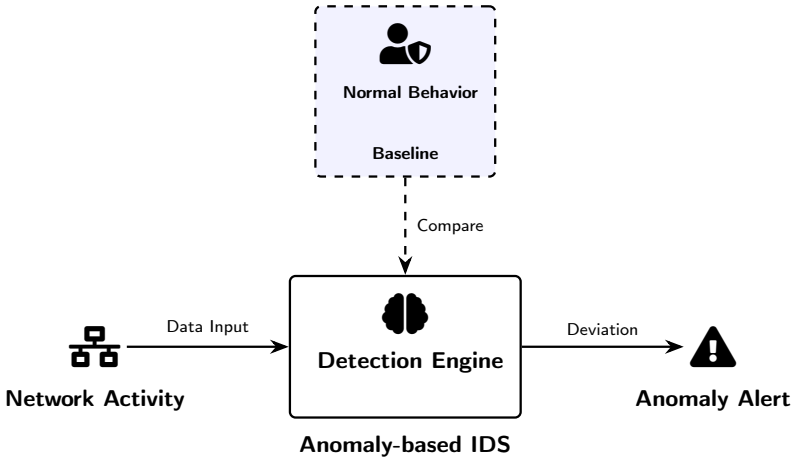


Figure 1.4: Architecture of the Anomaly-based IDS

rely on manually engineered features and statistical learning techniques to model data patterns.

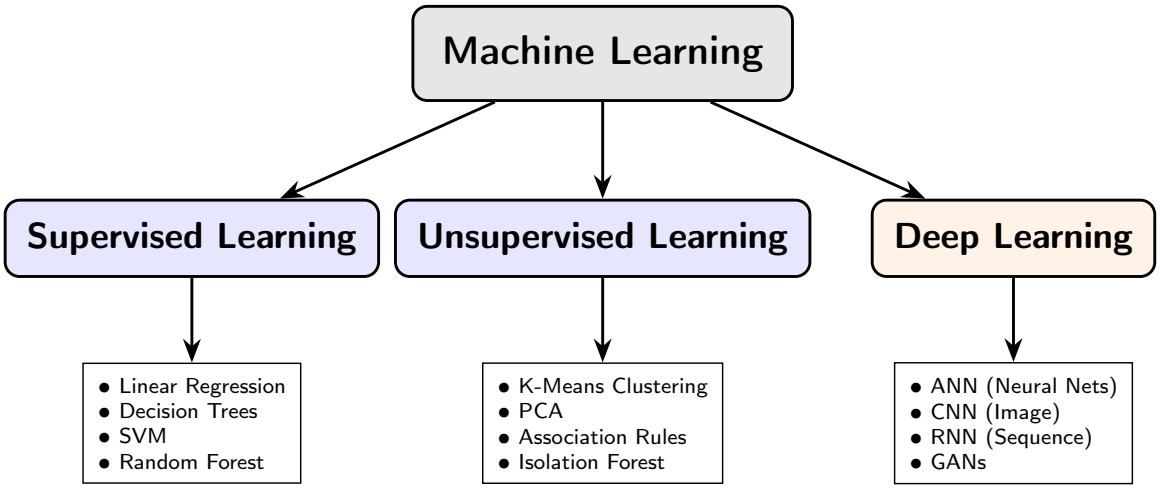


Figure 1.5: Taxonomy of ML and DL Algorithms.

DL is a specialized subset of ML that employs Artificial Neural Network (ANN) with multiple hidden layers to automatically learn hierarchical feature representations directly from large volumes of data. Figure 1.5 illustrates the categorization of traditional supervised and unsupervised ML techniques alongside DL architectures. Unlike conventional ML methods, DL models can automatically extract complex and high-level features, enabling them to capture intricate patterns and nonlinear relationships within the data. Architectures such as Convolutional Neural Network (CNN), Recurrent Neural Network (RNN), and transformers have demonstrated significant success across domains including computer vision, Natural Language Processing (NLP), and cybersecurity.

In the context of IDS, ML and DL techniques are widely used to analyze network traffic and system activities, detect anomalous behavior, identify malicious patterns,

and classify different types of cyberattacks. These approaches improve the ability of IDS to adapt to dynamic environments, recognize evolving threats, and enhance overall detection accuracy and robustness.

1.2.5 Transformers

Transformers are a class of DL models designed to process sequential data by using self-attention, a mechanism that enables the model to determine how strongly each element in a sequence relates to every other element when generating its representation. Rather than processing input elements sequentially, self-attention computes the relevance of each element with respect to all other elements in the sequence, allowing the model to focus on the most informative contextual information regardless of its position. This enables transformers to capture both short-range and long-range dependencies more effectively than traditional recurrent architectures. Furthermore, transformers process all input elements in parallel, resulting in faster training and better scalability for large datasets [64].

The self-attention mechanism operates by projecting each input element into three vector representations known as the query (Q), key (K), and value (V). Attention scores are calculated by measuring the similarity between the query of one element and the keys of all other elements in the sequence. These scores are then normalized using the Softmax function and used as weights to compute a weighted combination of the value vectors. Consequently, the representation of each element is enriched with contextual information from the most relevant parts of the input sequence [34].

A transformer architecture is typically composed of stacked encoder and/or decoder layers, each containing multi-head self-attention modules and position-wise feed-forward neural networks. Multi-head attention extends the self-attention mechanism by performing multiple attention operations in parallel, with each attention head learning different relationships and contextual features from distinct representation subspaces. The outputs of all attention heads are then concatenated and projected to form a richer representation of the input. This enables transformers to simultaneously capture diverse semantic and structural patterns within sequential data [27].

Due to their ability to automatically learn contextual representations and complex dependencies, transformers have become widely adopted in domains such as NLP, computer vision, speech processing, and cybersecurity. In the context of IDS, transformers can analyze network traffic sequences and learn behavioral patterns associated with both normal and malicious activities, thereby improving the detection of sophisticated and previously unseen cyber attacks [27].

1.2.6 TL

TL is a ML approach in which knowledge learned from a previously trained model on a source task or domain is reused and adapted to a different but related target task. Instead of training a model entirely from scratch, TL leverages pre-learned representations, patterns, and features acquired from large datasets, significantly reducing training time, computational cost, and the amount of labeled data required for the new task [61]. Figure 1.6 shows the difference between the traditional approach, which develops isolated models for each task, and the TL, that instead leverages knowledge (weights and features) from a pre-trained model (Task 1) to improve learning efficiency and performance on a related target task (Task 2). TL is most widely and effectively applied in DL, where neural networks learn hierarchical feature representations that can be transferred across tasks [67].

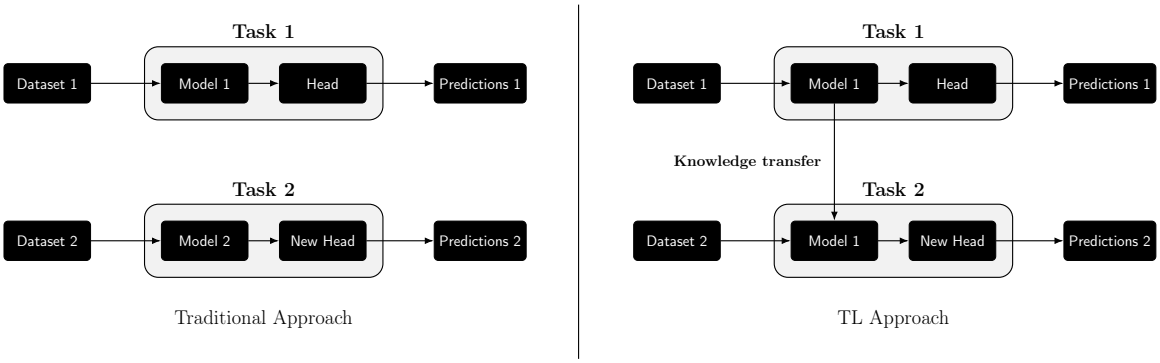


Figure 1.6: A comparison between the traditional ML workflow and the TL paradigm.

This approach is particularly effective when the target domain has limited training data or when collecting and labeling data is expensive and time-consuming. By transferring generalized knowledge from one domain to another, DL models can achieve faster convergence and improved performance while maintaining adaptability to new environments [50].

1.2.7 Class Imbalance

Class imbalance is a common challenge in IDS datasets, where instances of normal network traffic significantly outnumber malicious or attack-related samples. This uneven distribution can bias ML and DL models toward the majority class [31].

Figure 1.7 shows the balanced set maintains an equal 50%/50% split between classes, whereas the imbalanced set exhibits a severe skewness (95% to 5%), highlighting the significant disparity between the majority and minority classes. This skewness causes the models to achieve high overall accuracy while performing poorly in detecting minority or rare attack types. As a result, models may fail to correctly identify infrequent but critical intrusions, leading to increased False Negative (FN) and reduced detection reliability [39]. Addressing class imbalance is therefore essential

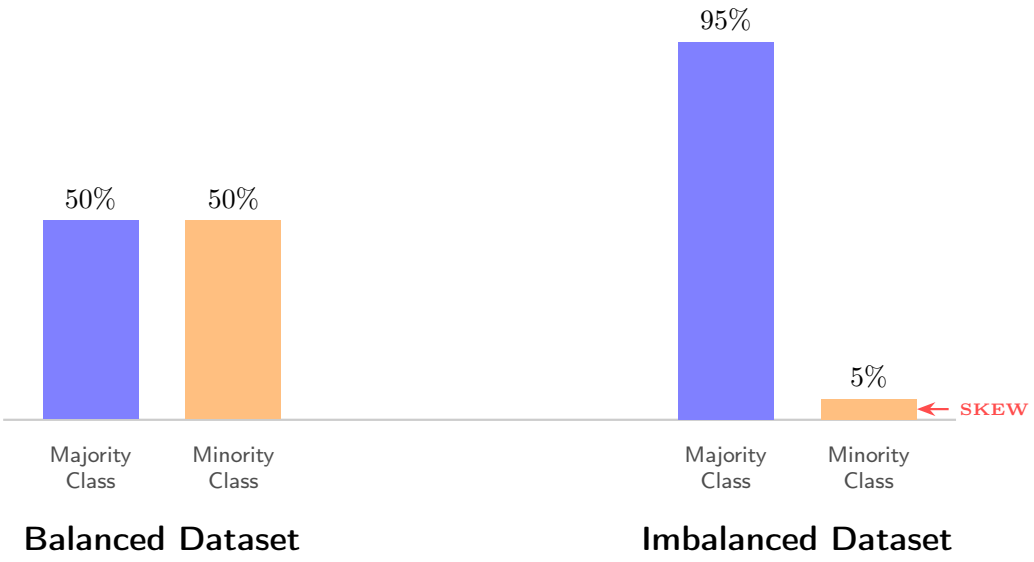


Figure 1.7: Class distributions between a Balanced Dataset and an Imbalanced Dataset.

for improving the robustness and effectiveness of IDS, particularly in the detection of sophisticated or emerging cyberattacks.

1.2.8 EC

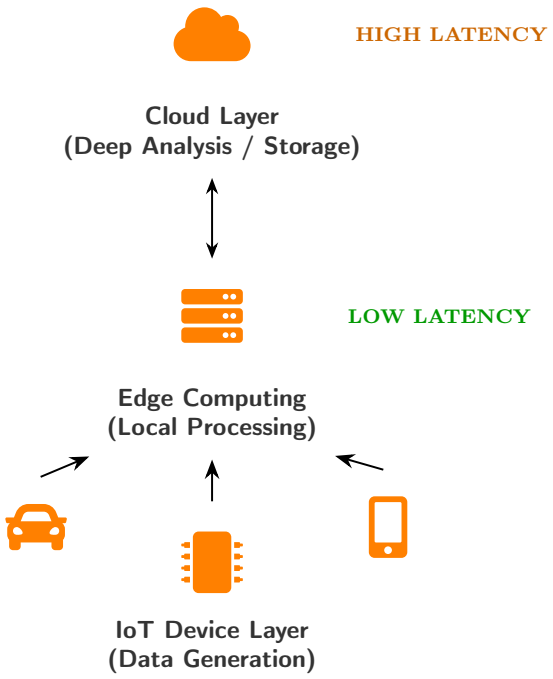


Figure 1.8: Architecture of EC in an IoT ecosystem

EC is a distributed computing paradigm in which data processing, storage, and analysis are performed closer to the data source, such as IoT devices, sensors, gateways, or local edge servers, rather than relying exclusively on centralized cloud in-

frastructures [56]. By bringing computational resources nearer to end devices, EC enables faster data processing and real-time decision making while reducing the need to transmit large volumes of data to remote data centers [23].

Figure 1.8 shows a three-tier architectural model of EC in an IoT ecosystem. The IoT Device Layer (bottom) generates raw data from diverse endpoints like vehicles, sensors, and mobile devices. This data is transmitted to the EC Layer (middle), where local servers perform immediate processing to ensure low latency and real-time responsiveness. Finally, the Cloud Layer (top) receives filtered data for long-term storage and high-latency deep analysis, optimizing overall network bandwidth and compute efficiency [70].

This approach significantly decreases communication latency, bandwidth consumption, and network congestion, making it particularly suitable for time-sensitive and resource-constrained applications. EC also enhances reliability, scalability, and privacy by allowing critical operations and sensitive data processing to occur locally, even in environments with limited or unstable network connectivity. In domains such as smart cities, industrial automation, healthcare, autonomous systems, and cybersecurity, EC supports efficient real-time monitoring, rapid threat detection, and low-latency system responses [63].

Chapter 2

State of the art

Cyber defense for IoT/IIoT and edge ecosystems has progressed along three tightly coupled directions: *(i)* data-driven intrusion detection, *(ii)* secure collaborative model management, and *(iii)* telemetry-driven operational detection. While the literature has moved from handcrafted rules and traditional ML pipelines to DL representation learning, TL for cross-domain robustness, and security-by-design mechanisms, there remain critical gaps in effectively adopting these paradigms for resource-constrained edge deployments. This chapter goes deeper into the analysis of existing work, highlighting the merits of available solutions and their limitations, thus introducing the specific needs that motivate the contributions of this thesis.

2.1 Intrusion Detection in IoT/IIoT

The rapid advancements in IoT and IIoT ecosystems have fundamentally transformed both consumer and industrial landscapes. These environments are characterized by massive device scale, strict resource constraints, heterogeneous communication protocols, and a strong coupling between cyber operations and physical processes. Consequently, they introduce unique and practically expansive attack surfaces.

2.1.1 Traditional Feature Selection and Classification

Classical IDSs transitioned from signature-based matching to anomaly-based systems [65]. Early data-driven efforts primarily relied on traditional feature selection techniques such as SVM integrated with Genetic Algorithm (GA) [15] or Logistic Regression (LR) heuristics [35] aimed at reducing dimensionality and computational cost. Farhan et al [62] proposed an approach that combines CNN and Long Short-Term Memory (LSTM) networks. This hybrid model aims to capture intricate features within network traffic data while ensuring a balanced representation of the information. In anomaly detection, Zhu et al. [73], proposed LSTM-based models that have been applied to the detection of time series anomalies. However, LSTM

requires a lot of labelled data and has limits in capturing long-distance dependencies. Aslahi-Shahri et al. [15] employed a novel hybrid methodology that integrated a SVM with a GA. Their primary objective was to enhance the efficiency of feature selection for the KDD CUP 99 [59] dataset. Specifically, this approach aimed to reduce the dimensionality of the dataset, which originally encompassed 41 features, ultimately distilling it into a more manageable set of 10 essential features. In another study by Alazzam et al. [2], they endeavoured to refine the feature selection methodology for NIDS by harnessing the power of a Pigeon Inspired Optimizer (PIO). Their technique efficiently reduced feature sets in KDD CUP 99 and NSL-KDD datasets, condensing them from 41 to 7 and 5 features, respectively.

While the state-of-the-art has highlighted various strategies for feature selection using both conventional and DL techniques, the effectiveness of traditional feature selection strongly depends on the quality and semantic relevance of the selected features. These conventional methods often rely on heuristic principles that fail to capture the complex contextual dependencies underlying network traffic.

These traditional feature selection and classification approaches have contributed significantly to the evolution of IDS by reducing dimensionality and improving computational efficiency. However, their dependence on handcrafted features and limited capability to model long-range contextual dependencies restrict their effectiveness against sophisticated and evolving cyber threats. These limitations highlight the need for more advanced intelligent models capable of automatically learning discriminative feature representations and capturing complex traffic behaviours for robust and adaptive intrusion detection.

This thesis addresses the identified research gap by leveraging BERT-based tokenization to capture the contextual and semantic characteristics of network traffic, while integrating a lightweight MLP classifier. In addition, the incorporation of SMOTE mitigates the issue of class imbalance by enhancing the representation of minority attack classes, thereby improving detection sensitivity and classification [9]. Furthermore, the proposed methodology is also validated in an IIoT environment, demonstrating its adaptability and effectiveness in handling the heterogeneous and resource-constrained nature of industrial network traffic [8].

2.1.2 Generalization and TL

To improve robustness under domain shift and reduce training costs, TL enables the reuse of learned representations across tasks and environments [51]. In scenarios where labelled data is scarce or expensive to obtain, TL provides an effective mechanism for generalizing knowledge acquired from related domains [69]. In IoT and EC environments, transfer mechanisms are particularly attractive because computationally intensive training can be performed on capable nodes, while lightweight models are deployed on resource-constrained edge devices [40].

In parallel, decentralized learning paradigms such as Federated Learning (FL) aim to exploit distributed data without centralizing raw traffic, thereby improving privacy and governance, although at the expense of orchestration complexity and robustness challenges [33, 41]. These developments have motivated the emergence of generalized intrusion detection frameworks capable of adapting across heterogeneous domains and deployment environments.

Several studies have investigated the application of TL for intrusion detection. B. Zong et al. [75] proposed a framework capable of transferring knowledge from a labelled source domain to a target domain with different statistical properties. Similarly, K. Alrawashdeh et al. [12] employed TL for cyber threat hunting by leveraging labelled data from related domains to improve detection performance. Nour Alhussien et al. [3] further examined the security implications of backdoor attacks on multi-source TL models for intrusion detection.

Recent works have also combined TL with decentralized and edge-oriented learning paradigms. Cheng et al. [25] introduced an efficient Federated Transfer Learning (FTL) framework for Mobile Edge Computing (MEC)-based intrusion detection by integrating FL with TL to preserve privacy and enhance training efficiency. Likewise, Jiazhen et al. [72] proposed a privacy-preserving anomaly-based IDS for IIoT networks using FL and instance-based TL to address Non-Independent and Identically Distributed (IID) data distributions. Their weighted rank aggregation mechanism achieved accuracies of 95.97% and 73.70% using AdaBoost and Random Forest (RF), respectively.

Other studies have focused on improving feature extraction and computational efficiency in heterogeneous IoT environments. Yutao et al. [71] presented the Visual Geometry Group (VGG)-RED approach, which applies TL to migrate feature weights across IoT intrusion detection datasets, achieving a 7–13% reduction in computational time. Mehedi et al. [42] proposed a DTL-based IDS integrating attribute selection and a Residual Network (ResNet) architecture, demonstrating improved accuracy and reliability in heterogeneous IoT environments.

Edge-enabled intrusion detection has also received significant attention. Saheed et al. [53] proposed the IoT-Defender framework, combining a Modified Genetic Algorithm (MGA) for feature selection with a fine-tuned LSTM deployed in an EC environment. Evaluated on BoT-IoT [37], UNSW-NB15 [46], and N-BaIoT [43] datasets using a Raspberry Pi platform, the framework achieved 99.41% accuracy and a 99.78% detection rate. Similarly, Masum et al. [45] proposed a NIDS classification framework based on TL using pre-trained VGG-16 weights extracted from ImageNet [26]. Their framework compared the performance of VGG-16, VGG-19 [17], MobileNet [57], ResNet-50, and Inception-V3, achieving 81.77% accuracy for binary classification and 78.39% for multi-class classification on the NSL-KDD 2009 dataset.

Saheed et al. [54] introduced an explainable and privacy-preserving Deep Neu-

ral Network (DNN) framework for anomaly detection in Cyber Physical Systems (CPS)-enabled IoT networks by integrating SHapley Additive exPlanations (SHAP) for interpretability. Evaluated on Edge-IIoTset [29] and X-IIoT intrusion detection [1] datasets, the framework achieved near-perfect accuracy while significantly reducing FP rates.

Despite these significant advancements, there is still margin for improvements within current TL-based intrusion detection frameworks. Existing approaches predominantly rely on architectures such as CNN, LSTM, VGG, and ResNet, which often struggle to generalize effectively across highly heterogeneous and dynamically evolving IoT environments. This limitation arises because these architectures primarily learn local spatial features or sequential dependencies, making it difficult to capture the global contextual relationships and long-range interactions that characterize complex network traffic. Consequently, their learned representations are more susceptible to performance degradation under domain shifts and out-of-distribution attack scenarios, where traffic distributions and attack characteristics differ from those encountered during training [27, 64]. A major challenge lies in the limited ability of these models to capture deeper semantic and contextual relationships within network traffic, particularly under domain shifts and out-of-distribution attack scenarios.

Building on this foundation, our appended works [10, 11] contribute toward addressing these limitations through the formalization and evaluation of a DTL-enabled transformer-based *supermodel* optimized for scalable deployment in EC-enabled IoT environments. The originality of the proposed TL-enabled *supermodel* resides in the integration of BERT-based semantic feature extraction with a lightweight MLP classifier and an edge-oriented deployment strategy. This design enables efficient intrusion detection while maintaining compatibility with resource-constrained edge devices. Furthermore, the framework enhances scalability, adaptability, and resilience against evolving and out-of-distribution attacks across heterogeneous IoT domains.

2.2 Secure and Collaborative Model Management in Distributed TL

Secure and distributed ML has been widely studied through FL, where participants collaboratively train models by exchanging parameter updates rather than raw data. Foundational work introduced communication-efficient decentralized optimization for large-scale ML [41], while later studies analyzed practical challenges related to privacy, system heterogeneity, and robustness [33]. Secure aggregation techniques further reduced information leakage by ensuring that only aggregated model updates are visible to the coordinator [22]. These approaches enable privacy-preserving collaborative training, but primarily focus on iterative parameter exchange and do not address persistent, searchable, and policy-governed sharing of complete model artifacts for

downstream TL [52].

To improve trust and coordination in distributed learning environments, several works have explored blockchain-assisted ML systems. Early frameworks such as BlockFL demonstrated how blockchain can coordinate FL participants and provide incentive alignment in decentralized training environments [36]. Subsequent studies examined decentralized governance, smart-contract–mediated coordination, and the scalability and security implications of blockchain-enabled learning pipelines [20]. While these approaches enhance auditability and participant accountability, they remain largely focused on coordinating training workflows rather than managing the secure publication, discovery, and reuse of trained model artifacts.

Parallel research has investigated decentralized storage and cryptographic mechanisms for secure data sharing. Blockchain platforms such as *Ethereum* provide programmable smart contracts that enable tamper-evident policy enforcement and transparent access control [68]. Complementarily, decentralized storage systems such as IPFS enable content-addressed distribution and resilient storage of large artifacts across Peer-to-Peer (P2P) networks [18, 74]. Recent blockchain–IPFS integration studies have demonstrated practical architectures for secure storage and verifiable data access in decentralized environments [76]. At the cryptographic level, authenticated encryption schemes such as AES-Galois Counter Mode (GCM) and elliptic-curve cryptography are widely used for efficient confidentiality, integrity protection, and key ownership verification [28, 58]. Proxy Re-Encryption (PRE) techniques further enable delegated access control by allowing a proxy to transform encrypted data for authorized recipients without revealing the underlying secret keys [16].

Building on these valuable approaches, our appended works C3 and J3 propose a secure framework for distributed model sharing that integrates model screening, node-local cryptography, decentralized storage, and blockchain-based metadata governance. Unlike existing FL and blockchain-assisted learning frameworks that primarily focus on iterative parameter exchange and decentralized coordination, the proposed approach addresses the persistent challenge of securely publishing, discovering, validating, and reusing complete model artifacts for downstream TL.

The framework overcomes several critical limitations in current distributed learning systems, including the absence of verifiable model provenance, weak access-control mechanisms, limited interoperability across heterogeneous nodes, and the lack of secure lifecycle management for transferable models. Furthermore, the integration of cryptographic mechanisms enables secure ownership verification, confidentiality preservation, and controlled delegated access to shared models without exposing sensitive information.

Through this design, the proposed framework establishes a scalable and privacy-preserving infrastructure for collaborative TL in distributed IoT and edge environments, enabling trusted dissemination and efficient reuse of pretrained intrusion de-

tection models across heterogeneous domains.

2.3 Telemetry-Driven Threat Detection in Virtualized Infrastructures

DDoS detection and mitigation have evolved significantly in recent years, spanning traditional statistical techniques, ML–based analytics, and more recent kernel-level observability approaches. Early work in this area focused on hybrid and anomaly-based methods for identifying volumetric attacks by combining volume-based features with entropy and feature statistics, demonstrating the advantage over simple threshold-based detection mechanisms in controlled network environments [21].

A substantial body of research has explored ML for DDoS detection, leveraging supervised and unsupervised models to identify attack patterns from network traffic characteristics. For example, comprehensive surveys of ML-based DDoS detection techniques underscore the effectiveness of classifiers such as RF, SVM, and DL models in distinguishing between normal and malicious traffic using datasets like CIC-DDoS 2019 [5] [24]. In IoT and cloud contexts, researchers have applied LSTM and ensemble models to capture temporal traffic features, achieving high accuracy under diverse threat scenarios [19]. However, these approaches typically rely on pre-collected flow statistics or packet captures, which may introduce latency and limit visibility into low-level system behavior.

In parallel, eBPF has emerged as a promising technology for lightweight, in-kernel observability and security monitoring. eBPF enables programmable telemetry collection within the operating system kernel without modifying source code, significantly reducing overhead compared to user-space capture mechanisms. Several studies have demonstrated the use of eBPF/eXpress Data Path (XDP) for DDoS detection and mitigation, especially in resource-constrained environments such as IoT edge devices, where rate-based blocking at the earliest point in the network stack can effectively drop malicious traffic while preserving legitimate flows [60]. Other work has combined eBPF with traditional IDS designs, using kernel-based filtering to pre-drop packets before more detailed inspection in user space [66].

Some prior works merge eBPF and ML to enhance intrusion detection performance. For instance, systems that use eBPF to gather packet-level features for training DT, RF, SVM, and similar classifiers illustrate the potential of kernel-level data as a rich input source for ML models [13].

Despite these advancements, existing research largely addresses DDoS detection in a fragmented manner, focusing either on kernel-level telemetry and in-kernel mitigation using eBPF, or on ML–based detection applied to network or flow-level data. Prior work typically deploys these techniques as standalone intrusion detection or mitigation systems, without integrating them into centralized security monitoring

frameworks.

Building on this foundation, the appended work C4 introduces a unified, modular architecture that forwards fine-grained eBPF telemetry to a SIEM platform, where ML–based analytics are applied for DDoS detection. By embedding eBPF-driven observability within a SIEM-centric monitoring pipeline, the proposed approach enables centralized analysis and correlation while preserving low-overhead data collection in a multi-VM environment.

Overall, the state-of-the-art demonstrates significant progress in specialized areas of cyber defense, but there remains a persistent "implementation gap" between high-level algorithmic advancements and the operational constraints of distributed IoT/IIoT and edge environments. This thesis bridges the gaps by proposing an integrated approach that harmonizes BERT-driven representation learning with blockchain-governed model sharing and eBPF-powered observability. By addressing the identified limitations in detection accuracy, collaborative trust, and monitoring efficiency, the subsequent appended papers detail a framework designed to secure the next generation of interconnected ecosystems.

Chapter 3

Thesis Contribution

J: Journal publication; **C:** Conference publication.

This thesis addresses cybersecurity challenges in modern interconnected environments, with a focus on IoT, IIoT, and EC systems. As these environments become increasingly distributed, heterogeneous, and data-driven, they face growing security risks such as expanded attack surfaces, evolving attack patterns, and the limitations imposed by resource-constrained devices. Traditional IDS approaches often struggle to adapt due to their dependence on static rules and handcrafted features.

To address these challenges, this thesis investigates advanced ML and DL techniques for intelligent and adaptive intrusion detection. The main contributions include the application of transformer-based models, including Google BERT, for contextual feature extraction, together with data balancing techniques such as SMOTE to mitigate class imbalance; the use of TL to improve adaptability and cross-domain generalization; the design of lightweight and scalable architectures for edge and IoT deployments; the development of a blockchain-based secure and auditable model-sharing protocol for DTL environments; and the implementation of telemetry-driven approaches for near real-time threat detection in virtualized and cloud infrastructures.

The appended publications address several key challenges in the field, including:

- Detection of minority and emerging attack classes in imbalanced datasets
- Robust intrusion detection in IIoT and OT networks
- Adaptive and scalable IDS deployment in decentralized environments
- Improved transferability and generalization across datasets and domains
- Confidential and traceable collaborative model sharing
- Low-overhead, near real-time telemetry-based threat detection

An overview of how each publication maps to these research challenges is presented in Table 1.1 in Chapter 1. Therefore, this chapter focuses on the specific technical contributions and author roles associated with each paper.

From a methodological perspective, the thesis demonstrates a clear progression from data-centric intrusion detection improvements (J1, C1, C2, J2), to secure and trust-centric distributed learning frameworks (C3, J3), and finally to operationally focused real-time defense mechanisms in virtualized infrastructures (C4). In particular, C3 introduces a blockchain-governed confidential model-sharing protocol, while J3 extends this framework with artifact screening and lineage-aware lifecycle management for secure model evolution.

Collectively, these works advance the state of the art in intelligent intrusion detection and cybersecurity for next-generation connected systems, providing both theoretical contributions and practically validated solutions across benchmark datasets and real-world implementations.

3.1 Publications appended to the thesis

The chapters presented in the second part of this thesis are based on the following published or submitted works. The main contributions of each publication are briefly summarized below.

J1 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Empowering Network Security: BERT Transformer Learning approach and MLP for Intrusion Detection in Imbalanced Network Traffic. *IEEE Access*, 2024

In this work, we proposed a novel transformer-based intrusion detection framework that addresses the challenges of detecting cyberattacks in highly imbalanced network traffic. Our approach integrates the SMOTE for class rebalancing, BERT for contextual feature extraction, and a MLP for downstream classification. A key innovation is the conversion of structured network flow features into textual representations, enabling BERT to learn semantically rich embeddings that capture complex traffic patterns more effectively than conventional handcrafted features. The model was rigorously validated on three widely used benchmark datasets: CIC-IDS 2017, University of New South Wales Network-Based 2015 (UNSW-NB 2015), and NSL-KDD 2009. Experimental results demonstrated that the integration of BERT and SMOTE substantially improves the detection of minority and difficult attack classes compared to baseline MLP models without advanced feature extraction and data balancing. The proposed framework achieved consistently strong results, reaching about 99% accuracy, precision, recall, and F1, and outperformed several state-of-the-art classifiers. This work highlights the effectiveness of combining transformer-based representation learning with lightweight neural classification for robust and scalable network intrusion detection, especially in environments with class imbalance and diverse attack types.

Author's contribution: The author conceived the main idea, designed and implemented the proposed BERT-SMOTE-MLP framework, performed the experimental analysis on multiple benchmark datasets, contributed to the interpretation of the results, and wrote the draft manuscript. The author was responsible for the development of advanced NLP-based feature extraction, data balancing strategies, and the overall evaluation and comparison with state-of-the-art methods.

C1 Z. Ali, A. Marotta, W. Tiberti, O. Odoardi, D. Cassioli, and P. Di Marco. Enhancing IIoT Security: BERT-Driven Intrusion Detection with MLP in Industrial Networks. In *2025 IEEE 11th World Forum on Internet of Things (WF-IoT)*, pages 1–7, 2025

In this work, we developed an advanced intrusion detection framework specifically designed for IIoT environments, addressing the unique security challenges posed by the convergence of OT and IT. The proposed approach integrates SMOTE for data balancing, Google BERT for contextual feature extraction, and a MLP classifier for robust intrusion detection. By transforming structured industrial network traffic into textual representations, the framework enables BERT to generate semantically rich embeddings that capture complex dependencies and behavioral patterns in IIoT traffic. The model was rigorously validated on three representative IIoT benchmark datasets: CIC-APT-IIoT-2024, WUSTL-IIoT-2021, and WUSTL-IIoT-2018. Experimental results demonstrated consistently excellent performance, with accuracy, precision, recall, and F1 approaching 100% across both multiclass and binary classification tasks. The framework proved highly effective in detecting Advanced Persistent Threats (APT) and minority attack classes, outperforming recent state-of-the-art classifiers and demonstrating strong generalization across diverse IIoT scenarios. This work highlights the effectiveness and scalability of transformer-based contextual feature learning combined with lightweight neural classification for enhancing cybersecurity resilience in real-world industrial networks.

Author's contribution: The author conceived the main idea, designed and implemented the BERT-SMOTE-MLP framework for IIoT security, conducted the experimental analysis on multiple IIoT benchmark datasets, interpreted the results, and wrote the manuscript. The author was responsible for the development of advanced contextual feature extraction, data balancing strategies, and for the comprehensive evaluation and comparison with state-of-the-art methods in the IIoT domain.

C2 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge. In *2025 IEEE 9th Forum on Research and Technologies for Society and Industry (RTSI)*,

pages 1–6, 2025

In this work, we introduced a lightweight intrusion detection framework for decentralized IoT environments, leveraging DTL at the edge to create a unified and compact *supermodel* for resource-constrained devices. The approach builds on the BERT-MLP intrusion detection model by integrating TL, enabling the aggregation and transfer of knowledge learned from heterogeneous datasets and multiple edge nodes. The architecture shifts the most demanding training tasks to capable edge devices, while IoT end devices receive only a small pre-trained model for efficient inference. The methodology combines SMOTE for class imbalance mitigation, BERT for contextual feature extraction, and an MLP classifier, together with replay-based fine-tuning and an updated classification head to prevent catastrophic forgetting and support new attack classes. The framework was validated on CIC-IDS 2017 and NSL-KDD 2009 datasets, emulating heterogeneous data collected at different edge nodes. Experimental results showed that the proposed supermodel maintains excellent classification performance (about 99% accuracy, precision, recall, and F1), outperforms several state-of-the-art classifiers, and remains highly compact (about 3 MB), making it suitable for real distributed IoT deployments. This work demonstrates that combining TL with transformer-based feature extraction provides an effective and scalable pathway for adaptive intrusion detection at the edge.

Author's contribution: The author conceived the main idea, given the assigned case of study of IoT in distributed EC scenarios, designed and implemented the TL-based BERT-MLP supermodel framework, performed the experimental analysis on multiple benchmark datasets, contributed to the interpretation of the results, and wrote the draft manuscript. The author was responsible for the integration of TL, replay-based fine-tuning, and the overall evaluation and comparison with state-of-the-art methods for lightweight and adaptive IoT security.

J2 Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios. *IEEE Internet of Things Journal*, 12(21):44882–44896, 2025

In this work, we substantially extends the C2 work by developing a comprehensive TL framework for intrusion detection in decentralized EC environments. Beyond introducing the TL-enabled BERT-MLP *supermodel*, the work formalizes a collaborative edge architecture in which standalone models trained at different edge nodes exchange knowledge to construct a unified *supermodel* while minimizing computational demands on IoT devices. The paper extends the original methodology with an in-depth analysis of deployment aspects, including computational load, scalability, and model distribution across edge and IoT lay-

ers. It further investigates the trade-offs between BERT-Base and BERT-Large, evaluating how model capacity affects feature representation, detection performance, and deployment feasibility. To improve generalization, the framework incorporates additional optimization strategies including dropout, L2 regularization, replay-based fine-tuning, an updated classification head, early stopping, and learning-rate scheduling. Extensive experiments on the CIC-IDS 2017 and NSL-KDD 2009 datasets demonstrate around 99% accuracy, precision, recall, and F1-score while maintaining a lightweight *supermodel* of approximately 3 MB suitable for real-world edge deployment. The work therefore provides a more complete, scalable, and deployment-oriented intrusion detection framework.

Author's contribution: The author conceived the extended research direction, designed the collaborative edge-based TL architecture, implemented the enhanced TL-enabled BERT-MLP framework, introduced the comparative analysis between BERT-Base and BERT-Large, integrated additional regularization and deployment-aware optimization strategies, performed the comprehensive experimental evaluation and scalability analysis, interpreted the results, and prepared the manuscript. The author was responsible for extending the C2 contribution into a complete journal study covering architecture design, optimization, deployment considerations, and state-of-the-art comparison.

C3 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. Confidential Model Sharing Protocol for Distributed Transfer Learning and Blockchain-Governed Access. Submitted to *10th IEEE Research and Technologies for Society and Industry (RTSI)* Conference (Accepted).

In this work, we establish the foundational blockchain-enabled protocol for confidential model exchange in distributed TL. The contribution is centered on secure publication and governed access to model artifacts in trustless edge environments. Specifically, model artifacts are encrypted and signed locally, published through IPFS, and registered on Ethereum with immutable metadata to support ownership verification and provenance. The protocol defines a requester-specific access workflow in which decryption capability is released only to authorized nodes, preserving confidentiality while maintaining discoverability and auditability. In addition, the design enables transparent lineage tracking through on-chain records, supporting accountable reuse of parent and derived models. Implementation-level validation on an end-to-end prototype (including Open Neural Network Exchange (ONNX)/Hierarchical Data Format 5 (H5) support, wallet-signed transactions, and smart-contract access control) confirms practical feasibility, energy efficiency, and economic viability for real distributed deployments.

Author's contribution: Given the research idea, the author conceived the

architecture and designed the secure model-sharing protocol, implemented the end-to-end prototype, conducted the experimental and system-level evaluation, contributed to the analysis of the security and performance outcomes, and wrote the draft manuscript. The author was responsible for integrating the cryptographic publication pipeline, IPFS-based artifact dissemination, and blockchain-mediated access-control and provenance workflow.

J3 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. A Blockchain-Enabled Framework for Confidential Model Sharing in Distributed Transfer Learning with Artifact Screening. Submitted to *IEEE Transactions on Information Forensics and Security* (Under Review).

In this work, we extend the protocol proposed in C3 into a broader blockchain-enabled framework for secure model lifecycle management in distributed TL. The main advancement is the integration of a pre-publication artifact-screening stage, which filters unsafe or potentially malicious model files before they enter the sharing pipeline, thereby addressing model supply-chain risk. The framework preserves the core confidentiality and access-governance guarantees (local Advanced Encryption Standard with a 256-bit key in Galois/Counter Mode (AES-256-GCM) encryption, Elliptic Curve Digital Signature Algorithm (ECDSA) signatures, IPFS storage, and Ethereum-based metadata control) while introducing stronger lifecycle functions. These include requester-specific delegated key access through an off-chain PRE-style mechanism and lineage-aware model synchronization over on-chain parent–child relations, enabling nodes to discover and optionally adopt validated updates with full provenance continuity. End-to-end prototype evaluation demonstrates that the extended framework remains practically deployable, energy efficient, and economically viable in trustless distributed environments.

Author's contribution: The author contributed to the definition of the framework, then designed and implemented the secure model publication and access-control pipeline, integrated the artifact screening procedure into the model-sharing workflow, and developed the blockchain–IPFS prototype for end-to-end validation. The author conducted the experimental and system-level evaluation, contributed to the analysis of the security and performance outcomes, and wrote the draft manuscript. The author was responsible for integrating cryptographic protection, smart-contract-based governance, delegated key-sharing mechanisms, and lineage-aware synchronization for secure model evolution.

C4 Z. Ali, W. Tiberti, A. Marotta and D. Cassioli. Real-Time DDoS Detection by Integrated eBPF Telemetry and Machine Learning-enhanced SIEM. Submitted to *26th International Conference on Intelligent Multimedia, Networking, and Security (IMNS)* Conference (Accepted).

In this work, we present a real-time DDoS detection framework that integrates kernel-level eBPF telemetry with ML-enhanced SIEM analytics in a virtualized multi-VM environment. The proposed architecture decouples telemetry collection from detection logic: XDP-attached eBPF programs capture fine-grained packets and connection-level events on monitored hosts with low overhead, while centralized SIEM processing performs event normalization, correlation, and near real-time inference. This design improves attack visibility without requiring modifications to protected applications or specialized hardware. The framework further introduces an isolation-aware deployment model that separates attack generation, workload execution, telemetry collection, and security analytics, thereby increasing robustness and deployment realism for cloud-native scenarios. End-to-end experimental evaluation under SYN-flood attack conditions demonstrates that kernel-derived telemetry provides discriminative features for accurate and timely DDoS detection while preserving service availability.

Author's contribution: The author co-conceived the research idea, designed and implemented the eBPF-to-SIEM detection pipeline, and developed the virtualized multi-node experimental setup for controlled attack evaluation. The author was responsible for telemetry engineering, ML-based detection integration, system-level testing, results analysis, and manuscript writing.

3.2 Publications not appended to the thesis

The following publications are not appended to the thesis but contain related materials and applications.

J4 Z. Ali, N. Aslam, A. Marotta, W. Tiberti, and D. Cassioli. A Systematic Review of Kernel-Level Security Mechanisms, Vulnerability Detection and Mitigation in Modern Operating Systems. *Sensors*, 26(8), 2026

In this work, we presented a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA)-based systematic review of kernel-level security mechanisms, vulnerability detection techniques, and mitigation strategies in modern OS. The study synthesized evidence from 30 peer-reviewed papers (2010–2024) and organized the state of the art into three core dimensions: prevention (e.g., Mandatory Access Control (MAC), memory protection, secure boot, and integrity monitoring), detection (e.g., static analysis, dynamic monitoring, and kernel fuzzing), and mitigation (e.g., Address Space Layout Randomization (ASLR)/Data Execution Prevention (DEP), privilege separation, live patching, and hardware-assisted defenses). The review identified key kernel attack vectors, including memory corruption, privilege escalation, rootkits, and race conditions, and highlighted persistent challenges related to cross-platform

generalization, evaluation standardization, and deployment complexity in containerized and virtualized environments. By providing a structured evidence synthesis, this work offers practical guidance for designing layered and adaptive kernel-security architectures.

Author's contribution: The author co-conceived the study, contributed to the review design and evidence synthesis, participated in validation and interpretation of the findings, provided technical resources and domain guidance, and co-authored the manuscript. The author was responsible for supporting the formulation of kernel-security research directions and ensuring the relevance of the synthesized results to practical OS defense.

C5 C. Aouiche, E. M. Aouiche, A. Aouiche, Z. Ali, Y. Alkhrjiah, S. Hamdi, A. Gaidi, A. Marotta, and D. Cassioli. Improving 6G Network Performance with Linear Regression and Stacking GBM. In *2025 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–6, 2025

In this work, we proposed a stacking-based ensemble learning approach for throughput prediction in 6G wireless networks, leveraging a Gradient Boosting Machine (GBM) as the meta-model and LR for robust imputation of missing or imbalanced samples. The methodology was validated on the real-world Lumos5G dataset [48], where the proposed model demonstrated strong predictive accuracy ($R^2 = 0.8482$, RMSE = 0.0948, MAE = 0.0600), outperforming individual supervised models and other ensemble configurations. Key contributions include the integration of LR-based imputation for handling missing data, the use of stacking ensemble architectures to enhance prediction reliability, and the application to real 5G/6G network data for practical relevance. This work highlights the effectiveness of advanced ensemble learning and imputation strategies for intelligent, adaptive optimization of next-generation wireless networks.

Author's contribution: The author contributed to the conception and design of the study, implemented the stacking GBM and LR-based imputation framework, performed the experimental analysis, interpreted the results, and contributed to the writing of the manuscript. The author was responsible for integrating ensemble learning and imputation strategies and for the comprehensive evaluation of the model's effectiveness for 6G network performance prediction.

J5 Z. Ali, A. Akram, N. Aslam, and M. S. Khurram. Supervised Learning Approach for Intrusion Detection in Unbalanced Network Traffic. *VFAST Transactions on Software Engineering*, 13(2):01–12, 2025

In this work, we proposed a supervised learning-based intrusion detection framework that leverages SMOTE to address class imbalance and SVM for classification. The methodology was validated on two widely used benchmark datasets,

CIC-IDS 2017 and NSL-KDD 2009, where SMOTE was applied to balance the class distribution and enhance the effectiveness of the IDS model. The SVM-based approach demonstrated exceptional accuracy (up to 99%) and strong performance across multiple intrusion types, outperforming conventional methods and highlighting the value of advanced ML for network security. This work underscores the importance of combining data balancing techniques with powerful classifiers to improve the detection of both common and rare attacks in modern IDS deployments.

Author's contribution: The author contributed to the conception and design of the study, implemented the SMOTE-SVM framework, performed the experimental analysis on benchmark datasets, interpreted the results, and wrote the manuscript. The author was responsible for integrating data balancing and Supervised Learning (SL) strategies and for the comprehensive evaluation of the IDS model's effectiveness.

Part II

Conclusion

Concluding remarks and future directions

This thesis addressed the problem of building accurate, adaptive, and practically deployable cyber-defense mechanisms for IoT, IIoT, and edge-enabled infrastructures. The central motivation was that conventional IDS pipelines often underperform in realistic settings because of class imbalance, heterogeneous traffic distributions, domain shift, and resource constraints. In addition, as distributed learning becomes more prevalent, secure and auditable model sharing emerges as a critical requirement, while operational cloud/virtualized systems demand low-overhead, near real-time detection pipelines.

To address these challenges, the thesis developed a coherent progression of contributions across seven studies. J1 established the core modeling rationale by combining contextual representation learning (BERT), class-imbalance mitigation (SMOTE), and lightweight downstream classification (MLP), showing strong and stable detection performance on widely used benchmarks under imbalance conditions. C1 then specialized this line for IIoT/OT environments and validated the same design principle on IIoT-focused datasets (WUSTL-IIoT-2018, WUSTL-IIoT-2021, CIC-APT-IIoT-2024), confirming practical relevance in industrial threat scenarios.

The second group of contributions, C2 and J2, introduced and formalized distributed TL for edge-centric deployment. By shifting heavy training tasks to capable edge nodes and transferring compact unified supermodels to resource-constrained IoT devices, these studies improved scalability and deployability while preserving high detection quality. Their experiments on CIC-IDS 2017 and NSL-KDD 2009, along with comparative analysis across BERT variants, clarified actionable trade-offs between model expressiveness, model size, and edge-side operational cost.

The third group of works, C3 and J3, extended the scope from detection performance to trust and governance in collaborative learning. C3 established a blockchain-governed protocol for confidential model sharing that integrates local encryption/signing, IPFS-based dissemination, and Ethereum-based metadata/access control. J3 advanced this baseline into a complete secure model lifecycle framework by introducing pre-publication artifact screening, requester-specific delegated key access, and lineage-aware synchronization. Together, these works showed that confidentiality,

provenance, and controlled reuse can be jointly achieved in trustless distributed settings with practical economic and energy feasibility.

Finally, C4 addressed real-time operational defense through a decoupled eBPF-to-SIEM architecture for virtualized multi-VM systems. By combining kernel-level telemetry collection with centralized ML-based analytics, the proposed pipeline enabled high-fidelity observability and timely DDoS detection without intrusive application instrumentation. This contribution complements the earlier model-centric studies by connecting algorithmic detection performance to deployment-oriented security monitoring.

Overall, this thesis contributes an end-to-end perspective on next-generation cyber defense: *(i)* robust intrusion detection under imbalance and heterogeneity (J1, C1), *(ii)* adaptive and lightweight transfer-learning deployment at the edge (C2, J2), *(iii)* secure and auditable collaborative model sharing (C3, J3), and *(iv)* operationally viable real-time detection in virtualized infrastructures (C4). The combined evidence indicates that integrating transformer learning, distributed transfer learning, blockchain-based trust mechanisms, and telemetry-driven analytics provides a scalable and practically grounded pathway for resilient cybersecurity in modern connected systems.

Cross-study synthesis and new insights

The results provide insights that extend beyond the numerical improvements reported in the individual papers. First, the gains obtained by J1 and C1 are explained by the interaction between contextual representation learning and explicit treatment of class imbalance: transformer-based embeddings capture dependencies among traffic features, while rebalancing prevents the classifier from being dominated by frequent benign or majority attack classes. The high scores therefore do not simply indicate a more powerful classifier; they show that representation quality and data distribution must be addressed together, particularly when minority attacks are the most security-critical events. The IIoT experiments further show that this principle is useful across industrial datasets, but also make clear that transferability is not automatic: changes in protocols, attack composition, and operating context must be evaluated explicitly rather than inferred from performance on a single benchmark.

The edge-oriented studies add a second insight: deployment efficiency is a systems-level property, not merely a consequence of reducing the number of model parameters. DTL and the *supermodel* design separate expensive training from constrained inference, allowing knowledge to be reused across domains while limiting the computational and communication burden at the device. The comparison of model variants consequently exposes a practical accuracy–resource trade-off that is hidden when only aggregate accuracy is reported. Similarly, the secure model-sharing studies demonstrate that model quality alone is insufficient for collaborative learning; confidentiality,

integrity, provenance, and controlled reuse are necessary conditions for making shared models operationally trustworthy. Finally, C4 shows that detection quality also depends on the observation architecture: decoupling kernel telemetry collection from centralized analytics improves deployability and responsiveness because monitoring can remain lightweight even when the analytical models evolve.

These findings should be interpreted together with the limitations of the evaluation. The experiments rely substantially on public benchmark datasets, whose class distributions, attack labels, and traffic-generation procedures may not fully represent continuously changing production environments. Very high accuracy or F1 values can therefore overstate operational performance when temporal drift, unseen attacks, cross-site variation, false-alarm cost, and calibration are not measured consistently. In addition, resource consumption and latency are not identical across papers and platforms, which limits direct comparison of deployment feasibility. The results motivate continued validation on temporally separated and multi-site data, open-world and zero-day settings, hardware-in-the-loop edge deployments, calibrated uncertainty and cost-sensitive metrics, and long-running operational pilots. Making these evaluations standard will help determine whether the observed improvements remain robust under realistic distribution shift and whether they translate into reduced analyst workload and safer response decisions.

Future research will focus on advancing intrusion detection into a mature, intelligence-driven capability that combines generative and reasoning models with principled operational safeguards. Generative AI will be employed to produce high-fidelity synthetic traffic and adversarial variations for training and stress testing, while large pretrained language models-integrated via Retrieval-Augmented Generation (RAG) over telemetry, configuration state, and threat intelligence-will be used to synthesize concise incident narratives, rank hypotheses, and produce prioritized, context-aware remediation recommendations. We will investigate agentic controllers that coordinate multi-step investigative workflows (targeted evidence collection, cross-source correlation, adaptive probing) under explicit human-in-the-loop policies, ensuring automation amplifies analyst effectiveness without undermining control, accountability, or traceability.

At the same time, emphasis will be placed on trust, privacy, and operational safety to enable real-world adoption. Explainable AI methods (feature attribution, counterfactual explanation, and provenance traces) will be integrated so that model decisions are auditable and diagnostically useful; Model Context Protocol (MCP)-style registries and verifiable provenance will support secure model distribution, lifecycle management, and policy-driven access. Finally, privacy-preserving update mechanisms (FL, secure aggregation, and differential privacy) will protect sensitive operational data. We will define safe closed-loop response patterns that combine automated mitigations with risk scoring, staged rollouts, and rapid rollback/approval

gates, and evaluate proposed systems using specific metrics to ensure solutions are effective, safe, and deployable, such as operational measures-detection latency, analyst workload and time-to-triage, false-alarm burden, mitigation safety, and robustness to adaptive adversaries-through long-running pilots, controlled red-team exercises, and reproducible benchmarks.

Bibliography

[1] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab. X-iiotid: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. *IEEE Internet of Things Journal*, 9(5):3962–3977, 2022.

[2] H. Alazzam, A. Sharieh, and K. E. Sabri. A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer. *Expert systems with applications*, 148:113249, 2020.

[3] N. Alhussien, A. Aleroud, R. Rahaeimehr, and A. Schwarzmann. Triggerability of backdoor attacks in multi-source transfer learning-based intrusion detection. In *2022 IEEE/ACM International Conference on Big Data Computing, Applications and Technologies (BDCAT)*, pages 40–47, 2022.

[4] M. L. Ali, K. Thakur, S. Schmeelk, J. Debello, and D. Dragos. Deep learning vs. machine learning for intrusion detection in computer networks: A comparative study. *Applied Sciences*, 15(4):1903, 2025.

[5] O. M. A. Ali, R. A. Hamaamin, B. J. Youns, and S. W. Kareem. Innovative machine learning strategies for ddos detection: A review. *UHD Journal of Science and Technology*, 8(2):38–49, 2024.

[6] Z. Ali, A. Akram, N. Aslam, and M. S. Khurram. Supervised Learning Approach for Intrusion Detection in Unbalanced Network Traffic. *VFAST Transactions on Software Engineering*, 13(2):01–12, 2025.

[7] Z. Ali, N. Aslam, A. Marotta, W. Tiberti, and D. Cassioli. A Systematic Review of Kernel-Level Security Mechanisms, Vulnerability Detection and Mitigation in Modern Operating Systems. *Sensors*, 26(8), 2026.

[8] Z. Ali, A. Marotta, W. Tiberti, O. Odoardi, D. Cassioli, and P. Di Marco. Enhancing IIoT Security: BERT-Driven Intrusion Detection with MLP in Industrial Networks. In *2025 IEEE 11th World Forum on Internet of Things (WF-IoT)*, pages 1–7, 2025.

[9] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Empowering Network Secu-

rity: BERT Transformer Learning approach and MLP for Intrusion Detection in Imbalanced Network Traffic. *IEEE Access*, 2024.

[10] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge. In *2025 IEEE 9th Forum on Research and Technologies for Society and Industry (RTSI)*, pages 1–6, 2025.

[11] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli. Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios. *IEEE Internet of Things Journal*, 12(21):44882–44896, 2025.

[12] K. Alrawashdeh and C. Purdy. Toward an online anomaly intrusion detection system based on deep learning. In *2016 15th IEEE international conference on machine learning and applications (ICMLA)*, pages 195–200. IEEE, 2016.

[13] N. Anand, S. MA, P. K. Aakula, R. B. Ponnuru, R. Patan, and C. R. P. Reddy. Enhancing intrusion detection against denial of service and distributed denial of service attacks: Leveraging extended berkeley packet filter and machine learning algorithms. *IET Communications*, 19(1):e12879, 2025.

[14] C. Aouiche, E. M. Aouiche, A. Aouiche, Z. Ali, Y. Alkhrjiah, S. Hamdi, A. Gaidi, A. Marotta, and D. Cassioli. Improving 6G Network Performance with Linear Regression and Stacking GBM. In *2025 International Conference on Recent Advances in Mathematics and Informatics (ICRAMI)*, pages 1–6, 2025.

[15] B. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi. A hybrid method consisting of GA and SVM for intrusion detection system. *Neural computing and applications*, 27:1669–1676, 2016.

[16] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Transactions on Information and System Security*, 9(1):1–30, 2006.

[17] A. Bagaskara and M. Suryanegara. Evaluation of VGG-16 and VGG-19 Deep Learning Architecture for Classifying Dementia People. In *2021 4th International Conference of Computer and Informatics Engineering (IC2IE)*, pages 1–4, 2021.

[18] J. Benet. IPFS - Content Addressed, Versioned, P2P File System. arXiv preprint arXiv:1407.3561, 2014.

[19] S. Berríos, S. Garcia, P. Hermosilla, and H. Allende-Cid. A machine-learning-based approach for the detection and mitigation of distributed denial-of-service attacks in internet of things environments. *Applied Sciences*, 15(11):6012, 2025.

[20] L. Bhatia and S. Samet. Decentralized federated learning: A comprehensive survey and a new blockchain-based data evaluation scheme. In *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, pages 289–296, 2022.

[21] P. Bojović, I. Bašičević, S. Ocovaj, and M. Popović. A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method. *Computers & Electrical Engineering*, 73:84–96, 2019.

[22] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 1175–1191, 2017.

[23] K. Cao, Y. Liu, G. Meng, and Q. Sun. An overview on edge computing research. *IEEE access*, 8:85714–85728, 2020.

[24] S.-R. Chen, S.-J. Chen, and W.-B. Hsieh. Enhancing machine learning-based ddos detection through hyperparameter optimization. *Electronics*, 14(16):3319, 2025.

[25] Y. Cheng, J. Lu, D. Niyato, B. Lyu, J. Kang, and S. Zhu. Federated transfer learning with client selection for intrusion detection in mobile edge computing. *IEEE Communications Letters*, 26(3):552–556, 2022.

[26] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *Proceedings of NAACL-HLT*, 2019.

[28] M. Dworkin. Recommendation for block cipher modes of operation: Galois/-counter mode (gcm) and gmac. NIST Special Publication 800-38D, 2007.

[29] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access*, 10:40281–40306, 2022.

[30] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, 28(1-2):18–28, 2009.

[31] N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[32] A. Javaid, Q. Niyaz, W. Sun, and M. Alam. A deep learning approach for network intrusion detection system. *Eai Endorsed Transactions on Security and Safety*, 3(9):21, 2016.

[33] P. Kairouz, H. B. McMahan, B. Avent, et al. Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1–2):1–210, 2021.

[34] J. D. M.-W. C. Kenton, L. K. Toutanova, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of naacL-HLT*, pages 4171–4186. Minneapolis, Minnesota, 2019.

[35] C. Khammassi and S. Krichen. A GA-LR wrapper approach for feature selection in network intrusion detection. *computers & security*, 70:255–277, 2017.

[36] H. Kim, J. Park, M. Bennis, and S.-L. Kim. Blockchained on-device federated learning. *IEEE Communications Letters*, 24(6):1279–1283, 2020.

[37] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

[38] M. V. Koroteev. Bert: a review of applications in natural language processing and understanding. *arXiv preprint arXiv:2103.11943*, 2021.

[39] J. L. Leevy, T. M. Khoshgoftaar, R. A. Bauder, and N. Seliya. A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):1–30, 2018.

[40] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie. Toward deep transfer learning in industrial internet of things. *IEEE Internet of Things Journal*, 8(15), 2021.

[41] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 54 of *Proceedings of Machine Learning Research*, pages 1273–1282, 2017.

[42] S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam. Dependable Intrusion Detection System for IoT: A Deep Transfer Learning Based Approach. *IEEE Transactions on Industrial Informatics*, 19(1):1006–1017, 2023.

[43] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, and Y. Elovici. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. *IEEE Pervasive Computing*, 17(3):12–22, 2018.

[44] M. Merenda, C. Porcaro, and D. Iero. Edge machine learning for ai-enabled iot devices: A review. *Sensors*, 20(9):2533, 2020.

[45] H. S. Mohammad Masum. A transfer learning with deep neural network approach for network intrusion detection. *International journal of intellligent computing research*, 12(1), 2021.

[46] N. Moustafa and J. Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)*, pages 1–6, 2015.

[47] B. Mukherjee, L. T. Heberlein, and K. N. Levitt. Network intrusion detection. *IEEE network*, 8(3):26–41, 1994.

[48] A. Narayanan, E. Ramadan, R. Mehta, X. Hu, Q. Liu, R. A. Fezeu, U. K. Dayalan, S. Verma, P. Ji, T. Li, F. Qian, and Z.-L. Zhang. Lumos5g dataset, 2021.

[49] Y. Otoum and A. Nayak. As-ids: Anomaly and signature based ids for the internet of things. *Journal of Network and Systems Management*, 29(3):23, 2021.

[50] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10):1345–1359, oct 2010.

[51] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[52] R. Ramya, D. Dhanasekaran, et al. Applications of ai/ml and blockchain in iot security. In *Internet of Things Security*, pages 219–240. Elsevier, 2026.

[53] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht. Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the internet of things networks with edge capabilities. *Applied Soft Computing*, 155:111434, 2024.

[54] Y. K. Saheed and S. Misra. CPS-IoT-PPDNN: a new explainable privacy preserving dnn for resilient anomaly detection in cyber-physical systems-enabled iot networks. *Chaos, Solitons & Fractals*, 191:115939, 2025.

[55] H. Satilmiş, S. Akleylek, and Z. Y. Tok. A systematic literature review on host-based intrusion detection systems. *Ieee Access*, 12:27237–27266, 2024.

[56] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu. Edge computing: Vision and challenges. *IEEE internet of things journal*, 3(5):637–646, 2016.

[57] D. Sinha and M. El-Sharkawy. Thin MobileNet: An Enhanced MobileNet Architecture. In *2019 IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 0280–0285, 2019.

[58] Standards for Efficient Cryptography Group (SECG). Sec 2: Recommended elliptic curve domain parameters, version 2.0. Standards Specification, 2010.

[59] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE symposium on computational intelligence for security and defense applications*, pages 1–6. Ieee, 2009.

[60] A. Tolay. ebpf-based real-time ddos mitigation for iot edge devices. *arXiv preprint arXiv:2508.00851*, 2025.

[61] L. Torrey and J. Shavlik. Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI Global Scientific Publishing, 2010.

[62] 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*, 2023.

[63] B. Varghese, N. Wang, S. Barbhuiya, P. Kilpatrick, and D. S. Nikolopoulos. Challenges and opportunities in edge computing. In *2016 IEEE international conference on smart cloud (SmartCloud)*, pages 20–26. IEEE, 2016.

[64] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[65] W. Stallings, L. Brown. Introduction to network-based intrusion detection systems, 2007. Accessed on July 15, 2024.

[66] S.-Y. Wang and J.-C. Chang. Design and implementation of an intrusion detection system by using extended bpf in the linux kernel. *Journal of Network and Computer Applications*, 198:103283, 2022.

[67] K. Weiss, T. M. Khoshgoftaar, and D. Wang. A survey of transfer learning. *Journal of Big data*, 3(1):9, 2016.

[68] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Yellow Paper, 2014. Accessed: 2026-03-16.

[69] Q. Yang. An introduction to transfer learning. In C. Tang, C. X. Ling, X. Zhou, N. J. Cercone, and X. Li, editors, *Advanced Data Mining and Applications*, pages 1–1, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[70] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang. A survey on the edge computing for the internet of things. *IEEE access*, 6:6900–6919, 2017.

[71] W. Yutao, L. Zhongtian, B. Yi, L. Jie, X. Fangzheng, and B. Yu. Internet of things intrusion detection system based on transfer learning. In *2022 IEEE 2nd International Conference on Electronic Technology, Communication and Information (ICETCI)*, pages 25–30, 2022.

[72] J. Zhang, C. Luo, M. Carpenter, and G. Min. Federated learning for distributed iiot intrusion detection using transfer approaches. *IEEE Transactions on Industrial Informatics*, 19(7):8159–8169, 2023.

[73] G. Zhu, H. Zhao, H. Liu, and H. Sun. A Novel LSTM-GAN Algorithm for Time Series Anomaly Detection. In *2019 Prognostics and System Health Management Conference (PHM-Qingdao)*, pages 1–6, 2019.

[74] T. Zhu, Y. Huang, Z. Liang, M. Qin, R. Niu, Y. Ma, and Q. Feng. Research on an on-chain and off-chain collaborative storage method based on blockchain and ipfs. *Future Internet*, 18(1):92, 2026.

[75] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In *International conference on learning representations*, 2018.

[76] Y. A. Zouaghi, M. Mahamdioua, A. Lahoulou, and S. Chettibi. Privacy preserving biometric authentication based on fully homomorphic encryption, blockchain, and IPFS data storage. *Multimedia Tools and Applications*, 84(34):42293–42319, 2025.

Part III

Appended Published Papers

Paper J1

Empowering Network Security BERT Transformer Learning Approach and MLP for Intrusion Detection in Imbalanced Network Traffic

Zeesshan Ali, Walter Tiberti, Andrea Marotta and Dajana Cassioli

Published in
IEEE Access

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 21 August 2024, accepted 16 September 2024, date of publication 20 September 2024, date of current version 2 October 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3465045

RESEARCH ARTICLE

Empowering Network Security: BERT Transformer Learning Approach and MLP for Intrusion Detection in Imbalanced Network Traffic

ZEESHAN ALI, (Graduate Student Member, IEEE), WALTER TIBERTI, (Member, IEEE), ANDREA MAROTTA, (Member, IEEE), AND DAJANA CASSIOLI, (Senior Member, IEEE)
Department of Information Engineering, Computer Science, and Mathematics (DISIM), University of L'Aquila, 67100 L'Aquila, Italy
Corresponding author: Zeeshan Ali (zeeshan.ali@graduate.univaq.it)

This work was supported by European Union (EU)—NextGenerationEU under Italian Ministry of University and Research (MUR) National Innovation Ecosystem under Grant ECS00000041—VITALITY—CUP E13C22001060006.

ABSTRACT Intrusion detection systems (IDS) stand as formidable guardians in network security, playing a pivotal role in identifying and mitigating potential threats. As our digital landscape evolves, the imperative for robust intrusion detection mechanisms grows exponentially. The significance of IDS extends beyond preserving the integrity, confidentiality, and availability of network resources. In the dynamic realm of evolving cyber threats, IDS acts as the frontline defender—constantly monitoring network traffic to pinpoint suspicious activities and preemptively mitigate security breaches. In this paper, we study the effectiveness of combining the potency of the transformer-based model known as Bidirectional Encoder Representations from Transformers (BERT) in conjunction with the Synthetic Minority Over-sampling Technique (SMOTE) and a Multi-Layer Perceptron (MLP) to enhance the classification tasks in Machine Learning (ML) based IDS. We tested our approach on well-known and recent datasets, demonstrating that it is possible to obtain very high accuracy and robust performance in the detection of various attack types even when the datasets are affected by class imbalance. Beyond these results, our research introduces a novel perspective by seamlessly integrating the interpretability and context awareness of BERT with the efficient classification of MLP. This novel approach holds promise for advancing intrusion detection capabilities and contributing to the broader cybersecurity community.

INDEX TERMS Bidirectional encoder representations from transformers (BERT), cybersecurity, multi-layer perceptron (MLP), network intrusion detection system (NIDS), imbalance dataset.

I. INTRODUCTION

The proliferation of computer networks and the increasing dependence on information technology have amplified the need for robust security measures. Intrusion detection systems (IDS) have emerged as critical components in ensuring the integrity and availability of computer systems by detecting and responding to unauthorized activities. IDS can be categorized into two primary types: Host-based IDS (HIDS) and Network-based IDS (NIDS). HIDS monitors activities within individual hosts, examining system logs, file integrity, and other host-specific information to identify potential security breaches [1]. Conversely, NIDS monitors network traffic, analyzes packets, and identifies suspicious patterns or anomalies that may indicate unauthorized access attempts or cyber-attack activities [2]. The FIGURE 1 depicts the general NIDS.

Furthermore, within IDS techniques, there are two main categories: signature-based and anomaly-based approaches. Signature-based IDS relies on predefined patterns or

The associate editor coordinating the review of this manuscript and approving it for publication was Mohammad S. Khan.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

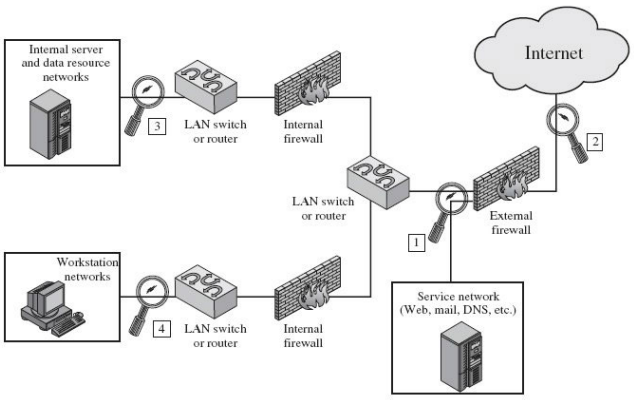


FIGURE 1. Network-based intrusion detection system (NIDS) from [5].

signatures representing known attacks, typically obtained through extensive research or by capturing historical attack patterns. However, signature-based IDS may need help to detect novel or unknown attacks that deviate from the established signatures [3]. Anomaly-based IDS, on the other hand, aims to identify deviations from normal behavior and raise alerts when unusual or suspicious activities occur. This makes it particularly effective in detecting novel or unknown attacks that deviate from the established signatures [4].

In cybersecurity, various attacks severely threaten computer system's integrity and availability. Distributed Denial of Service (DDoS) attacks, for instance, flood target systems with overwhelming traffic, rendering them inaccessible to legitimate users. Advanced Persistent Threats (APTs) represent sophisticated and persistent attacks, often orchestrated by well-funded adversaries, to breach network defenses and gain unauthorized access to sensitive information. Traditional rule-based IDS relies on predetermined patterns and conditions to identify potential security threats in network traffic or system logs, and they have long been used to identify and mitigate security threats. Still, they often struggle to keep pace with the evolving sophistication of cyberattacks. This is where Machine Learning (ML) and Artificial Intelligence (AI) come into play. ML-based IDS leverages the power of advanced algorithms and computational intelligence to enhance the security posture. Unlike rule-based systems, ML-based IDS can learn and adapt from large datasets, making them well-suited to identify novel and previously unseen threats.

These systems employ various ML techniques, such as supervised, unsupervised, and deep learning (DL), to analyze network traffic, system logs, and other relevant data sources. By doing so, they can autonomously detect unusual patterns, anomalies, or suspicious behaviors that might indicate a cyber intrusion. ML-IDS reduces false positives, enhances threat detection accuracy, and offers real-time responses to security incidents. Furthermore, they allow security teams to stay one step ahead of attackers by continuously learning and adapting to new attack vectors.

As we delve into the intricate landscape of intrusion detection, our methodological choices carry deliberate significance. Central to our approach is the meticulous selection of datasets, with a focus on leveraging widely recognized benchmarks to ensure the robustness of our proposed methodology. In this pursuit, we turn our attention to the CIC-IDS 2017 [6], UNSW-NB 2015 [7] and NSL-KDD 2009 [8] datasets, choices made with strategic intent. These datasets, well-regarded within the intrusion detection research community, hold a pivotal role in our study's foundation. The deliberate use of the datasets ensures not only a robust evaluation of our proposed methodology but also alignment with established benchmarks. The CIC-IDS 2017 dataset encapsulates a diverse array of cyber threats, including but not limited to Denial of Service (DoS), Distributed Denial of Service (DDoS), Port Scan, and Malware attacks. These attack types represent a comprehensive spectrum of challenges that modern networks face, making the dataset an invaluable resource for testing the resilience of intrusion detection models. On the other hand, the NSL-KDD 2009 dataset encompasses a rich tapestry of intrusion scenarios, featuring various attack types such as Probe, R2L (Remote to Local), U2R (User to Root), and DoS attacks. Similarly, the UNSW-NB 2015 dataset offers a broad range of network traffic behaviors, including exploits, analysis, fuzzers, backdoors, and reconnaissance attacks, providing an extensive framework for evaluating the efficacy of intrusion detection systems against both traditional and contemporary cyber threats. Each attack type presents a unique set of challenges, demanding nuanced detection mechanisms to safeguard against increasingly sophisticated cyber threats. Our selection of well-known datasets not only facilitates a comprehensive evaluation of our proposed methodology but also diverges from conventional benchmarks in the intrusion detection research community. This departure from established norms ensures a fresh perspective and a more innovative assessment of the efficacy of our approach.

These datasets have been instrumental in setting standards for evaluating IDS, enabling meaningful comparisons with other state-of-the-art approaches as discussed in the results section IV. Class imbalance is another prevalent challenge encountered in IDS, where the number of instances representing normal behavior significantly outweighs those depicting attacks. This class imbalance can hinder the detection performance of ML algorithms, as they tend to favor the majority class. To address this issue, the Synthetic Minority Over-sampling Technique (SMOTE) has been widely employed to generate synthetic samples, thus balancing the class distribution and improving the effectiveness of classification models [9].

Natural language processing (NLP) is a branch of AI that empowers computers to comprehend, interpret, and generate human language meaningfully and practically [10]. Bidirectional Encoder Representations from Transformers (BERT) is a powerful NLP model developed by Google in 2018. BERT utilizes a deep transformer architecture [11] and

has achieved state-of-the-art performance on various NLP tasks, including text classification, named entity recognition, question-answering, capturing contextual relationships, and generating high-quality representations of textual data [12]. BERT can understand the context of log messages and identify subtle patterns or anomalies that rule-based systems might miss. By harnessing BERT capabilities, IDS can transcend the limitations of rule-based systems. Incorporating BERT into IDS leverages its formidable tokenization and feature extraction capabilities, enhancing the effectiveness of processing network traffic data.

Multi-Layer Perceptron (MLP) is an artificial neural network (ANN) architecture that offers several advantages over the other traditional ML classifiers, making it an important choice in various ML tasks, including IDS. MLP employs non-linear activation functions, allowing it to capture intricate, non-linear patterns [13] in network flow data. This is especially crucial in IDS, where network threats often exhibit sophisticated and evolving behaviors. Traditional classifiers like logistic regression or linear support vector machines are fundamentally linear [14]. Linear models can face difficulties capturing intricate relationships within data, particularly when dealing with high-dimensional data, which is often the case in network traffic analysis. In contrast, MLP exhibits a remarkable ability to manage high-dimensional data, a crucial advantage in network traffic analysis. This is especially relevant because network traffic data typically involves extracting numerous features from packets and flows, employing models that can effectively handle this essential complexity, ultimately leading to enhanced performance compared to linear classifiers.

The methods employed in this study, including SMOTE for handling class imbalance, BERT for feature extraction, and MLP for model training, offer significant potential for broader applications. In cybersecurity, these techniques can be extended to enhance various security measures beyond intrusion detection, such as malware classification, phishing detection, and anomaly detection in network traffic. For instance, SMOTE's capability to generate synthetic samples can improve model robustness in scenarios with imbalanced datasets, a common challenge in cybersecurity. Similarly, BERT's powerful language representation can be adapted for analyzing security-related texts, such as threat reports or logs, to identify emerging threats more effectively. Moreover, the application of these methods is not limited to cybersecurity. In other fields, such as finance or healthcare, SMOTE and BERT could be used to tackle imbalanced datasets and extract meaningful features from complex textual data, respectively. In finance, these techniques might be applied to fraud detection or risk assessment, while in healthcare, they could be utilized for patient monitoring or medical record analysis. The ability of these methods to improve model performance and handle diverse types of data underscores their versatility and potential for cross-domain adaptation.

This paper presents our contributions to addressing the aforementioned challenges as follows:

1) We assess the effectiveness of SMOTE in combination with BERT's tokenization and feature extraction capabilities. This fusion tackles class imbalance within the dataset and extracts meaningful representations from textual data associated with network traffic. BERT's powerful language model is leveraged to capture contextual information and generate robust feature representations, while SMOTE addresses the class imbalance issue by generating synthetic samples for the minority class.
2) We propose an IDS model that efficiently handles the downstream classification by combining the aforementioned methods with MLP and ANN. The BERT-SMOTE fusion produces improved feature representations, which the MLP and ANN components are made to use to accurately and efficiently classify network traffic data.
3) Using three widely used intrusion detection datasets (the CIC-IDS 2017, UNSW-NB 2015 and NSL-KDD 2009 datasets), we examined our proposed model and found that it significantly outperformed comparable state-of-the-art methods in terms of accuracy, precision, recall, and F1-score.

The subsequent sections of the document are structured in the following manner: Section II presents a comprehensive discussion of the relevant and recently published literature. Section III explains the proposed model. The results are presented in Section IV. Section V serves as the concluding part of the paper. Lastly, Section VI offers a comprehensive discussion of our findings and their implications while outlining potential future research directions in this domain.

II. RELATED WORK

NIDS can be viewed as a supervised anomaly detection task and phrased as a classification challenge. Several research papers have been proposed recently to extract characteristic patterns from attack intrusions and effectively distinguish them from typical network traffic. These approaches span a spectrum from conventional statistical techniques to state-of-the-art DL-based methods. Using feature selection is a pre-processing technique in machine learning aimed at reducing computational costs [15]. Its primary objective is to eliminate irrelevant features while simultaneously preserving or improving the performance of IDS. Farhan et al. [16] has introduced an approach that combines Convolutional Neural Networks (CNNs) and Long Short-Term Memory (LSTM) networks. This hybrid model aims to capture intricate features within network traffic data while ensuring a balanced representation of the information. Furthermore, it utilizes LSTM to identify various attacks based on the features extracted from the data. In anomaly detection, Zhu et al. [17], proposed LSTM-based models that have been applied to the detection of time series anomalies. However, because LSTM requires a lot of labelled data and has limits in capturing long-

distance dependencies, we did not take it into consideration. On the other hand, BERT's pre-trained language model is a good option since it can efficiently extract contextual information and produce strong feature representations. Nguyen et al. [18] has focused on improving the domain adaptation capabilities of flow-based Network Intrusion Detection Systems (NIDS) by integrating Natural Language Processing (NLP) techniques and the Bidirectional Encoder Representations from Transformers (BERT) framework. This approach addresses the limitations of traditional Machine Learning (ML) classifiers in adapting to different domains, achieving positive results across CIDDS-001 and CIDDS-002 datasets that contain flow samples from a small business environment.

Aslahi-Shahri et al. [19] employed a novel hybrid methodology that integrated a support vector machine with a genetic algorithm. Their primary objective was to enhance the efficiency of feature selection for the KDD CUP 99 [8] dataset. Specifically, this approach aimed to reduce the dimensionality of the dataset, which originally encompassed 41 features, ultimately distilling it into a more manageable set of 10 essential features. Their experiment's results demonstrated their innovative technique's effectiveness by increasing true positive rates while minimizing false positives. In another study by Alazzam et al. [20], they endeavoured to refine the feature selection methodology for NIDS by harnessing the power of a Pigeon Inspired Optimizer (PIO). Their technique efficiently reduced feature sets in KDD CUP 99 and NSL-KDD datasets, condensing them from 41 to 7 and 5 features, respectively. Furthermore, this innovative approach showcased notable achievements, including elevated true positive rates and accuracy, while reducing the time required to construct decision trees. Khammassi et al. [21] implemented a methodological framework in their research, where logistic regression was the feature selection technique, supplemented by a trained model to ascertain the most suitable subset for NIDS. This approach yielded commendable results, manifesting as a heightened detection accuracy while utilizing a reduced set of 18 features from the KDD CUP 99 dataset. Nevertheless, it is imperative to acknowledge that the efficacy of feature selection-based methodologies is intricately linked to the selected features' intrinsic quality and semantic relevance. Consequently, these techniques may only ensure optimal outcomes in specific data scenarios, particularly in imbalanced distributions. Moreover, it is worth noting that prevalent feature selection strategies often rely on heuristic principles and evaluation metrics, potentially constraining their ability to effectively learn and capture intricate interrelationships among features in the data. Zhang et al. [22] introduce a novel model designed to address the challenge of class imbalance in supervised learning. Their approach combines the SMOTE with under-sampling for clustering using a Gaussian Mixture Model (GMM) to tackle the class imbalance problem through resampling effectively. By employing this resampling technique, the model aims to rebalance the dataset and mitigate the impact of class imbalance. Mimura and Tanaka [23] introduced a network interface packet analysis approach that primarily relied on source and destination IP information within Natural Language Processing (NLP). Their investigation revealed that this method could have substantially improved anomaly detection. The authors reported that the best-performing metric for their approach was an F1-score of 98%. In another study, Zhang et al. [24] proposed a Parallel Cross Convolutional Neural Network (PCCNN) to extract feature representations while preventing the neglect of few-sample categories. The PCCNN architecture is intended to overcome the limitations of conventional methods, which often need help handling datasets containing classes with very few samples. By leveraging the PCCNN, the researchers seek to improve the ability to learn meaningful representations from limited data. It is worth noting that these commonly used methods typically rely on labelled data for rebalancing the dataset or learning features. However, in situations where labels are insufficient or scarce, there is a risk of overfitting due to the limited diversity in the training data. Joloudari et al. [25] proposed a model based on Convolutional Neural Network (CNN) and SMOTE to handle class imbalance binary datasets more effectively. Their findings show that the combined CNN-SMOTE achieved 99.08% accuracy on 24 class imbalanced datasets. While the related work has highlighted various strategies for feature selection, class imbalance mitigation, and using conventional and DL techniques, our paper introduces a novel paradigm for IDS. We harnessed the power of SMOTE to handle the class imbalance, BERT transformer-based model for tokenization, and feature extraction with MLP for subsequent classification tasks.

III. PROPOSED METHODOLOGY

The proposed research framework, as depicted in FIGURE 2, consists of several vital steps to address the classification task. The stages in our proposed model are illustrated in the algorithm 1.

A. DATASET SOURCES

In the pursuit of developing a robust and effective IDS, our research methodology incorporates two prominent publicly available datasets: the CIC-IDS 2017 [6] and the NSL-KDD 2009 [8]. The selection of these datasets is motivated by their widespread recognition within network security and intrusion detection, as well as their diverse characteristics that collectively enable comprehensive model evaluation and validation. These datasets have played a pivotal role in benchmarking IDS over the years, facilitating fair comparisons among various methodologies. The continued prevalence of both datasets is evident through their adoption by a substantial number of recent studies [26], [27], [28], underscoring their enduring relevance.

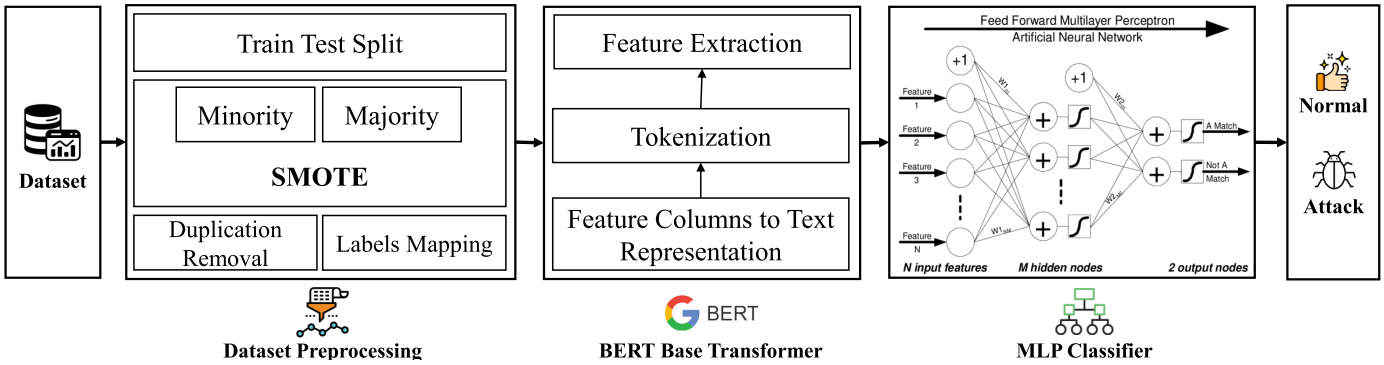


FIGURE 2. Proposed BERT and MLP model for IDS.

TABLE 1. CIC-IDS 2017 dataset.

No	Attacks	Attacks Found	Original Records	Preprocess (Duplicate Records Removal)	SMOTE (Training Records) *: Oversampling **: Undersampling
1	Normal	Normal	2,273,097	176,613	500,000 **
2	Brute Force	FTP-Patator SSH-Patator	7,938 5,897	2,005 2,678	12,000 *
3	DoS	DoS GoldenEye DoS Hulk DoS Slowhttptest DoS Slowloris Heartbleed Bot	10,293 231,073 5,499 5,796 11 1,966	7 58,224 271 411 N/A 13	205,000 *
4	Web Attacks	Web Attack - Brute Force Web Attack - SQL Injection Web Attack - XSS	1,507 21 652	37 N/A N/A	10,000 *
5	Infiltration	Infiltration	36	N/A	1,500 *
6	Port Scan	Port Scan	158,930	68,111	100,000 *
7	DDoS	DDoS	128,027	11	150,000 *

1) CIC-IDS 2017 DATASET

The dataset is an openly accessible repository that comprises raw pcap network traces collected under controlled conditions over five days. These traces encompass various types of attacks co-occurring with normal network traffic. The dataset incorporates a wide range of cyberattacks, including Denial of Service (DoS), Distributed Denial of Service (DDoS), Infiltration, Brute Force, Port Scan, Web attack, and Botnet activities as illustrated in TABLE 1. These attacks are examined across a feature set consisting of 79 distinct features.

2) UNSW-NB 2015 DATASET

The UNSW-NB 2015 dataset [7] was developed using the IXIA PerfectStorm tool in the Cyber Range Lab at UNSW Canberra to simulate a combination of real-world normal activities and synthetic attack behaviors. Tcpdump captured 100 GB of raw network traffic, resulting in Pcap files. This dataset includes nine attack families: Fuzzers, Analysis, Backdoors, DoS, Exploits, Generic, Reconnaissance, Shellcode, and Worms. Our analysis focuses on binary classification as illustrated in TABLE 2. Argus and Bro-IDS tools, along with twelve algorithms, were used to generate 49 features with class labels. The dataset comprises 2,540,047 records stored across four CSV files.

3) NSL-KDD 2009 DATASET

Complementing our research arsenal, the dataset is another essential component of our data-driven methodology. This dataset is derived from the original KDD Cup 1999 dataset, which has been refined to address certain limitations and challenges. The NSL-KDD dataset provides a well-balanced representation of network traffic data, classifying it into four primary categories: Normal, DoS (Denial of Service), Probe, U2R (User to Root) and R2L (Root to Local) types. This dataset encompasses a substantial training set with 125,973 records and an additional testing set containing 22,544 records as illustrated in TABLE 3. Notably, it encompasses 42 distinct features facilitating comprehensive analysis and benchmarking of intrusion detection models.

B. DATASET PREPROCESSING

1) DUPLICATION REMOVAL & LABELS MAPPING

Duplicate samples are instances in the dataset that are identical in all attributes. Removing duplicates is crucial to avoid biased ML models and to ensure that each instance contributes unique information to the analysis. In intrusion

TABLE 2. UNSW-NB 2015 dataset.

No	Labels	Original Records	Preprocess (Duplicate Records Removal)	SMOTE (Training Records) *: Oversampling **: Undersampling
1	Normal	2,218,764	259,415	100,000 **
2	Attack	321,283	232,437	150,000 *

TABLE 3. NSL-KDD 2009 dataset.

No	Labels	Training Set	Testing Set	SMOTE (Training Records) *: Oversampling
1	Normal	67,343	9,711	Not Augmented
2	DoS	45,927	7,460	Not Augmented
3	Probe	11,656	2,421	Not Augmented
4	R2L	995	2,885	8,000 *
5	U2R	52	67	5,000 *

detection datasets, labels often represent different types of network traffic or activities, such as normal traffic or various intrusion categories. Label mapping involves transforming these categorical labels into numerical values. Given the potential disparity between the normal and abnormal network data.

2) SYNTHETIC MINORITY OVER-SAMPLING TECHNIQUE (SMOTE)

The SMOTE is employed to rebalance the dataset instances, thereby mitigating the impact of class imbalance. This step ensures a balanced representation of different classes, thereby improving the overall effectiveness of the IDS. It creates synthetic instances in the minority class to balance class distribution, preventing the model from being biased towards the majority class. Additionally, undersampling is applied to reduce the size of the majority class and we also considered other approaches, including focus loss, to deal with class imbalance. With the focal loss strategy, the loss function is given varying weights according to the distribution of the classes. Compared to focused loss, SMOTE is a more resilient technique that is better able to manage noisy data and outliers. This is important since the minority class in our datasets is noisy and small. To evaluate the performance of the machine learning model, the preprocessed datasets were divided into training and testing subsets using an 80-20 split. Specifically, 80% of the data was allocated to the training set, while the remaining 20% was reserved for testing.

C. PROPOSED MODEL

1) BIDIRECTIONAL ENCODER REPRESENTATIONS FROM TRANSFORMERS (BERT)

A BERT transformer learning approach based on transformers is applied to extract semantic features from flow events effectively. The architectural diagram of BERT is illustrated in FIGURE 3. It has two main parts: Input Embedding and Output Embedding.

In input embedding, multi-head attention helps BERT understand how different words in a sentence relate to each other. It pays attention to various parts of the sentence to capture important information. After multi-head attention, the original input to the transformed information normalizes the result. This step helps maintain the original information's relevance. The feed-forward further processes the information using a neural network with multiple layers. In output embedding, masked multi-head attention focuses on understanding each word's context while considering the surrounding words. It is particularly useful for tasks like language understanding and translation. Finally, the model projects the output into a format suitable for the specific task. This often involves converting the output into probabilities for different word predictions. Using transformers in our proposed approach lets us capture intricate patterns and relationships within the data, enhancing the representation learning process. During feature extraction, we further preprocessed our network flow data after the train-test split, which involves converting the feature columns of the training and testing set into a text representation by concatenating the values of each row into a single string. This process involves taking each row of network traffic features and joining them with spaces to form a coherent text string. For instance, a row with feature values [src-ip, dst-ip, src-port, dst-port, protocol] would be converted to a string like "192.168.0.1 10.0.0.2 12345 80 TCP". This transformation is essential because we are utilizing Google's BERT model, which is designed to process and analyze text data. By converting the numerical and categorical feature data into a text format, we enable the BERT model to effectively interpret and learn from the input features, leveraging its pre-trained language understanding capabilities to capture complex patterns and relationships within the network traffic data. This text representation allows BERT to apply its powerful NLP techniques to our dataset, enhancing the overall performance of our IDS. All textual features from the IDS datasets were fed into Google BERT for tokenization and feature extraction. Tokenization is a fundamental process in NLP that involves converting raw text into a format that can be understood and processed by ML models. In our work with Google BERT for IDS, tokenization was carried out using the BERT base-uncased

Algorithm 1 Algorithm of Proposed Model

procedure Main(Dataset)
    Step 1: Acquire the dataset.
    Step 2: Preprocess the dataset:
        a. Remove duplicate samples.
        b. Map labels to numerical values.
        c. Apply SMOTE for minority oversampling and majority undersampling to balance the dataset.
        d. Split the dataset into training and testing sets.
    Step 3: Tokenization and Feature Extraction using BERT:
        a. Feature columns to text representation.
        b. Initialize BERT tokenizer.
        c. Tokenize text data using BERT.
        d. Extract BERT embeddings as features.
    Step 4: Build and Train MLP Classifier:
        a. Initialize the MLP classifier.
        b. Define the architecture of the MLP.
        c. Training the MLP classifier on the BERT extracted semantic features.
    Step 5: Evaluate the Model:
        a. Predict labels on the test data using the trained MLP classifier.
        b. Calculate evaluation metrics (e.g., accuracy, precision, recall, F1-score).
    Step 6: Analyze and Interpret Results:
        a. Examine the model's performance and interpret the findings.
    Step 7: Conclusion and Discussion:
        a. Summarize the results and implications of the model.
end procedure

pre-trained tokenizer. This is the base version of BERT. Its contextual embeddings have 12 layers (transformer blocks) and 768 hidden units (dimensionality). The ''uncased'' part means it is trained on uncased text, so it doesn't differentiate between lowercase and uppercase letters. It's a good choice when memory resources are limited, as it's smaller compared to larger variants like bert-large, breaking down the text into subword tokens. During tokenization, we perform several key steps. Firstly, the raw text data is split into individual tokens, which are typically words or subwords. Special tokens are then added to the tokenized text. For instance, a classification token ([CLS]) is added at the beginning of the sequence, and a separator token ([SEP]) is added at the end. FIGURE 4 provides an example of the tokenization process using the BERT-Base tokenizer. Here's a step-by-step explanation of the process shown. Input Sentence: The sentence being tokenized is ''A pioneering insight into machine learning techniques''. Step 1 (Tokenization): The sentence is first broken down into tokens. The BERT tokenizer may split words into smaller subword units when the full word is not in its vocabulary. In this example, the sentence is split into the following tokens: a, pioneering, insight, into, ##chine, learning, and techniques. Note that the word ''machine'' is split into two tokens: ##chine and its preceding part 'ma' (not shown), suggesting that the tokenizer has split the word after ''ma'' and then formed the subword token ##chine. Step 2 (Adding Special Tokens): The [CLS] token is added at the beginning, and the [SEP] token is added at the end of the token sequence. These special tokens are used for specific purposes in BERT models: [CLS] (ID: 100) is often used for sentence-level tasks, representing the entire sequence. [SEP] (ID: 101) is used to mark the end of the input sequence or separate different sentences. Step 3 (Conversion to IDs): Each token is converted to its corresponding ID based on the BERT vocabulary. For instance: a is converted to 4322, pioneering is converted to 3455, insight is converted to 11345, into is converted to 24431, ##chine is converted to 22411, learning is converted to 34523, techniques is converted to 22933. The [CLS] token is assigned an ID of 100, and [SEP] is assigned an ID of 101. The final sequence of token IDs will be 100, 4322, 3455, 11345, 24431, 22411, 34523, 22933, and 101. Finally, the input sentence ''A pioneering insight into machine learning techniques.'' is tokenized into words and subwords. Special tokens [CLS] and [SEP] are added. The sequence is then converted into corresponding token IDs, making it ready for input into a BERT model. This process allows BERT to effectively process and understand the sentence. These special tokens help BERT understand the structure of the input text and are crucial for certain tasks like sentence classification. The tokenized sequence is then adjusted to a uniform length. If the sequence is longer than the specified maximum length, it is truncated to fit within this limit. Conversely, if the sequence is shorter, it is padded with a fixed length of 128 to reach the required length. This padding ensures that all input sequences have the same length, which is necessary for efficient batch processing in ML models.

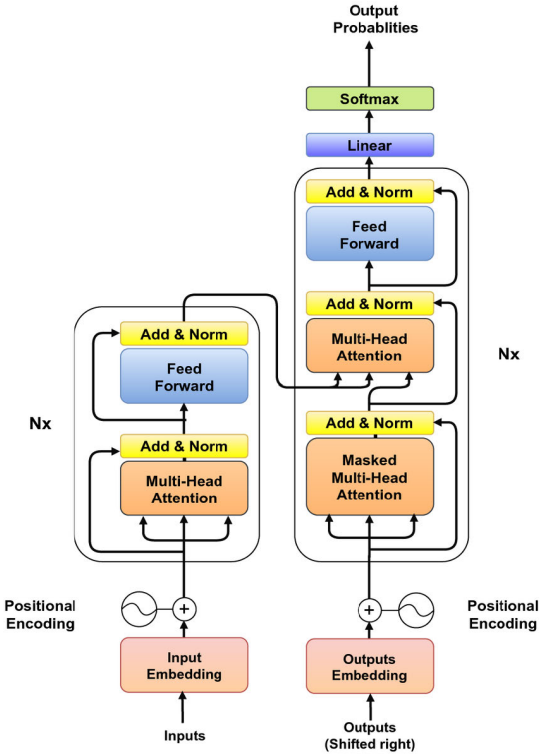


FIGURE 3. Architecture of bidirectional encoder representations from transformers (BERT) from [11].

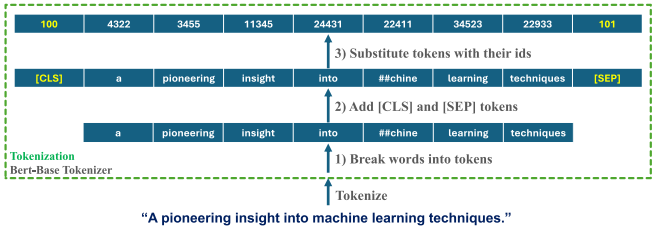


FIGURE 4. Google BERT tokenization process.

Additionally, an attention mask is created alongside the tokenized sequence. This mask indicates which tokens are actual words and which are padding. Through these steps, tokenization transforms raw text into a structured and consistent format that BERT can process, enabling it to perform tasks like text classification, sentiment analysis, and in our case, IDS. The BERT model is applied batch-wise to process the tokenized input data efficiently. For each batch, BERT computes embeddings for the tokens, and we specifically extract the representation of the classification token, which encapsulates the aggregated contextual information for the entire input sequence. This classification token representation effectively captures the semantics of the network flow event. The extracted features are of shape 768, representing the dimensionality of the contextual embeddings generated by the BERT base model. BERT processes the input text data to generate high-dimensional embeddings that capture the semantic context of the input. These embeddings represent the features used by the subsequent MLP model. *Anomaly BERT* is another version of the Google BERT architecture that is intended primarily for anomaly detection applications. As a customized model, *Anomaly BERT* might need additional resources for training and optimization. Making use of the pre-trained BERT basic architecture streamlines the implementation procedure and lowers the amount of computing power needed for training and intensive fine-tuning. Although *Anomaly BERT* is a promising strategy for anomaly detection, our method has advantages in terms of flexibility in downstream models, adaptability, ease of implementation, and transferability of information.

2) MULTI-LAYER PERCEPTRON (MLP)

An MLP is a feedforward artificial neural network comprising layers of interconnected neurons, often called nodes or units [29]. The input layer receives the features as input, i.e., the BERT-extracted semantic features that serve as the input to our downstream classifier. The BERT model was employed to tokenize and extract meaningful features from the text representations. The output from BERT, a set of contextualized embeddings, was then fed into the MLP. The MLP processes these embeddings to learn patterns and make predictions. One or more hidden layers can be placed between the input and output layers. Each hidden layer contains several neurons (or units). These hidden layers allow the network to learn complex patterns and representations from the input data. We used the MLP classifier configured with a single hidden layer consisting of 100 neurons, trained for a maximum of 200 iterations, and initialized with a random seed of 42. Additionally, we applied the Rectified Linear Unit (ReLU) activation function in the hidden layer. The ReLU activation function is a non-linear function commonly used in neural networks. It returns zero for any negative input and passes through any positive input. ReLU introduces non-linearity into the network, allowing it to learn and represent complex relationships in the data. This activation function has gained popularity due to its effectiveness in mitigating and vanishing the gradient problem, which may occur with other activation functions. ReLU has become a cornerstone in modern neural network architectures by facilitating efficient learning.

$$\text{ReLU}(x) = \begin{cases} x, & \text{if } x \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Our proposed model represents a holistic approach to data analysis and classification, combining data preprocessing, advanced tokenization, and a powerful classification algorithm to yield accurate and reliable results. By following this comprehensive research framework, our study aims to leverage the strengths of SMOTE for class balance, transformers for semantic feature extraction, and the specified MLP configuration for classification, thereby contributing to improved performance and robustness in network intrusion detection which underscores our commitment to addressing

TABLE 4. System configuration.

Component	Specification
GPU	NVIDIA A40
Environment	Jupyter Notebook
Programming Language	Python v3.10.9
RAM	206.09 GB
GPU Memory	45.00 GB
Disk	115.00 GB

the research problem effectively and advancing the state-of-the-art in this domain.

D. SYSTEM CONFIGURATION

In this study, we conducted our ML experiments on a high-performance computing platform with a carefully configured system. Our system comprises an NVIDIA A40 GPU, which offers exceptional parallel processing capabilities for efficient DL model training. We utilized the Jupyter Notebook environment and Python v3.10.9 to facilitate interactive and collaborative development. With a substantial 206.09 GB of RAM and 45.00 GB of GPU memory, our system is well-suited for handling extensive datasets and complex ML tasks. Furthermore, we had 115.00 GB of disk space for data storage and management. This robust system configuration was pivotal in achieving accurate and efficient results in our research endeavors.

IV. RESULTS

We formally describe the results obtained from our experimental analysis on three distinct datasets: CIC-IDS 2017, UNSW-NB 2015 and NSL-KDD 2009. These datasets were subjected to in-depth review and analysis to assess the performance of various machine-learning models and intrusion detection techniques. The outcomes of these evaluations will be elucidated in the subsequent sections.

A. EVALUATION METRICS

The proposed approach was subjected to a comprehensive evaluation using five distinct metrics:

1) ACCURACY

Accuracy is a widely used metric for classification tasks and represents the overall effectiveness of the classification process. It is the ratio of correctly classified instances (True Positives and True Negatives) to the total number of instances.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2}$$

2) PRECISION

Precision is a metric that measures the accuracy of positive predictions made by the classification model. It is the True Positives (TPs) ratio to the sum of True Positives and False Positives (FPs). In other words, precision tells us the percentage of instances predicted as positive (abnormal) that were correct.

$$\text{Precision} = \frac{TP}{TP + FP} \tag{3}$$

3) RECALL

Recall, also known as sensitivity or true positive rate, measures the ability of the model to identify all positive instances correctly. It is the True Positives (TPs) ratio to the sum of True Positives and False Negatives (FNs). Recall tells us the percentage of actual positive instances (abnormal) correctly identified by the model.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{4}$$

4) F1-SCORE

The F1-Score is the harmonic mean of precision and recall. It provides a balanced measure of the model's performance between precision and recall. The F1-Score is especially useful when dealing with class imbalance datasets, where one class is more dominant than the other.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5}$$

5) CONFUSION MATRIX

The confusion matrix in TABLE 5 provides a comprehensive view of the model's performance by showing the counts of True Positives (TPs), True Negatives (TNs), False Positives (FPs), and False Negatives (FNs). Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class.

TABLE 5. Confusion matrix.

Actual	Predicted	
	Positive	Negative
Positive	True Positives (TP)	False Negatives (FN)
Negative	False Positives (FP)	True Negatives (TN)

B. EXPERIMENTAL ANALYSIS

In this section, we provide an in-depth performance evaluation of our model, leveraging the power of BERT and MLP for multiclass and binary classification tasks. We conducted these evaluations using three benchmark datasets: CIC-IDS 2017, UNSW-NB 2015 and NSL-KDD 2009.

1) PERFORMANCE EVALUATION OF CIC-IDS 2017 DATASET

In our study, we conducted evaluations of the dataset using an MLP model under two conditions: first, without the application of the SMOTE and Google BERT embeddings as illustrated in the TABLE 6; and second, with both SMOTE and Google BERT. The initial evaluation served as a baseline to understand the model's performance on the original dataset distribution, while the second evaluation aimed to assess the improvements gained through data augmentation and advanced feature extraction techniques.

TABLE 6. Performance evaluation of multiclass classification using MLP on the CIC-IDS 2017 Dataset without SMOTE and Google BERT.

Labels	Accuracy	Precision	Recall	F1-Score
Normal	100.00	100.00	100.00	100.00
Bot	40.00	98.00	40.00	57.00
DDoS	100.00	100.00	100.00	100.00
DoS Golden Eye	98.00	99.00	98.00	99.00
DoS Hulk	99.00	100.00	99.00	99.00
DoS Slowhttptest	99.00	90.00	99.00	94.00
DoS Slowloris	99.00	99.00	99.00	99.00
FTP-Patator	99.00	100.00	99.00	99.00
Heartbleed	100.00	100.00	100.00	100.00
Infiltration	25.00	100.00	25.00	40.00
PortScan	100.00	99.00	100.00	99.00
SSH-Patator	95.00	96.00	95.00	96.00
WebAttackBruteForce	12.00	97.00	12.00	21.00
WebAttackSQLInjection	0.00	0.00	0.00	0.00
WebAttackXSS	0.01	50.00	0.01	0.01

TABLE 7. Performance evaluation (multiclass classification) on CIC-IDS 2017 dataset using SMOTE and google BERT.

No	Labels	Accuracy	Precision	Recall	F1-Score
0	Normal	99.00	99.44	99.46	99.45
1	Brute Force	99.00	99.44	98.81	99.13
2	DoS	99.00	98.89	98.75	98.82
3	Web Attacks	100.00	97.30	99.75	98.51
4	Infiltration	100.00	100.00	100.00	100.00
5	Port Scan	100.00	99.58	99.65	99.62
6	DDoS	100.00	99.81	99.78	99.80

TABLE 8. Performance evaluation (binary classification) on CIC-IDS 2017 dataset using SMOTE and google BERT.

Labels	Precision	Recall	F1-Score	Accuracy
Normal	99.24	99.45	99.35	99.45
Attack	99.53	99.35	99.44	99.35
Overall Accuracy				99.40
Macro Average	99.39	99.40	99.40	
Weighted Average	99.40	99.40	99.40	

TABLE 7 and TABLE 8 represents the results of the binary and multi-classification model applied to a dataset with seven different network traffic classes using the application of SMOTE, Google BERT and MLP. In addition, FIGURE 5 and FIGURE 6 present the multiclass and binary confusion matrices, respectively. The model performed exceptionally well, achieving high accuracy, with all classes having accuracy rates between 99% and 100%. Precision was consistently above 97%, indicating low false positives, while recall was mainly above 98%, indicating the model's ability to capture actual positive instances. The F1-Scores, which balance precision and recall, were also notably high, ranging from 98.51% to a perfect 100%. This excellent performance suggests that the model accurately classifies network traffic into the specified categories. Additionally, the multiclass Receiver Operating Characteristic (ROC) with Area Under the Curve (AUC) is presented in FIGURE 7. It is a metric that quantifies the overall discriminative power of a model in distinguishing between the classes across different thresholds. In the ROC-AUC curve, the y-axis represents the True Positive Rate (TPR) ranging from 0.995 to 1.000,

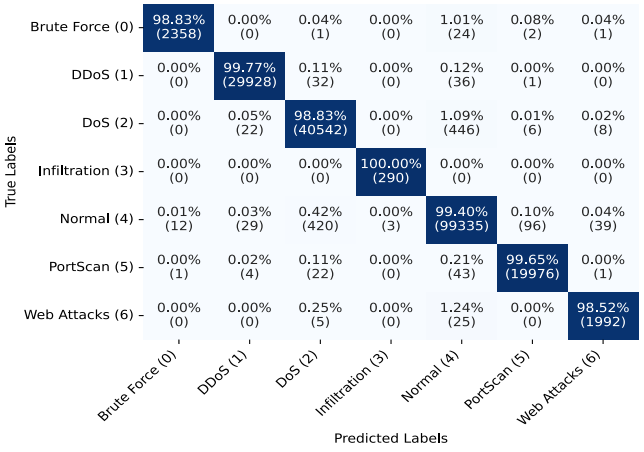


FIGURE 5. Confusion matrix of multiclass classification on CIC-IDS 2017 dataset.

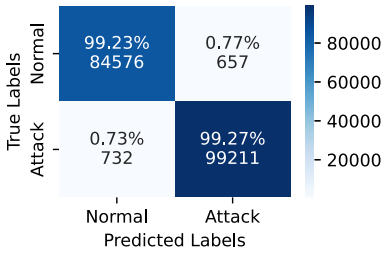


FIGURE 6. Confusion matrix of binary classification on CIC-IDS 2017 dataset.

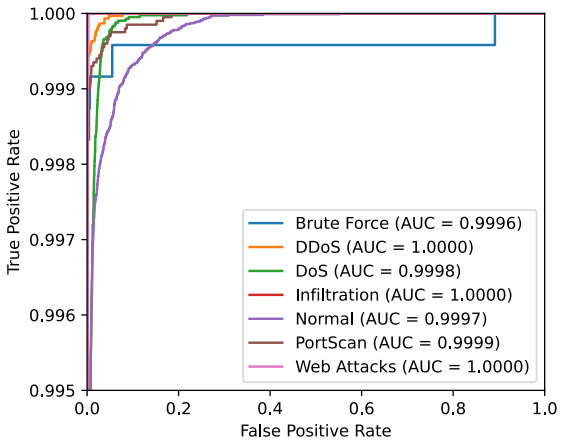


FIGURE 7. ROC-AUC Curve of CIC-IDS 2017 dataset.

while the x-axis represents the False Positive Rate (FPR) ranging from 0.0 to 1.0. Also, the multiclass precision-recall is presented in FIGURE 8. In the precision-recall curve, the y-axis represents precision, which is the proportion of true positive samples out of all predicted positive instances in each class, and the x-axis represents recall, which is the proportion of true positive samples (correctly classified instances) out of all actual positive instances, ranging from 0.97 to 1.00. The curve's trajectory shows how precision and recall change as the decision threshold varies.

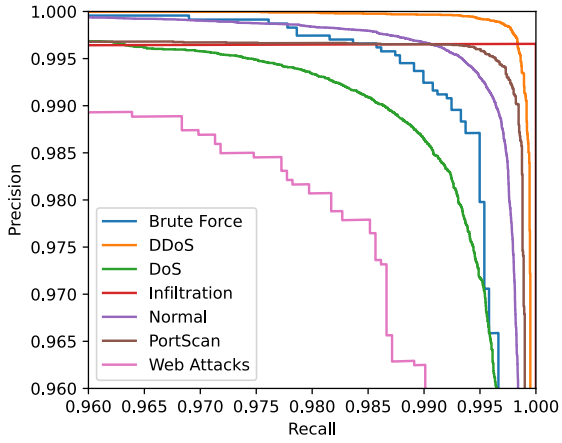


FIGURE 8. Class-wise precision-recall curves of CIC-IDS 2017 dataset.

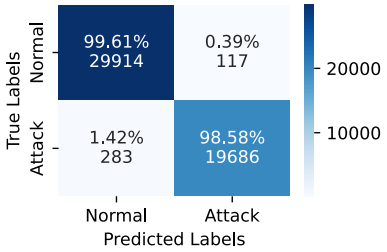


FIGURE 9. Confusion matrix of binary classification on UNSW-NB 2015 dataset.

TABLE 9. Performance evaluation (binary classification) on UNSW-NB 2015 dataset.

Labels	Precision	Recall	F1-Score	Accuracy
Normal	99.00	100.00	99.00	99.20
Attack	99.00	99.00	99.00	99.20
Overall Accuracy				99.00
Macro Average	99.00	99.00	99.00	
Weighted Average	99.00	99.00	99.00	

2) PERFORMANCE EVALUATION OF UNSW-NB 2015 DATASET

The TABLE 9 and FIGURE 9 provides a comprehensive overview of the model's performance in binary classifications using the UNSW-NB 2015 dataset, achieving a remarkable 99.00% in precision, recall, and F1-Score for both the Normal and Attack classes. This high precision indicates that the model accurately identifies instances of each class with minimal false positives, while the perfect recall reflects its ability to detect virtually all relevant instances without missing any. The overall accuracy of 99.00% further underscores the model's effectiveness, confirming that it correctly classifies nearly all instances across both classes. Additionally, the consistent macro and weighted averages of 99.00% for precision, recall, and F1-Score highlight the model's robust performance and balanced handling of both classes, regardless of class distribution. These results collectively demonstrate the model's strong generalization capabilities and reliability in distinguishing between normal and attack instances within the dataset.

3) PERFORMANCE EVALUATION OF NSL-KDD 2009 DATASET

In our study, we conducted evaluations of the dataset using an MLP model under two conditions: first, without the application of the SMOTE and Google BERT embeddings as illustrated in the TABLE 10; and second, with both SMOTE and Google BERT. The initial evaluation served as a baseline to understand the model's performance on the original dataset distribution, while the second evaluation aimed to assess the improvements gained through data augmentation and advanced feature extraction techniques. TABLE 11 and TABLE 12 provides a comprehensive overview of the model's performance in both binary and multiclass classifications using the NSL-KDD 2009 dataset, which encompasses five distinct network traffic classes. FIGURE 10 and FIGURE 11 present the corresponding multiclass and binary confusion matrices, enriching our understanding of the model's performance. We conducted a comparative analysis of our proposed BERT-MLP model against state-of-the-art classifiers, the results of which are summarized in the following TABLE 15 for a comprehensive understanding of their performance. Notably, our model demonstrates outstanding accuracy across various attack classes. For DoS attacks, it achieves an impressive accuracy of 99.83%, showcasing its proficiency in correctly identifying such attacks. Furthermore, the high precision of 99% in this category indicates that it is overwhelmingly accurate when our model predicts DoS attacks. Moving on to the Probe class, our model excels with a 98.47% accuracy, and its precision and recall both stand at 99%. This suggests that it consistently identifies Probe attacks with high confidence. In the case of R2L attacks, the model performs admirably, achieving an accuracy of 97.06% and maintaining high precision, recall, and F1 score, all at 97%. These results highlight the model's effectiveness in identifying instances of unauthorized access and classifying them correctly. For the U2R class, although the precision is relatively lower at 89%, indicating some false positives, the recall stands at a respectable 96%, suggesting that it can detect most U2R attacks. The F1-score of 92% in this category shows a reasonable balance between precision and recall, indicating good overall performance. It is worth noting that the high precision and recall values for the other classes demonstrate the model's suitability for detecting and classifying various types of network attacks with a high degree of accuracy. Additionally, the multiclass ROC-AUC is presented in FIGURE 12. To improve the visualization of curves, the y-axis represents the TPR ranging from 0.9970 to 1.00, while the x-axis FPR ranges from 0.0 to 1.0. Also, the multiclass precision-recall is presented in FIGURE 13. The y-axis represents precision ranges from 0.90 to 1.0, and the x-axis represents recall ranges from 0.50 to 1.00.

We tested our proposed methodology using a variety of classifiers, such as MLP, LSTM, and CNN, to assess its performance. We found that the MLP classifier performed the best, although CNN and LSTM displayed encouraging results. This is probably because the feature representations

TABLE 10. Performance evaluation of multiclass classification using MLP on the NSL-KDD 2009 dataset without SMOTE and google BERT.

No	Labels	Accuracy	Precision	Recall	F1-Score
0	Normal	96.91	70.00	97.00	81.00
1	DoS	82.71	97.00	83.00	89.00
2	Probe	65.47	85.00	65.00	74.00
3	R2L	08.25	34.00	08.00	13.00
4	U2R	28.36	31.00	28.00	29.00

TABLE 11. Performance evaluation (multiclass classification) on NSL-KDD 2009 dataset using SMOTE and google BERT.

No	Labels	Accuracy	Precision	Recall	F1-Score
0	Normal	99.40	99.00	99.00	99.00
1	DoS	99.83	99.00	98.00	99.00
2	Probe	98.47	100.00	99.00	99.00
3	R2L	97.06	97.00	97.00	97.00
4	U2R	96.00	89.00	96.00	92.00

TABLE 12. Performance evaluation (binary classification) on NSL-KDD 2009 dataset.

Labels	Precision	Recall	F1-Score	Accuracy
Normal	99.40	99.41	99.41	99.41
Attack	99.36	99.36	99.36	99.36
Overall Accuracy				99.38
Macro Average	99.38	99.38	99.38	
Weighted Average	99.38	99.38	99.38	

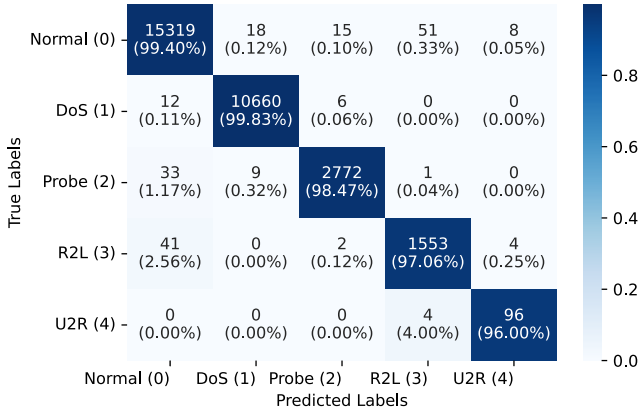


FIGURE 10. Confusion matrix of multiclass classification on NSL-KDD 2009 dataset.

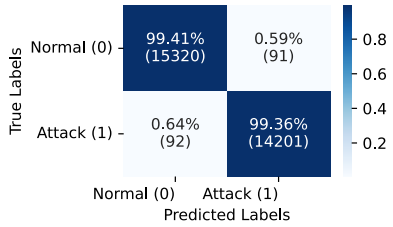


FIGURE 11. Confusion matrix of binary classification on NSL-KDD 2009 dataset.

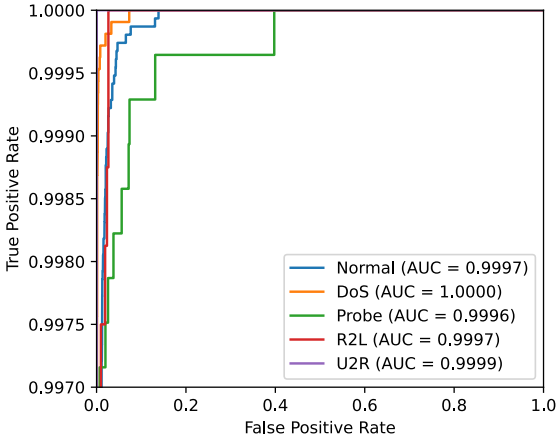


FIGURE 12. ROC-AUC Curve of NSL-KDD 2009 dataset.

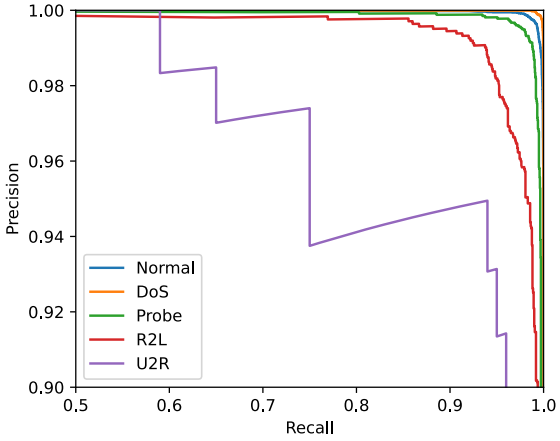


FIGURE 13. Class-wise precision-recall curves of NSL-KDD 2009 dataset.

produced by BERT contain complicated patterns that the MLP can capture with accuracy. Because of its stability and simplicity, the MLP was a good fit for this task, enabling it to identify anomalies and classify network traffic data with accuracy.

4) COMPARISON OF PROPOSED MODEL WITH THE STATE-OF-THE-ART CLASSIFIERS USING CIC-IDS 2017, UNSW-NB 2015 AND NSL-KDD 2009 DATASET

We conducted a comparative analysis of our proposed BERT-MLP model against state-of-the-art classifiers, the results of which are summarized in the following TABLE 13, TABLE 14 and TABLE 15 for a comprehensive understanding of their performance. The proposed model utilizing Google BERT for feature extraction combined with an MLP for classification offers several advantages over existing state-of-the-art approaches. Firstly, BERT's advanced NLP capabilities enable it to capture nuanced and complex features from the dataset. The proposed model achieves a peak accuracy, precision and recall on the dataset CIC-IDS 2017 of 99.39%, and an exceptional F1-Score of 99.99%. This surpasses the performance of Convolutional Neural Networks (CNNs) reported by [30], which achieved an accuracy of 98.79% and an F1-Score of 98.77%. It also outperforms the Self-Organizing Map (SOM) used by [31], which yielded a significantly lower accuracy

TABLE 13. Performance evaluation with state-of-the-art classifiers using the CIC-IDS 2017 dataset.

Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed	BERT and MLP	**99.39**	**99.39**	**99.39**	**99.99**
[30]	Convolutional Neural Network (CNN)	98.79	98.80	98.77	98.77
[31]	Self-Organizing Map (SOM)	80.00	77.40	81.80	80.00
[16]	CNN-Long Short Term Memory (LSTM)	98.56	98.50	99.00	99.25
[16]	CNN-Recurrent Neural Networks (RNN)	97.24	97.50	97.25	96.50
[16]	CNN-Gated Recurrent Unit (GRU)	96.12	96.25	96.25	95.50

TABLE 14. Performance evaluation with state-of-the-art classifiers using the UNSW-NB 2015 dataset.

Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed	BERT and MLP	**99.20**	**99.06**	**99.61**	**99.34**
[32]	Artificial Neural Network (ANN)	86.71	81.54	98.06	89.04
[33]	Decision Tree	89.66	87.22	95.17	91.02
[34]	AdaBoost	92.29	92.30	92.29	92.30
[34]	Random Forest	95.17	95.18	95.18	95.18
[35]	Linear Regression	97.86	97.25	96.28	96.49

TABLE 15. Performance evaluation with state-of-the-art classifiers using the NSL-KDD 2009 dataset.

Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Proposed	BERT and MLP	**99.33**	**99.34**	**99.33**	**99.33**
[36]	Decision Tree	97.40	97.40	97.10	97.20
[31]	SOM	91.00	91.00	91.40	91.00
[37]	Autoencoder	79.74	82.22	79.74	76.47
[28]	Ensemble Learning-Deep Neural Network (DNN)	98.90	99.90	98.71	99.31
[38]	Non-Symmetric Deep Autoencoder (NDAE)	85.42	100.00	85.42	87.37

of 80.00% and an F1-Score of 80.00%. Additionally, the proposed model exceeds the performance of CNN-Long Short Term Memory (LSTM) networks from [16], which achieved an accuracy of 98.56% and an F1-Score of 99.25%, as well as CNN-Recurrent Neural Networks (RNN) and CNN-Gated Recurrent Unit (GRU) models from the same study, with accuracies of 97.24% and 96.12%, respectively.

Our proposed model on the UNSW-NB 2015 dataset achieves an accuracy of 99.20%, precision of 99.06%, recall of 99.61%, and an F1-Score of 99.34%. These results notably outperform the Artificial Neural Network (ANN) by [32], which only achieved an accuracy of 86.71% and an F1-Score of 89.04%. Similarly, the Decision Tree model by [33] has lower performance metrics, with an accuracy of 89.66% and an F1-Score of 91.02%. Moreover, the AdaBoost classifier reported by [34] achieved an accuracy of 92.29% and an F1-Score of 92.30%, while the Random Forest model performed better, with an accuracy of 95.17% and a consistent F1-Score of 95.18%. However, both still fall short compared to the proposed approach. The Linear Regression model by [35] also lags behind, with an accuracy of 97.86% and an F1-Score of 96.49%.

On the NSL-KDD 2009 dataset, the proposed model for classification exhibits remarkable performance with accuracy, recall, and F1-Score of 99.33% and precision of 99.34%, it surpasses the Decision Tree model by [36], which achieved an accuracy of 97.40% and an F1-Score of 97.20%. The Self-Organizing Map (SOM) by [31] lags significantly behind, with an accuracy and F1-Score of 91.00%. The Autoencoder by [37] shows even lower performance metrics, with an accuracy of 79.74% and an F1-Score of 76.47%. Although the Ensemble Learning-Deep Neural Network (DNN) by [28] demonstrates competitive precision at 99.90% and an F1-Score of 99.31%, its accuracy of 98.90% is still lower than that of the proposed model. Additionally, the Non-Symmetric Deep Autoencoder (NDAE) by [38] achieves an accuracy of 85.42% with a perfect precision score of 100.00% but has a lower F1-Score of 87.37%, indicating an imbalanced performance.

The key advantages of the proposed approach lie in its utilization of BERT, which enhances the model's ability to extract complex features from the dataset, leading to higher precision and recall. This allows for a more balanced and comprehensive classification performance, particularly in managing diverse data patterns. The robustness and scalability of the BERT-MLP combination contribute to its consistent outperformance, showcasing its potential as a suitable solution for NIDS.

V. CONCLUSION

In conclusion, this paper has introduced a novel approach to bolstering the capabilities of IDS by harnessing the power of the transformer-based model BERT in conjunction with an MLP for classification tasks. The ever-evolving digital landscape necessitates robust and effective IDS mechanisms to safeguard network resources' integrity, confidentiality, and availability. As cyber threats evolve in complexity and scale,

IDS serves as the front line of defense, tirelessly monitoring network traffic to pinpoint suspicious activities and mitigate potential security breaches. The methodology presented in this paper has been examined thoroughly using two prominent and diverse datasets, CIC-IDS 2017, UNSW-NB 2015 and NSL-KDD 2009, each recognized as state-of-the-art benchmarks within the intrusion detection research community. These datasets present unique challenges, reflecting real-world scenarios and contributing to the depth and validity of our analysis. To address the challenge of class imbalance in these datasets, meticulous preprocessing and the application of the Synthetic Minority Over-sampling Technique (SMOTE) were employed. The experimental results on the CIC-IDS 2017, NSL-KDD 2009 and UNSW-NB 2015 datasets have demonstrated exceptional performance enhancements, with an impressive accuracy of 99%, delivering flawless precision, recall, and F1-scores for various intrusion types. Similarly, on the NSL-KDD 2009 dataset, our approach has exhibited outstanding results, achieving an accuracy of 99.83%. These findings underscore the efficacy of the proposed BERT-MLP approach in enhancing intrusion detection systems across diverse and state-of-the-art datasets. In an era where the imperative for resilient intrusion detection systems remains undeniably clear, the approach presented in this paper represents a significant stride towards fortifying our digital defenses. By harnessing state-of-the-art machine learning techniques and utilizing datasets at the forefront of intrusion detection research, we can better adapt to the ever-changing landscape of cyber threats, providing organizations and individuals with a more robust and effective means of safeguarding their network assets. As the digital realm continues to evolve, research and innovation in intrusion detection systems will remain crucial in ensuring the security and stability of our interconnected world.

VI. DISCUSSION AND FUTURE DIRECTIONS

One of the primary limitations encountered during this research was the availability of GPU resources. Applying transformer-based models, such as BERT, and deep learning techniques, like the MLP, is computationally intensive and demands substantial GPU resources. This constraint may limit the scalability and real-time applicability of the proposed approach, particularly for organizations or environments with limited access to high-performance computing infrastructure. Future work should explore optimization strategies, including model compression, distributed computing, or hardware acceleration methods, to alleviate this limitation and make the approach more scalable and accessible. Smart Network Interface Cards (NICs) can enhance network performance by integrating processing capabilities directly into the network interface. This approach offloads intensive computational tasks from the central CPU to the NIC, which improves efficiency and reduces latency. By incorporating Smart NICs, our IDS can handle real-time data processing more effectively, addressing computational challenges and facilitating smoother deployment in various network environments. We extensively evaluated the CIC-IDS 2017, UNSW-NB 2015 and NSL-KDD 2009 datasets, which may only encompass part of the spectrum of real-world network traffic scenarios. The performance of the proposed BERT-MLP approach may vary when applied to different datasets or in diverse network environments. Future work should validate the approach on broader datasets to assess its generalizability and adaptability.

While acknowledging these constraints is vital for transparency, our primary focus remains charting the path forward. Looking to the future, we are committed to refining and expanding the applicability of the proposed BERT-MLP approach within diverse network security scenarios. We aim to enhance the IDS scalability, generalizability, and real-world effectiveness by addressing these challenges through ongoing research. Our dedication to advancing the state of intrusion detection in an ever-evolving digital landscape is exemplified by our pursuit of these promising future directions.

REFERENCES

[1] Z. Wang and Y. Zhu, "A centralized HIDS framework for private cloud," in *Proc. 18th IEEE/ACIS Int. Conf. Softw. Eng., Artif. Intell., Netw. Parallel/Distrib. Comput. (SNPD)*, Jun. 2017, pp. 115–120.

[2] T. Zhang and S. Bao, "A novel deep neural network model for computer network intrusion detection considering connection efficiency of network systems," in *Proc. 4th Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Jan. 2022, pp. 962–965.

[3] S. Jin, J.-G. Chung, and Y. Xu, "Signature-based intrusion detection system (IDS) for in-vehicle CAN bus network," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2021, pp. 1–5.

[4] S. Bhadauria and T. Mohanty, "Hybrid intrusion detection system using an unsupervised method for anomaly-based detection," in *Proc. IEEE Int. Conf. Adv. Netw. Telecommun. Syst. (ANTS)*, Dec. 2021, pp. 1–6.

[5] W. Stallings and L. Brown. (2007). *Introduction to Network-Based Intrusion Detection Systems*. Accessed: Jul. 15, 2024. [Online]. Available: https://www.informit.com/articles/article.aspx?p=782118

[6] I. Sharafaldin, A. Habibi Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.

[7] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, Nov. 2015, pp. 1–6.

[8] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Secur. Defense Appl.*, Jul. 2009, pp. 1–6.

[9] V. Surya and M. M. Selvam, "An effective machine learning approach for loT intrusion detection system based on SMOTE," in *Proc. 6th Int. Conf. Electron., Commun. Aerosp. Technol.*, Dec. 2022, pp. 905–911.

[10] Z. Zong and C. Hong, "On application of natural language processing in machine translation," in *Proc. 3rd Int. Conf. Mech., Control Comput. Eng. (ICMCCE)*, Sep. 2018, pp. 506–510.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–15.

[12] S. Ravichandiran, *Getting Started With Google BERT: Build and Train State-of-the-Art Natural Language Processing Models Using BERT*. Birmingham, U.K.: Packt Publishing Ltd, 2021.

[13] P. I. Dalatu, A. Fitrianto, and A. Mustapha, "A comparative study of linear and nonlinear regression models for outlier detection," in *Proc. 2nd Int. Conf. Recent Adv. Soft Comput. Data Mining (SCDM)*, Bandung, Indonesia. Cham, Switzerland: Springer, Aug. 2017, pp. 316–326. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-51281-5_32#citeas
[14] A. Widodo and S. Handoyo, "The classification performance using logistic regression and support vector machine (SVM)," *J. Theor. Appl. Inf. Technol.*, vol. 95, no. 19, pp. 5184–5193, 2017. [Online]. Available: https://www.researchgate.net/publication/320409761_The_classification_performance_using_logistic_regression_and_support_vector_machine_SVM#fullTextFileContent
[15] S. Hafeez and N. Kathirisetty, "Effects and comparison of different data pre-processing techniques and ML and deep learning models for sentiment analysis: SVM, KNN, PCA with SVM and CNN," in *Proc. 1st Int. Conf. Artif. Intell. Trends Pattern Recognit. (ICAITPR)*, Mar. 2022, pp. 1–6.
[16] 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," *Digit. Commun. Netw.*, vol. 10, no. 1, pp. 190–204, Feb. 2024.
[17] G. Zhu, H. Zhao, H. Liu, and H. Sun, "A novel LSTM-GAN algorithm for time series anomaly detection," in *Proc. Prognostics Syst. Health Manage. Conf. (PHM-Qingdao)*, Oct. 2019, pp. 1–6.
[18] L. G. Nguyen and K. Watabe, "Flow-based network intrusion detection based on BERT masked language model," in *Proc. 3rd Int. CoNEXT Student Workshop*, Dec. 2022, pp. 7–8.
[19] B. M. Aslahi-Shahri, R. Rahmani, M. Chizari, A. Maralani, M. Eslami, M. J. Golkar, and A. Ebrahimi, "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Comput. Appl.*, vol. 27, no. 6, pp. 1669–1676, Aug. 2016.
[20] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Syst. Appl.*, vol. 148, Jun. 2020, Art. no. 113249.
[21] C. Khammassi and S. Krichen, "A GA-LR wrapper approach for feature selection in network intrusion detection," *Comput. Secur.*, vol. 70, pp. 255–277, Sep. 2017.
[22] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on SMOTE and Gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107315.
[23] M. Mimura and H. Tanaka, "Reading network packets as a natural language for intrusion detection," in *Proc. 20th Int. Conf. Inf. Secur. Cryptol. (ICISC)*, Seoul, South Korea. Cham, Switzerland: Springer, Dec. 2017, pp. 339–350. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-78556-1_19#citeas
[24] Y. Zhang, X. Chen, D. Guo, M. Song, Y. Teng, and X. Wang, "PCCN: Parallel cross convolutional neural network for abnormal network traffic flows detection in multi-class imbalanced network traffic flows," *IEEE Access*, vol. 7, pp. 119904–119916, 2019.
[25] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective class-imbalance learning based on SMOTE and convolutional neural networks," *Appl. Sci.*, vol. 13, no. 6, p. 4006, Mar. 2023.
[26] A. A. E. Donkol, A. G. Hafez, A. I. Hussein, and M. M. Mabrook, "Optimization of intrusion detection using likely point PSO and enhanced LSTM-RNN hybrid technique in communication networks," *IEEE Access*, vol. 11, pp. 9469–9482, 2023.
[27] R. Zhour, C. Khalid, and K. Abdellatif, "Hybrid intrusion detection system based on random forest, decision tree and multilayer perceptron (MLP) algorithms," in *Proc. 10th Int. Conf. Wireless Netw. Mobile Commun. (WINCOM)*, Oct. 2023, pp. 1–5.
[28] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, Jul. 2023.
[29] B. B. Borisenko, S. D. Erokhin, A. S. Fadeev, and I. D. Martishin, "Intrusion detection using multilayer perceptron and neural networks with long short-term memory," in *Proc. Syst. Signal Synchronization, Generating Process. Telecommun. (SYNCHROINFO)*, Jun. 2021, pp. 1–6.
[30] M. A. Siddiqi and W. Pak, "Tier-based optimization for synthesized network intrusion detection system," *IEEE Access*, vol. 10, pp. 108530–108544, 2022.
[31] J. Ables, T. Kirby, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, and M. Seale, "Creating an explainable intrusion detection system using self organizing maps," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, Dec. 2022, pp. 404–412.
[32] S. M. Kasongo and Y. Sun, "Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset," *J. Big Data*, vol. 7, no. 1, p. 105, Dec. 2020.
[33] R. A. Disha and S. Waheed, "A comparative study of machine learning models for network intrusion detection system using UNSW-NB 15 dataset," in *Proc. Int. Conf. Electron., Commun. Inf. Technol. (ICECIT)*, Sep. 2021, pp. 1–5.
[34] N. Pansari, S. Srivastava, R. H. Raghavendra, and M. Agarwal, "Attack classification using machine learning on UNSW-NB 15 dataset using XGBoost feature selection & ablation analysis," in *Proc. IEEE 9th Int. Conf. Converg. Technol. (I2CT)*, vol. 710, Apr. 2024, pp. 1–9.
[35] U. C. Akuthota and L. Bhargava, "Evaluation of machine learning models for intrusion detection with the UNSW-NB15 dataset," in *Proc. IEEE Silchar Subsect. Conf. (SILCON)*, Nov. 2023, pp. 1–5.
[36] R.-F. Hong, S.-C. Horng, and S.-S. Lin, "Machine learning in cyber security analytics using NSL-KDD dataset," in *Proc. Int. Conf. Technol. Appl. Artif. Intell. (TAAI)*, Nov. 2021, pp. 260–265.
[37] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in *Proc. IEEE 13th Int. Conf. Anti-Counterfeiting, Secur., Identificat. (ASID)*, Oct. 2019, pp. 41–45.
[38] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.



ZEESHAN ALI (Graduate Student Member, IEEE) received the Bachelor of Science (B.S.) degree in computer science, in 2021, and the Master of Science (M.S.) degree in computer science, in 2023. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, Italy. He performing research activities under the supervision of Dr. Dajana Cassioli. These academic achievements were obtained from the Department of Computer Science and Information Technology, National College of Business Administration and Economics (NCBAE), Lahore, Pakistan. He was a National Startup Founder with Punjab Information Technology Board (PITB), Lahore. His research interests include cybersecurity, artificial intelligence, machine learning, the Internet of Things, and software quality assurance.



WALTER TIBERTI (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in information and communications technology from the University of L'Aquila, working on embedded systems security, in particular, wireless sensor network security. He is currently an Assistant Professor with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila. He was a Visiting Researcher with the Research Centre in Real-Time and Embedded Computing Systems (CISTER), Porto, Portugal. He is part of the technical committee member of IEEE conferences. He is involved in Italian initiatives to promote nationwide cyber-security education and training. His research work is focused on software security, network security, and cryptography.



ANDREA MAROTTA (Member, IEEE) received the M.Sc. degree in computer engineering and the Ph.D. degree in information and communications technology from the University of L'Aquila, Italy, in 2015 and 2019, respectively.

He is currently an Assistant Professor with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila. He was a Visiting Researcher with the Group for Research on Wireless (GROW), Instituto Superior Técnico/University of Lisbon, Research Institute of Communication, Information and Perception Technologies (TeCIP), Scuola Superiore di Studi Universitari e di Perfezionamento Sant'Anna, Pisa, and the Computer Networks Laboratory (NetLab), University of California at Davis. His research interests include network reliability, SDM optical networks, multi-access edge computing, 5G software defined access, and CoMP coordinated scheduling. He actively participates in the TPC of IEEE Communications Society Conferences.



DAJANA CASSIOLI (Senior Member, IEEE) is currently an Associate Professor in telecommunications engineering with the University of L'Aquila, Italy. Her research interests include wireless communications, 5G/B5G networks, and cybersecurity. She is also the Chair of the IEEE ComSoc RCC SiG on Propagation Channels for 5G&B and the Diversity, Equity, and Inclusion Activity Coordinator of the IEEE Italy Section. She is the Past Chair of the IEEE WIE AG Italy Section (2016–2022) and IEEE VT06/COM19 Italy Chapter (2011–2017). Since 2015, she has been the Coordinator of the University of L'Aquila Node, CINI National Laboratory of Cybersecurity, where she led the CyberEquality WG (2020–2021). She has been awarded the ERC StG VISION (Video-Oriented UWB-Based Intelligent Ubiquitous Sensing), in 2010, and the ERC PoC Grant MobIle health-Care system for monitoring toxicity and symptoms in cAncer patients Receiving diseasE-oriented therapy (iCARE), in 2016. She was the CEO (2014–2018) and in 2019, of the spin-off of the University of L'Aquila, Smartly: Natives of Smart Living Srl. She served as the IEEE EUROCON 2023 WIE Chair, the IEEE ICC 2023 CISS Co-Chair, the PIMRC2018 Industry Co-Chair, the RTSI WIE Chair, in 2018, 2019, and 2020, MELECON2020 and MetroInd 4.0, and a TPC Member of several international conferences (ICC, PIMRC, VTC, and GLOBECOM). She is also an Associate Editor of *Electronic Letters* (IET) and IEEE COMMUNICATIONS LETTERS and an Executive Editor of *Internet Technology Letters* (Wiley) and *Transactions on Emerging Telecommunication Technologies*. In 2000, she was the Summer Manager of the Wireless Systems Research Department, AT&T Laboratories-Research, NJ, USA. She participated in defining the standard channel model for the IEEE 802.15.4 standard, in 2005. In 2022, she was a Visiting Short-Term Scholar with the University of Southern California, LA, USA.

Open Access funding provided by 'Università degli Studi dell'Aquila' within the CRUI CARE Agreement

Paper C1

Enhancing IIoT Security BERT-Driven Intrusion Detection with MLP in Industrial Networks

Zeeshan Ali, Andrea Marotta, Walter Tiberti, Otorino Odoardi, Dajana Cassioli and Piergiuseppe Di Marco

In proceedings of
11th IEEE World Forum on Internet of Things (WF-IoT)

Enhancing IIoT Security: BERT-Driven Intrusion Detection with MLP in Industrial Networks

Z. Ali*, A. Marotta*, W. Tiberti*, O. Odoardi†, D. Cassioli*, P. Di Marco*
*DISIM - University of L'Aquila, L'Aquila, Italy
†S.EL.ME.C. Srl, Chieti, Italy
Email: zeeshan.ali@graduate.univaq.it

2025 IEEE 11th World Forum on Internet of Things (WF-IoT) | 979-8-3315-1522-5/25/$31.00 ©2025 IEEE | DOI: 10.1109/WF-IoT64238.2025.11270748

***Abstract*—Intrusion Detection Systems (IDS) are pivotal in securing Industrial Internet of Things (IIoT) networks, where the convergence of operational technology (OT) and information technology (IT) has heightened vulnerability to sophisticated cyber threats. The complexity, scale, and heterogeneity of IIoT environments necessitate advanced threat detection mechanisms capable of identifying and mitigating attacks in real-time. In this study, we propose a novel deep learning-based IDS framework that integrates Google BERT for advanced feature extraction and a Multi-Layer Perceptron (MLP) for robust classification. Our methodology is evaluated using three IIoT-specific benchmark datasets: CIC APT IIoT 2024, WUSTL IIoT 2021, and WUSTL IIoT 2018, which encompass diverse attack scenarios and network behaviors. To address the prevalent issue of class imbalance in these datasets, we employ the Synthetic Minority Over-sampling Technique (SMOTE), enhancing the model's ability to learn and generalize minority attack patterns effectively. Experimental results demonstrate that our proposed framework significantly improves the detection accuracy of cyber threats in IIoT environments. The system's scalability, efficiency, and real-time applicability make it a promising solution for safeguarding OT networks against evolving cyber threats. This study provides a practical and high-performance approach to enhancing the resilience of IIoT ecosystems.**

***Index Terms*—Intrusion Detection System (IDS), Industrial IoT, Operations Technology, Bidirectional Encoder Representations from Transformers (BERT), Cybersecurity, Multi-Layer Perceptron (MLP), Imbalanced Dataset.**

I. Introduction

THE rapid adoption of Industrial Internet of Things (IIoT) technologies has revolutionized operational technology (OT) environments, enabling real-time monitoring, automation, and data-driven decision-making. However, this increased connectivity has exposed IIoT systems to sophisticated cyber threats [1], making them prime targets for malicious actors. Unlike traditional IT networks, OT systems oversee critical industrial operations, where security breaches can result in severe disruptions, financial losses, and safety hazards. Consequently, Intrusion Detection Systems (IDS) have become crucial for proactively detecting and mitigating cyber threats in IIoT environments [2].

Traditional IDS solutions primarily rely on rule-based or signature-based techniques, which often fail to detect emerging threats, including zero-day attacks [3]. Although traditional Machine Learning (ML) approaches offer improvements, they require extensive feature engineering, domain expertise, and struggle with the high dimensionality and dynamic nature of IIoT traffic. They also face limitations in handling imbalanced datasets typical in cybersecurity scenarios, leading to performance degradation against minority-class attacks.

Several recent works have proposed ML-based solutions to address these challenges. Bushnag et al. [4] introduced a two-level ML-based IDS using Random Forest, achieving high accuracy but highlighting limitations in speed and adaptability. Mao et al. [5] developed a federated incremental learning IDS, focusing on data privacy and continuous learning, yet resource optimization remains challenging. Arsalan et al. [6] presented a lightweight CNN achieving high accuracy, but its performance under evolving attacks was untested. Similarly, Kumar et al. [7] deployed ML models at the fog layer, attaining impressive accuracy on standard datasets but were limited by reliance on traditional feature extraction methods.

While transfer learning (TL) techniques [8] and hybrid DL/ML strategies [9], [10] have improved detection accuracy, they still fail to fully exploit the rich contextual relationships inherent in IIoT traffic data. Despite their advantages, TL and ML-based intrusion

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

detection methods in IIoT environments face several challenges. One major limitation of TL techniques is their struggle with domain adaptation, especially in highly dynamic and heterogeneous IIoT networks. These models often require extensive fine-tuning to generalize effectively across varying network conditions, making them less adaptable to real-world deployments. Similarly, hybrid ML/DL approaches, while enhancing detection capabilities, often fall short in capturing complex contextual dependencies within network traffic. Many of these models still rely on handcrafted feature engineering or traditional statistical methods, which may not be sufficient for identifying subtle and evolving attack patterns.

Motivated by these challenges, our work proposes a novel DL-based IDS framework that integrates Google BERT (Bidirectional Encoder Representations from Transformers) for advanced meaningful context-aware feature extraction directly from raw, unstructured data with a Multi-Layer Perceptron (MLP) classifier for intrusion classification tailored specifically for IIoT environments, ultimately improving detection accuracy and adaptability in evolving cyber threat landscapes. Moreover, our proposed solution integrates the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance, significantly improving detection performance against minority attack classes. We provide a comprehensive evaluation of the proposed approach using three benchmark datasets (CIC-APT-IIoT-2024, WUSTL-IIoT-2021, WUSTL-IIoT-2018), covering diverse attack scenarios related to the IIoT network conditions. Experimental results demonstrate that the proposed IDS performance is excellent for all benchmark datasets, overcoming the limitations of state-of-the-art classifiers applied to IIoT networks.

II. System Model

We consider the reference architecture depicted in Figure 1, specifically tailored for intrusion detection within IIoT environments. This architecture comprises interconnected industrial components such as sensors, valves, pumps, motors, and Remote Terminal Units (RTUs), which communicate with Programmable Logic Controllers (PLCs), microcontrollers, and Supervisory Control and Data Acquisition (SCADA) systems. Communication occurs over a combination of Industrial Ethernet and fieldbus technologies (e.g., Profibus, Profinet) with the associated gateways.

The Intrusion Detection System (IDS) plays a critical role in securing the industrial network infrastructure. Positioned strategically between the IIoT network and external connections, the IDS continuously monitors network traffic to detect and classify potential cyber threats. To facilitate detailed traffic monitoring, dedicated network probes may be placed at multiple strategic locations and various hierarchical levels within the OT domain. These probes sniff traffic within their respective segments and forward the captured data through an Ethernet connection to the IDS for further analysis.

Attackers may attempt to infiltrate the IIoT infrastructure from several points. External attackers can exploit vulnerabilities by accessing the network remotely via the internet, bypassing conventional firewall protections. Internally, attackers may directly compromise local components such as sensors, RTUs, PLCs, or field gateways, exploiting weaknesses in the operational network.

III. Proposed Intrusion Detection System

The IDS training involves several structured phases, building on state-of-the-art machine learning techniques to achieve high accuracy in threat detection. Initially, as shown in Figure 1, raw network traffic data collected from the IIoT environment undergoes a rigorous preprocessing pipeline. This pipeline includes duplication removal to ensure data integrity, label mapping to standardize classification tasks, and addressing class imbalance issues using the Synthetic Minority Oversampling Technique (SMOTE). Such preprocessing results in balanced and representative datasets, significantly improving the classifier's training quality and generalization capability.

Following data preprocessing, our system employs Google BERT (Bidirectional Encoder Representations from Transformers) to perform advanced feature extraction [11]. Specifically, raw numerical network features from the datasets are first tokenized and transformed into textual representations compatible with BERT's input requirements. These tokenized inputs pass through BERT's transformer architecture, where multiple self-attention layers effectively capture complex contextual relationships and hidden dependencies inherent in IIoT network traffic. This step produces high-dimensional embeddings rich in meaningful patterns, significantly surpassing traditional feature extraction methods that typically require domain-specific, manual feature engineering. Finally, the embeddings generated by BERT feed into a Multi-Layer Perceptron (MLP) classifier.

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

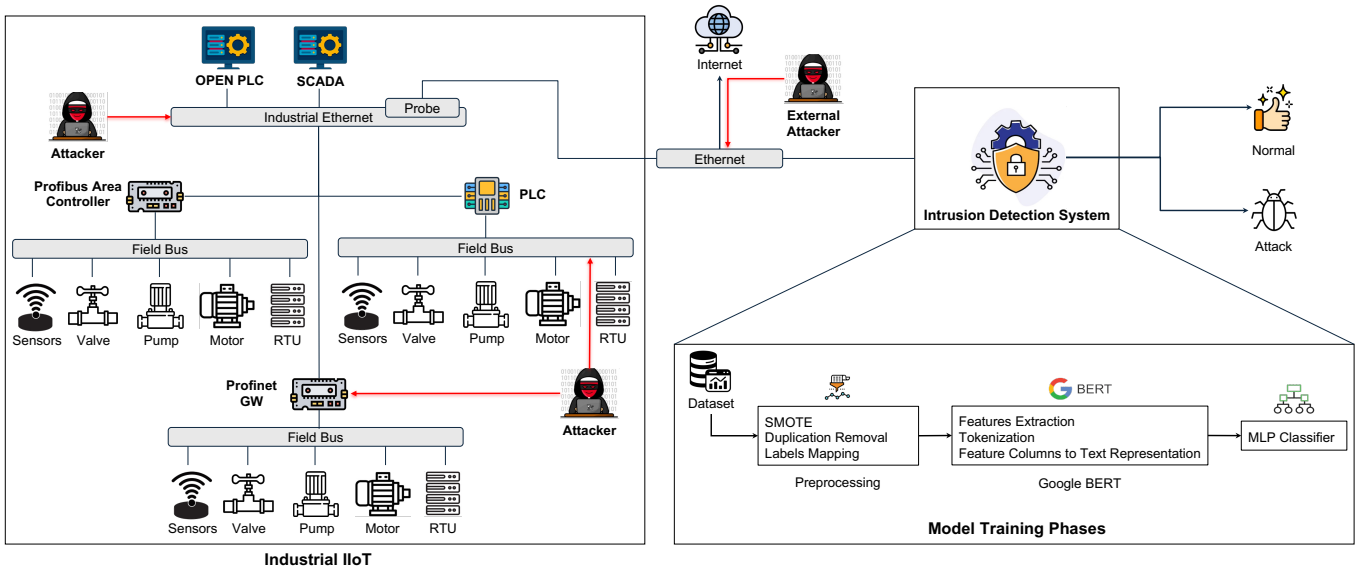


Fig. 1. Proposed System Model

The MLP consists of a single hidden layer containing 256 neurons activated by a Rectified Linear Unit (ReLU), introducing non-linear capabilities necessary for capturing intricate traffic patterns. The classifier employs an adaptive learning rate strategy, dynamically adjusting its step sizes during backpropagation to enhance training efficiency. Training proceeds over a maximum of 200 iterations to ensure robust learning of distinct patterns between normal and attack scenarios.

IV. Validation Setup

To validate the effectiveness and applicability of the proposed IDS described in the previous section, we conducted extensive experiments utilizing three diverse and representative IIoT benchmark datasets: CIC-APT-IIoT-2024 [12], WUSTL-IIoT-2021 [13], and WUSTL-IIoT-SCADA-2018 [14]. These datasets, as detailed in Table I, cover a wide range of realistic IIoT network traffic patterns, including normal operational behaviors and sophisticated cyber threats. Specifically, CIC-APT-IIoT-2024 comprises advanced persistent threats (APT) scenarios captured within a realistic IIoT infrastructure, whereas WUSTL-IIoT-2021 and WUSTL-IIoT-2018 datasets provide real-world traffic collected from SCADA and IIoT networks, encompassing various types of cyberattack strategies and operational conditions. The inclusion of these datasets ensures a comprehensive evaluation of our IDS model's performance, demonstrating its robustness and generalizability across diverse IIoT network environments.

A. Performance Metrics

The proposed approach was thoroughly evaluated using five distinct metrics: Accuracy, Precision, Recall, F1-Score, and the Confusion Matrix. In evaluating the performance of a classification model, the True Positive (TP) refers to instances where the model correctly identifies actual positive cases, while the True Negative (TN) denotes cases where the model accurately classifies negative instances. False Positive (FP) occurs when the model mistakenly labels a negative instance as positive, and False Negative (FN) describes situations where the model fails to detect a positive case, incorrectly classifying it as negative.

1) Accuracy: Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$ measures the overall effectiveness of the classifier. It is defined as the ratio of correctly predicted instances to the total number of predictions, providing a general indication of model performance.

2) Precision: Precision $\frac{TP}{TP+FP}$ measures the accuracy of positive predictions. It is defined as the proportion of true positive instances among all instances predicted as positive, reflecting the model's ability to minimize false positives.

3) Recall: Recall $\frac{TP}{TP+FN}$ measures the model's ability to identify actual positive instances. It is defined as the proportion of true positives among all actual positives, highlighting the model's sensitivity.

4) F1-Score: The F1-Score $(2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision}+\text{Recall}})$ provides a balanced evaluation by combining precision and recall. It is defined as their harmonic mean and is

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

TABLE I
DATASET IDENTIFICATION

Dataset	No	Labels	Records	SMOTE *: Oversampling **: Undersampling	Other Information	
CIC-APT-IIoT-2024	1	Normal	21,598,215	500,000 **	**No. of Features**	70
	2	Collection	460	2,500 *	**Dataset Size**	09.14 GB
	3	Cleanup	192	2,000 *	**Attack Category**	Advanced Persistent Threats
	4	Discovery	138	1,500 *	**Total Records**	21,599,219
	5	Credential Access	58	1,000 *	**Attack Records**	1,004
	6	Command and Control	56	1,000 *		
	7	Persistence	44	1,500 *		
	8	Exfiltration	42	1,500 *		
	9	Lateral Movement	14	1,000 *		
WUSTL-IIOT-2021	1	Normal	1,107,448	500,000 **	**No. of Features**	41
	2	Command Injection	259	1,500 *	**Dataset Size**	390.80 MB
	3	DoS	78,305	100,000 *	**Total Records**	1,194,464
	4	Reconnaissance	8,240	15,000 *	**Attack Records**	87,016
	5	Backdoor	212	1,500 *		
WUSTL-IIoT-2018 ICS SCADA	1	Normal	6,634,581	-	**No. of Features**	7
	2	Attack	403,402	-	**Dataset Size**	189.90 MB
					Duplicate Records	6,975,024
					Total No. of Records	7,037,983

particularly effective for assessing model performance on imbalanced datasets.

5) Confusion Matrix: A confusion matrix is a performance evaluation tool for classification models that outlines how many predictions fall into four categories: TP, TN, FP, and FN. It provides a clear snapshot of the model's ability to correctly distinguish between different classes.

V. Results and Discussion

The results in Table II demonstrate the effectiveness of the approach across various classes of attacks and normal traffic. For the CIC-APT-IIoT-2024 dataset, which contains advanced persistent threats (APT), the overall accuracy reaches 99.977%, with normal traffic detection performing exceptionally well at 99.992% accuracy. Notably, several attack categories, including "Credential Access", "Command and Control", "Discovery", and "Lateral Movement", achieve 100% accuracy, indicating that all positive instances were correctly classified without any false negatives. The precision and recall values for most attack types are also near or at 100%, ensuring a balanced detection of both attack and normal traffic. The lowest precision value is observed in "Cleanup" (98.044%), which suggests a slight tendency for false positives in this category. However, the F1-scores consistently exceed 99%, reflecting the robustness of the model in accurately detecting all threats. Figure 2 presents the multiclass confusion matrix, further illustrating the model's classification performance.

For the WUSTL-IIoT-2021 dataset, the approach maintains high detection accuracy (99.980%), demonstrating strong generalization capabilities across different IIoT environments. The model successfully identifies normal traffic with 99.987% accuracy while also excelling in detecting complex cyber threats. For example, the "Command Injection" attack achieves an impressive precision of 99.313% and recall of 99.318%, resulting in an F1-score of 99.331%. Similarly, "DoS" attacks are effectively detected with 99.988% accuracy, ensuring minimal misclassifications. The "Backdoor" attack, while still achieving a high F1-score of 98.315%, shows a slightly lower precision (96.687%), indicating some false positives in this category. Figure 3 presents the multiclass confusion matrix, providing further insight into classification effectiveness.

Lastly, in the WUSTL-IIoT-2018 dataset, the model continues to deliver exceptional results in a binary classification setting. Both normal traffic and attack categories achieve a near-perfect accuracy of 99.984%. The recall for attacks reaches 100%, meaning all attack instances were correctly classified with no false negatives, while the precision remains high at 99.958%, resulting

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

in an overall F1-score of 99.979%. This indicates that the model not only detects all attacks but also maintains an extremely low false positive rate, which is critical for real-world IIoT. Figure 4 presents the binary confusion matrix, highlighting the effectiveness of the proposed approach in distinguishing between normal and attack traffic. The results presented in Table III presents a comparative analysis of our proposed approach against state-of-the-art classifiers from existing studies.

For the CIC-APT-IIoT-2024 dataset, which contains sophisticated advanced persistent threats (APT) scenarios, our proposed approach clearly outperforms existing solutions, achieving an overall accuracy of 99.9765%. This result significantly surpasses the accuracy reported by existing studies, such as the ensemble model combining Random Forest and XGBoost from [15] (accuracy: 98.00%) and the PCA-based enhanced Deep Neural Network proposed in [16] (accuracy: 95.20%). Moreover, our model yields exceptionally high precision, recall, and F1-scores (all approximately 99.98%), demonstrating robust and reliable performance in precisely distinguishing normal traffic from diverse sophisticated cyber-attacks.

For the WUSTL-IIoT-2021 dataset, our model again showcases remarkable performance, achieving an accuracy of 99.9733%. Compared to classifiers such as Naïve Bayes [17] (accuracy: 97.95%) and CNN [18] (accuracy: 99.94%), our proposed method achieves

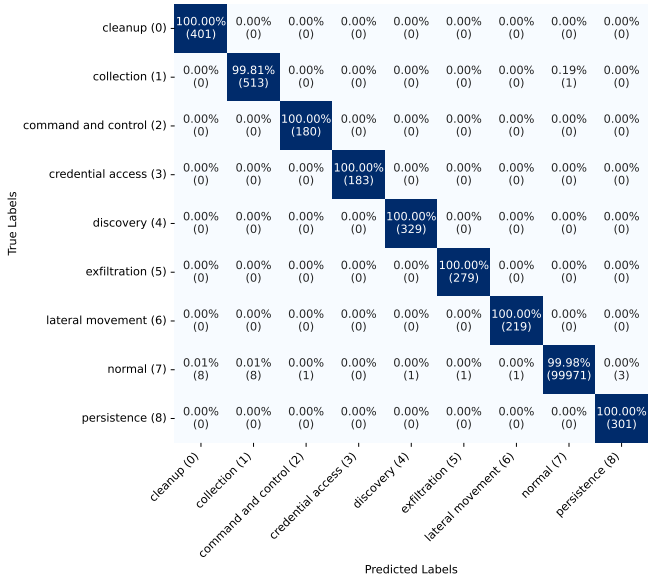


Fig. 2. Confusion Matrix of Multiclass Classification on CIC-APT-IIoT-2024 Dataset

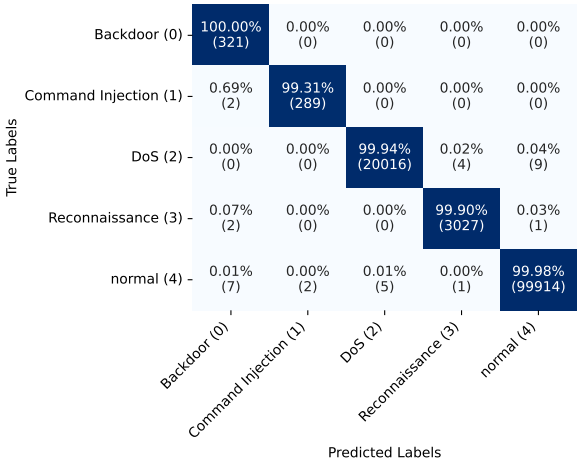


Fig. 3. Confusion Matrix of Multiclass Classification on WUSTL-IIoT-2021 Dataset

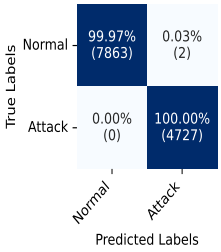


Fig. 4. Confusion Matrix of Binary Classification on WUSTL-IIoT-2018 Dataset

better balance between precision and recall (both at approximately 99.97%), resulting in an impressive F1-score of 99.9733%. Notably, our IDS demonstrates superior performance in terms of recall, minimizing false negatives across challenging attack categories like "Backdoor" and "Command Injection." Although the SVM+LDA method [19] reported a nominal accuracy of 100%, it has lower recall and F1-score values, highlighting that our model offers a more balanced and reliable detection of intrusions, especially in real-world scenarios characterized by class imbalance.

The evaluation on the WUSTL-IIoT-2018 dataset confirms the superior performance of our proposed IDS with a remarkable overall accuracy of 99.9841%. This result clearly outperforms other recent approaches like LDA [22] (accuracy: 93.11%), Deep Q-Learning [21] (accuracy: 99.36%), RNN [23] (accuracy: 99.09%), and Transformer Neural Network [24] (accuracy: 98.54%). Importantly, our model achieves perfect recall (100%) on attack detection, meaning it effectively identifies every instance of intrusion attempts without false neg-

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

TABLE II
PERFORMANCE EVALUATION OF PROPOSED APPROACH USING CIC-APT-IIOT-2024, WUSTL-IIOT-2021, WUSTL-IIOT-2018 DATASETS

Dataset	Labels	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CIC-APT-IIoT 2024	Normal	99971	2405	1	23	99.977	99.999	99.977	99.988
	Cleanup	401	101991	8	0	99.992	98.044	100.00	99.012
	Collection	513	101878	8	1	99.991	98.464	99.805	99.130
	Command & Control	180	102219	1	0	99.999	99.448	100.00	99.723
	Credential Access	183	102217	0	0	100.00	100.00	100.00	100.00
	Discovery	329	102070	1	0	99.999	99.697	100.00	99.848
	Exfiltration	279	102120	1	0	99.999	99.643	100.00	99.821
	Lateral Movement	219	102180	1	0	99.999	99.545	100.00	99.772
	Persistence	301	102096	3	0	99.997	99.013	100.00	99.504
WUSTL-IIOT 2021	Normal	99914	23661	10	15	99.980	99.990	99.985	99.987
	Backdoor	321	123268	11	0	99.991	96.687	100.00	98.315
	Command Injection	289	123307	2	2	99.997	99.313	99.313	99.313
	DoS	20016	103566	5	13	99.985	99.975	99.935	99.955
	Reconnaissance	3027	120565	5	3	99.994	99.835	99.901	99.868
WUSTL-IIoT 2018	Normal	7863	4727	0	2	99.984	100.00	99.975	99.987
	Attack	4727	7863	2	0	99.984	99.958	100.00	99.979

TABLE III
PERFORMANCE EVALUATION OF PROPOSED APPROACH WITH STATE-OF-THE-ART CLASSIFIERS USING CIC-APT-IIOT-2024, WUSTL-IIOT-2021, WUSTL-IIOT-2018 DATASETS

Dataset	Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CIC-APT-IIoT-2024	Proposed	BERT and MLP	**99.9765**	**99.9768**	**99.9765**	**99.9766**
	[15]	Ensemble Model (RF and XGBoost)	98.00	99.00	95.00	96.00
	[16]	PCA-based Enhanced DNN	95.20	93.10	92.70	92.60
WUSTL-IIOT-2021	Proposed	BERT and MLP	**99.9733**	**99.9735**	**99.9733**	**99.9733**
	[17]	Naïve Bayes	97.95	89.09	98.87	93.31
	[18]	CNN	99.94	94.74	92.29	93.41
	[19]	SVM+LDA	100	98.48	92.54	94.84
	[20]	Autoencoders	99.95	99.71	99.72	96.31
	[21]	Deep Q-Learning	98.62	99.23	98.62	99.23
WUSTL-IIoT-2018	Proposed	BERT and MLP	**99.9841**	**99.9577**	**100.00**	**99.9788**
	[22]	LDA	93.11	94.00	93.00	93.00
	[21]	Deep Q-Learning	99.36	99.40	99.38	99.36
	[23]	RNN	99.09	97.12	96.45	95.05
	[24]	Transformer Neural Network	98.54	98.70	98.42	98.61

atives. The very high precision (99.9577%) and F1-score (99.9788%) underscore the model's exceptional capability in minimizing false positives, making it highly practical for securing critical industrial networks. Overall, the comprehensive performance analysis presented demonstrates the effectiveness, reliability, and robustness of our BERT and MLP-based IDS. These outstanding results indicate that our proposed approach significantly surpasses other state-of-the-art methods across all considered metrics, emphasizing its practical applicability and superiority in enhancing cybersecurity resilience in real-world industrial IoT.

VI. CONCLUSIONS

In this study, we presented a deep learning-based Intrusion Detection System (IDS) specifically tailored to secure Industrial Internet of Things (IIoT) environments. Our approach utilizes Google BERT for advanced contextual feature extraction combined with an MLP classifier, significantly enhancing the detection of sophisticated cyber threats in complex IIoT network scenarios. The model's performance was thoroughly evaluated using three benchmark datasets (CIC-APT-IIoT-2024, WUSTL-IIoT-2021, and WUSTL-IIoT-2018), demonstrating superior accuracy, precision, recall, and F1-score compared to recent state-of-the-

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

art methodologies. The results confirmed that our approach effectively addresses class imbalance through SMOTE and consistently identifies various IIoT-specific attacks. Future research directions include evaluating the model's generalizability with additional diverse IIoT datasets and further optimization of computational efficiency to enable real-time deployments in operational industrial networks.

ACKNOWLEDGEMENT

This work has been partially funded by Abruzzo Region through POR FESR Abruzzo 2021/2027 Programme – Measure 1.1.1.1 (CUP C79J24000060007) and by the EU NGEU under the Italian MUR through the National Innovation Ecosystem grant (ECS00000041 - VITALITY - CUP E13C22001060006), the partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART" - CUP E83C22004640001) and the project ISP5G+ (CUP D33C22001300002), which is part of the SERICS program (PE00000014).

REFERENCES

[1] S. Berger, O. Bürger, and M. Röglinger, "Attacks on the industrial internet of things – development of a multi-layer taxonomy," *Computers & Security*, vol. 93, p. 101790, 2020.

[2] M. M. Rahman, S. A. Shakil, and M. R. Mustakim, "A survey on intrusion detection system in iot networks," *Cyber Security and Applications*, vol. 3, p. 100082, 2025.

[3] Q. Liu, V. Hagenmeyer, and H. B. Keller, "A review of rule learning-based intrusion detection systems and their prospects in smart grids," *IEEE Access*, vol. 9, pp. 57 542–57 564, 2021.

[4] A. Bushnag, "Intrusion two-level detection system using machine learning techniques for industrial internet of things (iiot)," in *2023 IEEE International Conference on Internet of Things and Intelligence Systems (IoTaIS)*, 2023, pp. 8–13.

[5] J. Mao, Z. Wei, B. Li, R. Zhang, and L. Song, "Toward ever-evolution network threats: A hierarchical federated class-incremental learning approach for network intrusion detection in iiot," *IEEE Internet of Things Journal*, vol. 11, no. 18, pp. 29 864–29 877, 2024.

[6] M. Arsalan, M. Mubeen, M. Bilal, and S. F. Abbasi, "1d-cnn-ids: 1d cnn-based intrusion detection system for iiot," in *2024 29th International Conference on Automation and Computing (ICAC)*, 2024, pp. 1–4.

[7] P. Kumar and I. Banerjee, "Attack and anomaly detection in iiot networks using machine learning techniques," in *2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, 2023, pp. 1–7.

[8] X. Liu, W. Yu, F. Liang, D. Griffith, and N. Golmie, "Toward deep transfer learning in industrial internet of things," *IEEE Internet of Things Journal*, vol. 8, no. 15, 2021.

[9] S. Aslam, M. M. Alshoweky, and M. Saad, "Binary and multiclass classification of attacks in edge iiot networks," in *Advances in Science and Engineering Technology International Conferences*, 2024, pp. 1–5.

[10] S. Phetlasy, S. Ohzahata, C. Wu, and T. Kato, "Applying smote for a sequential classifiers combination method to improve the performance of intrusion detection system," in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 255–258.

[11] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli, "Empowering network security: Bert transformer learning approach and mlp for intrusion detection in imbalanced network traffic," *IEEE Access*, vol. 12, pp. 137 618–137 633, 2024.

[12] E. Ghiasvand, S. Ray, S. Iqbal, S. Dadkhah, and A. A. Ghorbani, "Cicapt-iiot: A provenance-based apt attack dataset for iiot environment," *arXiv preprint arXiv:2407.11278*, 2024.

[13] M. Zolanvari, "Wustl-iiot-2021," 2021. [Online]. Available: https://dx.doi.org/10.21227/yftq-n229

[14] M. Teixeira, M. Zolanvari, and R. Jain, "Wustl-iiot-2018," 2020. [Online]. Available: https://dx.doi.org/10.21227/kzgp-7t84

[15] S. Narkedimilli, S. Makam, A. V. Sriram, S. P. Mallellu, M. Sathvik, and R. R. V. Prasad, "Enhancing iot network security through adaptive curriculum learning and xai," 2025. [Online]. Available: https://arxiv.org/abs/2501.11618

[16] M. K. V. Priya, H. A. Glory, and V. S. S. Sriram, "Enhanced deep learning for iiot threat intelligence: Revealing advanced persistent threat attack patterns," in *Applications and Techniques in Information Security*, V. Shankar Sriram, A. G. H., G. Li, and S. R. Pokhrel, Eds. Springer Nature, 2025, pp. 201–217.

[17] S. Ismail, S. Dandan, D. W. Dawoud, and H. Reza, "A comparative study of lightweight machine learning techniques for cyber-attacks detection in blockchain-enabled industrial supply chain," *IEEE Access*, 2024.

[18] A. M. Eid, B. Soudan, A. B. Nassif, and M. Injadat, "Enhancing intrusion detection in iiot: optimized cnn model with multi-class smote balancing," *Neural Computing and Applications*, vol. 36, no. 24, pp. 14 643–14 659, 2024.

[19] I. Tareq, S. Amin, B. M Elbagoury, and E.-S. M. El-Horabty, "Machine learning for detecting internet of things network cyber-attacks," *International Journal of Intelligent Computing and Information Sciences*, vol. 24, no. 2, pp. 18–27, 2024.

[20] H. Cui, T. Xue, Y. Liu, and B. Liu, "Transferable intrusion detection model for industrial internet based on deep learning: Iids model combining hybrid deep learning model and transfer learning," in *Proceedings of the 2024 3rd International Conference on Cryptography, Network Security and Communication Technology*, 2024, pp. 107–113.

[21] F. Mesadieu, D. Torre, and A. Chennamaneni, "Leveraging deep reinforcement learning technique for intrusion detection in scada infrastructure," *IEEE Access*, vol. 12, 2024.

[22] A. Alzahrani and T. H. H. Aldhyani, "Design of efficient based artificial intelligence approaches for sustainable of cyber security in smart industrial control system," *Sustainability*, vol. 15, no. 10, 2023.

[23] O. Yousuf and R. N. Mir, "Ddos attack detection in internet of things using recurrent neural network," *Computers and Electrical Engineering*, vol. 101, p. 108034, 2022.

[24] S. Y. Diaba, T. Anafo, L. A. Tetteh, M. A. Oyibo, A. A. Alola, M. Shafie-Khah, and M. Elmusrati, "Scada securing system using deep learning to prevent cyber infiltration," *Neural Networks*, vol. 165, pp. 321–332, 2023.

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:26:57 UTC from IEEE Xplore. Restrictions apply.

Paper C2

A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge

Zeeshan Ali, Walter Tiberti, Andrea Marotta and Dajana Cassioli

In proceedings of
25th IEEE Research and Technologies for Society and Industry (RTSI)

A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge

Zeeshan ALI
Department of Information Engineering, Computer Science and Mathematics
University of L'Aquila
L'Aquila, Italy
zeeshan.ali@graduate.univaq.it

Walter TIBERTI
Department of Information Engineering, Computer Science and Mathematics
University of L'Aquila
L'Aquila, Italy
walter.tiberti@univaq.it

Andrea MAROTTA
Department of Information Engineering, Computer Science and Mathematics
University of L'Aquila
L'Aquila, Italy
andrea.marotta@univaq.it

Dajana CASSIOLI
Department of Information Engineering, Computer Science and Mathematics
University of L'Aquila
L'Aquila, Italy
dajana.cassioli@univaq.it

2025 IEEE 9th Forum on Research and Technologies for Society and Industry (RTSI) | 979-8-3315-9788-7/25/$31.00 ©2025 IEEE | DOI: 10.1109/RTSI64020.2025.11212527

***Abstract*—This paper aims to tackle the cyber threats posed by decentralized settings, such as edge computing, by exploiting the Transfer Learning (TL) approach to create a lightweight Intrusion Detection System (IDS) for resource-constrained IoT devices. The proposed approach shifts the training complexity to the edge, allowing resource-constrained IoT devices to use less resources because they receive from the edge a single (unified) pre-trained *supermodel*. This *supermodel* ensures real-time adaptability across multiple datasets and edge environments and achieves 99% accuracy, precision, recall, and F1-score, while reducing latency and increasing scalability. Beyond TL, the proposed approach combines BERT-based semantic feature extraction, MLP classification, and SMOTE class imbalance compensation for improved performance.**

***Index Terms*—Intrusion Detection System (IDS), Edge Computing, Bidirectional Encoder Representations from Transformers (BERT), Cybersecurity, Multi-Layer Perceptron (MLP), Transfer Learning (TL), Imbalanced Dataset.**

I. INTRODUCTION

Edge computing shifts data processing closer to the source, reducing latency, optimizing bandwidth, and enhancing energy efficiency. Hence, it represents the ideal architecture for IoT applications, smart cities, and autonomous systems [1]. However, its distributed nature introduces security challenges that require robust Intrusion Detection Systems (IDS). Main challenges in edge computing IDS include computational constraints, dataset heterogeneity, and real-time detection needs.

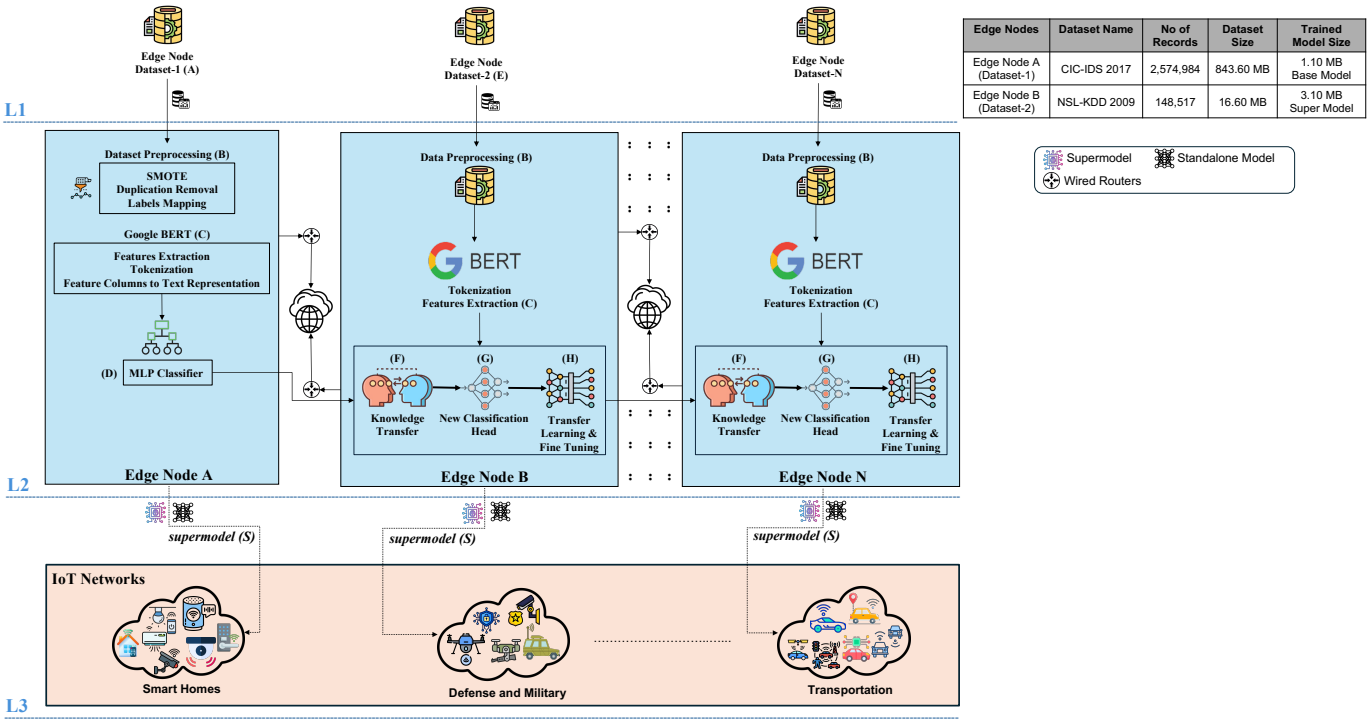
Machine Learning (ML)-based IDS leverage supervised, unsupervised, and deep learning techniques to analyze network traffic and detect anomalies. Unlike rule-based systems, ML-based IDS can adapt to emerging threats, but class imbalance remains a significant challenge, often mitigated by the Synthetic Minority Over-sampling Technique (SMOTE) [2]. IDS capabilities can be further enhanced by incorporating Bidirectional Encoder Representations from Transformers (BERT) to improve feature extraction, while a classifier like Multi-Layer Perceptron (MLP) can reliably capture non-linear patterns in network data [3]. We have already shown, in our previous work [4], the effectiveness of integrating these techniques to create a powerful IDS. However, in resource-constrained edge environments, it is crucial to reduce computational costs and data requirements. This can be achieved by adopting Transfer Learning (TL) which allows reusing pre-trained models in different nodes.

In this study, we enhance the previously proposed IDS by introducing a Transfer Learning mechanism at the edge, that creates a unified *supermodel* by aggregating data from multiple sources. The resulting TL-based IDS reveals to be particularly suited for distributed IoT deployments because the resource-constrained IoT devices are not required to train the model using large datasets, but instead receive a lightweight and highly compact *supermodel*, which has been created at the more powerful edge devices and has a size of only 3.10 MB.

With respect to our previous work [4], the novelty of this paper lies in the introduction of TL, which represents a great improvement, making our IDS particularly effective in heterogeneous and dynamic environments such as edge computing and IoT. In fact, our TL approach enables a continuous refinement and adaptation of the IDS models, thus dynamically optimizing security while utilizing resources efficiently. This efficiency ensures real-time intrusion detection with minimal computational overhead, facilitating practical deployment in distributed IoT environments. By leveraging the knowledge acquired from diverse edge nodes, the *supermodel* can recognize transferable patterns across various intrusion scenarios, enhancing the adaptability of the IDS to novel and evolving threats.

The proposed IDS architecture is validated against two widely adopted IDS datasets, CIC-IDS 2017 and NSL-KDD 2009, that encompass a wide range of attack types and network behaviors, typically observed also in edge computing

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.



Edge Nodes	Dataset Name	No of Records	Dataset Size	Trained Model Size
Edge Node A (Dataset-1)	CIC-IDS 2017	2,574,984	843.60 MB	1.10 MB Base Model
Edge Node B (Dataset-2)	NSL-KDD 2009	148,517	16.60 MB	3.10 MB Super Model

Fig. 1. Proposed Transfer-Learning BERT and MLP *supermodel* for IDS in a decentralized IoT environment.

scenarios. The experimental results demonstrate significant gains in accuracy, precision, recall, and F1-score over state-of-the-art methods, highlighting the model's adaptability across diverse intrusion scenarios within a unified framework.

II. SYSTEM MODEL

The proposed architecture is illustrated in Figure 1, where we identify three layers (L) moving from the top to the bottom:

- **L1** It represents the "Data Acquisition Layer," responsible of gathering the relevant observation data and composing the datasets to be fed to L2;
- **L2** It is the "Edge Layer," formed by a multitude of powerful edge devices in charge of the pre-processing of the datasets acquired by L1 to train its own model. Every edge node is independent from the others in this training task, that follows the same pipeline for all, to ensure consistency and compatibility of "local models."
- **L3** It consists of IoT devices, that cannot train a model themself due to their constrained resources.

The novelty of our approach lies in the introduction of TL in the L2-Edge Layer: whenever a trained model is available at one node, e.g., edge node A, it is advertised among all other edge nodes, that may acquire such a trained model, greatly benefiting from this knowledge to create a *supermodel* by extending its own model through TL.

Without loss of generality, we assume that the "Edge Node A," trained on Dataset-1, makes available its model, referred to as the *Base Model* (BM), to other edge nodes. The "Edge Node B," initially trained on Dataset-2, decides to benefit of TL from the "Edge Node A." The presented methodology can be easily generalized to a greater number of edge nodes exploiting the TL mechanism and periodically launching the process by sharing its own model.

In Figure 1, the different boxes depicted in L2 represent the edge nodes (labeled respectively "Edge Node A," "Edge Node B," ..., "Edge Node N"), each box embedding exactly the same block diagram describing the steps performed to train each node's **Base Model (BM)**, i.e., the dataset preprocessing (Step B), tokenization and feature extraction by Google BERT (Step C), the classification using the MLP model (Step D). This process has been defined in our prior work [4], where these steps have been amply described. However, for the convenience of the reader, we summarize these steps in Algorithm 1. To create the **supermodel** at an edge node, e.g., **B**, from the **BM** created at a different edge node, e.g., **BM-A** created at the edge node **A**, the new dataset created at the edge node **B** by collecting the data acquired by the end devices (Step E) undergoes the same pre-processing steps (Step B and C) to ensure consistency and compatibility. However, instead of running the MLP classifier (Step D) locally at the edge node **B**, the knowledge already available at the edge node A is seamlessly integrated into the edge node **B** model (Step F) by means of the New Classification Head (Step G), followed by Transfer Learning and Fine-Tuning (Step H) to adapt the model to the new dataset. This process culminates in forming a comprehensive model, denoted as the *supermodel* (Step S), which is periodically shared with end devices. The whole

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.

Algorithm 1 Algorithm for the creation of the base-model at the edge node **J** (**BM-J**) as in [4]

procedure MAIN(Dataset-J)
  Input:
    Dataset-J
  Output:
    Base model **BM-J**: A classifier capable of making predictions on Dataset-J.
  Step 1: Load the Dataset-J and perform the *Dataset Preprocessing (B)*
  Step 2: Tokenization and Feature Extraction using Google *BERT Base Transformer (C)*
  Step 3: Build, Train and Test *MLP Classifier (D)*:
    a. Training the MLP classifier on the BERT extracted train features of the Dataset-J.
    b. Base model **BM-J** is created and tested on the Dataset-J test features.
end procedure

Algorithm 2 Algorithm for Transfer Learning from **BM-X** to *supermodel* **SM-Y** for two edge nodes **X** and **Y**

procedure MAIN(BM-X, Dataset-Y)
  Input:
    BM-X: Load the base model created at the edge node **X**
    Dataset-Y: Acquiring Dataset-Y at the edge node **Y**
  Output:
    supermodel **SM-Y**: A classifier capable of making predictions on both datasets **X** and **Y**.
  Step 1: Load the Dataset-Y and perform the *Dataset Preprocessing (B)*
  Step 2: Tokenization and Feature Extraction using Google *BERT Base Transformer (C)*
  Step 3: Base Model *Knowledge Transfer (F)* using TL and Fine-Tuning:
    a. The base model knowledge from **BM-X** is transferred using the proposed Transfer Learning (TL) approach.
    b. A new classification head (G) is added to handle new classes in Dataset-Y, followed by fine-tuning.
    c. The *supermodel* **SM-Y**, enhanced via TL & fine-tuning (H), is created and tested.
end procedure

procedure is summarized in Algorithm 2.

This innovative approach has two advantages. First, it represents a novel means of *enhancing classification capabilities* by disseminating the knowledge gained in the central learning process across edge devices, making the best usage of their highly available resources. Second, the transfer of the *supermodel* is lightweight compared to the full dataset, thus best matches the capabilities of resource-constrained devices in terms of both storage and computational capacity. The evidence of the small size of the *supermodel* to be transferred, stems from our validation on real datasets, as described in Sec. IV and summarized in the table in the top right corner of Fig. 1. The full dataset's size used to train the BM was about 1 GB, while the size of dataset used to train the *supermodel* was 16.6 MB. Both datasets are too large to be managed by the IoT devices, whereas the transferred BM is very small, i.e. 1.1 MB, and the size of the *supermodel* (SM) shared with the IoT devices is about 3 MB. The BM could be easily transferred to the IoT devices, but will offer a limited detection capability, since it is trained over a specific dataset. The SM, instead, is slightly larger, but covers additional attacks and is more robust. The performance evaluation is presented in Sec. IV.

III. TRANSFER LEARNING IMPLEMENTATION

The TL loads the pre-trained BM on dataset-2. To preserve the knowledge the BM has gained from its initial training, i.e., to combat the well-known "catastrophic forgetting" problem [5], a specific strategy has been designed: the "replay" technique. By reintroducing a subset of dataset-1 during training on dataset-2, the replay technique ensures the *supermodel* retains the knowledge from dataset-1 while learning from the new one. To map string labels (i.e categorical values) to integer encodings, the BM top layers are involved, and in particular the *TensorFlow's StringLookup layer* within the MLP. Commonly integrated into ML preprocessing pipelines, the StringLookup layer offers a seamless approach to managing categorical data in numerical format to process and learn from categorical information, e.g., class labels, thereby improving compatibility with neural network architectures. The TensorFlow's StringLookup layer solves errors or misinterpretations due to "Out-Of-Vocabulary (OOV)" tokens, i.e., string class labels not seen during the base model training phase. The StringLookup layer specifies a unique OOV token as the integer representation assigned to any string not found in dataset-1. All unseen labels are mapped to this OOV token. The dataset-2 features are loaded, combined with a subset (20%) of the dataset-1, and then scaled. The *supermodel* efficiency is enhanced by updating the classification head to effectively accommodate the consolidated class categories using optimization techniques, like e.g., *early stopping* and *learning rate scheduler*. Early stopping prevents overfitting and ensures the *supermodel* maintains strong generalization capabilities on unseen data, by monitoring the model's per-

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.

TABLE I
CIC-IDS 2017 DATASET (REFERRED TO AS DATASET-1 IN OUR SYSTEM MODEL)

No	Attacks	Attacks Found	Original Records	Augmented Records
1	BENIGN	BENIGN	2,096,484	Not Augmented
2	Brute Force	FTP-Patator, SSH-Patator	9,152	12,000
3	DoS	GoldenEye, Hulk, Slowhttptest, Slowloris, Heartbleed, Bot	195,712	205,000
4	Web Attacks	Brute Force, SQL Injection, XSS	2,143	10,000
5	Infiltration	Infiltration	36	1,500
6	Port Scan	Port Scan	90,819	100,000
7	DDoS	DDoS	128,016	150,000

TABLE II
NSL-KDD 2009 DATASET (REFERRED TO AS DATASET-2 IN OUR SYSTEM MODEL)

No	Labels	Training Set	Testing Set	Augmented Records
1	Normal	67,343	9,711	Not Augmented
2	DoS	45,927	7,460	Not Augmented
3	Probe	11,656	2,421	Not Augmented
4	R2L	995	2,885	8,000
5	U2R	52	67	5000

formance on a validation dataset throughout training, and stopping the training as soon as the performance on the validation set deteriorates. A learning rate scheduler dynamically adjusts the step size during optimization, starting with a higher learning rate to accelerate convergence, and decreasing it as the model approaches optimal parameters, to fine-tune and avoid overshooting.

IV. VALIDATION RESULTS

To test the effectiveness of the proposed methodology, we emulate the heterogeneity of datasets collected from different edge nodes by adopting two widely used datasets: the CIC-IDS 2017 [6] as the dataset-1 at the "Edge Node A," and the NSL-KDD 2009 [7] as the dataset-2 at the "Edge Node B." These comprehensive datasets have been widely used in previous studies focused in IDS for edge computing in IoT networks [8]–[10], because they offer 79 and 42 distinct features, forming a rich foundation for an in-depth analysis. The selection of these datasets is motivated by their widespread recognition within network security and intrusion detection, as well as their varied characteristics that collectively enable comprehensive model evaluation and validation. Details about the attacks covered by these datasets are illustrated in Table I and Table II, where DoS stands for Denial of Service, DDoS for Distributed DoS, R2L for Root to Local and U2R for User to Root attacks. The attack types represented in these datasets are commonly encountered in distributed networks where edge devices process data closer to the source, making it critical to detect and mitigate such threats in real-time. The last columns of Table I and Table II, labeled "Augmented Records," report the number of records for each attack class augmented by using SMOTE to address class imbalance and improve the model's performance in detecting minority class intrusions [4]. Both datasets present significant class imbalance issues, e.g, the BENIGN class in the CIC-IDS 2017 dataset has over 2 million records, whereas classes such as Web Attack and Infiltration have only 2,143 and 36 records, respectively (Table I). The NSL-KDD 2009 dataset (Table II) demonstrates a pronounced imbalance in the R2L and U2R classes. Without addressing this class imbalance problem, the model would likely struggle to learn effective patterns for these minority classes, further reducing the efficacy of the IDS.

To assess the ML model's efficacy, 80% of the data was allocated for training purposes, while the remaining 20% was reserved for rigorous testing. This deliberate partitioning ensures a representative distribution of instances across both sets, maintaining a proportional representation of different classes in both the training and testing phases. As detailed in [4], we eliminate duplicate samples to ensure the dataset's integrity by preventing redundancy and bias, which could otherwise skew ML model outcomes. We then perform label mapping, i.e., the conversion of categorical labels into numerical values, to facilitate accurate processing and analysis by ML models. These preprocessing steps lay a strong foundation for developing robust and unbiased IDS frameworks. The experiments make use of the most widely used BERT-Base architecture for feature extraction at both edge nodes. BERT-Base consists of 12 transformer layers, 12 attention heads, and a hidden size of 768, making it computationally efficient while still capturing meaningful contextual representations. The experiments are conducted on the high-performance computing platform, optimized for handling large datasets and complex tasks, shown in Figure 2, consisting of an NVIDIA A40 GPU to create two virtual machines for the "Edge Node A" and "Edge Node B," respectively.

The presented evaluation is based on five well-established metrics: Confusion Matrix, Accuracy, Precision, Recall & F1-Score [4], that are shown in Figure 3 and 4, with multiclass confusion matrices presented in Figure 5 and 6.

Across both datasets, the BERT-Base model demonstrates high accuracy, particularly in classifying BENIGN traffic and common attack types like DDoS and Port Scan. However, it shows variability in detecting more complex attacks, such as web-based attacks and infiltration on the CIC-IDS 2017 dataset, and faces challenges with the R2L and U2R categories on the NSL-KDD 2009 dataset. Despite high precision, indicating few false positives, BERT-Base struggles with false negatives, especially in scenarios involving less frequent attack types. When applied to the NSL-KDD 2009 dataset, BERT-Base performs slightly better overall, particularly in terms

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.

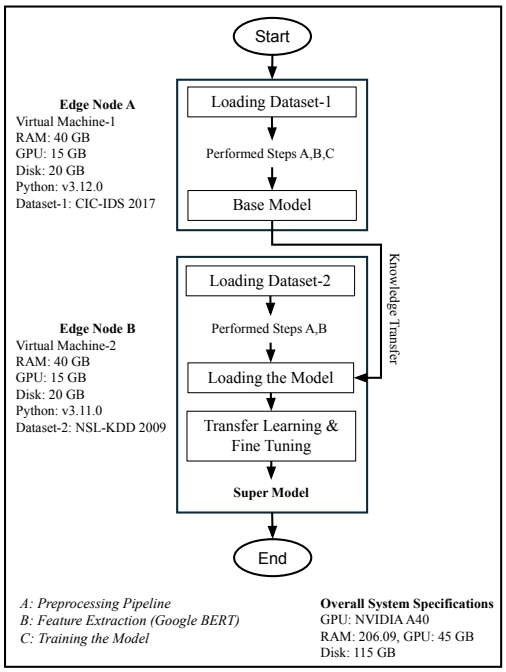


Fig. 2. Distributed System Architecture using Transfer Learning Approach

of accuracy and precision, due to its better balance between model complexity and the dataset's characteristics.

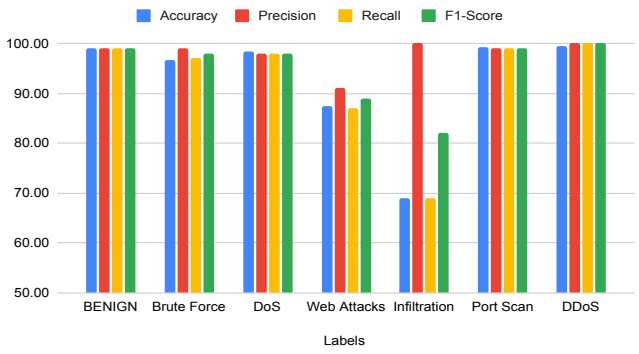


Fig. 3. Performance Evaluation of BERT-Base *supermodel* on the CIC-IDS 2017 Dataset

The effectiveness of the proposed model is demonstrated by the comparison in Table III of its accuracy, precision, recall, and F1-score against several state-of-the-art classifiers evaluated on the same datasets.

For the CIC-IDS 2017 dataset, our model achieves an accuracy of 99.39%, higher than the other models. The closest competitor, a CNN-based approach [11], achieves an accuracy of 98.79%. Our model excels in terms of precision, recall, and F1-score, which are at 99.39%, surpassing other methods.

On the NSL-KDD 2009 dataset, our model also shows superior performance, with an accuracy of 99.33%, compared to other methods like the Ensemble Learning-Deep Neural

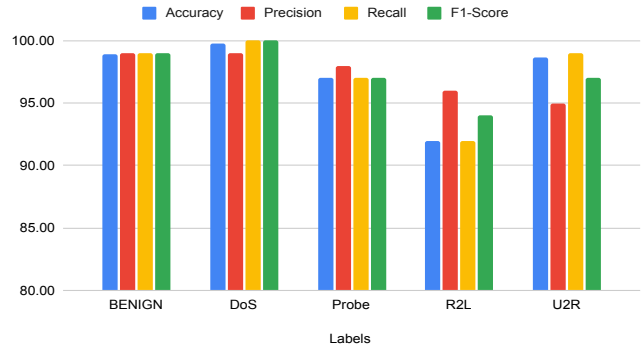


Fig. 4. Performance Evaluation of BERT-Base *supermodel* on the NSL-KDD 2009 Dataset

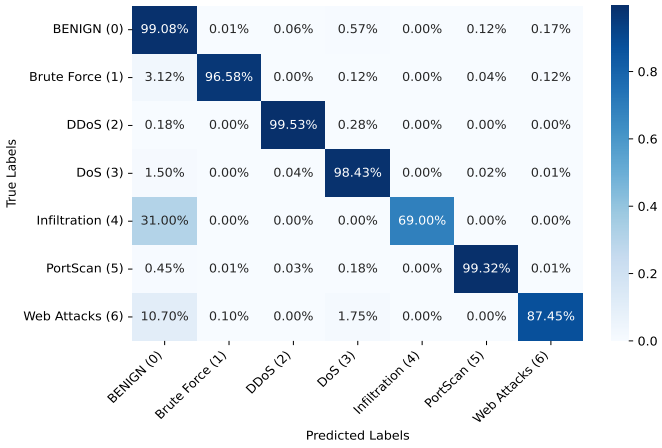


Fig. 5. Confusion Matrix of Multiclass Classification using BERT-Base *supermodel* on CIC-IDS 2017 Dataset.

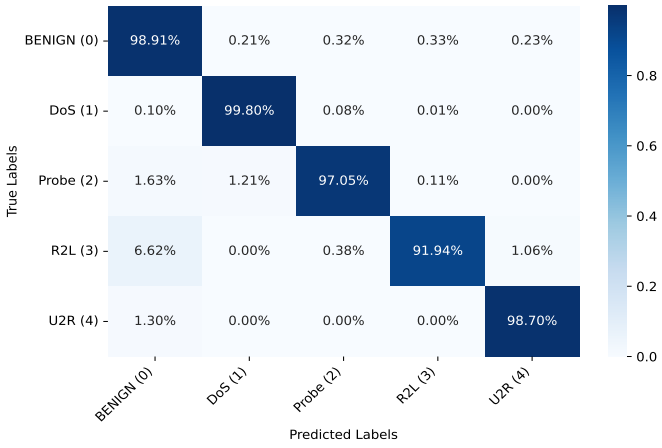


Fig. 6. Confusion Matrix of Multiclass Classification using BERT-Base *supermodel* on NSL-KDD 2009 Dataset.

Network (DNN) [12], which achieved 98.90%. Our model

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.

TABLE III
PERFORMANCE EVALUATION OF TL-BASED SUPERMODEL WITH STATE-OF-THE-ART CLASSIFIERS USING CIC-IDS 2017 AND NSL-KDD 2009 DATASETS

Dataset	Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CIC-IDS 2017	Proposed	BERT and MLP	**99.39**	**99.39**	**99.39**	**99.99**
	[11]	Convolutional Neural Network (CNN)	98.79	98.80	98.77	98.77
	[13]	Self-Organizing Map (SOM)	80.00	77.40	81.80	80.00
	[14]	CNN-Long Short Term Memory (LSTM)	98.56	98.50	99.00	99.25
	[14]	CNN-Recurrent Neural Networks (RNN)	97.24	97.50	97.25	96.50
	[14]	CNN-Gated Recurrent Unit (GRU)	96.12	96.25	96.25	95.50
NSL-KDD 2009	Proposed	BERT and MLP	**99.33**	**99.34**	**99.33**	**99.33**
	[15]	Decision Tree	97.40	97.40	97.10	97.20
	[13]	SOM	91.00	91.00	91.40	91.00
	[16]	Autoencoder	79.74	82.22	79.74	76.47
	[12]	Ensemble Learning-Deep Neural Network (DNN)	98.90	99.90	98.71	99.31
	[17]	Non-Symmetric Deep Autoencoder (NDAE)	85.42	100.00	85.42	87.37

maintains a balanced accuracy, recall, and F1-score of 99.33%. These results underscore the robustness of the proposed *supermodel*, which maintains high precision, recall, and F1-score, even in heterogeneous data distributions.

V. CONCLUSION

We presented a groundbreaking approach to fortify IDS by seamlessly integrating TL into edge computing. Harnessing the power of BERT for semantic feature extraction, and SMOTE to address class imbalance, we train an MLP by the enriched feature data using a TL strategy. A single unified *supermodel* is developed, that shows exceptional adaptability and proficiency across various intrusion scenarios, thanks to the heterogeneous datasets collected from different edge nodes. For future work, we plan to enhance our system by incorporating IoT-specific datasets to better reflect real-world traffic and attack scenarios. We also aim to deploy and evaluate the model on actual edge devices (e.g., Raspberry Pi) to assess inference time, energy use, and resource consumption. To improve trust and transparency, we will integrate explainable AI techniques like SHAP for interpreting model decisions. Additionally, we will explore continual or federated learning to adapt the model to evolving IoT threats.

ACKNOWLEDGMENT

This work has been partially funded by the European Union - NextGenerationEU under the Italian Ministry of University and Research (MUR) through the National Innovation Ecosystem grant (ECS00000041 - VITALITY - CUP E13C22001060006), the partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART" - CUP E83C22004640001) and the project ISP5G+ (CUP D33C22001300002), which is part of the SERICS program (PE00000014).

REFERENCES

[1] B. Panchali, "Edge computing- background and overview," in *2018 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, 2018, pp. 580–582.

[2] T. Sapra, Shubhama, and S. Meena, "A novel approach to handle imbalanced dataset in machine learning," in *2023 IEEE 8th International Conference for Convergence in Technology (I2CT)*, 2023, pp. 1–5.

[3] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity," *Applied Sciences*, vol. 13, no. 13, p. 7507, Jun 2023.

[4] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli, "Empowering network security: Bert transformer learning approach and mlp for intrusion detection in imbalanced network traffic," *IEEE Access*, vol. 12, pp. 137 618–137 633, 2024.

[5] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[6] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." *ICISSp*, vol. 1, pp. 108–116, 2018.

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, 2009.

[8] A. Kumar, S. Kumar, V. Kumar, A. Kumari, A. Saini, and S. Gupta, "Edge computing based ids detecting threats using machine learning and pycaret," in *2023 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, 2023.

[9] P. Mahadevappa, R. K. Murugesan, R. Al-amri, R. Thabit, A. H. Al-Ghushami, and G. Alkawsi, "A secure edge computing model using machine learning and ids to detect and isolate intruders," *MethodsX*, vol. 12, p. 102597, 2024.

[10] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," *Computers and Electrical Engineering*, vol. 100, p. 107901, 2022.

[11] M. A. Siddiqi and W. Pak, "Tier-based optimization for synthesized network intrusion detection system," *IEEE Access*, vol. 10, 2022.

[12] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for iot network using ensemble-learning-based deep neural network," *IEEE Internet of Things Journal*, vol. 10, no. 13, 2023.

[13] J. Ables, T. Kirby, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, and M. Seale, "Creating an explainable intrusion detection system using self organizing maps," in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022, pp. 404–412.

[14] 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*, 2023.

[15] R.-F. Hong, S.-C. Horng, and S.-S. Lin, "Machine learning in cyber security analytics using nsl-kdd dataset," in *2021 International Conference on Technologies and Applications of Artificial Intelligence (TAAI)*, 2021, pp. 260–265.

[16] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A deep learning approach for network intrusion detection based on nsl-kdd dataset," in *2019 IEEE 13th International Conference on Anti-counterfeiting, Security, and Identification (ASID)*, 2019, pp. 41–45.

[17] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41–50, 2018.

Authorized licensed use limited to: University of L'Aquila. Downloaded on March 24,2026 at 14:27:44 UTC from IEEE Xplore. Restrictions apply.

Paper J2

Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios

Zeesshan Ali, Walter Tiberti, Andrea Marotta and Dajana Cassioli

Published in
IEEE Internet of Things (IoT)

Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios

Zeeshan Ali, *Graduate Student Member, IEEE*, Walter Tiberti, *Member, IEEE*, Andrea Marotta, *Member, IEEE*, and Dajana Cassioli, *Senior Member, IEEE*

***Abstract*—The rapidly evolving landscape of cyber threats poses significant challenges to network security, particularly in decentralized environments, such as *edge computing*. This article proposes an enhanced intrusion detection system (IDS) architecture that integrates transfer learning (TL) to create a unified *supermodel*, enabling adaptability and scalability for detecting emerging threats across diverse datasets (DS). The key contributions of this study include: 1) leveraging bidirectional encoder representations from transformer (BERT)-based feature extraction to enhance the semantic understanding of intrusion patterns; 2) employing an multilayer perceptron (MLP) classifier refined through TL to improve classification performance; 3) addressing class imbalance using synthetic minority over-sampling technique (SMOTE); and 4) optimizing model deployment by distributing the heaviest computational tasks for the creation of the unified *supermodel* across the edge nodes with the available capacity, thereby reducing latency and enabling accurate real-time threat detection by resource constrained IoT devices. The proposed TL-enabled *supermodel* is periodically updated and shared with the IoT devices, ensuring robust and adaptive security mechanisms without the need for extensive local training. Our experimental evaluation on CIC-IDS 2017 and NSL-KDD 2009 DS demonstrates the effectiveness of the approach, achieving 99% accuracy, precision, recall, and F1-score. Our results highlight the scalability, efficiency, and real-world applicability of our IDS framework, reinforcing its role in fortifying network security within highly dynamic cyber threat landscapes.**

***Index Terms*—Bidirectional encoder representations from transformers (BERTs), cybersecurity, edge computing, imbalanced dataset, intrusion detection system (IDS), multilayer perceptron (MLP), transfer learning (TL).**

I. Introduction

THE INTRODUCTION of edge computing marks a significant departure from traditional cloud-based architectures, offering a more responsive and efficient approach to data processing and analysis. This paradigm shift is driven by the need for real-time decision-making and the proliferation of devices generating vast amounts of data at the network's periphery. Edge computing capitalizes on the strategic placement of computing resources at the network's edge, fostering a distributed model well-suited for applications requiring instantaneous responses. In the context of IoT, smart cities, and autonomous systems, the integration of edge computing enhances the overall system performance by enabling data processing at the source. This approach reduces latency, optimizes bandwidth usage, and saves energy by minimizing the transmission of large volumes of raw data to centralized servers [1]. As a consequence, edge computing is particularly beneficial for scenarios where swift and localized data analysis is crucial, such as real-time monitoring of sensor data or rapid decision-making in autonomous vehicles.

The dynamic nature of edge environments, characterized by many interconnected devices and varying network conditions, underscores the importance of robust security measures. Intrusion detection becomes paramount in ensuring the integrity of data flows and the confidentiality of sensitive information within these distributed ecosystems. As edge computing continues its ascendancy, striking a balance between performance optimization and security becomes a central concern for architects and developers. Intrusion detection systems (IDSs) have emerged as critical components in ensuring the integrity and availability of computer systems by detecting and responding to unauthorized activities. However, the specific challenges inherent in decentralized environments, such as limited computational resources, dynamic network conditions, and the diversity of connected devices [2], underscore the crucial need for innovative approaches specifically tailored to enhance the effectiveness of intrusion detection mechanisms in these scenarios.

Machine learning (ML)-based IDS leverages the power of advanced algorithms and computational intelligence to enhance security postures. Unlike rule-based systems, ML-based IDS can learn and adapt from large datasets (DS), making them well-suited to identify novel and previously unseen threats. These systems employ various ML techniques, such as supervised, unsupervised, and deep learning (DL), to analyze network traffic, system logs, and other relevant data sources. By doing so, they can autonomously detect unusual patterns, anomalies, or suspicious behaviors that might indicate a cyber intrusion.

Transfer learning (TL) is a transformative approach in DL that focuses on reusing knowledge acquired from one domain or task to improve performance on a different but

Received 9 July 2025; accepted 4 August 2025. Date of publication 11 August 2025; date of current version 24 October 2025. This work was supported in part by the European Union—NextGenerationEU through the Italia Ministry of University and Research National Innovation Ecosystem under Grant ECS00000041 and through VITALITY under Grant CUP E13C22001060006; in part by the Partnership on "Telecommunications of the Future" under Grant PE00000001 and through the Program "RESTART" under Grant CUP E83C22004640001; and in part by the Project ISP5G+—CUP under Grant D33C22001300002, which is part of the SERICS Program under Grant PE00000014. *(Corresponding author: Zeeshan Ali.)*

The authors are with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, 67100 L'Aquila, Italy (e-mail: zeeshan.ali@graduate.univaq.it; walter.tiberti@univaq.it; andrea.marotta@univaq.it; dajana.cassioli@univaq.it).

Digital Object Identifier 10.1109/JIOT.2025.3597892

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

related task. Instead of training a model from scratch, TL leverages pretrained models, which have already learned complex patterns from large DS, and adapts them to new tasks through fine-tuning. This approach accelerates the training process and reduces reliance on extensive labeled data, making it highly effective in scenarios with limited resources or data availability. TL has become a cornerstone in natural language processing (NLP), enabling highly accurate models with significantly reduced computational costs.

Recent advancements in large-language models (LLMs), such as bidirectional encoder representations from transformers (BERTs), have revolutionized NLP by enabling deep contextual understanding of complex data patterns. BERT is a powerful NLP model developed by Google in 2018 that has achieved state-of-the-art performance on various NLP tasks, such as text classification, question-answering, and capturing contextual relationships. Leveraging this capability, our work introduces a novel application for intrusion detection in edge computing environments. Although BERT and similar models have been used primarily in textual and semantic tasks, our approach is among the first to harness their powerful feature extraction capabilities to enhance network intrusion detection. By applying BERT to structured network traffic data, we uncover nuanced behavioral patterns that traditional models may overlook. This innovative integration not only boosts detection performance but also brings the strength of advanced DL to resource-constrained systems, marking a significant step forward in intelligent, real-time threat mitigation.

Network threats often exhibit sophisticated and evolving behaviors; thus, IDS should be able to capture intricate, nonlinear patterns in network flow data [3]. This is made possible by a particular artificial neural network (ANN) architecture that employs the nonlinear activation functions, i.e., the multilayer perceptron (MLP). Compared to linear classifiers, MLP exhibits a remarkable ability to manage high-dimensional data, a crucial advantage in network traffic analysis.

Edge computing networks, which involve IoT networks of resource-constrained devices, cannot rely on computationally intensive models. At the same time, highly dynamic and heterogeneous environments demand scalable and adaptable solutions to address evolving cyber threats. Additionally, the distributed nature of data collection at edge nodes results in diverse and nonuniform DS, complicating the task of creating a unified and comprehensive training model. Ensuring the generalizability of IDS models across varying DS and intrusion scenarios remains a critical challenge.

The above challenges highlight the pressing need for innovative approaches to develop scalable, adaptable, and resource-efficient IDS tailored to edge computing environments.

Our study evaluates a pioneering application of TL within IDS tailored for edge computing environments. In edge computing, where diverse devices collaborate and share data, our approach capitalizes on this distributed information to create a robust, adaptive intrusion detection model. The novelty of our approach lies in aggregating data from multiple devices to construct a comprehensive and diverse dataset. This combined dataset trains a unified *supermodel* synthesizing insights across the different sources. Once trained, the *supermodel* is deployed to IoT devices, continuously providing them with updated data for further model's refinement and adaptation over time.

An alternative, valuable approach to enhance adaptability is federated learning (FL). FL allows multiple edge nodes to collaboratively train models without sharing raw data, thereby improving privacy and scalability. Potential benefits of FL in IDS include privacy-preserving adaptability, reduced communication overhead, and enhanced global threat awareness. However, FL underlies challenges, such as aggregation complexity and adversarial security risk. Instead, our proposed TL approach offers a unique advantage in enhancing the precision and effectiveness of intrusion detection by capitalizing on the collective knowledge of diverse devices to identify and respond to novel threats in real-time, while optimizing resource utilization.

The contributions of this article can be summarized as follows.

1) We extend the effectiveness of BERT-based feature extraction with synthetic minority over-sampling technique (SMOTE) to handle DS' class imbalance to improve detection accuracy.
2) We compare the performance achieved by the proposed IDS using the most widely used variants of BERT architecture, namely, BERT-Base and BERT-Large. BERT-Base, with a vector size of 768, offers a lightweight and faster alternative suitable for resource-constrained environments, whereas BERT-Large, with a vector size of 1024, captures more complex representations, potentially enhancing detection performance.
3) We introduce a novel TL—enabled IDS architecture to create a unified *supermodel* that leverages knowledge acquired from diverse edge nodes to recognize transferable patterns across various intrusion scenarios, enhancing the adaptability of the IDS to novel and evolving threats. The *supermodel* is highly compact and lightweight, with a size of only 3.10 MB, particularly suitable for deployment on resource-constrained IoT devices. This efficiency ensures real-time intrusion detection with minimal computational overhead, facilitating practical deployment in distributed environments.
4) We enhance the downstream classification performance of network traffic data, by applying an MLP, aligning with the high scalability and low latency requirements of edge computing.
5) We validate our TL–based IDS model using two widely adopted IDS DS, CIC-IDS 2017, and NSL-KDD 2009, which, although not edge-specific, encompass a wide range of attack types and network behaviors. Our experimental results demonstrate significant gains in accuracy, precision, recall, and F1-score over state-of-the-art methods, highlighting the model's adaptability across diverse intrusion scenarios and its promising applicability to distributed settings.

The subsequent sections of the document are structured in the following manner. Section II presents a comprehensive discussion of the relevant and recently published literature.

Sections III and IV explain the proposed system model. The results are shown in Section V, which also offers a comprehensive discussion of our findings and their implications while outlining potential future research directions in this domain. Section VI serves as the concluding part of the paper.

II. Related Work

IDS can be viewed as a supervised anomaly detection task and phrased as a classification challenge. Recently, several research papers focused on extracting characteristic patterns from intrusion attacks to effectively detect them from typical network traffic. These approaches span various techniques, from conventional statistical to state-of-the-art DL-based approaches.

Feature selection is a preprocessing technique in ML aimed at reducing computational costs [4]. Its primary objective is to eliminate irrelevant features while simultaneously preserving or improving the performance of IDS.

To capture intricate features within network traffic data, a hybrid approach combining convolutional neural networks (CNNs) and long–short-term memory (LSTM) networks has been proposed by Dini et al. [3]. This method primarily focuses on centralized environments and is particularly effective in handling feature complexities; however, it is not flexible enough to generalize across diverse DS and to manage class imbalances in intrusion detection tasks. LSTM-based models have also been used for time series anomaly detection [5]. However, LSTM presents some limitations in capturing long-distance dependencies and has a stringent requirement for large amounts of labeled data.

To further enhance the efficiency of feature selection, Aslahi-Shahri et al. [6] employed a novel hybrid methodology that integrates a support vector machine (SVM) with a genetic algorithm (GA) to reduce the dimensionality of the dataset. Their experiment has shown how to distill the KDD CUP 99 dataset [7], of 41 features, into a more manageable subset of 10 essential features, and demonstrated the effectiveness of their innovative technique in increasing true positive rates while minimizing false positives. Although this technique is effective in streamlining feature sets and improving IDS performance, superior performance has been demonstrated by NLP methods aiming to directly improve feature representation, rather than relying solely on dimensionality reduction [8].

A powerful NLP method for feature extraction is BERT, which does not merely select existing features but generates new semantically rich feature representations from raw data [9]. By using BERT, the ability of models to understand complex patterns in the data is enhanced, which is especially crucial for the nuanced detection of intrusions.

Another effective approach for anomaly detection based on NLP has been introduced by Mimura and Tanaka [10]. This method leverages network interface packet analysis and primarily relies on source and destination IP information. The best-performing metric for this approach was an F1-score of 98%. However, decentralized networks, like, e.g., IoT, present unique challenges for intrusion detection, including the heterogeneity of devices, constrained computational and energy resources, and the vast volume of data generated in real-time.

In decentralized edge environments, traditional feature selection methods struggle with adaptability to evolving threats and scalability. Latif et al. [11] introduced a Deep TL-based IDS for heterogeneous industrial IoT (IIoT) networks, employing a tri-layer architecture that combines CNNs, GA and ensemble techniques. The framework achieved a 100% attack detection accuracy across 14 types of cyberattacks, demonstrating high effectiveness in IIoT environments. The integration of CNNs with GA and ensemble techniques offers robust detection capabilities and high accuracy. However, the use of CNNs and GAs might impose significant computational overhead, which is challenging in edge environments with limited resources.

The detection performance of ML-based IDS is strongly influenced by the *class imbalance* of DS. A novel model designed to address the challenge of class imbalance in supervised learning has been proposed by Zhang et al. [12]. Their approach combines the SMOTE with under-sampling for clustering using a Gaussian mixture model (GMM) to tackle the class imbalance problem effectively through resampling. The model introduced by Joloudari et al. [16] achieved 99.08% accuracy over 24 imbalanced DS by effectively handling imbalanced binary DS by CNN and SMOTE.

In scenarios where labeled data is scarce or expensive to obtain, a powerful approach is the *TL*, as it allows models to generalize knowledge acquired from different sources and synthesized in high-resource domains, ready to be shared with low-resource ones [17]. Applications of TL span a wide range of fields, including NLP, computer vision, and bioinformatics, where tasks, such as object recognition, language translation, and disease diagnosis have benefited greatly from this methodology. Unlike traditional learning approaches, which often require large amounts of task-specific data and computational resources, TL leverages pretrained models or features, significantly reducing the need for extensive data collection and training from scratch. Zong et al. [13] proposed an approach for intrusion detection that can transfer knowledge from a source domain with labeled data to a target domain with different statistical properties. In another study, Alrawashdeh and Purdy [14] employed TL in cyber threat hunting by leveraging labeled data from a source domain to enhance detection performance in a different but related target domain. Alhussien et al. [18] examined the impact of backdoor attacks on multisource TL models for intrusion detection. It finds that while backdoors in TL models are feasible, they have a lesser effect than on traditional ML models. Cheng et al. [19] introduced an efficient federated TL (FTL) framework for mobile-edge computing (MEC) intrusion detection, combining FL for privacy preservation and TL for enhanced training efficiency. Zhang et al. [15] proposed a privacy-preserving anomaly-based IDS for IIoT networks, employing FL and instance-based TL to address nonindependent and identically distributed (Non-IID) data challenges. A rank aggregation algorithm with weighted voting

is proposed, resulting in significant performance improvements, with 95.97% and 73.70% accuracy for AdaBoost and Random Forest. Yutao et al. [20] presented a visual geometry group (VGG)-RED approach for IoT intrusion detection, employing TL to migrate feature weights from the IoT intrusion detection DS. Results show a 2% accuracy improvement and 7%–13% reduction in time, demonstrating the model's efficacy in enhancing malicious attack detection in IoT environments while optimizing computational resources. Heterogeneous IoT environments are also targeted by Mehedi et al. [21], who proposed a deep TL-based IDS featuring effective attribute selection and a robust residual network (ResNet) model. The system outperforms existing methods, demonstrating enhanced accuracy, efficiency, and dependability. Saheed et al. [22] proposed the IoT-Defender framework, which combines a modified GA (MGA) for feature selection and a fine-tuned LSTM network within an edge computing environment for IoT intrusion detection. Evaluated on BoT-IoT [23], UNSW-NB15 [24], and N-BaIoT [25] DS using a Raspberry Pi, the model achieved 99.41% accuracy and 99.78% detection rate. Masum [26] proposed a network intrusion detection system (NIDS) classification framework using TL from the VGG-16 model, comparing its performance with VGG-19 [27], mobile network (MobileNet) [28], ResNet-50, and Inception-V3. By extracting features from ImageNet [29] pretrained weights and applying a deep neural network (DNN) for classification on the NSL-KDD 2009 dataset, the framework achieved an accuracy of 81.77% for binary and 78.39% for multiclass classification. Saheed and Misra [30] presented an explainable, privacy-preserving DNN framework for anomaly detection in cyber–physical systems (CPSs) enabled IoT networks, integrating shapley additive explanations (SHAPs) for interpretability. Tested on Edge-IIoTset [31] and X-IIoTID [32] DS, the model achieved near-perfect accuracy, demonstrating its effectiveness in intrusion detection while reducing false positives.

Compared to existing state-of-the-art IDS frameworks, including those in [22], [30], and [26], the originality of our proposed TL-enabled *supermodel* resides on the integration of BERT-based semantic feature extraction and a scalable edge-oriented deployment strategy. While prior works employ traditional deep models like ResNet, LSTM, and VGG for intrusion detection, our system introduces a unified architecture that leverages language-based embeddings to capture deeper contextual patterns of network intrusions.

The related work analysis has highlighted several valuable strategies for feature selection and class imbalance mitigation, often relying on conventional and DL techniques. In our previous study, we proposed an IDS framework leveraging SMOTE to handle class imbalance through oversampling and undersampling, a BERT transformer-based model for tokenization and feature extraction, and an MLP for classification, achieving robust results on individual DS [8]. We demonstrated that the BERT model excels in capturing nuanced contextual information and robust feature representations from network traffic data.

Building on this foundation, this article extends our prior work by introducing a TL-based paradigm for IDS tailored to the unique challenges of edge computing employed in IoT networks. The integration of TL to our framework enables the creation of a unified *supermodel* that generalizes knowledge across multiple DS, improving scalability and adaptability to evolving intrusion patterns across multiple DS. Unlike CNN-LSTM hybrids, our methodology prioritizes generalization and feature diversity, ensuring enhanced accuracy and resilience in detecting novel threats across different scenarios captured in the DS. This advancement ensures enhanced attack prediction capabilities while addressing the dynamic and decentralized nature of edge environments for IoT.

In summary, while several studies have explored SVM, GA, CNN, GMM and LSTM for feature extraction and intrusion classification, there remains a significant gap, as discussed in Table I. This study aims to bridge this gap by introducing a novel IDS architecture that integrates TL into a unified *supermodel*. The proposed work leverages the BERT-based transformer model for the feature extraction to improve the semantic understanding of intrusion patterns and employs a DL-based MLP classifier refined through TL to enhance classification performance. Hence, it offers a more comprehensive solution to the challenges identified in the literature.

III. System Model

The system model is shown in Fig. 1, along with the network architecture for the intelligent edge deployment of the proposed TL-based IDS in IoT scenarios.

The proposed architecture includes three layers.

1) The lower layer consists of multiple IoT networks, each serving a specific application domain, such as medical applications, Industry 5.0 and automotive; the IoT devices do not have sufficient resources to train their own ML model for intrusion detection and rely on the "Edge Layer" to offload this and other heavy computational tasks.
2) Lying on top of the IoT networks, the "Edge Layer" is made up of a multitude of powerful edge devices in charge of running heavy computational tasks on behalf of IoT devices belonging to the served IoT network; among these heavy tasks, edge nodes run the preprocessing of the DS acquired from the served IoT networks to train its own model, referred to as *standalone model (SAM)*. Every edge node is independent of the others in this training task, consisting of the same pipeline for all nodes to ensure consistency and compatibility of "local SAMs." The edge nodes are all connected through Internet and can apply the TL to improve their IDS model by acquiring the embeddings from different DS shared by other edge nodes and creating the *TL supermodel*. The details are provided in Section IV.
3) The top layer represents the "Data Acquisition Layer," responsible for collecting the relevant observation data and composing the DS to be fed to the "Edge Layer" for both the SAM and the *TL supermodel*.

TABLE I
COMPARISON OF EXISTING APPROACHES AND PROPOSED WORK

Author(s)	Approach	Addressed Challenge(s)	Limitation(s)	Proposed Work's Advantage
[3]	CNN-LSTM Hybrid for Feature Extraction	Feature extraction in centralized environments	Limited Generalizability	Our work generalizes across multiple datasets and enhances flexibility.
[6]	SVM with Genetic Algorithm for Feature Selection	Dimensionality reduction and feature selection	Does not leverage NLP techniques for feature representation	Our work employs BERT for richer, semantic feature extraction.
[10]	Packet Analysis for Intrusion Detection	Anomaly detection in network traffic	Focused on source and destination IPs, limited to network packet analysis	Our approach uses BERT to capture more complex, semantic intrusion patterns.
[11]	Deep TL-based IDS for Industrial IoT	High accuracy in IIoT networks	Computationally heavy for edge environments	Our approach optimizes computational tasks by distributing across edge nodes.
[12]	SMOTE with Under-sampling for Class Imbalance	Class imbalance in datasets	Might not address all forms of class imbalance effectively	We combine SMOTE with TL to address class imbalance more robustly and efficiently.
[13]	TL-based IDS for Knowledge Transfer Across Domains	Knowledge transfer for intrusion detection	Limited generalization across diverse datasets	Our unified *supermodel* adapts to multiple datasets and intrusion patterns.
[14]	Transfer Learning for Cyber Threat Hunting	Enhancing detection performance with limited target domain data	Focuses only on transferring knowledge from related domains	Our approach broadens the scope, creating a more adaptable *supermodel* across datasets.
[15]	Privacy-Preserving TL for IIoT	Privacy in Federated Learning with TL	Focuses on privacy but does not directly address performance issues	Our work balances with performance and enhancing detection capabilities.
Proposed	TL-based IDS with BERT Feature Extraction and MLP Classifier	Scalable, adaptable IDS for edge computing in IoT environments	Limited by previous methods' inability to handle evolving threats in decentralized environments	Enhanced scalability adaptability and feature richness through BERT and TL integration.

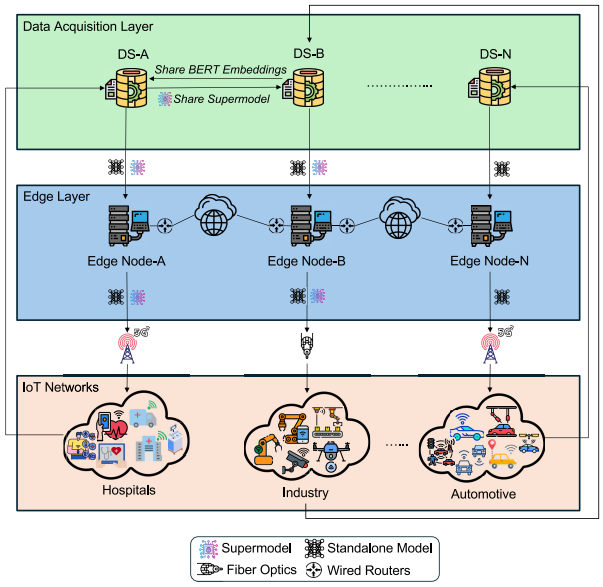


Fig. 1. System model and network architecture for intelligent edge deployment of the proposed TL-based IDS.

The novelty of our approach lies in the introduction of TL in the Edge Layer: whenever a trained model is available at one node, it is advertised among all other edge nodes, that may acquire such a trained model, greatly benefiting from this knowledge to extend its own model by creating a *supermodel* through TL. Hence, as explained in the following, our method leverages the knowledge gained from an initial dataset, i.e., the base model (BM), built on one edge node to enhance performance by acquiring new knowledge from another dataset available at a different edge node. Certainly, the presented methodology can be applied to any number of nodes participating in the TL process.

IV. TRANSFER LEARNING METHODOLOGY

The principle diagram depicted in Fig. 2 illustrates a collaborative edge-based TL framework involving two edge nodes, "edge node A (EN-A)" and "edge node B (EN-B)," designed to support both SAM training and TL-based *supermodel* for their IoT devices, as described in the following.

A. Standalone Model's Training

Every edge node is able to train its own SAM by applying the procedure summarized in Algorithm 1. This includes dataset preprocessing (Step Ⓟ) and feature extraction using Google BERT (Step Ⓖ), followed by the classification using an MLP model. Although these steps have been extensively described in [8], for the convenience of the reader, we summarize the main elements below.

With reference to the EN-B box in the lower part of Fig. 2, the data preprocessing pipeline (Step Ⓟ) performs tasks, such as SMOTE, duplication removal, and label mapping. Then, to ensure that the input representations capture contextual semantic information and to reduce the risk of overfitting to specific dataset characteristics, our methodology uses Google BERT. In the present work, we extend the analysis we proposed in [8] by testing two variants of BERT.

1) *BERT-Large-Uncased* consisting of 24 layers, 16 attention heads, and a hidden size of 1024.
2) *BERT-Base-Uncased* with its 12 layers, 12 attention heads, and a hidden size of 768.

Input sequences are tokenized with a maximum length of 128, applying truncation and padding. Feature extraction is performed using the CLS token representations from the last

Algorithm 1 Algorithm for the Creation of the SAM as in [8]

procedure MAIN(Dataset-A (DS-A))
  Input:
    DS-A
  Output:
    Base model **BM**: A classifier capable of making predictions on DS-A.
  Step 1: Load the DS-A and perform the *Dataset Preprocessing*
  Step 2: Tokenization and Feature Extraction using Google *BERT Base Transformer*
  Step 3: Build, Train and Test *MLP Classifier*:
    a. Training the MLP classifier on the BERT extracted train features of the DS-A.
    b. Base model **BM** is created and tested on the DS-A test features.
end procedure

hidden layer, processed in batches of 100 samples to optimize memory usage.

Our MLP model is configured with a single hidden layer containing 100 neurons, a maximum iteration limit of 200, and a fixed random state of 42 to ensure consistent results. Finally, the MLP classifier applies multiple regularization techniques to the extracted features. We incorporate dropout layers with a rate of 0.5 to randomly deactivate neurons during training, preventing the model from becoming overly reliant on specific patterns. Additionally, early stopping with a patience value of 10 is employed to monitor validation performance and halt training when overfitting is detected, ensuring that the model learns meaningful representations rather than memorizing the training data.

Furthermore, L2 regularization (also known as weight decay) is applied to the MLP classifier to help prevent overfitting. This technique works by adding a penalty term to the loss function that is proportional to the sum of the squared values of the model's weights. In simple terms, the model is discouraged from assigning large values to the weights during training, which helps reduce the complexity of the learned patterns. As shown in (1), the total loss includes both the original classification loss and the L2 penalty term. Here $L_{original}$ is the typical loss, w_i are the individual weights of the model, and α is a hyperparameter that controls how strongly the penalty is applied

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{original}} + \alpha \sum_i w_i^2. \tag{1}$$

In our case, we set $\alpha = 0.0001$, which means the regularization is relatively mild but still sufficient to push the model toward smaller weights. This helps the model generalize better by reducing its sensitivity to noise or outliers in the training data. In effect, L2 regularization acts like a "weight smoother," promoting simpler models that are more likely to perform well on new, unseen data.

The SAM can be directly shared with the IoT network(s) connected to the edge node for intrusion detection or may be used as the BM for the creation of the *supermodel* as explained in Section IV-B.

Algorithm 2 Algorithm for TL From *BM-X* to *Supermodel SM-Y* for Two Edge Nodes **X** and **Y**

procedure MAIN(BM-X, Dataset-Y (DS-Y))
  Input:
    BM-X: Load the BM created at the edge node **X**
    DS-Y: Acquiring DS-Y at the edge node **Y**
  Output:
    supermodel **SM-Y**: A classifier capable of making predictions on both datasets **X** and **Y**.
  Step 1: Load the DS-Y and perform the *Dataset Preprocessing* Ⓟ
  Step 2: Tokenization and Feature Extraction using Google *BERT Base Transformer* Ⓖ
  Step 3: BM *Knowledge Transfer* Ⓚ using TL and fine-tuning:
    a. The BM knowledge from **BM-X** is transferred using the proposed TL approach.
    b. A new classification head Ⓗ is added to handle new classes in DS-Y, followed by fine-tuning.
    c. The *supermodel* **SM-Y**, enhanced via TL & fine-tuning Ⓣ, is created and tested Ⓢ.
end procedure

B. Supermodel's Creation

The procedure for the *supermodel*'s creation is summarized in Algorithm 2 and is shown in the upper box of the principle diagram of Fig. 2, the one related to EN-A. The *supermodel*'s creation relies on a BM, which is exactly the SAM generated at the edge node that hosts the TL, for instance the EN-A, applying the procedure described in Section IV-A. This SAM can be either used as a BM for the TL or deployed, through the transfer gate (TG) indicated in Fig. 2, to the IoT devices served by EN-A as an IDS based solely on the knowledge acquired by processing the DS-A. Let's assume, without any loss of generality, that EN-B aims to participate in the TL, thus it replies positively to the EN-A advertisement. EN-B has already trained its own SAM using DS-B and has stored the corresponding Google BERT embeddings obtained at Step Ⓖ. These Google BERT embeddings are then shared with EN-A via the receiving gate (RG) shown in Fig. 2 to create the *supermodel* and enhance the IDS performance through TL.

Once the TL engine in EN-A has acquired the BM-A from the BM gate and the Google BERT embeddings from the RG interfaced to EN-B, as indicated in Fig. 2, knowledge transfer is performed (Step Ⓚ), a new classification head is introduced (Step Ⓗ), and fine-tuning is applied (Step Ⓣ) to build the *supermodel* (Step Ⓢ) based on both DS-A and DS-B. Next, this *supermodel* is shared through the sending gate (SG) and received by EN-B via its RG. Now, the *supermodel* is ready to be deployed on EN-B's and EN-A's IoT devices through their own TG.

The challenge behind the TL approach is to load the pretrained *BM* on DS-B while preserving the knowledge

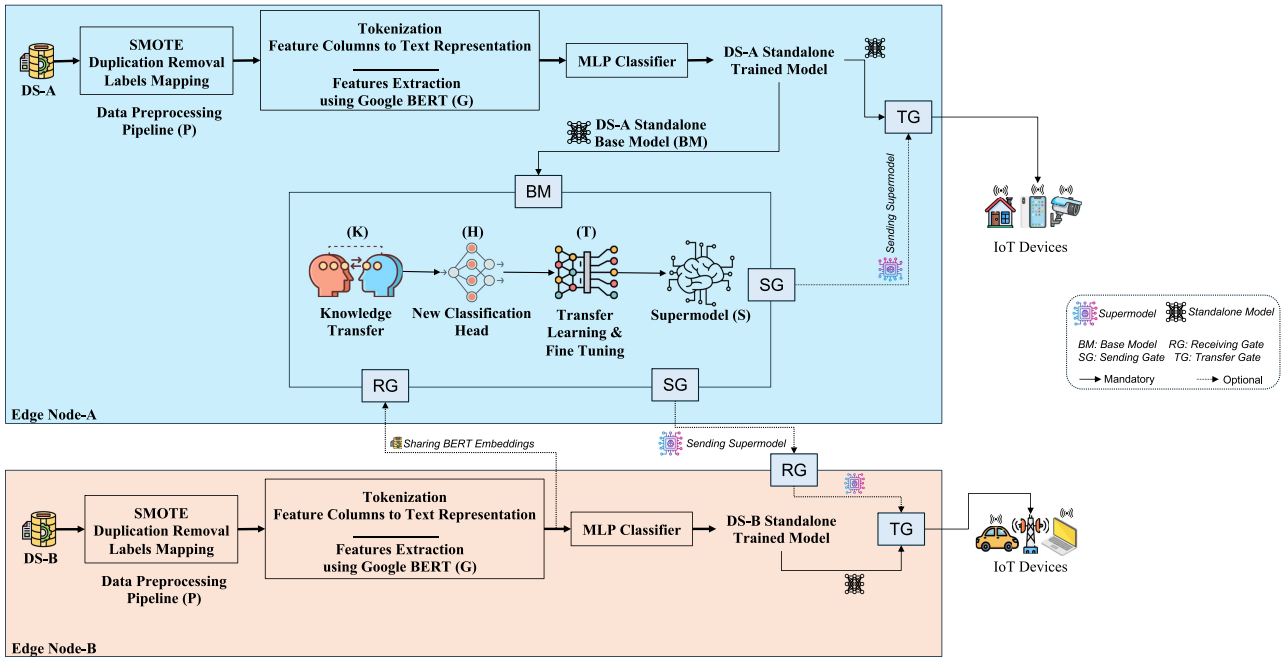


Fig. 2. Principle diagram of the proposed TL methodology for IDS in a decentralized IoT environment.

gained from its initial training without incurring the well-known "catastrophic forgetting" problem: when models are trained on new data, there is a risk they might overwrite or "forget" previously learned information [33]. To combat this issue, we employ a strategy designed on purpose, i.e., the "replay" technique, which operates by reintroducing a subset of DS-A during training on DS-B. This ensures the *supermodel* retains the knowledge from DS-A while learning from the new one.

Our practical implementation starts by unfreezing the *BM* top layers and loading the TensorFlow's StringLookup layer within the MLP, which is responsible for mapping string labels (i.e., categorical values) to integer encodings. Commonly integrated into preprocessing pipelines for ML models, the StringLookup layer offers a seamless approach to managing categorical data in a numerical format. By leveraging this layer, the model can effectively process and learn from categorical information, such as class labels, thereby improving compatibility with neural network architectures. We introduce the TensorFlow's StringLookup layer to cope with errors or misinterpretations in the model predictions due to "out-of-vocabulary (OOV)" tokens, i.e., string class labels not seen during the BM training phase. Actually, the StringLookup layer allows the specification of a unique OOV token, which is an integer representation assigned to any string not found in the DS-A. Essentially, all unseen labels are mapped to this OOV token. The DS-B features are loaded, combined with a subset (20%) of the DS-A, and then scaled. To enhance the *supermodel* efficiency, right after merging the OOV tokens from both DS, we update the classification head to effectively accommodate the consolidated class categories.

To avoid potential overfitting that can result from unfreezing the BM's top layers, we employ optimization techniques, such as *early stopping* and *learning rate scheduler* to enhance the generalizability of the model. Early stopping is a regularization method that monitors the model's performance on a validation dataset throughout training. If performance in the validation set deteriorates, training is halted early to prevent overfitting, ensuring that the *supermodel* maintains strong generalization capabilities on unseen data. *Learning rate scheduler* dynamically adjusts the step size during optimization. Initially, a higher learning rate accelerates convergence, and as the model approaches optimal parameters, the learning rate is decreased to fine-tune and avoid overfitting. Our experimental results (Section V) confirm that these strategies effectively mitigate overfitting, enabling the IDS to maintain high accuracy across multiple DS while demonstrating strong adaptability to real-world network traffic.

C. Computational Load and Scalability in Edge Deployments

The efficient deployment of IDSs in edge computing environments requires balancing model complexity with resource constraints. Our proposed TL-enabled IDS architecture addresses this challenge by offloading computationally intensive tasks to edge nodes, thereby reducing the overhead on resource-limited IoT end devices.

All edge nodes are able to independently train and deploy their own SAMs while optionally participating in a collaborative TL process. This approach fully benefits from the hierarchical and distributed architecture of edge computing where heavy training tasks are performed at the edge nodes and the sole *precomputed lightweight model*, be it the SAM or

TABLE II
EDGE NODES, DATASET CHARACTERISTICS, AND MODEL SIZE

Edge Nodes	Dataset Name	No of Records	Dataset Size	Trained Model Size
Edge Node A	CIC-IDS 2017 (DS-A)	2,574,984	843.60 MB	1.10 MB (Base Model)
Edge Node B	NSL-KDD 2009 (DS-B)	148,517	16.60 MB	3.10 MB (Super Model)

the *supermodel*, is transferred to IoT devices. These are only involved in the basic task of sharing relevant raw data, i.e., the DS-X, thus saving energy and resources.

This innovative approach has a twofold advantage. On one hand, it represents a novel means of *enhancing classification capabilities* by disseminating across IoT devices the knowledge gained in the central learning process, then it becomes particularly effective in distributed environments. On the other hand, the transfer of the *supermodel* is lightweight compared to the full dataset, thus it best matches the capabilities of resource-constrained devices in terms of both storage and computational capacity.

Additionally, the final *supermodel* consolidates *essential knowledge* from multiple edge nodes while discarding unnecessary parameters, optimizing storage efficiency.

As shown in Table II, the full dataset's size used to train the BM is about 1 GB, while the size of DS-B used to train the *supermodel* is 16.6 MB, which could be too large to be managed by the IoT devices. Instead, the BM size is very small, i.e., 1.1 MB, and the size of the *supermodel* shared with the N-end devices is about 3.10MB, which significantly reduces memory and computational demands in the deployment phase. Hence, the computational load is entirely at edge nodes, utilizing BERT-based feature extraction and an MLP classifier.

Concerning the scalability of the proposed approach, it is worth to highlight that the *supermodel* is periodically updated and shared across edge IoT devices, ensuring adaptability to evolving threats. Increasing the number of edge nodes allows parallel fine-tuning, enhancing detection capabilities without increasing central processing overhead. By leveraging TL, each new dataset improves the *supermodel* without requiring full retraining from scratch.

V. VALIDATION RESULTS

We validate the proposed methodology by running experiments on the distributed system architecture shown in Fig. 3, implemented on a high-performance computing platform optimized to handle large DS and complex tasks.

We utilized an NVIDIA A40 GPU shared across two virtual machines, designated as EN-A and EN-B. The BM training was conducted on EN-A and the trained model was shared with EN-B for TL and fine-tuning. This process resulted in a highly optimized *supermodel* which was then deployed to IoT devices for real-time intrusion detection, ensuring efficient and scalable threat mitigation in distributed environments. The source code for our TL-based supermodel is available on (https://github.com/drzeeshanaliofficial/Deep-Transfer-Learning-for-Intrusion-Detection-in-Edge-Computing-Scenarios), and the README contains detailed descriptions of the Python files.

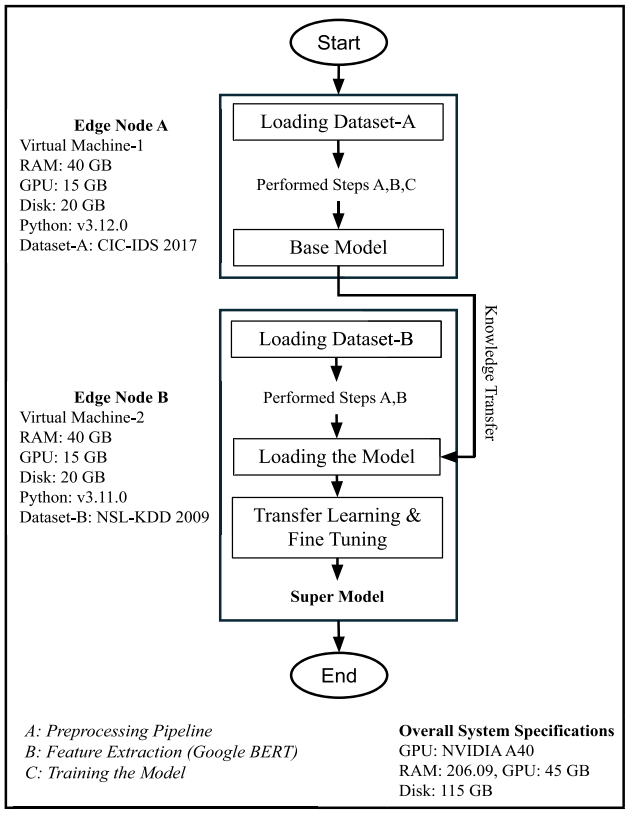


Fig. 3. Distributed system architecture used for the TL experiments.

A. Input Datasets

To emulate the heterogeneity of DS collected from different edge nodes, we adopt two DS widely used in IDS for edge computing in IoT networks [34], [35], [36]: the CIC-IDS 2017 [37] as the DS-A at the EN-A and the NSL-KDD 2009 [7] as the DS-B at the EN-B.

The CIC-IDS 2017 dataset, collected by the Canadian Institute for Cybersecurity, is widely used for NIDS since it contains a variety of network traffic data labeled as both normal and attack traffic, associated to the most common attack types, such as DoS, DDoS, port scanning, and more. Traffic data was captured in a controlled environment, simulating real-world scenarios, and includes detailed information, such as packet and flow-level features, resulting in the 79 distinct features shown in Table III, which form a rich foundation for our in-depth analysis.

The NSL-KDD dataset is a modified version of the original KDD Cup 1999 dataset, and contains a set of labeled network traffic data, with each instance representing a network connection characterized by a variety of features (up to 42), such as connection duration, protocol type, and error rates, listed in

TABLE III
FEATURES OF CIC-IDS 2017 DATASET

F#	Feature Name	F#	Feature Name	F#	Feature Name	F#	Features
1	Destination Port	21	Fwd IAT Total	41	Pkt Length Mean	61	Bwd Avg Pkt/Bulk
2	Flow Duration	22	Fwd IAT Mean	42	Pkt Length Std	62	Bwd Avg Bulk Rate
3	Total Fwd Pkt	23	Fwd IAT Std	43	Pkt Len Variance	63	Subflow Fwd Packs
4	Total Bwd Pkt	24	Fwd IAT Max	44	FIN Flag Count	64	Subflow Fwd Bytes
5	Total Len of Fwd Pkt	25	Fwd IAT Min	45	SYN Flag Count	65	Subflow Bwd Packs
6	Total Len of Bwd Pkt	26	Bwd IAT Total	46	RST Flag Count	66	Subflow Bwd Bytes
7	Fwd Pkt Len Max	27	Bwd IAT Mean	47	PSH Flag Count	67	Init Win bytes Fwd
8	Fwd Pkt Len Min	28	Bwd IAT Std	48	ACK Flag Count	68	Init Win bytes Bwd
9	Fwd Pkt Len Mean	29	Bwd IAT Max	49	URG Flag Count	69	Act Data Pkt Fwd
10	Fwd Pkt Len Std	30	Bwd IAT Min	50	CWE Flag Count	70	Min Seg Size Fwd
11	Bwd Pkt Len Max	31	Fwd PSH Flags	51	ECE Flag Count	71	Active Mean
12	Bwd Pkt Len Min	32	Bwd PSH Flags	52	Down/Up Ratio	72	Active Std
13	Bwd Pkt Len Mean	33	Fwd URG Flags	53	Average Pkt Size	73	Active Max
14	Bwd Pkt Len Std	34	Bwd URG Flags	54	Avg Fwd Seg Size	74	Active Min
15	Flow Bytes/s	35	Fwd Header Len	55	Avg Bwd Seg Size	75	Idle Mean
16	Flow Pkt/s	36	Bwd Header Len	56	Fwd Header Len. 1	76	Idle Std
17	Flow IAT Mean	37	Fwd Pkt/s	57	Fwd Avg Bytes/Bulk	77	Idle Max
18	Flow IAT Std	38	Bwd Pkt/s	58	Fwd Avg Pkt/Bulk	78	Idle Min
19	Flow IAT Max	39	Min Pkt Len	59	Fwd Avg Bulk Rate	79	Label

TABLE IV
FEATURES OF NSL-KDD 2009 DATASET

F#	Feature name	F#	Feature name	F#	Feature name
1	Duration	15	Su attempted	29	Same srv rate
2	Protocol type	16	Num root	30	Diff srv rate
3	Service	17	Num file creations	31	Srv diff host rate
4	Flag	18	Num shells	32	Dst host count
5	Src bytes	19	Num access files	33	Dst host srv count
6	Dest bytes	20	Num outbound cmds	34	Dst host same srv rate
7	Land	21	Is host login	35	Dst host diff srv rate
8	Wrong fragment	22	Is guest login	36	Dst host same src port rate
9	Urgent	23	Count	37	Dst host srv diff host rate
10	Hot	24	Srv count	38	Dst host error rate
11	Num failed logins	25	error rate	39	Dst host srv error rate
12	Logged in	26	Srv error rate	40	Dst host error rate
13	Num compromised	27	error rate	41	Dst host srv error rate
14	Root shell	28	Srv error rate	42	label

Table IV. The dataset includes both normal and attack traffic, with the attacks categorized into various types, such as DoS, R2L, U2R, and probing.

Both DS provide a comprehensive range of network flow and session-level features that closely align with the types of telemetry and behavior patterns typically observable in IoT devices and systems. For instance, features like flow duration, packet lengths, byte counts, interarrival times (IATs), and header statistics in CIC IDS 2017 are directly reflective of network traffic characteristics at the edge, where lightweight monitoring tools capture such parameters for real-time anomaly detection. Similarly, the NSL-KDD dataset includes key indicators, such as source/destination bytes, protocol type, service usage, and connection flags—features that are equally relevant in edge networks where constrained devices often interact over specific services and protocols. Moreover, both DS incorporate behavioral and statistical insights (e.g., login attempts, connection error rates, session flags) that are crucial for detecting intrusion or abnormal activity in distributed edge settings. The presence of such overlapping low-level and session-based features ensures that the analytical models trained on these DS can generalize well to edge scenarios, making their inclusion not only appropriate but essential for studying security in edge computing contexts.

The attacks covered by these DS are detailed in Tables V and VI, where DoS stands for Denial of Service, DDoS for Distributed DoS, R2L for Root to Local and U2R for User to Root attacks. These attack types are commonly encountered in distributed networks where IoT devices process data closer to the source, making it critical to detect and mitigate such threats in real-time to ensure the security and reliability of edge networks. Our choice of these DS is motivated by their widespread recognition within network security and intrusion detection, as well as their varied characteristics that collectively enable comprehensive model evaluation and validation. However, while they cover a wide range of attack types and normal traffic, we acknowledge that they may not capture all possible variations in network conditions that could be encountered in different edge environments, thus generating potential biases. To address this concern and to assess the model's performance and robustness in real-world scenarios, ensuring its generalizability across diverse edge environments,

TABLE V
CIC-IDS 2017 DATASET (REFERRED TO AS DS-A IN OUR SYSTEM MODEL)

No	Attacks	Attacks Found	Original Records	Augmented Records
1	BENIGN	BENIGN	2,096,484	Not Augmented
2	Brute Force	FTP-Patator SSH-Patator	9,152	12,000
3	DoS	DoS GoldenEye DoS Hulk DoS Slowhttptest DoS Slowloris Heartbleed Bot	195,712	205,000
4	Web Attacks	Web Attack - Brute Force, Web Attack - SQL Injection, Web Attack - XSS	2,143	10,000
5	Infiltration	Infiltration	36	1,500
6	Port Scan	Port Scan	90,819	100,000
7	DDoS	DDoS	128,016	150,000

TABLE VI
NSL-KDD 2009 DATASET (REFERRED TO AS DS-B IN OUR SYSTEM MODEL)

No	Labels	Training Set	Testing Set	Augmented Records
1	Normal	67,343	9,711	Not Augmented
2	DoS	45,927	7,460	Not Augmented
3	Probe	11,656	2,421	Not Augmented
4	R2L	995	2,885	8,000
5	U2R	52	67	5000

our future work will involve validating the model on additional DS collected from various edge nodes across different network environments.

B. Dataset Preprocessing

Both DS present a significant class imbalance, e.g., the BENIGN class in the CIC-IDS 2017 dataset (Table V) has over 2 million records, whereas classes, such as Web Attack and Infiltration have only 2143 and 36 records, respectively, and the NSL-KDD 2009 dataset (Table VI), demonstrates a pronounced imbalance in the R2L and U2R classes, with only 995 and 52 samples in the training set, respectively, compared to tens of thousands of samples in the "Normal" and "DoS" classes. Without addressing this class imbalance problem, the model would likely struggle to learn effective patterns for these minority classes, further reducing the efficacy of the IDS. To address this issue and improve the model's performance in detecting minority class intrusions, we augment the number of records for each attack class by using SMOTE [8], resulting in the "Augmented Records" listed in the last columns of Tables V and VI. This approach ensures that our IDS remains robust and effective, even when faced with highly imbalanced data distributions. While other techniques, such as ADASYN and cost-sensitive learning, are valid options, they often introduce additional complexity and require careful tuning of hyperparameters to achieve optimal results. Furthermore, methods like ADASYN can sometimes result in the oversynthesis of minority class instances, which may negatively impact the model's generalization ability. In contrast, SMOTE offers a more straightforward and well-established approach that balances performance enhancement with the preservation of data distribution.

As detailed in our prior work [8], we eliminate duplicate samples to ensure the dataset's integrity by preventing redundancy and bias, which could otherwise skew ML model outcomes. We then perform label mapping, i.e., the conversion of categorical labels into numerical values, to facilitate accurate processing and analysis by ML models. These preprocessing steps lay a strong foundation for developing robust and unbiased IDS frameworks.

As an upgrade of our previous work [8], we process the DS by using the most widely used BERT architectures, i.e., BERT-Base and BERT-Large, for feature extraction at both edge nodes A and B. BERT-Base consists of 12 transformer layers, 12 attention heads, and a hidden size of 768, making it computationally efficient while still capturing meaningful contextual representations. On the other hand, BERT-Large features 24 transformer layers, 16 attention heads, and a hidden size of 1024, enabling it to learn more complex patterns at the cost of increased computational demands.

Finally, to assess the ML model's efficacy, the preprocessed DS underwent stratified division into distinct training and testing subsets. Specifically, 80% of data was allocated for training purposes, while the remaining 20% was reserved for rigorous testing. This deliberate partitioning ensures a representative distribution of instances across both sets, maintaining a proportional representation of different classes in both the training and testing phases.

C. Evaluation Metrics

The evaluation is based on the five metrics Accuracy, Precision, Recall, F1-Score and Confusion Matrix, expressed as functions of the key evaluation terms defined in Table VII.

Accuracy represents the overall effectiveness of the classification process, and is calculated as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{2}$$

Precision is used to assess the accuracy of positive predictions in terms of the percentage of instances correctly

TABLE VII
DEFINITIONS OF TERMS: TP, TN, FP, FN

Term	Definition
True Positive (TP)	Correctly identified positive instances
True Negative (TN)	Correctly identified negative instances
False Positive (FP)	Instances wrongly identified as positive
False Negative (FN)	Instances wrongly identified as negative

TABLE VIII
CONFUSION MATRIX

Actual	Predicted	
	Positive	**Negative**
Positive	True Positives (TP)	False Negatives (FN)
Negative	False Positives (FP)	True Negatives (TN)

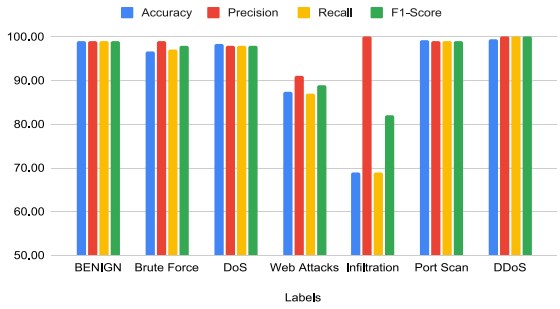


Fig. 4. Performance evaluation of BERT-base super model on the CIC-IDS 2017 dataset.

predicted by the model as positive (e.g., abnormal)

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{3}$$

Recall, also known as sensitivity or true positive rate, measures the ability of the model to identify all positive instances correctly

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{4}$$

Especially useful when dealing with imbalanced DS, where one class is more dominant than the others, the F1-Score provides a balanced measure of the model's performance between precision and recall

$$\text{F1-Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{5}$$

Finally, the confusion matrix, defined in Table VIII, provides a comprehensive view of the model's performance. Each row of the matrix represents the instances in an actual class, while each column represents the instances in a predicted class.

D. Performance Evaluation

The performance evaluation results are shown in Fig. 4–Fig. 7, with multiclass confusion matrices presented in Fig. 8–Fig. 11.

Across both DS, the BERT-based BM demonstrates high accuracy, particularly in classifying BENIGN traffic and common attack types like DDoS and Port Scan. However, it shows variability in detecting more complex attacks, such as Web-based attacks and infiltration on the CIC-IDS 2017 dataset,

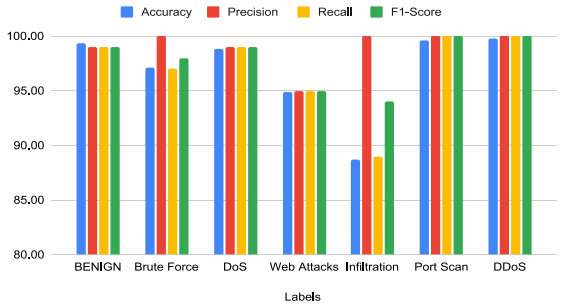


Fig. 5. Performance evaluation of BERT-large-super model on the CIC-IDS 2017 dataset.

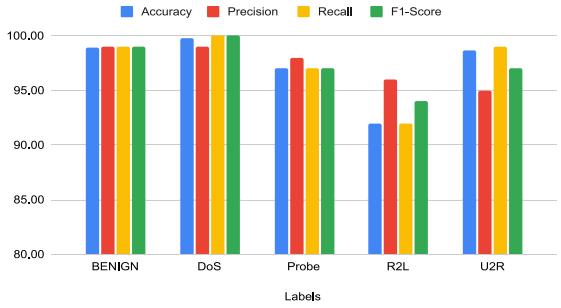


Fig. 6. Performance evaluation of BERT-base super model on the NSL-KDD 2009 dataset.

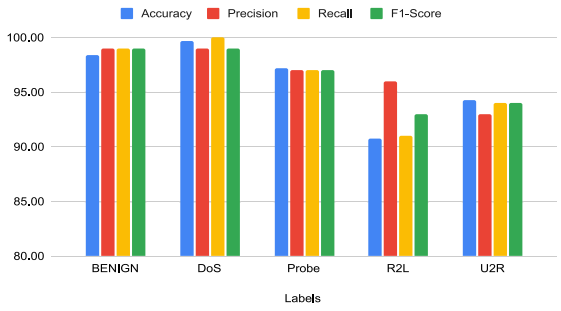


Fig. 7. Performance evaluation of BERT-large-super model on the NSL-KDD 2009 dataset.

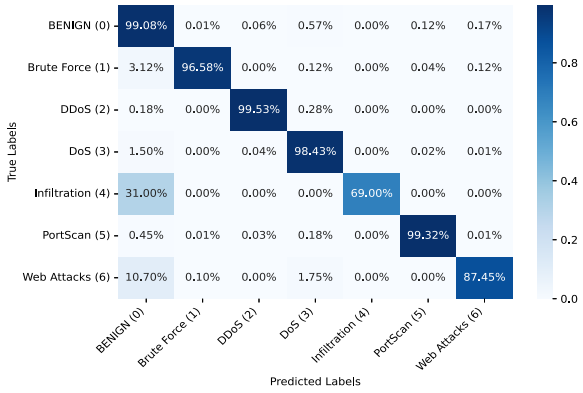


Fig. 8. Confusion matrix of multiclass classification using BERT-base *supermodel* on CIC-IDS 2017 dataset.

and faces challenges with the R2L and U2R categories on the NSL-KDD 2009 dataset. Despite high precision, indicating few false positives, BERT-Base struggles with false negatives, especially for less frequent attack types.

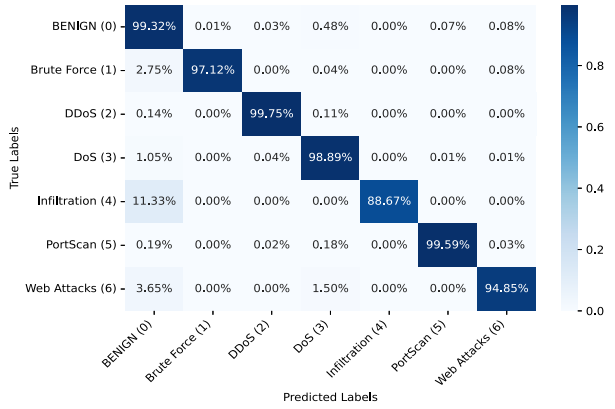


Fig. 9. Confusion matrix of multiclass classification using BERT-large-*supermodel* on CIC-IDS 2017 dataset.

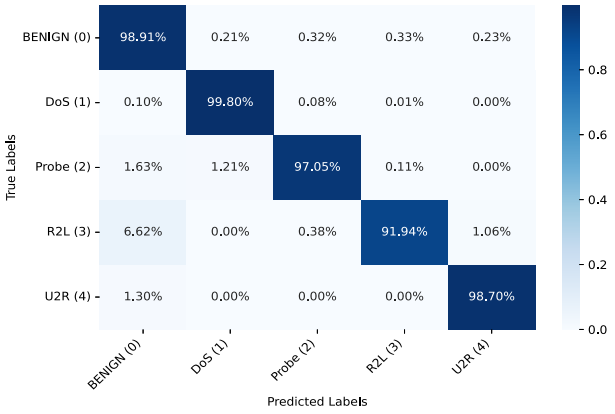


Fig. 10. Confusion matrix of multiclass classification using BERT-base *supermodel* on NSL-KDD 2009 dataset.

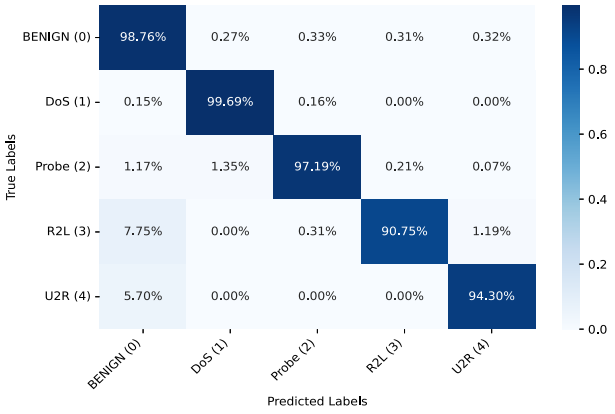


Fig. 11. Confusion matrix of multiclass classification using BERT-large-*supermodel* on NSL-KDD 2009 dataset.

On the other hand, the BERT-Large model consistently outperforms BERT-Base on the CIC-IDS 2017 dataset, achieving superior results in accuracy, precision, recall, and F1-Score across most categories, particularly for Web attacks and infiltration. However, when applied to the NSL-KDD 2009 dataset, BERT-Base performs slightly better overall, particularly in terms of accuracy and precision, due to its better balance between model complexity and the dataset's characteristics.

Our experimental results indicate that BERT-Large-based *supermodel* performs better on the higher number of features, as seen with CIC-IDS 2017, which contains 79 features, while BERT-Base excels on DS with fewer features, such as NSL-KDD 2009, which has 42 features. This suggests that the increased model capacity of BERT-Large is beneficial for capturing complex feature interactions in high-dimensional data, whereas BERT-Base *supermodel* remains effective in scenarios with lower feature complexity, offering a more efficient alternative for resource-constrained environments.

To evaluate the effectiveness of our proposed BERT and MLP model, Table IX compares its accuracy, precision, recall, and F1-score against several state-of-the-art classifiers on the same CIC-IDS 2017 and NSL-KDD 2009 DS. For the CIC-IDS 2017 dataset, our model achieved an accuracy of 99.39%, which is higher than the other models compared. The closest competitor, a CNN-based approach [38], achieved an accuracy of 98.79%. Additionally, our model excels in terms of precision, recall, and F1-score, all of which are also at 99.39%, surpassing other methods, such as the CNN-LSTM [39] hybrid model that reaches 98.56% accuracy and 99.25% F1-score. On the NSL-KDD 2009 dataset, our model also shows superior performance, with an accuracy of 99.33%, compared to other methods like the Ensemble Learning-DNN [40], which achieved 98.90%. Our model maintains a balanced accuracy, recall, and F1-score of 99.33%.

To address potential concerns regarding sequential dependency between edge nodes, we clarify that the proposed TL framework does not enforce strict sequential execution. While EN-B benefits from the pretrained model generated at EN-A, it is capable of performing standalone training and intrusion detection using its own dataset. The performance evaluation of the proposed TL-based approach, leveraging both BERT-Base and BERT-Large models at individual edge nodes, is detailed in Table X. The standalone BERT-Base and BERT-Large models consistently demonstrate high accuracy across both nodes, achieving 99.37% and 98.94% accuracy at EN-A, and 99.33% and 99.35% at EN-B, respectively. The *supermodel* BERT-Large, designed to handle data from both nodes, achieves competitive results with 99.23% accuracy at Node A, 98.38% at Node B, and an aggregated accuracy of 99.33% when tested across EN-A and EN-B. These results underscore the robustness of the proposed *supermodel*, which maintains generalization, high precision, recall, and F1-score, even in heterogeneous data distributions without limiting the autonomy of individual nodes.

E. Real-World Applicability of Proposed Approach

The model is designed for distributed edge environments, where intrusion detection must occur in real-time with minimal computational overhead. By leveraging TL, the model adapts to different network conditions without requiring extensive retraining for each new dataset or device. The unified *supermodel* is only 3.10 MB in size, making it highly efficient for deployment on IoT devices with limited storage and processing capabilities. Traditional DL-based IDS models are often too computationally expensive for IoT devices, whereas

TABLE IX
PERFORMANCE EVALUATION OF TL-BASED *Supermodel* WITH STATE-OF-THE-ART CLASSIFIERS USING CIC-IDS 2017 AND NSL-KDD 2009 DS

Dataset	Study	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
CIC-IDS 2017	Proposed	BERT and MLP	**99.39**	**99.39**	**99.39**	**99.99**
	[38]	Convolutional Neural Network (CNN)	98.79	98.80	98.77	98.77
	[41]	Self-Organizing Map (SOM)	80.00	77.40	81.80	80.00
	[39]	CNN-Long Short Term Memory (LSTM)	98.56	98.50	99.00	99.25
	[39]	CNN-Recurrent Neural Networks (RNN)	97.24	97.50	97.25	96.50
	[39]	CNN-Gated Recurrent Unit (GRU)	96.12	96.25	96.25	95.50
NSL-KDD 2009	Proposed	BERT and MLP	**99.33**	**99.34**	**99.33**	**99.33**
	[42]	Decision Tree	97.40	97.40	97.10	97.20
	[41]	SOM	91.00	91.00	91.40	91.00
	[43]	Autoencoder	79.74	82.22	79.74	76.47
	[40]	Ensemble Learning-Deep Neural Network (DNN)	98.90	99.90	98.71	99.31
	[44]	Non-Symmetric Deep Autoencoder (NDAE)	85.42	100.00	85.42	87.37

TABLE X
PROPOSED APPROACH PERFORMANCE EVALUATION AT INDIVIDUAL NODES

Individual Nodes (A and B)	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Standalone BERT-Base at Node A	99.37	99.37	99.37	99.37
Standalone BERT-Base at Node B	99.33	99.34	99.33	99.33
Standalone BERT-Large at Node A	98.94	98.96	98.94	98.94
Standalone BERT-Large at Node B	99.35	99.34	99.35	99.34
Supermodel BERT-Large at Node A	99.23	99.23	99.23	99.23
Supermodel BERT-Large at Node B	98.38	98.38	98.38	98.38
Supermodel BERT-Large at Node A & B	99.33	99.34	99.33	99.33

our BERT-MLP-based approach optimizes performance while maintaining lightweight deployment. The performance evaluation (Table IX) demonstrates that the proposed approach outperforms multiple SOTA IDS solutions in terms of accuracy, precision, recall, and F1-score across two benchmark DS. These results validate the model's robustness across diverse network traffic scenarios, making it suitable for real-world cybersecurity applications in IoT networks, smart cities, industrial control systems, and autonomous vehicles. The TL mechanism enables the model to continuously learn from new edge nodes, allowing it to adapt to previously unseen attack patterns without requiring complete retraining. This adaptability is crucial for zero-day attack detection, ensuring that the IDS remains effective in evolving network environments.

However, deploying the proposed TL-enabled IDS in real-world edge environments presents several challenges, including hardware constraints, network latency, security vulnerabilities, and adaptability to evolving threats. IoT devices often have limited computational power and storage, necessitating lightweight models; our optimized 3.10 MB *supermodel* facilitates efficient deployment, but further compression techniques, such as quantization and pruning, can enhance feasibility. Additionally, as the *supermodel* is periodically shared with end devices, network bandwidth limitations and latency issues must be addressed through asynchronous updates and differential synchronization techniques to ensure real-time adaptability. Furthermore, given the highly dynamic nature of cyber threats, continuous model updating is essential, and while TL allows adaptability across DS, integrating FL and online incremental learning can further enhance real-time detection capabilities.

There also security concerns, such as poisoning attacks, adversarial perturbations, and model inversion risks, that require robust countermeasures, including adversarial training, secure aggregation in federated settings, and blockchain-based trust mechanisms to maintain IDS reliability.

Adversarial attacks involve manipulating input data to either evade detection or degrade model performance. The two primary types are evasion attacks, where adversaries craft adversarial samples to bypass detection, and poisoning attacks, where malicious data is injected into the training set to compromise model accuracy.

To enhance the robustness of our TL-enabled IDS, several countermeasures can be employed, such as adversarial training, defensive distillation, feature squeezing and ensemble learning. While our current work does not explicitly focus on adversarial robustness, these countermeasures provide a foundation for enhancing the security of TL-enabled IDS in edge deployments and will be addressed in future work.

One of the primary limitations encountered during this research was the computational intensity associated with transformer-based models, such as BERT, and DL techniques, like the MLP. While essential for robust performance, this demand for substantial GPU resources may hinder scalability and real-time applicability, especially in environments or organizations with limited access to high-performance computing infrastructure. Future work should explore optimization strategies, distributed computing, or hardware acceleration to overcome this constraint, specifically focusing on integrating edge computing resources. Leveraging edge computing capabilities can distribute the computational load to local devices, enhancing scalability and real-time responsiveness in resource-constrained settings. Acknowledging these computational challenges is crucial for transparency and a comprehensive understanding of the proposed approach's practical implications, ensuring the security and stability of our interconnected world in the face of digital evolution.

VI. Conclusion

Our research presents a groundbreaking approach to fortifying IDS by seamlessly integrating TL into edge computing. Our methodology harnesses the power of BERT for semantic feature extraction, utilizing both BERT-Base and BERT-Large architectures. To address the class imbalance, we applied SMOTE. The enriched feature data is utilized to train an MLP using a TL collaborative strategy, resulting in the development of a joint *supermodel* demonstrating exceptional adaptability and proficiency across various intrusion scenarios mapped by heterogeneous DS collected from different edge nodes. In real-world scenarios, the model's accuracy could be challenged by sudden shifts in network traffic patterns or the introduction of novel intrusion techniques that deviate significantly from those observed during training. For example, a DDoS attack with previously unseen traffic characteristics or a sophisticated phishing attempt involving encrypted traffic may introduce anomalies that the model has not encountered, potentially resulting in misclassification. Similarly, unexpected spikes in benign traffic or the emergence of new attack vectors could also reduce the model's effectiveness. To mitigate these challenges and ensure consistent performance, we propose incorporating threat intelligence feeds that provide real-time updates on emerging threats. By integrating this information, the model can stay informed of new attack patterns and adjust accordingly. Furthermore, training the model on data from diverse sources and environments, rather than relying solely on data from local edge nodes, can enhance its ability to generalize. This approach will make the model more resilient to variations in traffic patterns across different network environments, as shown by our validation experiments on two widely used DS.

References

[1] B. Panchali, "Edge computing-background and overview," in *Proc. Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, 2018, pp. 580–582.

[2] G. Kaur and R. S. Batth, "Edge computing: Classification, applications, and challenges," in *Proc. 2nd Int. Conf. Intell. Eng. Manag. (ICIEM)*, 2021, pp. 254–259.

[3] P. Dini, A. Elhanashi, A. Begni, S. Saponara, Q. Zheng, and K. Gasmi, "Overview on intrusion detection systems design exploiting machine learning for networking cybersecurity," *Appl. Sci.*, vol. 13, no. 13, p. 7507, Jun. 2023.

[4] S. Hafeez and N. Kathirisetty, "Effects and Comparison of different Data pre-processing techniques and ML and deep learning models for sentiment analysis: SVM, KNN, PCA with SVM and CNN," in *Proc. 1st Int. Conf. Artif. Intell. Trends Pattern Recognit. (ICAITPR)*, 2022, pp. 1–6.

[5] G. Zhu, H. Zhao, H. Liu, and H. Sun, "A novel LSTM-GAN algorithm for time series anomaly detection," in *Proc. Prognostics Syst. Health Manag. Conf. (PHM-Qingdao)*, 2019, pp. 1–6.

[6] B. Aslahi-Shahri et al., "A hybrid method consisting of GA and SVM for intrusion detection system," *Neural Comput. Appl.*, vol. 27, pp. 1669–1676, Aug. 2016.

[7] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *Proc. IEEE Symp. Comput. Intell. Security Def. Appl.*, 2009, pp. 1–6.

[8] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli, "Empowering network security: Bert transformer learning approach and MLP for intrusion detection in imbalanced network traffic," *IEEE Access*, vol. 12, pp. 137618–137633, 2024.

[9] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[10] M. Mimura and H. Tanaka, "Reading network packets as a natural language for intrusion detection," in *Proc. 20th Int. Conf. Inf. Security Cryptol. (ICISC)*, 2018, pp. 339–350.

[11] 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," *J. Netw. Comput. Appl.*, vol. 221, Jan. 2024, Art. no. 103784.

[12] H. Zhang, L. Huang, C. Q. Wu, and Z. Li, "An effective convolutional neural network based on smote and gaussian mixture model for intrusion detection in imbalanced dataset," *Comput. Netw.*, vol. 177, Aug. 2020, Art. no. 107315.

[13] B. Zong et al., "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–19.

[14] K. Alrawashdeh and C. Purdy, "Toward an online anomaly intrusion detection system based on deep learning," in *Proc. 15th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, 2016, pp. 195–200.

[15] J. Zhang, C. Luo, M. Carpenter, and G. Min, "Federated learning for distributed IIoT intrusion detection using transfer approaches," *IEEE Trans. Ind. Informat.*, vol. 19, no. 7, pp. 8159–8169, Jul. 2023.

[16] J. H. Joloudari, A. Marefat, M. A. Nematollahi, S. S. Oyelere, and S. Hussain, "Effective class-imbalance learning based on smote and convolutional neural networks," *Appl. Sci.*, vol. 13, no. 6, p. 4006, 2023.

[17] Q. Yang, "An introduction to transfer learning," in *Proc. Adv. Data Min. Appl.*, 2008, p. 1.

[18] N. Alhussien, A. Aleroud, R. Rahaeimehr, and A. Schwarzmann, "Triggerability of backdoor attacks in multi-source transfer learning-based intrusion detection," in *Proc. IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol. (BDCAT)*, 2022, pp. 40–47.

[19] Y. Cheng, J. Lu, D. Niyato, B. Lyu, J. Kang, and S. Zhu, "Federated transfer learning with client selection for intrusion detection in mobile edge computing," *IEEE Commun. Lett.*, vol. 26, no. 3, pp. 552–556, Mar. 2022.

[20] W. Yutao, L. Zhongtian, B. Yi, L. Jie, X. Fangzheng, and B. Yu, "Internet of Things intrusion detection system based on transfer learning," in *Proc. IEEE 2nd Int. Conf. Electron. Technol., Commun. Inf. (ICETCI)*, 2022, pp. 25–30.

[21] S. T. Mehedi, A. Anwar, Z. Rahman, K. Ahmed, and R. Islam, "Dependable intrusion detection system for IoT: A deep transfer learning based approach," *IEEE Trans. Ind. Informat.*, vol. 19, no. 1, pp. 1006–1017, Jan. 2023.

[22] Y. K. Saheed, O. H. Abdulganiyu, and T. A. Tchakoucht, "Modified genetic algorithm and fine-tuned long short-term memory network for intrusion detection in the Internet of Things networks with edge capabilities," *Appl. Soft Comput.*, vol. 155, Apr. 2024, Art. no. 111434.

[23] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.

[24] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Proc. Mil. Commun. Inf. Syst. Conf. (MilCIS)*, 2015, pp. 1–6.

[25] Y. Meidan et al., "N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders," *IEEE Pervasive Comput.*, vol. 17, no. 3, pp. 12–22, Jul.–Sep. 2018.

[26] H. S. M. Masum, "A transfer learning with deep neural network approach for network intrusion detection," *Int. J. Intelll. Comput. Res.*, vol. 12, no. 1, pp. 1–9, 2021.

[27] A. Bagaskara and M. Suryanegara, "Evaluation of VGG-16 and VGG-19 deep learning architecture for classifying dementia people," in *Proc. 4th Int. Conf. Comput. Informat. Eng. (IC2IE)*, 2021, pp. 1–4.

[28] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An enhanced MobileNet architecture," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, 2019, pp. 280–285.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2009, pp. 248–255.

[30] Y. K. Saheed and S. Misra, "CPS-IoT-PPDNN: A new explainable privacy preserving DNN for resilient anomaly detection in cyber-physical systems-enabled IoT networks," *Chaos, Solitons & Fractals*, vol. 191, Feb. 2025, Art. no. 115939.

[31] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and H. Janicke, "Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281–40306, 2022.

[32] M. Al-Hawawreh, E. Sitnikova, and N. Aboutorab, "X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial Internet of Things," *IEEE Internet Things J.*, vol. 9, no. 5, pp. 3962–3977, Mar. 2022.
[33] J. Kirkpatrick et al., "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci.*, vol. 114, no. 13, pp. 3521–3526, 2017.
[34] A. Kumar, S. Kumar, V. Kumar, A. Kumari, A. Saini, and S. Gupta, "Edge computing based IDS detecting threats using machine learning and PyCaret," in *Proc. Int. Conf. Comput. Intell. Sustain. Eng. Solut. (CISES)*, 2023, pp. 668–673.
[35] P. Mahadevappa, R. K. Murugesan, R. Al-amri, R. Thabit, A. H. Al-Ghushami, and G. Alkawsi, "A secure edge computing model using machine learning and IDS to detect and isolate intruders," *MethodsX*, vol. 12, Jun. 2024, Art. no. 102597.
[36] H. Bangui and B. Buhnova, "Lightweight intrusion detection for edge computing networks using deep forest and bio-inspired algorithms," *Comput. Elect. Eng.*, vol. 100, May 2022 Art. no. 107901.
[37] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in *Proc. ICISSP*, 2018, pp. 108–116.
[38] M. A. Siddiqi and W. Pak, "Tier-based optimization for synthesized network intrusion detection system," *IEEE Access*, vol. 10, pp. 108530–108544, 2022.
[39] 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," *Digit. Commun. Netw.*, vol. 10, pp. 190–204, Feb. 2024.
[40] A. Thakkar and R. Lohiya, "Attack classification of imbalanced intrusion data for IoT network using ensemble-learning-based deep neural network," *IEEE Internet Things J.*, vol. 10, no. 13, pp. 11888–11895, Jul. 2023.
[41] J. Ables et al., "Creating an explainable intrusion detection system using self organizing maps," in *Proc. IEEE Symp. Ser. Comput. Intell. (SSCI)*, 2022, pp. 404–412.
[42] R.-F. Hong, S.-C. Horng, and S.-S. Lin, "Machine learning in cyber security analytics using NSL-KDD dataset," in *Proc. Int. Conf. Technol. Appl. Artif. Intell. (TAAI)*, 2021, pp. 260–265.
[43] C. Zhang, F. Ruan, L. Yin, X. Chen, L. Zhai, and F. Liu, "A deep learning approach for network intrusion detection based on NSL-KDD dataset," in *Proc. IEEE 13th Int. Conf. Anti-Counterfeiting, Security, Identif. (ASID)*, 2019, pp. 41–45.
[44] N. Shone, T. N. Ngoc, V. D. Phai, and Q. Shi, "A deep learning approach to network intrusion detection," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 2, no. 1, pp. 41–50, Feb. 2018.



Zeeshan Ali (Graduate Student Member, IEEE) received the B.S. and M.S. degrees in computer science from the National College of Business Administration and Economics, Lahore, Pakistan, in 2021 and 2022, respectively. He is currently pursuing the Ph.D. degree with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila, L'Aquila, Italy, supervised by Dr. D. Cassioli.

He is a Cybersecurity Researcher with the Department of Information Engineering, Computer Science, and Mathematics, University of L'Aquila. He has hands-on experience with state-of-the-art cybersecurity tools and advanced technologies, including large-language models and generative AI. He previously served as the National Startup Founder with the Punjab Information Technology Board, Lahore. His research interests encompass cybersecurity, artificial intelligence, machine learning, the Internet of Things, and software quality assurance.



Walter Tiberti (Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering and the Ph.D. degree in information and communication technology from the University of L'Aquila, L'Aquila, Italy, in 2013, 2016, and 2020, respectively, where his research focused on embedded systems security, particularly the security of wireless sensor networks.

He is currently an Assistant Professor with the Department of Information Engineering, Computer Science, and Mathematics, University of LŠAquila. He has also been a Visiting Researcher with the Research Center in Real-Time and Embedded Computing Systems, Porto, Portugal. His research interests include software security, network security, and cryptography.

Dr. Tiberti serves as a technical committee member for various IEEE conferences and is actively involved in national initiatives aimed at advancing cybersecurity education and training in Italy.



Andrea Marotta (Member, IEEE) received the M.Sc. degree in computer engineering and the Ph.D. degree in information and communications technology from the University of L'Aquila, L'Aquila, Italy, in 2015 and 2019, respectively.

He is Assistant Professor (Tenure) with the Department of Information Engineering, Computer Science and Mathematics, University of L'Aquila. He was Visiting Researcher with the Group for Research on Wireless, Instituto Superior Técnico/University, Lisbon, Portugal, and the Research Institute of Communication, Information and Perception Technologies, Scuola superiore di studi universitari e di perfezionamento Sant'Anna, Pisa, Italy, and the Computer Networks Laboratory, University of California at Davis, Davis, CA, USA. He performs researches on optical and mobile networks, network reliability, and multiaccess-edge computing.

Dr. Marotta actively participates in the TPC of IEEE Communications Society conferences.



Dajana Cassioli (Senior Member, IEEE) is an Associate Professor of Telecommunications Engineering with the University of L'Aquila, L'Aquila, Italy, where she served as the Elected Head for the Study Program in "Telecommunications Engineering: Advanced Technologies and Services" from 2021 to 2024. She co-authored more than 100 articles published in the most renowned international journals, magazines, and conference proceedings. She was a Summer Manager with the Wireless Systems Research Department, AT&T Laboratories-Research, NJ, USA, in 2000. She was a Short-Term Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, hosted by Prof. Andy Molisch, in 2022. Her main research interests are in wireless communications, 5G/B5G networks and cybersecurity.

Dr. Cassioli is the Chair of the IEEE ComSoc RCC SiG on Propagation Channels for 5G&B, the Chair of the SiG on Distributed and Massive MIMO of the IEEE P1944 Standard for Channel Models of Wireless Systems and the Diversity, Equity and Inclusion Activity Coordinator of the IEEE Italy Section. She is the Past Chair of the IEEE WIE AG Italy Section from 2016 to 2022 and the IEEE VT06/COM19 Italy Chapter from 2011 to 2017. Since 2015, she is the Coordinator of the University of L'Aquila Node of the CINI National Laboratory of Cybersecurity, where she led the CyberEquality WG from 2020 to 2021. She has been awarded the ERC StG VISION (Video-oriented UWB-based Intelligent Ubiquitous Sensing) in 2010 and the ERC PoC Grant iCARE (MobIle health-Care system for monitoring toxicity and symptoms in Cancer patients Receiving Disease-Oriented Therapy) in 2016. She was the CEO in 2014–2018, and 2019 of the spin-off of the University of L'Aquila "Smartly: Natives of Smart Living srl," which designs and markets advanced ICT solutions to improve the quality of life. She served as the Co-Chair for Globecom 2025 WC, ICC 2024 MWN, ICC 2023 CISS, PIMRC2018 Industry, RTSI WIE Chair in 2018, 2019, and 2020, MELECON2020 and MetroInd4.0, and the TPC member of several International Conferences, including ICC, PIMRC, VTC, and GLOBECOM. She participated in the definition of the standard channel model for the IEEE 802.15.4 standard in 2005. She is an Associate Editor of *IET Electronic Letters*, and an Executive Editor of WILEY INTERNET TECHNOLOGY LETTERS and *Transactions on Emerging Telecommunications Technology* and served as an Associate Editor for IEEE COMMUNICATIONS LETTERS from 2018 to 2022.

Part IV

Appended Papers
Under-review/Accepted

Paper C3

Confidential Model Sharing Protocol for Distributed Transfer Learning and Blockchain-Governed Access

Zeeshan Ali, Walter Tiberti, Andrea Marotta and Dajana Cassioli

Accepted in
10^{th} IEEE Research and Technologies for Society and Industry (RTSI)

Confidential Model Sharing Protocol for Distributed Transfer Learning and Blockchain-Governed Access

Zeeshan Ali*, Walter Tiberti, Andrea Marotta, and Dajana Cassioli
Department of Information Engineering, Computer Science, and Mathematics (DISIM)
University of L'Aquila, L'Aquila, Italy

***Abstract*—One of the main challenges of collaborative transfer learning (TL) is the *secure* model sharing across distributed edge nodes. In this paper, we propose a novel protocol for the secure publication of models and controlled access in distributed environments. Any model artifact is locally encrypted and signed by the owner node, for integrity and authenticity verification, and then published to the InterPlanetary File System (IPFS) under a unique content identifier (CID). The CID and relevant model metadata are then registered immutably on an Ethereum smart contract via the publisher's blockchain wallet. The proposed protocol also introduces a blockchain-mediated access-request workflow in which a requesting node registers its public key on-chain, such that it can use its private key to locally recover the symmetric decryption key generated by the model owner, which records the access grant on-chain. The requester then verifies the publisher's digital signature before decrypting. Our experimental evaluation demonstrates that the protocol supports transparent model lineage, selective access control, and end-to-end confidentiality, making it well-suited to secure TL in trustless distributed environments. We also provide considerations on the economic viability and energy efficiency of the proposed approach.**

***Index Terms*—Transfer Learning, Secure Model Sharing, Blockchain, End-to-End Encryption**

I. Introduction

In collaborative TL, participants shall exchange trained model artifacts without exposing intellectual property (IP) or relying on trusted centralized intermediaries [1], [2]. Its widespread adoption across edge and distributed intelligence environments, however, has created a practical need for secure model exchange among mutually entrusted nodes [3]. In many real-world deployments, a node may train a *basemodel* for a specific task and later share it with other participants for inference reuse, domain adaptation, or fine-tuning. However, direct model sharing exposes several security and governance problems. First, the model artifact itself may encode valuable IP and therefore cannot be released in plaintext. Second, a centralized repository introduces a single point of failure and weakens availability in geographically distributed settings. Third, traditional access control mechanisms provide limited transparency regarding ownership, provenance, model lineage, and permission updates. These limitations become more severe when multiple nodes iteratively refine shared models and when each participant needs cryptographic proof of authorship and integrity. To our knowledge, these issues have not yet been adequately addressed, preventing widespread use of TL for greater protection against cyber threats. To improve trust and coordination in distributed learning environments, several works have explored blockchain-assisted ML systems. Early frameworks such as BlockFL demonstrated how blockchain can coordinate federated learning (FL) participants and provide incentive alignment in decentralized training environments [4]. Subsequent studies examined decentralized governance, smart-contract–mediated coordination, and the scalability and security implications of blockchain-enabled learning pipelines [5]. While these approaches enhance auditability and participant accountability, they remain largely focused on coordinating training workflows rather than managing the secure publication, discovery, and reuse of trained model artifacts. Decentralized storage systems such as IPFS enable content-addressed distribution and resilient storage of large artifacts across peer-to-peer (P2P) networks [6]. Recent blockchain–IPFS integration studies have demonstrated practical architectures for secure storage and verifiable data access in decentralized environments [7].

This work proposes a secure model-sharing protocol combining local cryptographic protection, IPFS-based distributed storage [6], and blockchain-based access control [8]. The proposed framework integrates three complementary layers for secure model dissemination in distributed environments. First, it separates data confidentiality from registry transparency by storing encrypted model artifacts in IPFS while recording searchable metadata and ownership evidence on-chain. Second, it employs a node-centric publication pipeline in which encryption, signature generation, and key retention occur locally before dissemination. Third, it introduces a blockchain-mediated access-request workflow combined with an off-chain ECDH-based key-wrapping mechanism [9], enabling model owners to selectively release the decryption key to approved requesters.

Consequently, models remain globally discoverable through their CID, while only authorized users can decrypt them. The proposed framework has been validated through experiments on a dedicated implementation of an end-to-end prototype supporting common model formats (e.g., ONNX and H5), encrypted IPFS publication, MetaMask-based transaction signing, and smart-contract-driven access management. The approach is demonstrated to be economically viable and energy efficient.

*Corresponding author: Zeeshan ALI

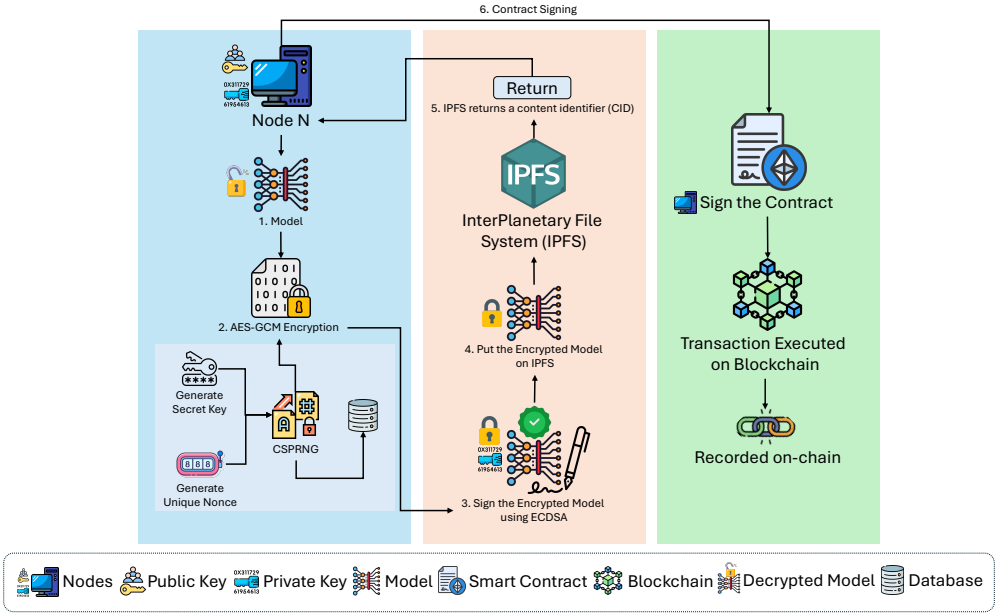


Fig. 1. Protocol architecture for end-to-end secure TL, model encryption, IPFS integration, and blockchain-based provenance.

II. Methodology

We consider a distributed TL environment composed of multiple autonomous edge nodes, as illustrated in Figure 1. Node N is an independent participant that may train a new *basemodel*, publish an encrypted model artifact, discover previously shared models, request access to a protected model, or fine-tune an existing parent model in accordance with the policy defined by the owner. The design assumption is that nodes do not fully trust one another, yet they require a common infrastructure for verifiable publication and controlled reuse of model assets. Each node maintains the four logically distinct resources listed in Table I. This decomposition is intentional: computation and sensitive key material remain local to the node, encrypted artifacts are disseminated through decentralized storage, and only compact, verifiable metadata are committed to the blockchain.

Before publishing or requesting a model, each node N_i, for $i = 1, 2, \ldots$, generates an asymmetric key pair (pk_i, sk_i) using the Elliptic-Curve Cryptography (ECC). In this work we adopted the standard curve secp256k1, defined in the Standards for Efficient Cryptography (SEC) [10]. The generated key pair is then encoded in Privacy-Enhanced Mail (PEM) format:

$$(pk_i, sk_i) \leftarrow \mathsf{KeyGen}_{\text{secp256k1}}. \tag{1}$$

The public key pk_i identifies the node N in the off-chain cryptographic layer.

When the node intends to publish a model, it creates a *local model package* $\mathcal{P} = \{M, \mathcal{A}\}$, where M is the original model artifact and $\mathcal{A}$ is the associated descriptive metadata that can include architecture descriptor, preprocessing information, training context, version, and deployment-specific attributes.

Before publication, the node generates a fresh symmetric encryption key K_M and a nonce η using a secure pseudorandom number generator (CSPRNG):

$$K_M \leftarrow \mathsf{CSPRNG}(256), \qquad \eta \leftarrow \mathsf{CSPRNG}(96). \tag{2}$$

and encrypts the plaintext model artifact M as $C_M = \mathsf{AES\text{-}GCM_Enc}(K_M, \eta, M)$, where C_M includes both the ciphertext and the authentication tag as produced by the Galois Counter Mode (GCM) mode.

Notice that the key K_M and the nonce are retained locally at the publisher node, hence, the IPFS receives only the encrypted model envelope and the decryption capability remains under the control of the originating node. To allow a requester to verify that the encrypted artifact was genuinely produced by the claimed publisher and has not been modified after publication, the publisher node computes a deterministic signature input Σ_M on the encrypted model envelope using its private key, as

$$\sigma_M = \mathsf{Sign}_{sk_i}(\mathcal{H}(\Sigma_M)), \tag{3}$$

TABLE I
Node's Resources

Resource	Description
Local model repository	Plaintext model artifacts storage
Local cryptographic state	ECC key pair and symmetric keys for model encryption
IPFS distributed storage	Encrypted model envelopes publication
Blockchain network	A MetaMask wallet to authorize smart-contract transactions for model registration and blockchain access management

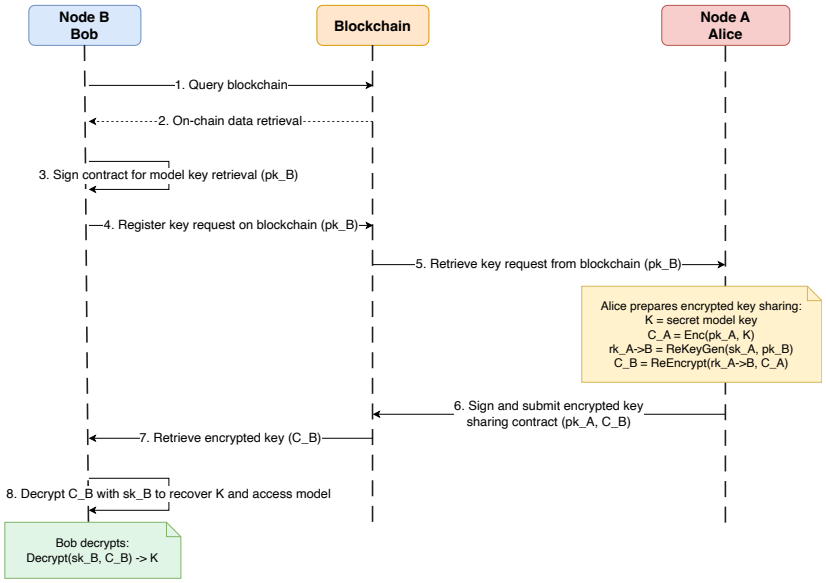


Fig. 2. Sequence diagram of the access control layer to show the interaction between data requester (Bob), data owner (Alice), and the immutable ledger for key management.

where $\mathcal{H}(\cdot)$ is instantiated with SHA-256 and the signing algorithm is Elliptic Curve Digital Signature Algorithm (ECDSA) over secp256k1. The σ_M is stored in the local database alongside K_M, the model identifier, the original file name, and the algorithm descriptors, and embedded in the encrypted envelope that is uploaded to IPFS, together with the publisher's public verification key.

After encryption and signing, the node packages the protected artifact into an encrypted envelope, denoted by

$$\mathcal{E}_M = \{\texttt{modelId}, C_M, \sigma_M, pk_i, \texttt{metadata}\}. \quad (4)$$

where `modelId` is a publisher-generated unique identifier for the model used to link the off-chain envelope to the on-chain registry record (i.e., id_M), and `metadata` is a structured descriptor that includes publication-time parameters as discussed in Table II, later registered on-chain.

The envelope is serialized as a `.enc.json` object and uploaded to IPFS, where the content is addressed by a hash-derived unique identifier $\mathsf{CID}(\mathcal{E}_M)$ and not by a server location. The CID is a cryptographic pointer to the encrypted envelope described as $\mathsf{CID}_M \leftarrow \mathsf{IPFS_Add}(\mathcal{E}_M)$, that becomes the reference used by the blockchain registry. The CID does not reveal the plaintext model parameters; it merely provides a globally resolvable handle to the encrypted content. The usage of IPFS has the advantage that, unlike centralized object storage, the dependency on a single administrative endpoint is reduced and resilience in distributed deployments is significantly improved.

A. Blockchain Registration and Global Discoverability

At this point, the publisher node registers the model on the blockchain, by providing a record for model M given by:

$$R_M = \{id_M, \mathsf{CID}_M, owner_M, t_M, task_M, domain_M \; ver_M, policy_M, parent_M\} \quad (5)$$

TABLE II
ON-CHAIN MODEL RECORD R_M DESCRIPTION

Field	Description
id_M	Unique identifier of the model
CID_M	Content Identifier (CID) of the model stored in IPFS
$owner_M$	Blockchain address of the model owner
t_M	Model type (e.g., CNN, Transformer, ResNet, LSTM, GAN)
$task_M$	Task type (e.g., Classification, Detection, Segmentation)
$domain_M$	Application domain (e.g., IIoT, Healthcare, AVs, IoT)
ver_M	Version of the model
$policy_M$	Access policy (e.g., Inference-only, TL allowed)
$parent_M$	Parent model identifier (for lineage tracking)

where the descriptive metadata and notations are presented in Table II.

The smart contract emits corresponding events when a model is deployed or when a fine-tuned descendant is registered.

Thereafter, the model becomes discoverable to other participants through blockchain queries over attributes such as task type, domain, model type, owner, version, or access policy. However, the retrieved object remains encrypted and cannot be decrypted without access to K_M. This distinction between discoverability and decryptability is central to the proposed design.

B. Blockchain-Mediated Model Access

To obtain a model for inference or fine-tuning, a requester node (e.g., Bob) first identifies a suitable model by querying the blockchain registry, as shown in the sequence diagram of Figure 2. The search criteria may include a target task, application domain, or an access policy indicating whether downstream TL is permitted. Suppose Node B (Bob) wishes to access a model published by Node A (e.g., Alice). Let the

TABLE III
MEASURED LATENCY BREAKDOWN OVER 30 RANDOMIZED MODEL RUNS.

Operation	Median (ms)	P95 (ms)	Notes
Key generation (secp256k1)	0.746	1.430	Per requester keypair generation
AES-256-GCM encryption	44.738	126.125	Strongly model-size dependent
ECDSA signing	89.473	188.457	Payload-envelope integrity
IPFS upload + CID return	329.813	876.116	Network and local IPFS daemon dependent
ECDH key wrapping	1.474	2.604	Per access grant
ECDH key unwrapping	1.337	3.282	Requester-side recovery
AES-256-GCM decryption	34.980	114.357	Decryption-side model recovery

TABLE IV
MEASURED ON-CHAIN COST BY ACTION TYPE ACROSS SEPOLIA AND POLYGON AMOY (POL) DEPLOYMENTS.

Network	Action	Avg. Gas Used	Avg. Fee (Native)	USD (Reference)	Main Cost Driver
Sepolia	Model registration ($n = 41$)	347,795	0.000701313 SepoliaETH	1.639539	Metadata string writes
Sepolia	Access request ($n = 11$)	328,858	0.000493286 SepoliaETH	1.153212	Public-key string storage
Sepolia	Access grant ($n = 5$)	635,524	0.000953286 SepoliaETH	2.228606	Encrypted-key payload storage
POL	Model registration ($n = 11$)	342,260	0.008898765 POL	0.000889	Metadata string writes
POL	Access request ($n = 10$)	327,665	0.008519290 POL	0.000851	Public-key string storage
POL	Access grant ($n = 10$)	632,142	0.016435692 POL	0.001642	Encrypted-key payload storage

selected model identifier be id_M, and let Node B's public key be pk_B. Node B submits an access request to the smart contract by providing the model identifier id_M and its public key pk_B, along with its blockchain identity. The request is recorded on-chain as $Q_{B\to A} = \{id_M, addr_B, pk_B, t_s\}$ and includes the requester's wallet address $addr_B$, public key pk_B, and timestamp t_s. After the transaction is confirmed, the model owner can view and manage the pending request through the blockchain-based request management interface. Upon receiving a valid request, the model owner Node A (Alice) evaluates the received key request. If access is approved, the owner performs node-local ECDH-based wrapping of the locally stored model key K_M using the requester's public key pk_B. This is a key-wrapping operation rather than true PRE, because the owner computes the transformation locally and no proxy re-encrypts an existing ciphertext. The owner then generates an ephemeral secret and derives a shared wrapping key, after which the model key is protected for the requester as $\widetilde{K}_{M\to B} = \mathsf{ECDH_Wrap}(K_M, pk_B)$.

The resulting protected key object contains the ephemeral public key, initialization vector, authentication tag, and ciphertext. This object is then submitted to the access-grant smart contract together with the model identifier, the requester's wallet address, and the requester's public key. The blockchain, therefore, stores an auditable grant record while the sensitive transformation itself remains off-chain. Once the key grant is published, Node B (Bob) retrieves the protected key object from the blockchain-accessible grant record. Using its own private key sk_B, Node B derives the corresponding shared secret, recovers the wrapped model key, and obtains the original symmetric decryption key by $K_M = \mathsf{ECDH_Unwrap}(\widetilde{K}_{M\to B}, sk_B)$. Node B then downloads the encrypted model envelope from IPFS using CID_M, verifies the embedded ECDSA signature using the publisher's public key pk_A, and decrypts the encrypted model payload using AES-256-GCM to obtain $M = \mathsf{AES\text{-}GCM_Dec}(K_M, \eta, C_M)$. If the model's access policy permits inference only, Node B uses the decrypted model in a read-only operational setting. If fine-tuning is allowed, Node B can adapt the model to local data and publish the resulting descendant model as a new registry entry with the original model referenced in the field $parent_M$. In this way, the blockchain captures model lineage across successive TL stages. This lineage structure enables at least three useful capabilities: transparent provenance analysis, traceability of collaborative TL, and policy-aware reuse of downstream models.

The proposed architecture satisfies the key requirements of secure model sharing in distributed TL: model confidentiality is enforced through node-local symmetric encryption, integrity and authenticity are provided through digital signatures, availability is improved through distributed IPFS storage, and ownership, discovery, access control, and lineage are made transparent through blockchain registration.

III. RESULTS AND DISCUSSION

This section evaluates the performance of the proposed secure TL pipeline with emphasis on execution latency and deployment feasibility. The implemented system integrates local cryptographic processing (AES-256-GCM, ECDSA-secp256k1, and ECDH-based key wrapping), decentralized artifact storage using IPFS, and smart-contract execution for model registration and access control.

Performance measurements were collected using an automated benchmark over $n = 30$ randomized runs with model artifacts between 20 MB and 100 MB. Each run executed the full processing pipeline used in the system: secp256k1 key generation, AES-256-GCM encryption, ECDSA-secp256k1 signing, ECDH-based key wrapping/unwrapping, AES-256-GCM decryption, and IPFS upload latency measurement. The generated dataset ranged from 24,129,721 bytes to 99,536,480 bytes, with a mean size of 60,724,332 bytes ($\approx 57.9\,\mathrm{MB}$). To make the variance explicit while keeping the table compact, we report both median and 95th percentile (P95) latencies in Table III; the randomized run design and fixed artifact-size range also support reproducibility. Measurements show

that cryptographic key-management operations introduce minimal overhead. Key generation and ECDH-based key wrapping/unwrapping complete within approximately ($\approx 1-3$ ms), even at the P95 level. In contrast, payload-dependent operations dominate the runtime. AES-256-GCM encryption and decryption scale with artifact size, while ECDSA signing reflects the cost of generating integrity proofs for the encrypted model envelope. The most significant contributor to latency is the IPFS publication stage. As shown in Table III, the IPFS upload and CID generation process reaches a median latency of ($\approx$ 329 ms) and a P95 of ($\approx$ 876 ms), reflecting network and local daemon variability. This confirms that storage and network transfer, rather than cryptographic processing, dominate the end-to-end pipeline delay. Importantly, the ECDH-based key wrapping operations remain stable and low-latency, supporting efficient access-grant workflows even for large model artifacts. To evaluate deployment behavior across blockchain environments, the protocol was executed on two Ethereum-compatible test networks: Sepolia and POL. This multi-network validation ensures that smart-contract execution, access-control workflows, and metadata registration behave consistently across different fee markets. Table IV reports the observed gas consumption and transaction costs for the primary protocol operations: model registration, access request submission, and access-grant authorization. For each action type, we collected the corresponding transactions generated during the experiments (sample size n in Table IV) and computed the averages from their on-chain receipts using a Python script. For each transaction, we read `gasUsed` and `effectiveGasPrice`, compute the native fee as fee $=$ `gasUsed` $\times$ `effectiveGasPrice`, and then report the mean `gasUsed` and mean fee over the n transactions. Gas usage remains largely consistent across networks because it is determined by the computational and storage complexity of the smart-contract logic, whereas the resulting monetary cost varies due to network-specific fee markets and token evaluations. In the experiments, POL produced significantly lower USD-denominated transaction costs than Sepolia while maintaining identical protocol behavior. The primary contributors to gas consumption are metadata string storage during model registration, requester public-key storage during access requests, and encrypted key-payload storage during access grants. For completeness, the values reported in Table IV were obtained from validation-time observations on test networks and should be interpreted as comparative engineering indicators rather than fixed production costs. Since testnet native tokens (Sepolia and POL) are not directly priced, the USD column is provided only as a reference conversion by applying contemporaneous mainnet token/USD spot valuations.

These findings demonstrate that while the protocol is functionally portable across Ethereum-compatible networks, economic efficiency is network-dependent. Consequently, practical deployment should treat blockchain selection and fee-policy tuning as part of overall system optimization alongside cryptographic workflow and storage design. The results depend on the current implementation, workload characteristics, and evaluation environment. Network-dependent measurements obtained on Sepolia and POL represent validation-time observations rather than fixed production deployment costs. To mitigate this limitation, the study reports operation-level timing, provides reproducible benchmarking scripts, and treats testnet cost figures as comparative indicators rather than deterministic production estimates.

IV. Conclusion and Future Work

This work presents a secure protocol for distributed model sharing that integrates node-local cryptography, IPFS-based distributed storage, and blockchain-based metadata governance. The framework also supports blockchain-mediated access requests and owner-authorized key sharing through an ECDH-based key-wrapping mechanism, allowing authorized nodes to obtain decryption capability without exposing publisher secrets or plaintext models on-chain. Experimental evaluation shows that cryptographic key-management operations introduce minimal latency, while payload size and IPFS publication dominate end-to-end delay. Cross-network experiments on Sepolia and POL demonstrate that the protocol is portable across Ethereum-compatible networks, although economic behavior varies by network. Future work will incorporate a mandatory pre-publication model analysis layer to mitigate malicious model dissemination.

Acknowledgement

This work is partially supported by the European Commission through the HORIZON-MSCA-SE EXARCH project (GA: 101236244).

References

[1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] A. Zeeshan, W. Tiberti, A. Marotta, and D. Cassioli, "Deep transfer learning for intrusion detection in edge computing scenarios," *IEEE Internet of Things Journal*, 2025.

[3] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[4] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.

[5] L. Bhatia and S. Samet, "Decentralized federated learning: A comprehensive survey and a new blockchain-based data evaluation scheme," in *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, 2022, pp. 289–296.

[6] J. Benet, "Ipfs - content addressed, versioned, p2p file system," arXiv preprint arXiv:1407.3561, 2014.

[7] Y. A. Zouaghi, M. Mahamdioua, A. Lahoulou, and S. Chettibi, "Privacy preserving biometric authentication based on fully homomorphic encryption, blockchain, and ipfs data storage," *Multimedia Tools and Applications*, vol. 84, no. 34, pp. 42 293–42 319, 2025.

[8] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger," Ethereum Yellow Paper, 2014, accessed: 2026-03-16.

[9] R. Haakegaard and J. Lang, "The elliptic curve diffie-hellman (ecdh)," *Online at https://koclab. cs. ucsb. edu/teaching/ecc/project/2015Projects/Haakegaard+ Lang. pdf*, vol. 5, p. 74, 2015.

[10] Certicom Research, "Standards for efficient cryptography - SEC 2: Recommended elliptic curve domain parameters," https://www.secg.org/sec2-v2.pdf.

Paper J3

A Blockchain-Enabled Framework for Confidential Model Sharing in Distributed Transfer Learning with Artifact Screening

Zeeshan Ali, Walter Tiberti, Andrea Marotta and Dajana Cassioli

Submitted to
IEEE Transactions on Information Forensics and Security

A Blockchain-Enabled Framework for Confidential Model Sharing in Distributed Transfer Learning with Artifact Screening

Zeeshan Ali, Walter Tiberti, Andrea Marotta, and Dajana Cassioli
Department of Information Engineering, Computer Science & Mathematics
University of L'Aquila, L'Aquila, Italy

***Abstract*—Securely sharing model artifacts across distributed edge nodes without exposing intellectual property remains a critical challenge in collaborative transfer learning, particularly in the absence of centralized trusted intermediaries. Novel cryptographic protocols for model screening, secure publication and controlled access in distributed environments are definitely needed. We propose an innovative framework that fulfills these requirements. In our design, after a successful screening to identify malicious issues, model artifacts are encrypted and signed locally, before publication to a decentralized, peer-to-peer distributed storage infrastructure. Then, specifically designed model records are created and registered immutably on Ethereum smart contracts. This approach preserves model confidentiality while enabling globally verifiable provenance and fine-grained access control. Downstream model reuse for inference or fine-tuning is enabled by a blockchain-mediated workflow: requesting node is granted model's access using a decryption key generated by model's owner and encrypted by requester public key, previously registered on-chain; the requester recovers the symmetric key locally by its private key, verifies the publisher's digital signature, and decrypts the model. Experimental evaluation demonstrates that the protocol is energy-efficient and economically viable, and supports transparent model lineage, selective access control, and end-to-end confidentiality, making it well-suited to secure transfer learning in trustless distributed environments.**

***Index Terms*—Transfer Learning, Secure Model Sharing, Blockchain, End-to-End Encryption**

I. Introduction

THE increasing adoption of transfer learning (TL) across edge and distributed intelligence environments has created a practical need for secure model exchange among mutually untrusted nodes [1], [2]. In many real-world deployments, a node may train a *Base Model* for a specific task and later share it with other participants for inference reuse, domain adaptation, or fine-tuning [3], [4]. However, direct model sharing exposes to several security and governance problems. First, the model artifact itself may encode valuable intellectual property and therefore cannot be released in plaintext. Second, a centralized repository introduces a single point of failure and weakens availability in geographically distributed settings. Third, traditional access control mechanisms provide limited transparency regarding ownership, provenance, model lineage, and permission updates. Fourth, the shared model may contain malicious content, thereby posing a potential threat to the user. These issues become more severe when multiple nodes iteratively refine shared models and when each participant needs cryptographic proof of authorship and integrity.

Several decentralized learning solutions have been proposed in the literature [5]. Training-time coordination is the primary focus of the solutions proposed in [6], [7], based on federated learning (FL), and of the secure aggregation approach in [8].

Generic decentralized storage and access control, such as, e.g., integrated blockchain and InterPlanetary File System (IPFS) architectures, have also been investigated, but the end-to-end lifecycle management of *model artifacts* is still under-addressed. In particular, prior systems rarely combine encrypted artifact publication, globally discoverable and verifiable metadata, cryptographic provenance/lineage, and owner-authorized, requester-specific key release required for downstream TL reuse [9].

Moreover, model sharing pipelines increasingly face supply-chain risks stemming from unsafe serialized artifacts, yet screening is typically treated as an external best practice rather than an integrated pre-publication gate [10].

This work addresses these challenges by proposing a secure model sharing protocol that combines local node-side cryptographic protection, distributed storage, and blockchain-based metadata registration and access coordination [11].

The proposed framework integrates three complementary layers for secure model dissemination in distributed environments. First, it separates data confidentiality from registry transparency by storing encrypted model artifacts in a distributed storage file system (DSFS), *de facto* remaining in the local node, while recording searchable metadata and ownership evidence on-chain. Second, it employs a node-centric publication pipeline in which screening/audit, encryption, signature generation, and key retention occur locally before dissemination. Third, it introduces a blockchain-mediated access-request workflow combined with an off-chain key-sharing mechanism, enabling model owners to selectively release the decryption key to approved requesters without republishing the model artifact. Moreover, by leveraging the on-chain lineage and descriptive metadata, the framework supports *model synchronization* in which nodes can periodically check for relevant, lineage-linked model updates and (optionally) upgrade deployed models in a provenance-preserving way. Consequently, models remain globally discoverable through their DSFS unique content identifier (CID), while only authorized users can decrypt them.

This work is partially supported by the European Commission through the HORIZON-MSCA-SE EXARCH project (GA: 101236244).

(Authors Email: zeeshan.ali1@graduate.univaq.it, walter.tiberti@univaq.it, andrea.marotta@univaq.it, dajana.cassioli@univaq.it)
Corresponding author: Zeeshan ALI

The main contributions of this work are summarized as follows:

- We propose a secure distributed model sharing framework where (1) model artifacts are encrypted locally, stored in a DSFS, and referenced on-chain through immutable blockchain metadata; (2) a pre-publication model screening/audit step evaluates plaintext model artifacts before encryption and blocks unsafe or malformed models from entering the sharing pipeline; and (3) a cryptographic publication pipeline ensures confidentiality, integrity, and publisher authenticity.
- We develop a blockchain-coordinated access workflow that supports model discovery, ownership verification, access requests, lineage tracking through parent–child relationships and model synchronization, allowing nodes to detect relevant lineage-linked updates via on-chain metadata and optionally switch to newer model versions.
- We implement an end-to-end prototype supporting common model formats (e.g., Open Neural Network Exchange (ONNX) and Hierarchical Data Format version 5 (H5)), encrypted IPFS publication, transaction signing by the widely used crypto wallet MetaMask, and smart-contract-driven access management. The cryptographic publication pipeline is implemented by the cryptographically secure Advanced Encryption Standard in Galois/Counter Mode (AES-256-GCM) [12] based on Cryptographically Secure Pseudorandom Random Number Generator (CSPRNG), combined with digital signatures based on Elliptic Curve Digital Signature Algorithm (ECDSA) over the Standards for Efficient Cryptography prime-field 256-bit Koblitz curve (secp256k1) [13]. Approved requesters obtain the model decryption key through an off-chain, requester-specific Elliptic-Curve Diffie–Hellman (ECDH)-based key-wrapping mechanism, which protects the symmetric model key without exposing the publisher's secret material or the plaintext model.

The remainder of this paper is organized as follows. Section II reviews the relevant literature on TL. Section III presents the system overview and the node's characteristics, while the secure model sharing framework is described in Section IV. Section V describes the on-chain model retrieval and access mechanisms, and Section VI introduces the blockchain-based model synchronization process. The prototype implementation is detailed in Section VII, and the experimental results and discussion are reported in Section VIII. Finally, Section IX concludes the paper and outlines future research directions.

II. Related Work

Secure and distributed machine learning (ML) has been widely studied through FL, where participants collaboratively train models by exchanging parameter updates rather than raw data. Foundational work introduced communication-efficient decentralized optimization for large-scale ML [5], while later studies analyzed practical challenges related to privacy, system heterogeneity, and robustness [2]. Secure aggregation techniques further reduced information leakage by ensuring that only aggregated model updates are visible to the coordinator [8]. These approaches enable privacy-preserving collaborative training, but primarily focus on iterative parameter exchange and do not address persistent, searchable, and policy-governed sharing of complete model artifacts for downstream TL.

To improve trust and coordination in distributed learning environments, several works have explored blockchain-assisted ML systems. Early frameworks such as BlockFL demonstrated how blockchain can coordinate FL participants and provide incentive alignment in decentralized training environments [6]. Subsequent studies examined decentralized governance, smart-contract–mediated coordination, and the scalability and security implications of blockchain-enabled learning pipelines [7]. While these approaches enhance auditability and participant accountability, they remain largely focused on coordinating training workflows rather than managing the secure publication, discovery, and reuse of trained model artifacts.

Parallel research has investigated decentralized storage and cryptographic mechanisms for secure data sharing. Blockchain platforms such as *Ethereum* provide programmable smart contracts that enable tamper-evident policy enforcement and transparent access control [11]. Complementarily, decentralized storage systems such as IPFS enable content-addressed distribution and resilient storage of large artifacts across peer-to-peer (P2P) networks [14]. Recent blockchain–IPFS integration studies have demonstrated practical architectures for secure storage and verifiable data access in decentralized environments [9]. At the cryptographic level, authenticated encryption schemes such as AES-GCM and elliptic-curve cryptography are widely used for efficient confidentiality, integrity protection, and key ownership verification [12], [13]. Proxy Re-Encryption (PRE) techniques provide a distinct form of delegated access control in which an intermediate proxy transforms ciphertexts for authorized recipients without learning the plaintext or secret keys [15]. Our framework does not rely on such proxy-side ciphertext transformation; instead, it uses node-local ECDH-derived key wrapping to protect the symmetric model key for each approved requester.

Table I summarizes the key elements of the main representative decentralized ML approaches and the proposed architecture, focusing on secure model artifact publication, verifiable on-chain metadata, delegated key access, and TL reuse support.

III. System Overview and Node Model

We consider a distributed TL environment composed of multiple autonomous edge nodes, each with the following capabilities:

- train a new *Base Model*,
- publish an encrypted model artifact,
- discover previously shared models,
- request access to a protected model,
- fine-tune an existing parent model in accordance with the policy defined by the owner.

The design assumption is that nodes do not fully trust one another, yet they require a common infrastructure for verifiable publication and controlled reuse of model assets.

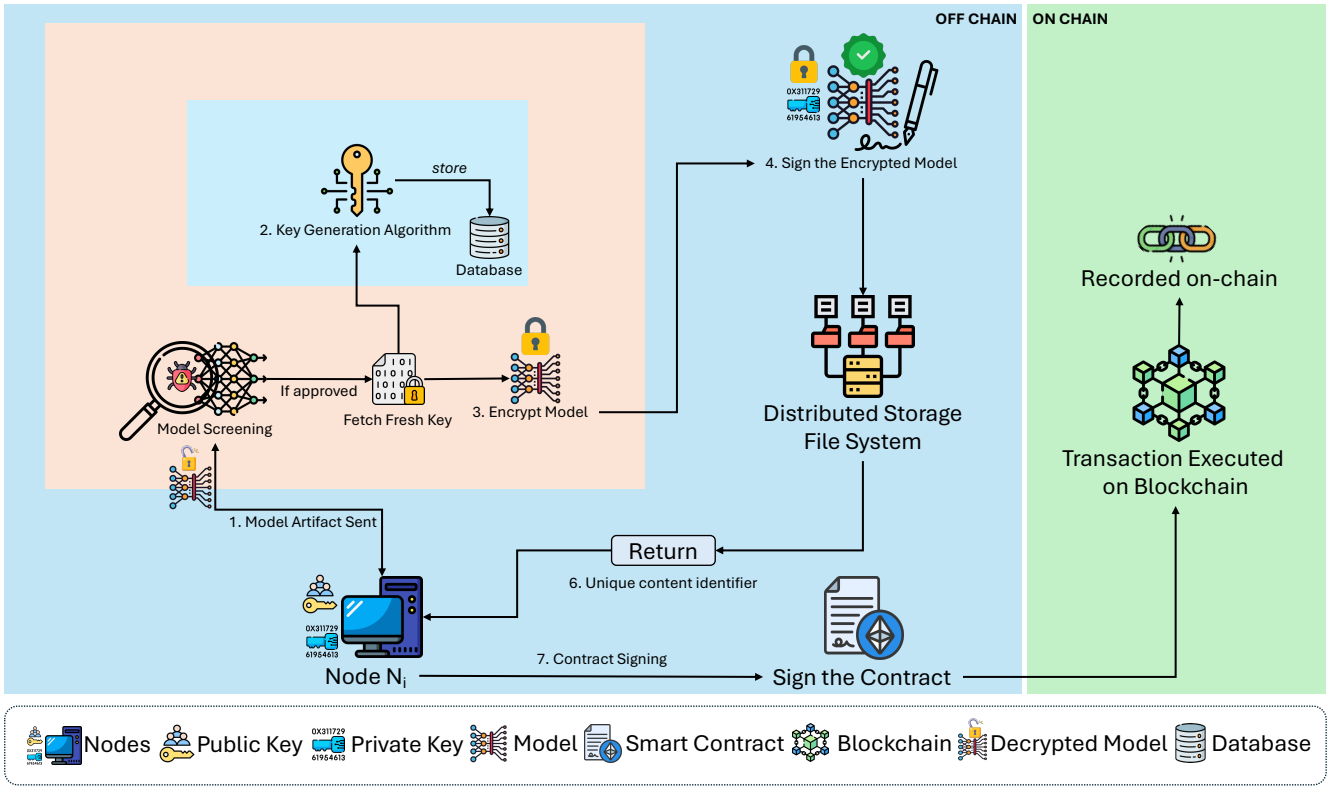


Fig. 1. General architecture of the proposed framework for end-to-end secure TL, including pre-publication model screening, model encryption, distributed storage integration, and blockchain-based provenance.

Each node maintains four logically distinct resources, i.e., (1) a local model repository in which plaintext model artifacts are stored before publication, (2) a local cryptographic state, including its elliptic-curve public/private key pair and the symmetric keys generated for model encryption, (3) interaction with a distributed storage layer to publish encrypted model envelopes, and (4) participation in a blockchain network through its wallet, which is used to authorize smart-contract transactions for model registration and access management on the blockchain. This decomposition is intentional: computation and sensitive key material remain local to the node, encrypted artifacts are disseminated through decentralized storage, and only compact, verifiable metadata are committed to the blockchain.

Each node N_i generates an asymmetric key pair denoted as:

$$(pk_i, sk_i) \leftarrow \mathsf{KeyGen}. \tag{1}$$

The private key sk_i is used to produce the signatures over encrypted model envelopes generated by node N_i, retained locally and never published, while the public key pk_i is published to enable signature verification and to facilitate requester-specific key exchange.

In parallel, the node also controls a blockchain account, which signs the smart-contract transaction that records model metadata on-chain. This dual-identity design separates the node's artifact-level cryptographic operations from the blockchain transaction authorization layer while still binding both to the same publisher node.

IV. Secure Model Sharing

The secure end-to-end TL model sharing is based on the general architecture of Figure 1, where we distinguish two layers:

TABLE I
COMPARISON OF EXISTING DECENTRALIZED ML SYSTEMS WITH THE PROPOSED ARCHITECTURE.

System	Encrypted Model	On-chain Metadata	Delegated Key Access	TL Reuse
Federated Learning (FL) [5]	×	×	×	×
Secure Aggregation FL [8]	×	×	×	×
BlockFL [6]	×	✓	×	×
Blockchain-based FL Surveys [7]	×	✓	×	×
Blockchain + IPFS Storage [9]	✓	✓	×	×
Proposed Protocol	✓	✓	✓	✓

1) the *model distribution layer*, labeled "OFF CHAIN," which includes pre-publication model screening, model encryption, and DSFS integration;
2) the *blockchain model registration layer*, labeled "ON CHAIN," which includes registration of model metadata, ownership information, versioning records, and cryptographic hashes on the blockchain.

The entire process, including both the model distribution and the publication transaction, is summarized in Algorithm 1.

Algorithm 1 Secure model distribution and registration workflow

Require: Local model file from the node repository; node key pair; nodes' wallet; access to a DSFS; blockchain registry

Ensure: Encrypted model is stored off-chain and registered on-chain with its CID and metadata

Off-chain

Select a local model file from the node repository
Screen the plaintext artifact and abort if the artifact is rejected
Generate a fresh symmetric key and nonce
Encrypt the model using a freshly generated key, and sign the resulting encrypted envelope using the node's private key
Package the encrypted payload as an envelope in a suitable format
Upload the encrypted envelope to a DSFS
Receive the resulting CID

On-chain

Use the wallet to sign the blockchain transaction that registers the model and its attributes
Wait for block confirmation
Register the model record in the immutable blockchain registry
return The encrypted model is publicly retrievable through the selected DSFS using the registered CID

A. Model Distribution Layer (Off Chain)

When a node N_i intends to publish a model artifact M, it first selects or uploads the unencrypted model file into its local storage, where it is stored along with its own descriptive metadata $\mathcal{A}$. The model M represents the learned parameters, while $\mathcal{A}$ captures the relevant contextual and descriptive information listed under $\mathcal{A}$ in Table II. The format is not imposed, since the methodology is format-agnostic and can be applied to any serialized model representation.

Before any cryptographic processing, the publisher node submits the *plaintext* model artifact M to a local *model screening* module that evaluates the artifact against a predefined acceptance policy. The purpose of screening is to prevent unsafe or malformed artifacts from entering the sharing pipeline and to ensure that only valid models proceed to encryption and publication. This screening step is executed before key generation and encryption; hence, if the artifact fails screening, the publication process is aborted and the artifact is rejected locally.

At this stage, the plaintext model remains exclusively on the publisher node. It is not sent to the blockchain and is not uploaded to DSFS without first being encrypted. This design reduces exposure during the publication phase and ensures that confidentiality is established before model dissemination.

If the artifact is approved, the node proceeds to generate a fresh key K_M and a nonce η using a random number generator (RNG) and then encrypts the plaintext model artifact M, using an encryption algorithm $\mathsf{E}(\cdot)$, to obtain the ciphertext C_M as

$$C_M = \mathsf{E}_{K_M,\eta}(M), \tag{2}$$

This process takes as input a secret key K_M, which controls access to the model, and a nonce η which ensures that each encryption instance is unique. Note that the decryption key K_M remains under the control of the originating node. This local-retention design is fundamental to selective sharing because it allows the owner to grant or deny future access to the decryption key independently of the public availability of the encrypted artifact.

However, confidentiality alone is insufficient in a distributed environment because the model sharing protocol should allow the requester to verify that the encrypted artifact was genuinely produced by the claimed publisher and was not modified after publication. For this reason, the publisher node N_i computes a deterministic signature input over the encrypted model envelope C_M and signs that input using its private key sk_i, thus binding the publisher's identity to the artifact, and protecting the signed fields against undetected modification.

TABLE II
DESCRIPTIVE METADATA FOR THE ON-CHAIN RECORD OF MODEL M.

Notation		Description
$\mathrm{id_M}$		Unique Model Identifier
CID_M		Unique Content Identifier on DSFS
$\mathrm{addr_A}$		Model owner wallet address (e.g., Alice)
$\mathcal{A}$	$\mathrm{t_M}$	Model type (e.g., Convolutional Neural Network (CNN), Transformers, Residual Network (ResNet), Long Short-Term Memory (LSTM), Generative Adversarial Network (GAN), etc.)
	$\mathrm{task_M}$	Task type (e.g., Classification, Detection, Segmentation, etc.)
	$\mathrm{domain_M}$	Application domain (e.g., Healthcare, IoT, Autonomous Vehicles, etc.)
	$\mathrm{ver_M}$	Version $(v_1, \ldots, v_n)$
	$\mathrm{policy_M}$	Access policy (e.g., inference-only or TL allowed)
	$parent_M$	Parent model identifier for lineage tracking (optional) (e.g., MODEL_ID_BaseModel)

The digital signature results:

$$\sigma_M = \mathsf{Sign}_{sk_i}\left(\mathcal{H}(C_M)\right), \tag{3}$$

where $\mathcal{H}(\cdot)$ is a cryptographic hash function that outputs a fixed-length digest of its input, and Sign_{sk_i} denotes the signing algorithm using the private key sk_i of node N_i.

The resulting signature σ_M is stored locally along with the decryption key K_M and the unique model identifier $\mathrm{id_M}$, generated by the publisher. This step provides two security guarantees. First, any recipient can verify integrity prior to decryption by checking the signature over the encrypted envelope. Second, since the signature is computed over the encrypted representation rather than over only the plaintext model hash, the recipient can detect tampering in the exact object stored in distributed storage.

Next, node N_i creates the encrypted envelope $\mathcal{E}_M$ to be serialized and uploaded to the DSFS:

$$\mathcal{E}_M = (\mathrm{id_M}, C_M, \eta, \mathsf{alg}, \sigma_M, pk_i, \mathcal{A}), \tag{4}$$

where pk_i is the publisher's public verification key and alg denotes the encryption algorithm used to produce the ciphertext. Here, the identifier $\mathrm{id_M}$ links the off-chain envelope to the corresponding on-chain registry record. The parameters that remain on-chain, capturing the persistent registry information associated with the published model artifact, are summarized in Table II.

Unlike centralized object storage, DSFS reduces reliance on a single administrative endpoint and improves resilience in distributed deployments. Basic requirements for the chosen DSFS are that it shall operate in a P2P manner and the content should be addressed not by a specific server location, but using a hash-derived unique identifier described as

$$\mathsf{CID}_M \leftarrow \mathsf{DSFS_Add}(\mathcal{E}_M). \tag{5}$$

This CID shall be a cryptographic pointer to the encrypted envelope, to become the reference used by the blockchain registry. The CID does not reveal the plaintext model parameters; it merely provides a globally resolvable handle to the encrypted content. Therefore, even though the encrypted model can be publicly discoverable, the confidentiality of the model remains protected by the symmetric key K_M.

This property is particularly advantageous for secure model sharing, as the artifact can be retrieved from any participating node or gateway as long as the content remains available, while the identifier remains stable and inherently tamper-evident.

B. Blockchain Model Registration Layer (On Chain)

Once the encrypted envelope has been stored in the DSFS and the corresponding CID has been obtained, the publisher node registers the model on the blockchain as:

$$R_M = \{\mathrm{id_M}, \mathsf{CID}_M, \mathrm{addr_A}, \mathcal{A}\}, \tag{6}$$

where the descriptive parameters for the on-chain record are described in Table II. The smart contract emits corresponding events when a model is deployed or when a fine-tuned descendant is registered.

After confirmation of the registration transaction, the model becomes globally discoverable to other participants through blockchain queries over attributes such as task type, domain, model type, owner, version, or access policy. Because the registry stores the CID, any node in the network can retrieve the encrypted envelope from DSFS once it gets the CID from the blockchain. However, the retrieved object remains encrypted and cannot be decrypted without access to K_M.

This distinction between discoverability and decryptability is central to the proposed design. The model registry is intentionally public and searchable, thereby supporting transparency and model reuse. In contrast, model confidentiality is preserved because the blockchain reveals only metadata and storage pointers, not the plaintext model or its decryption key. As a result, the architecture supports broad visibility of available models without sacrificing access control over the underlying asset.

V. On Chain Model Retrieval and Access

The access control layer is based on the architecture illustrated in Figure 2, showing the interactions between data requester (Bob), data owner (Alice), and the immutable ledger for key management.

To obtain a model for inference or fine-tuning, the requester Node B (e.g., Bob) first identifies a suitable model by querying the blockchain registry. The search criteria may include a target task, application domain, or an access policy indicating whether downstream TL is permitted.

Suppose Node B wishes to access the model identified by $\mathrm{id_M}$ published by Node A (e.g., Alice). Hence, Node B submits an access request to the smart contract which is recorded on-chain in the form:

$$Q_{B\to A} = \{\mathrm{id_M}, \mathrm{addr_B}, pk_B\}. \tag{7}$$

where $\mathrm{id_M}$ is the selected model identifier, pk_B and $\mathrm{addr_B}$ are Node B's public key and blockchain wallet address, respectively.

After the transaction is confirmed, Node A can view and manage the pending request through the blockchain-based request management interface.

If Node A approves the access, it transforms the locally stored model key K_M into a requester-specific protected payload using the requester's public key pk_B. Notice that: 1) the model is not re-uploaded, 2) the symmetric key K_M originally generated and used by Node A to store the encrypted model C_M is not disclosed.

Specifically, the owner derives an ephemeral shared wrapping key $\widetilde{K}_{M\to B}$ based on K_M and encrypted with pk_B, after which the model key is protected for the requester as

$$\widetilde{K}_{M\to B} = \mathsf{Wrap}_{pk_B}(K_M). \tag{8}$$

where $\mathsf{Wrap}(\cdot)$ denotes authenticated key wrapping, implemented in the prototype with AES-256-GCM over an ECDH-derived secret. The object $\widetilde{K}_{M\to B}$ is then submitted to the access-grant smart contract together with the model identifier $\mathrm{id_M}$, the requester's wallet address $\mathrm{addr_B}$, and the requester's public key pk_B as

$$\mathcal{R}_{M\to B} = \left(\mathrm{id}_M,\ \mathrm{addr}_B,\ pk_B,\ \widetilde{K}_{M\to B}\right). \tag{9}$$

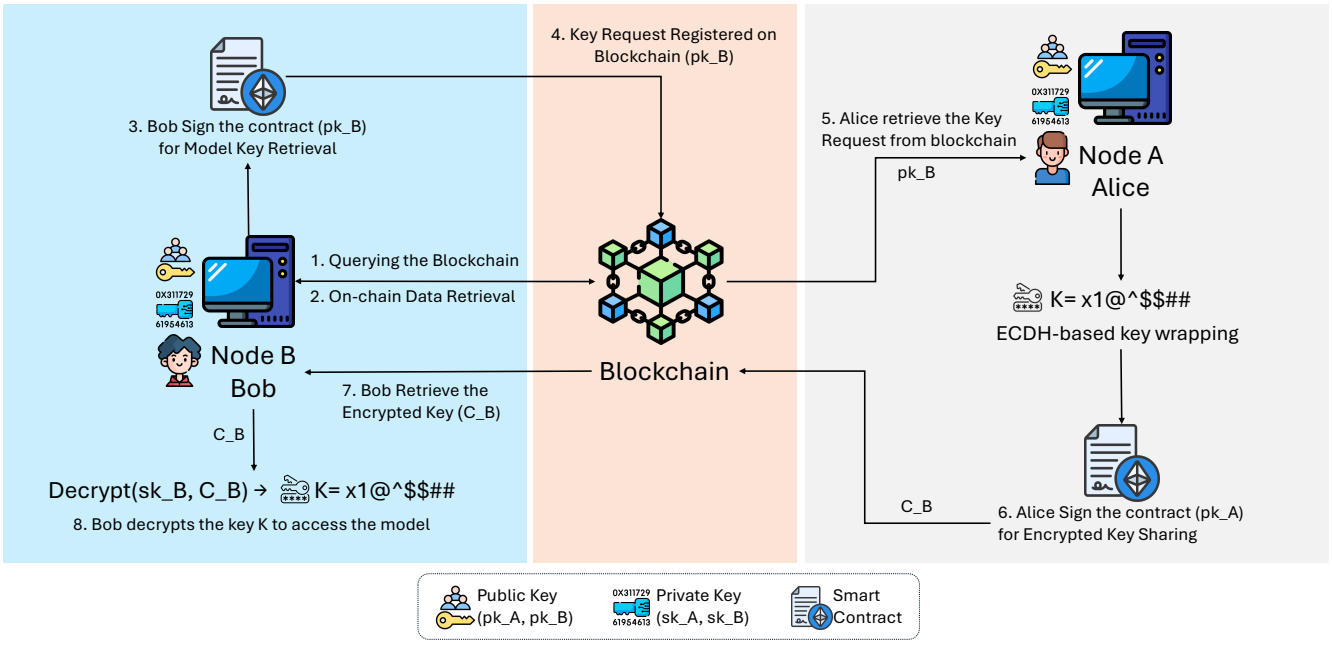


Fig. 2. Model retrieval and access control protocol: demonstrating the interaction between data requester (Bob), data owner (Alice), and the immutable ledger for key management.

The blockchain, therefore, stores an auditable grant record while the sensitive transformation itself remains off-chain. This mechanism has the advantage of preserving the owner's control over the original key material because only a requester-specific protected representation is released. Moreover, the grant remains publicly auditable at the metadata level, since the blockchain records that access was granted and to whom, without disclosing the encrypted model key.

From the created blockchain-accessible grant record, Node B can now retrieve the protected key object, derive the corresponding shared secret, recover the wrapped model key, and obtain the original symmetric decryption key K_M, using its own private key sk_B and the decryption procedure $\mathsf{Unwrap}(\cdot)$:

$$K_M = \mathsf{Unwrap}_{sk_B}(\widetilde{K}_{M\to B}). \tag{10}$$

Node B then downloads the encrypted model envelope from DSFS using CID_M, verifies the embedded signature using the publisher's public key pk_A, and decrypts the encrypted model payload using the corresponding symmetric key:

$$M = \mathsf{D}_{K_M,\eta}(C_M). \tag{11}$$

where $D(\cdot)$ denotes the decryption operation. If the model's access policy permits inference only, Node B uses the decrypted model in a read-only operational setting. If fine-tuning is allowed, Node B may adapt the model to local data and publish the resulting descendant model as a new registry entry with the original model referenced in the $\mathrm{parent_M}$ field. In this way, the blockchain captures model lineage across successive TL stages, as explained in the following section.

VI. Blockchain-based ML Model Synchronization

An important methodological feature of the architecture is the explicit parent-child linkage across model generations. A *Base Model* is registered with an empty parent identifier, whereas a *Fine-Tuned Model* references the identifier of the parent model from which it was derived. This applies whether the parent model originated at the same node or at a different collaborating node. Therefore, if Node B fine-tunes a model first published by Node A, the resulting blockchain record stores Node A's model identifier as the parent of Node B's new model.

This lineage structure enables at least three useful capabilities: transparent provenance analysis, traceability of collaborative TL, and policy-aware reuse of downstream models. Since each new model is separately encrypted, signed, uploaded to DSFS, and registered on-chain, the confidentiality and provenance guarantees of the base publication process are preserved for all descendant models as well.

Beyond passive traceability, the lineage record also enables *model synchronization* for nodes running deployed models. As illustrated in Figure 3, a node executing an active model may periodically run (e.g., every $\approx 10\,\mathrm{s}$) an *update checker* that queries the on-chain registry for newly registered models.

If a relevant update is detected, the node is notified and can optionally approve the upgrade; upon approval, the node initiates the standard retrieval/access workflow described in Sec. V, to obtain the encrypted artifact and its decryption capability, and then switches execution from the current model version to the new one.

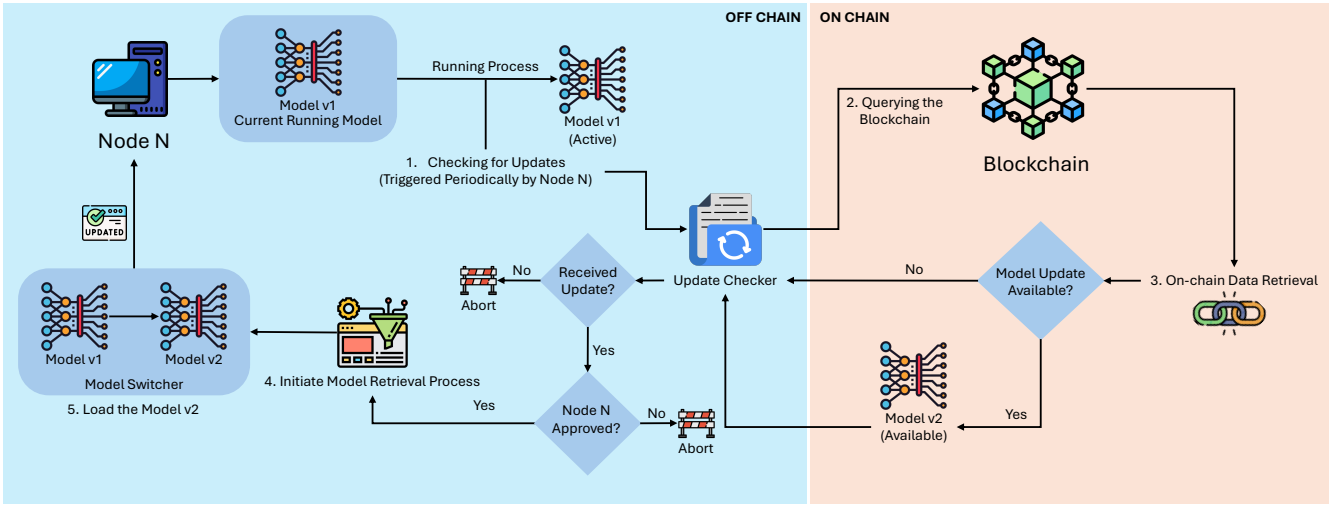


Fig. 3. Workflow for Blockchain-based ML Model Synchronization. The architecture details the transition from Model v1 to v2, highlighting the automated update checker that bridges off-chain execution with on-chain data retrieval to ensure model provenance.

VII. Prototype Implementation

The proposed architecture satisfies the key requirements of secure model sharing in distributed TL: **model confidentiality** is enforced through node-local symmetric encryption; **integrity** and **authenticity** are provided through digital signatures, **availability** is improved through DSFS, and **ownership**, **discovery**, **access control**, and **lineage** are made transparent through blockchain registration.

We implemented the proposed architecture in a prototype to test the performance of the proposed solution and analyze its economic viability and energy efficiency. The implemented system, shown in Figure 4, integrates model screening (*ModelAudit* by Promptfoo), local cryptographic processing (AES-256-GCM, ECDSA-secp256k1, and ECDH-based key wrapping), decentralized artifact storage using IPFS, and smart-contract execution for model registration and access control.

The local storage of model artifacts are implemented using formats like, e.g,. ONNX, H5, .pkl etc, but the methodology is format-agnostic and can be extended to other serialized model representations.

In the prototype, we implement the model screening using *ModelAudit* via Promptfoo's wrapper command `promptfoo scan-model` [10]. ModelAudit is a lightweight static security scanner for ML model files that performs format-aware checks to detect risks such as malicious code embedded in serialized artifacts (e.g., pickled payloads), unsafe framework constructs (e.g., suspicious TensorFlow operations or Keras Lambda layers), embedded executables within archives/binaries, hidden credentials, and suspicious network indicators (URLs/IPs) that could enable exfiltration, among other indicators [10]. Table III summarizes representative public and free tools for artifact screening, and serves to motivate the choice of a lightweight, local, pre-publication screening gate. We selected *ModelAudit* because it aligns closely with this trust boundary: it can be executed locally as a deterministic *gating* step, it performs format-aware checks beyond a single serialization family.

The key generation is performed locally using the `Node.js` cryptographic API. The keys are encoded in `Privacy-Enhanced Mail (PEM)` format, where

TABLE III
PUBLIC AND FREE TOOLS FOR *static* MODEL ARTIFACT SCREENING, AND THEIR SUITABILITY AS A PRE-PUBLICATION GATE IN OUR PROTOCOL.

Tool	Coverage	Key security checks / outputs	Best used for	Fit†
ModelAudit (Promptfoo) [10]	Multi-format; local paths and Hugging Face URLs	Format-aware static checks (e.g., malicious serialization patterns, unsafe framework constructs, embedded executables/archives, secrets, and network indicators)	Supply-chain screening of model files before deployment/publication	**High**
ModelScan (Protect AI) [16]	H5, Pickle, SavedModel	Detects unsafe code and serialization attack patterns across common model formats	Baseline local scanning for known risky serialization behaviors	Medium
Picklescan [17]	Pickle/PyTorch archives; local/URL/HF	Flags suspicious imports/globals in pickled artifacts to reduce unsafe deserialization risk	Quick, pickle-focused triage for PyTorch ecosystems	Medium
Hugging Face Hub pickle vetting [18]	Hub-hosted pickle files	Surfaces the imports used by pickled artifacts and encourages safer alternatives (e.g., SafeTensors)	Lightweight visibility when consuming Hub-hosted pickled models	Low-Medium

†*Fit* is judged against our pipeline requirement: a local, file-based screening step on plaintext artifacts before encryption.

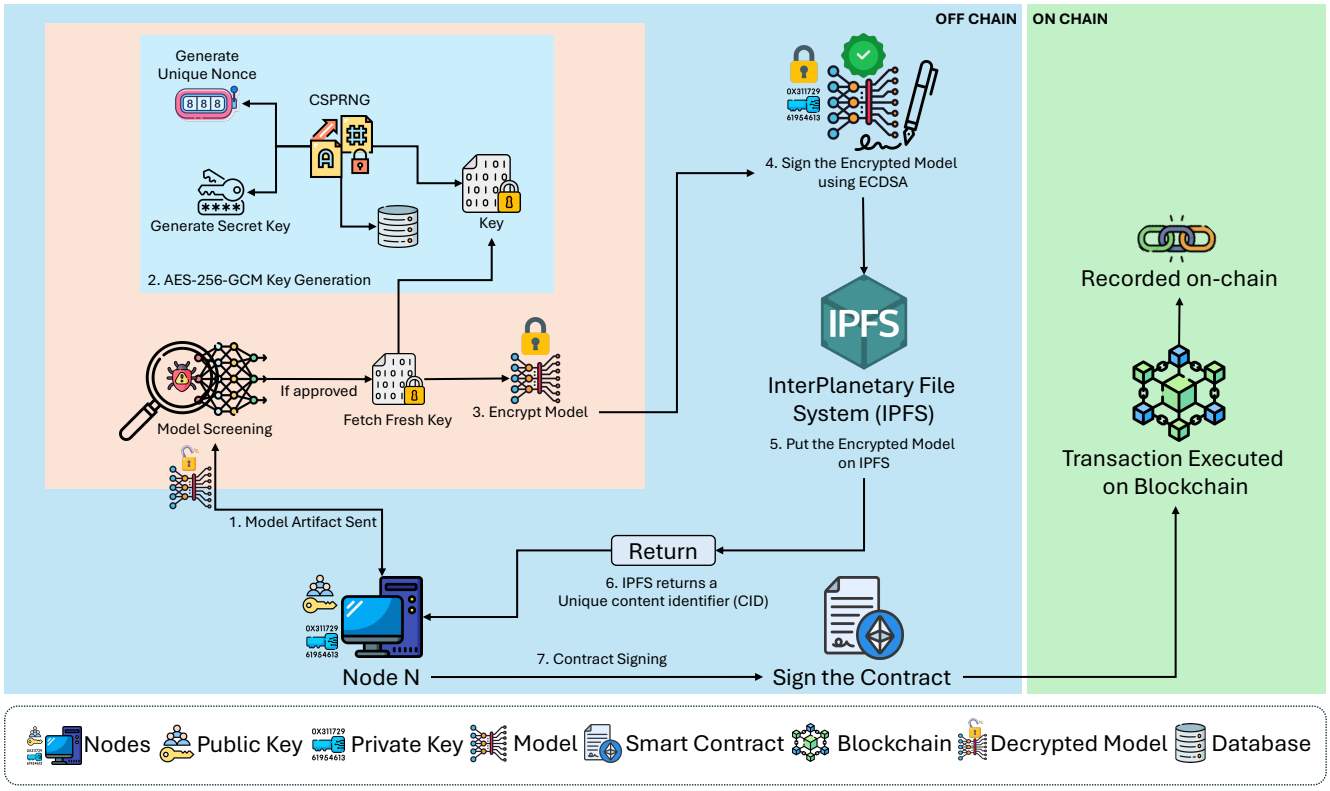


Fig. 4. Protocol implementation for end-to-end secure TL, including pre-publication model screening using *ModelAudit*, model encryption (*AES-256-GCM*), signing the encrypted envelope (*ECDSA*), P2P DSFS integration (*IPFS*), and blockchain-based provenance.

the public key follows the `SubjectPublicKeyInfo (SPKI)` structure and the private key follows the `Public-Key Cryptography Standards v8 (PKCS8)` format.

The model is encrypted using AES-256-GCM, offering a strong balance of performance and security that makes it particularly well-suited for large files such as ML models. AES with a 256-bit key length provides a very high level of resistance against brute-force attacks, ensuring long-term confidentiality even against powerful adversaries. In this implementation, the encryption key is generated using a CSPRNG, ensuring high entropy and unpredictability. Additionally, a unique nonce (initialization vector) is used for each encryption process to maintain security guarantees. These values are generated as:

$$K_M \leftarrow \mathsf{CSPRNG}(256), \qquad \eta \leftarrow \mathsf{CSPRNG}(96). \tag{12}$$

The plaintext model artifact M is then encrypted as

$$C_M = \mathsf{AES\text{-}256\text{-}GCM_E}_{K_M,\eta}(M). \tag{13}$$

Being GCM highly parallelizable, it performs exceptionally well on modern hardware, enabling fast processing of large files without sacrificing security. Together, AES-256 and GCM, combined with secure key generation and proper use of nonce, provide a robust and efficient solution that ensures both confidentiality and integrity in a single streamlined step.

The node stores the raw key material and associated cryptographic metadata in its local `PostgreSQL` database [19]. This choice is motivated by PostgreSQL's strong reliability and data integrity guarantees, which are essential for safeguarding sensitive cryptographic assets. Its support for Atomicity, Consistency, Isolation, and Durability (ACID) compliant transactions ensures consistency during key storage and retrieval operations, preventing corruption or partial updates that could compromise security. In addition, PostgreSQL provides advanced indexing and query optimization capabilities, enabling efficient and precise retrieval of cryptographic metadata even at scale. Its mature ecosystem, extensibility, and proven performance in high-throughput environments further contribute to its suitability for securely managing structured cryptographic data. Collectively, these properties make PostgreSQL a robust and dependable backend for handling sensitive key material in distributed systems.

The signature process, as defined in (3), begins with the deterministic construction of C_M. A cryptographic digest is then computed as $H = \mathsf{SHA256}(C_M)$, ensuring a fixed-length, collision-resistant representation of the data. This digest is subsequently signed using the ECDSA over the secp256k1 curve, producing the signature σ_M.

To verify model authenticity, a requester reconstructs C_M, recomputes the digest $H = \mathsf{SHA256}(C_M)$, and verifies σ_M using the publisher's public key. Successful verification guarantees both the integrity of the envelope and the authenticity

of its origin, ensuring that the model has not been altered and was indeed signed by the legitimate publisher. The choice of ECDSA over secp256k1 is motivated by its strong security guarantees and operational efficiency. ECDSA achieves a high level of cryptographic security with relatively small key sizes, reducing storage and communication overhead compared to traditional schemes such as Rivest–Shamir–Adleman (RSA). The secp256k1 curve is widely adopted and optimized for performance, enabling fast signature generation and verification while maintaining resistance against known cryptographic attacks. These properties make it particularly suitable for decentralized and resource-constrained environments, where efficient, high-throughput authentication and non-repudiation are essential.

The encrypted model envelope $\mathcal{E}_M$ in (4) is serialized as a `.enc.json` object before the DSFS upload. `JSON` provides a structured, portable, and interoperable format for storage and transmission. This simplifies parsing, validation, and cross-platform compatibility, while ensuring that all necessary cryptographic components are consistently preserved and easily retrievable during decryption and verification.

To support decentralized storage of encrypted model artifacts, we adopt IPFS [14], a P2P DSFS that enables content-addressable storage and retrieval. Unlike traditional centralized storage solutions, IPFS identifies files based on their cryptographic hash (CID), ensuring data integrity and tamper resistance. This property is particularly important for ML artifacts, as it guarantees that any retrieved model exactly matches the originally uploaded encrypted version. Additionally, IPFS improves availability and resilience by distributing data across multiple nodes, reducing reliance on a single point of failure. The use of IPFS also aligns with the decentralized nature of blockchain systems, as only the lightweight CID needs to be stored on-chain, minimizing storage costs while enabling efficient and verifiable access to off-chain data. This combination of on-chain metadata and off-chain storage provides a scalable, secure, and cost-effective solution for managing encrypted model artifacts.

The on chain record described in (6) is submitted to the smart contract of the model registry via the *MetaMask wallet* [20], allowing the transaction to be transparently verified and immutably recorded on-chain, thus facilitating secure user interaction with the blockchain. MetaMask also offers browser extension/mobile app to let users interact with blockchain networks directly from their device. MetaMask acts as a non-custodial wallet that manages users' private keys locally, ensuring that sensitive credentials are never exposed to the application or external servers. It provides a standardized and widely adopted mechanism for signing and submitting transactions to smart contracts, thereby enhancing both security and usability. Its compatibility with multiple blockchain networks and support for cryptographic signing, ensure that all transactions are authenticated, tamper-resistant, and transparently verifiable on-chain. This design choice simplifies user access while maintaining strong security guarantees in a decentralized environment.

In our implementation, the node's access request to the smart contract described by (7) includes only the model identifier and the requester's public key as the required fields in the current contract design.

The requester-specific key sharing functionality is realized through a secp256k1 ECDH-derived shared secret combined with AES-256-GCM wrapping of the model key. This is not PRE: no proxy receives a re-encryption key, and no third party transforms ciphertext from one recipient to another. Instead, the owner performs the wrapping locally after approving an access request, and the requester unwraps the protected model key locally with its own private key. ECDH allows the two parties to independently compute a common secret over an insecure channel using their private-public key pairs, without ever transmitting the secret directly. This design provides efficient key distribution, strong elliptic-curve-based security, reduced computational overhead compared to re-encrypting large model artifacts, and scalable requester-specific access to encrypted models in distributed systems.

Finally, update notifications are not broadcast for every new registry entry; instead, the update checker filters candidates by matching the active model's descriptive attributes stored on-chain, i.e., *task type*, *application domain*, and *model type*. An update is considered relevant only when a newly available model record matches these attributes and is linked to the same lineage (e.g., a fine-tuned descendant referencing the current model as its parent).

VIII. Results and Discussion

This section presents the performance evaluation of the proposed secure TL pipeline with emphasis on execution latency, model screening efficacy and deployment feasibility.

The reported results were recorded on 20/03/2026 with the current implementation, workload characteristics, and evaluation environment. In particular, network-dependent measurements obtained on Sepolia and Polygon Amoy (POL) represent validation-time observations rather than fixed production deployment costs. To mitigate this limitation, the study reports operation-level timing, provides reproducible benchmarking scripts, and treats testnet cost figures as comparative indicators rather than deterministic production estimates.

A. Measured Latency

Performance measurements were collected using an automated benchmark over $n = 30$ runs with randomly generated model artifacts. Each run executed the full processing pipeline used in the system: secp256k1 key generation, AES-256-GCM encryption, ECDSA-secp256k1 signing, ECDH-based key wrapping/unwrapping, AES-256-GCM decryption, and IPFS upload latency measurement.

To approximate realistic deployment scenarios, artifact sizes were configured in the 20-100 MB range. The generated dataset ranged from 24,129,721 bytes to 99,536,480 bytes, with a mean size of 60,724,332 bytes (≈ 57.9 MB). Table IV summarizes the observed median and 95th percentile (P95) latencies.

The measurements show that cryptographic key-management operations introduce minimal overhead. Key generation and ECDH-based key wrapping/unwrapping

complete within approximately 1-3 ms, even at the P95 level. In contrast, payload-dependent operations dominate the runtime. AES-256-GCM encryption and decryption scale with artifact size, while ECDSA signing reflects the cost of generating integrity proofs for the encrypted model envelope.

The most significant contributor to latency is the IPFS publication stage. As shown in Table IV, the IPFS upload and CID generation process reaches a median latency of ($\approx$ 329 ms) and a P95 of ($\approx$ 876 ms), reflecting network and local daemon variability. This confirms that storage and network transfer, rather than cryptographic processing, dominate the end-to-end pipeline delay. Importantly, the ECDH-based key wrapping operations remain stable and low-latency, supporting efficient access-grant workflows even for large model artifacts.

B. Model Screening - Vulnerability Findings

We evaluated the performance of the selected *ModelAudit* by scanning 10 model artifacts available at [21], of which 5 are safe and 5 are malicious. The security-scan outcome for the 5 safe model artifacts, i.e. safe_model.onnx, safe_data.pkl, safe_model.h5, safe_model.pt and safe_model.safetensors, returned no vulnerabilities or severity measure, and all passed the screening.

On the contrary, Table V reports the security-scan outcome for the 5 malicious model artifacts considered, spanning the formats used in our prototype (ONNX, H5, PyTorch `.pt`, and pickle-based artifacts). Every malicious artifact was flagged by at least one detector. While this artifact suite does not constitute a comprehensive benchmark, it provides a practical validation that the integrated pre-publication model screening gate can separate benign artifacts from intentionally crafted risky ones *before* the artifacts are encrypted and disseminated through IPFS.

The severity tuple shown as **C/W/I** represents **C** = Critical findings, **W** = Warning findings, and **I** = Informational findings. In our workflow, *critical* findings are treated as deployment blockers (publication aborts), *warning* findings require human review or stricter policy checks, and *informational* findings provide context about potentially risky constructs even when an immediate exploit is not confirmed.

The malicious artifacts show distinct attack surfaces that are relevant to model supply-chain security. In the following we analyze impact of the detected vulnerability per-artifact, which are summarized in Table V.

- `malicious_system_call.pkl`: if an untrusted pickle is deserialized (e.g., as part of automated loading), the attacker can trigger arbitrary OS command execution, enabling credential theft, key exfiltration, or full compromise of the publisher node.
- `path_traversal.zip`: path traversal can overwrite configuration files or drop malicious payloads in sensitive locations, and the embedded execution primitive can then be used to run attacker-controlled commands, yielding a high-risk compromise path in ingestion pipelines that automatically unpack and load artifacts.
- `malicious_eval.pt`: when treated as a "model checkpoint" and loaded without strict controls, the artifact can execute attacker-controlled code and compromise the runtime environment; this is particularly concerning because `.pt` files are frequently exchanged as ML artifacts.
- `malicious_lambda.h5`: model reconstruction can evaluate attacker-provided expressions, which can be abused to run arbitrary code, read local files, or exfiltrate secrets from the host environment during loading or inference initialization.
- `malicious_custom_op.onnx`: custom operators extend the runtime with external implementations; if an inference environment resolves such operators via untrusted plugins or packages, it can introduce malicious code (or trigger denial-of-service/undefined behavior), undermining the integrity of downstream inference.

Operationally, these findings justify our design choice of running *ModelAudit* on plaintext artifacts as a *mandatory pre-publication model screening*, because it reduces the likelihood that a malicious model is encrypted, replicated via IPFS, and subsequently referenced on-chain. Preventing registration in the first place therefore limits propagation and minimizes the window in which untrusted artifacts can spread across the ecosystem.

C. Cross-Network Validation Reference

To evaluate deployment behavior across blockchain environments, the protocol was executed on two Ethereum-compatible test networks: Sepolia [22] and POL [23]. This multi-network validation ensures that smart-contract execution, access-control workflows, and metadata registration behave consistently across different fee markets.

Table VI reports the observed gas consumption and Figure 5 shows the transaction costs for the primary protocol operations: model registration, access request submission, and access-grant authorization. These values were obtained from validation-time observations on test networks and should be interpreted as comparative engineering indicators rather than fixed production costs. USD estimates were derived

TABLE IV
MEASURED LATENCY BREAKDOWN OVER 30 RANDOMIZED MODEL RUNS.

Operation	Median ($\approx$ ms)	P95 ($\approx$ ms)	Notes
Key generation (secp256k1)	0.746	1.430	Per requester keypair generation
AES-256-GCM encryption	44.738	126.125	Strongly model-size dependent
ECDSA signing	89.473	188.457	Payload-envelope integrity
IPFS upload + CID return	329.813	876.116	Network and local IPFS daemon dependent
ECDH key wrapping	1.474	2.604	Per access grant
ECDH key unwrapping	1.337	3.282	Requester-side recovery
AES-256-GCM decryption	34.980	114.357	Decryption-side model recovery

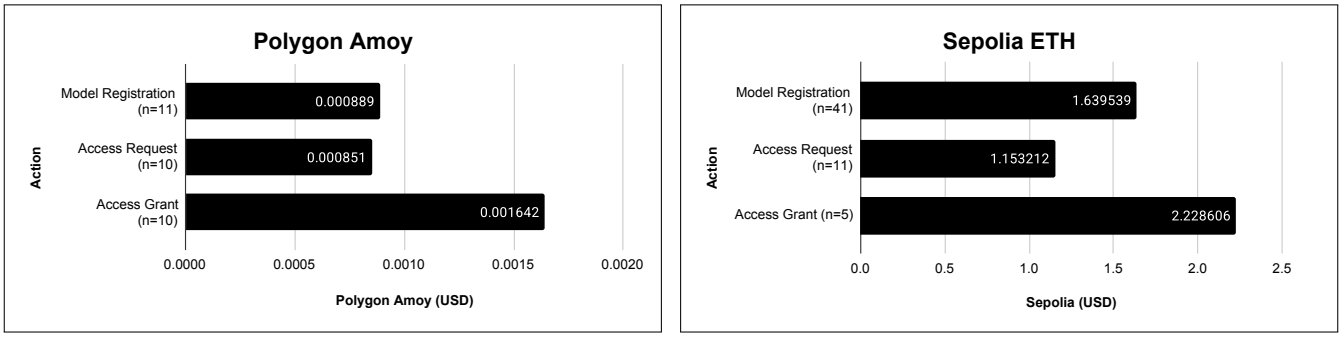


Fig. 5. Average transaction cost by action type (USD) across POL and Sepolia deployments.

using benchmark-time native-token reference valuations (SepoliaETH and POL proxies). The results indicate that gas usage remains largely consistent across networks because it is determined by the computational and storage complexity of the smart-contract logic. However, the resulting monetary cost varies due to network-specific fee markets and token valuations.

In the measured experiments, POL produced significantly lower USD-denominated transaction costs than Sepolia while maintaining identical protocol behavior. The primary contributors to gas consumption are metadata string storage during model registration, requester public-key storage during access requests, and encrypted key-payload storage during access grants. These findings demonstrate that while the protocol is functionally portable across Ethereum-compatible networks, economic efficiency is network-dependent. Consequently, practical deployment should treat blockchain selection and fee-policy tuning as part of overall system optimization alongside cryptographic workflow and storage design.

IX. Conclusion and Future Work

This work presents a secure protocol for distributed model sharing that integrates node-local cryptography, distributed storage, and blockchain-based metadata governance. The proposed workflow has been tested and validated on a prototype implemented on purpose. In our implementation, model artifacts are encrypted using CSPRNG-generated AES-256-GCM keys, signed with ECDSA-secp256k1, stored as encrypted content on IPFS, and registered on-chain through verifiable metadata and access-policy records. This separation of encrypted payload storage and auditable blockchain metadata enables secure model discoverability, ownership traceability, and controlled reuse for inference and TL in untrusted environments. In addition, the publication pipeline incorporates a pre-publication model screening/audit step that evaluates plaintext artifacts locally before encryption and IPFS upload, helping prevent unsafe or malformed models from entering the sharing workflow. The framework also supports blockchain-mediated access requests and owner-authorized key sharing through requester-specific ECDH-based key wrapping, allowing authorized nodes to obtain decryption capability without exposing publisher secrets or plaintext models on-chain.

Experimental evaluation with artifacts' sizes in the range of $\approx 20 - 100$ MB shows that cryptographic key-management operations introduce minimal latency, while payload size and IPFS publication dominate end-to-end delay. Cross-network experiments on Sepolia and POL demonstrate that the protocol is portable across Ethereum-compatible networks, although economic behavior varies by network. In our measurements, POL yielded lower USD-denominated registration costs than Sepolia, indicating that blockchain selection is an important deployment parameter influenced by network congestion, fee dynamics, and token valuation.

Future work will focus on fine-grained and time-bounded access policies, efficient revocation and key-rotation mechanisms. Additionally, we plan to expand the model synchro-

TABLE V
Security scan summary for the 5 malicious model artifacts evaluated [21].

Model	Class	Scan Decision	Severity (C/W/I)	Detected Vulnerability Types
malicious_custom_op.onnx	Malicious	Flagged	0/1/1	Custom ONNX operator detected; untrusted custom domain `evil.domain`
malicious_eval.pt	Malicious	Flagged	1/0/1	REDUCE invokes `builtins.eval`; unsafe PyTorch pickle deserialization context (CVE-2025-32434)
malicious_lambda.h5	Malicious	Flagged	1/1/1	Lambda layer contains dangerous Python code; suspicious `eval` configuration string
malicious_system_call.pkl	Malicious	Flagged	1/0/0	REDUCE invokes `posix.system` (OS command execution primitive)
path_traversal.zip	Malicious	Flagged	2/0/0	Archive path traversal (`../../etc/evil.txt`); REDUCE invokes `posix.system`

TABLE VI
MEASURED ON-CHAIN COST BY ACTION TYPE ACROSS SEPOLIA AND POLYGON AMOY DEPLOYMENTS.

Network	Action	Avg. Gas Used	Avg. Fee (Native)	Main Cost Driver
Sepolia	Model registration ($n = 41$)	347,795	0.000701313 SepoliaETH	Metadata string writes
Sepolia	Access request ($n = 11$)	328,858	0.000493286 SepoliaETH	Public-key string storage
Sepolia	Access grant ($n = 5$)	635,524	0.000953286 SepoliaETH	Encrypted-key payload storage
POL	Model registration ($n = 11$)	342,260	0.008898765 POL	Metadata string writes
POL	Access request ($n = 10$)	327,665	0.008519290 POL	Public-key string storage
POL	Access grant ($n = 10$)	632,142	0.016435692 POL	Encrypted-key payload storage

nization workflow with configurable update policies and safe rollback for deployed nodes, and to validate the protocol at larger scale using real TL workloads and adversarial evaluation settings.

REFERENCES

[1] S. J. Pan and Q. Yang, "A Survey on Transfer Learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, 2010.

[2] P. Kairouz, H. B. McMahan, B. Avent *et al.*, "Advances and Open Problems in Federated Learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] A. Zeeshan, W. Tiberti, A. Marotta, and D. Cassioli, "Deep Transfer Learning for Intrusion Detection in Edge Computing Scenarios," *IEEE Internet of Things Journal*, 2025.

[4] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli, "A Lightweight Intrusion Detection System for IoT Based on Deep Transfer Learning at the Edge," in *2025 IEEE 9th Forum on Research and Technologies for Society and Industry (RTSI)*, 2025, pp. 1–6.

[5] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-Efficient Learning of Deep Networks from Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*, ser. Proceedings of Machine Learning Research, vol. 54, 2017, pp. 1273–1282.

[6] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained On-Device Federated Learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.

[7] L. Bhatia and S. Samet, "Decentralized Federated Learning: A Comprehensive Survey and a New Blockchain-based Data Evaluation Scheme," in *2022 Fourth International Conference on Blockchain Computing and Applications (BCCA)*, 2022, pp. 289–296.

[8] K. Bonawitz, V. Ivanov, B. Kreuter, A. Marcedone, H. B. McMahan, S. Patel, D. Ramage, A. Segal, and K. Seth, "Practical Secure Aggregation for Privacy-Preserving Machine Learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, 2017, pp. 1175–1191.

[9] Y. A. Zouaghi, M. Mahamdioua, A. Lahoulou, and S. Chettibi, "Privacy preserving biometric authentication based on fully homomorphic encryption, blockchain, and IPFS data storage," *Multimedia Tools and Applications*, vol. 84, no. 34, pp. 42 293–42 319, 2025.

[10] Promptfoo, "ModelAudit Scanners," https://www.promptfoo.dev/docs/model-audit/scanners/, accessed: 2026-04-17.

[11] G. Wood, "Ethereum: A Secure Decentralised Generalised Transaction Ledger," Ethereum Yellow Paper, 2014, accessed: 2026-03-16.

[12] M. Dworkin, "Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC," NIST Special Publication 800-38D, 2007.

[13] Standards for Efficient Cryptography Group (SECG), "SEC 2: Recommended Elliptic Curve Domain Parameters, Version 2.0," Standards Specification, 2010.

[14] J. Benet, "IPFS - Content Addressed, Versioned, P2P File System," arXiv preprint arXiv:1407.3561, 2014.

[15] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved Proxy Re-Encryption Schemes with Applications to Secure Distributed Storage," *ACM Transactions on Information and System Security*, vol. 9, no. 1, pp. 1–30, 2006.

[16] Protect AI, "ModelScan: Protection Against Model Serialization Attacks," https://github.com/protectai/modelscan, accessed: 2026-04-17.

[17] Picklescan Contributors, "picklescan: Security scanner detecting suspicious actions in Python Pickle files," https://pypi.org/project/picklescan/, accessed: 2026-04-17.

[18] Hugging Face, "Pickle Scanning (Hub Security Documentation)," https://huggingface.co/docs/hub/security-pickle, accessed: 2026-04-17.

[19] PostgreSQL Global Development Group, "Postgresql: The world's most advanced open source relational database," https://www.postgresql.org/, 2025, accessed: 2026-04-22.

[20] MetaMask, "Metamask developer documentation," https://docs.metamask.io/, 2026, accessed: 2026-04-22.

[21] Promptfoo, "Modelaudit comparisons," https://github.com/promptfoo/modelaudit-comparisons, 2026, accessed: 2026-04-20.

[22] Ethereum Foundation, "Sepolia test network," https://ethereum.org/en/developers/docs/networks/#sepolia, 2023, accessed: 2026-04-11.

[23] Polygon Labs, "Polygon amoy testnet (pos) documentation," https://docs.polygon.technology/pos/reference/rpc-endpoints/, 2024, sepolia-anchored Polygon PoS testnet (chain ID 80002, gas token POL). Accessed: 2026-04-11.



Zeeshan ALI is a cybersecurity researcher and Ph.D. student in the Department of Information Engineering, Computer Science, and Mathematics at the University of L'Aquila, Italy, supervised by Prof. Dr. Dajana Cassioli. He holds both B.S. (2017–2021) and M.S. (2021–2022) degrees in Computer Science from the National College of Business Administration and Economics (NCBAE), Lahore, Pakistan. He has hands-on experience with state-of-the-art cybersecurity tools and advanced technologies, including large language models (LLMs) and generative AI. He previously served as a national startup founder at the Punjab Information Technology Board (PITB), Lahore. His research interests encompass cybersecurity, artificial intelligence, machine learning, the Internet of Things, and software quality assurance.



Walter TIBERTI is an Assistant Professor at the Department of Information Engineering, Computer Science and Mathematics of the University of L'Aquila, where he received his B.Sc, M.Sc. degree in Computer Engineering and his Ph.D. in Information and Communications Technology working on embedded systems security, in particular, wireless sensor network security. He was a visiting researcher at the Research Centre in Real-Time and Embedded Computing Systems (CISTER) of Porto (Portugal). He is part of the technical committee member of IEEE conferences and he is involved in Italian initiatives to promote nationwide cyber-security education and training. His research work is focused on software security, network security and cryptography.



Andrea MAROTTA is Assistant Professor (Tenure) at the Department of Information Engineering, Computer Science and Mathematics of the University of L'Aquila, Italy. He received his M.Sc. degree in Computer Engineering and his Ph.D. in Information and Communications Technology from the University of L'Aquila, in 2015 and 2019, respectively. He was visiting researcher at the Group for Research on Wireless (GROW) at Instituto Superior Técnico/University of Lisbon, at the research Institute of Communication, Information and Perception Technologies (TeCIP) of Scuola superiore di studi universitari e di perfezionamento Sant'Anna in Pisa, and at the Computer Networks Lab (NetLab) of the University of California, Davis. He actively participates in the TPC of IEEE Communications Society conferences. He performs research on Optical and Mobile Networks, Network Reliability, and Multi-Access Edge Computing.



Dajana CASSIOLI is (Senior Member, IEEE) is *Associate Professor* of Telecommunications Engineering with the University of L'Aquila, L'Aquila, Italy, where she served as the Elected Head for the Study Program in "Telecommunications Engineering: Advanced Technologies and Services" from 2021 to 2024. She co-authored more than 100 articles published in the most renowned international journals, magazines, and conference proceedings. She was a Summer Manager with the Wireless Systems Research Department, AT&T Laboratories-Research, NJ, USA, in 2000. She was a Short-Term Visiting Scholar with the University of Southern California, Los Angeles, CA, USA, hosted by Prof. Andy Molisch, in 2022. Her main research interests are in wireless communications, 5G/B5G networks and cybersecurity. Dr. Cassioli is the Chair of the SiG on Distributed and Massive MIMO of the IEEE P1944 Standard for Channel Models of Wireless Systems and the *Diversity, Equity and Inclusion Activity Coordinator* of the IEEE Italy Section. She is the Past Chair of the IEEE WIE AG Italy Section from 2016 to 2022 and the IEEE VT06/COM19 Italy Chapter from 2011 to 2017. She served as Chair of the IEEE ComSoc RCC SiG on Propagation Channels for 5G&B from 2021 to 2025. Since 2015, she is the Coordinator of the University of L'Aquila Node of the CINI National Laboratory of Cybersecurity, where she led the CyberEquality WG from 2020 to 2021. She has been awarded the *ERC StG VISION* (Video-oriented UWB-based Intelligent Ubiquitous Sensing) in 2010 and the *ERC PoC Grant iCARE* (MobIle health-Care system for monitoring toxicity and symptoms in Cancer patients Receiving Disease-Oriented Therapy) in 2016. She was the CEO in 2014–2018, and 2019 of the spin-off of the University of L'Aquila "Smartly: Natives of Smart Living srl," which designs and markets advanced ICT solutions to improve the quality of life. She served as the Co-Chair for Globecom 2025 WC, ICC 2024 MWN, ICC 2023 CISS, PIMRC2018 Industry, RTSI WIE Chair in 2018, 2019, and 2020, MELECON2020 and MetroInd4.0, and the TPC member of several International Conferences, including ICC, PIMRC, VTC, and GLOBECOM. She participated in the definition of the standard channel model for the IEEE 802.15.4 standard in 2005. She is an *Associate Editor* of IET Electronic Letters, and an *Executive Editor* of WILEY INTERNET TECHNOLOGY LETTERS and TRANSACTIONS ON EMERGING TELECOMMUNICATIONS TECHNOLOGY and served as an *Associate Editor* for IEEE COMMUNICATIONS LETTERS from 2018 to 2022.

Paper C4

Real-Time DDoS Detection by Integrated eBPF Telemetry and ML-enhanced SIEM

Zeeshan Ali, Andrea Marotta, Walter Tiberti, Carlo Centofanti, Andrea Marotta and Dajana Cassioli

Accepted in

2026 International Conference on Intelligent Multimedia, Networking, and Security (IMNS)

Real-Time DDoS Detection by Integrated eBPF Telemetry and Machine Learning-enhanced SIEM

Zeeshan Ali*, Andrea Marotta, Walter Tiberti, Carlo Centofanti, Dajana Cassioli
Department of Information Engineering Computer Science and Mathematics (DISIM)
University of L'Aquila, L'Aquila, Italy
*Corresponding author: zeeshan.ali1@graduate.univaq.it

***Abstract*—Distributed Denial-of-Service (DDoS) attacks remain one of the most disruptive threats to modern web services, overwhelming application resources and degrading service availability. This paper presents a lightweight, virtualized system architecture for real-time DDoS detection that combines kernel-level telemetry collection with machine learning (ML) based analysis. The proposed architecture enables fine-grained, low-overhead log collection without modifying the web applications because the network and application-level events generated during normal and attack traffic are captured directly at the kernel layer by means of an extended Berkeley Packet Filter (eBPF). The collected logs are then processed within a Security Information and Event Management (SIEM) platform, where ML–based detection models analyze traffic patterns and behavioral features to identify DDoS attacks in near real-time. This architecture improves visibility into attack characteristics while maintaining minimal performance impact on the protected services. The proposed system demonstrates how eBPF-based observability, when integrated with SIEM and ML techniques, can provide an effective, scalable, and modular approach for DDoS detection in virtualized environments. The design is particularly suited for cloud and multi-VM deployments, offering enhanced security monitoring, faster attack detection, and improved operational resilience.**

***Index Terms*—Distributed DDoS Detection, Security Information and Event Management (SIEM), Kernel-level Observability, Real-time Network Anomaly Detection, Splunk, eBPF-based Network Telemetry**

I. INTRODUCTION

Distributed Denial-of-Service (DDoS) attacks continue to pose a significant threat to the availability and reliability of web-based services. By flooding a target with a massive volume of malicious traffic, attackers can exhaust server resources, disrupt legitimate user access, and cause substantial financial and reputational damage. Despite advances in network security, DDoS attacks remain difficult to detect and mitigate due to their evolving attack patterns, high traffic volumes, and increasing use of legitimate-looking requests that closely resemble normal behavior [1].

Traditional DDoS detection mechanisms often rely on packet sampling, flow-level statistics, or application-layer logging [2]. While these approaches provide partial visibility, they frequently suffer from high overhead, limited granularity, or delayed detection [3].

Furthermore, many existing solutions require intrusive instrumentation or specialized network hardware [4], reducing their practicality in modern cloud and virtualized environments. As organizations increasingly adopt virtual machines (VMs) and cloud-native architectures, there is a growing need for lightweight, flexible, and high-fidelity monitoring solutions that can operate efficiently at scale [5].

Recent advancements in eBPF have enabled safe and programmable observability directly within the operating system kernel. eBPF allows fine-grained monitoring of network and system events with minimal performance overhead, making it a promising technology for security telemetry collection [6]. However, eBPF alone does not provide detection or decision-making capabilities; its effectiveness depends on how collected data is processed and analyzed.

SIEM platforms, on the other hand, provide centralized capabilities for log aggregation, event correlation, and security alerting across diverse system components. [7]

In this paper, we propose an integrated DDoS detection architecture that bridges this gap by combining eBPF-based kernel-level telemetry with a SIEM platform enhanced by ML–based analytics. In the proposed system, web applications are deployed on a protected VM, while eBPF programs continuously collect network and application-level events during both normal and attack scenarios. These logs are forwarded in near real-time to a SIEM instance hosted on a separate VM, ensuring isolation between application services and security analysis. The ML model is deployed within the SIEM to analyze traffic patterns and behavioral features, enabling the identification of DDoS attacks with high accuracy and low detection latency.

The proposed design emphasizes modularity, scalability, and low overhead, making it suitable for cloud, edge, and multi-VM deployments. By leveraging kernel-level observability and centralized intelligence, the system enhances attack visibility while preserving the performance of the protected web services. Our experimental setup demonstrates that integrating eBPF telemetry with SIEM-driven ML enables timely and reliable DDoS detection, without requiring application modifications or specialized hardware.

Based on the proposed architecture and experimental evaluation, the key contributions of this paper are as follows:

- We design and implement a decoupled kernel-to-SIEM security architecture that enables real-time DDoS detection without executing detection logic on the pro-

tected host. Unlike prior eBPF-based IDS approaches that embed inference locally, our architecture externalizes analytics while preserving fine-grained kernel visibility.

- We develop an XDP-attached eBPF telemetry framework that extracts connection- and packet-level features directly from the Linux kernel with minimal processing overhead. The framework enables high-fidelity observability without packet mirroring, user-space capture, or application instrumentation.
- We propose a structured telemetry abstraction layer that transforms low-level eBPF map data into SIEM-compatible event streams, enabling centralized correlation, temporal aggregation, and ML inference across distributed virtualized environments.
- We introduce an isolation-aware multi-VM deployment model that separates attack generation, workload execution, telemetry collection, and security analytics. This design improves robustness, reduces attack surface expansion, and reflects realistic cloud-native security architectures.
- We provide an end-to-end experimental evaluation of the proposed pipeline, analyzing detection accuracy, traffic-separability characteristics, and real-time inference behavior under volumetric SYN-flood DDoS attacks. Our results demonstrate that kernel-level telemetry provides discriminative features suitable for centralized ML-based DDoS detection without degrading service availability.

The subsequent sections of the document are structured in the following manner: Section II presents a comprehensive discussion of the relevant and recently published literature. Section III explains the proposed system model. The results are shown in Section V, which also offers a comprehensive discussion of our findings and their implications. Section VI serves as the concluding part of the paper.

II. RELATED WORK

DDoS detection and mitigation have evolved significantly in recent years, spanning traditional statistical techniques, ML–based analytics, and more recent kernel-level observability approaches. Early work in this area focused on hybrid and anomaly-based methods for identifying volumetric attacks by combining volume-based features with entropy and feature statistics, demonstrating the advantage over simple threshold-based detection mechanisms in controlled network environments [8].

A substantial body of research has explored ML for DDoS detection, leveraging supervised and unsupervised models to identify attack patterns from network traffic characteristics. For example, comprehensive surveys of ML-based DDoS detection techniques underscore the effectiveness of classifiers such as Random Forest (RF), Support Vector Machine (SVM), and Deep Learning (DL) models in distinguishing between normal and malicious traffic using datasets like CIC-DDoS 2019 [9] [10]. In IoT and cloud contexts, researchers have applied LSTM and ensemble models to capture temporal traffic features, achieving high accuracy under diverse threat scenarios [11]. However, these approaches typically rely on pre-collected flow statistics or packet captures, which may introduce latency and limit visibility into low-level system behavior.

In parallel, eBPF has emerged as a promising technology for lightweight, in-kernel observability and security monitoring. eBPF enables programmable telemetry collection within the operating system kernel without modifying source code, significantly reducing overhead compared to user-space capture mechanisms. Several studies have demonstrated the use of eBPF/XDP for DDoS detection and mitigation, especially in resource-constrained environments such as IoT edge devices, where rate-based blocking at the earliest point in the network stack can effectively drop malicious traffic while preserving legitimate flows [12]. Other work has combined eBPF with traditional IDS designs, using kernel-based filtering to pre-drop packets before more detailed inspection in user space [13].

Some prior works merge eBPF and ML to enhance intrusion detection performance. For instance, systems that use eBPF to gather packet-level features for training Decision Tree (DT), RF, Support Vector Machine (SVM), and similar classifiers illustrate the potential of kernel-level data as a rich input source for ML models [14].

Despite these advancements, existing research largely addresses DDoS detection in a fragmented manner, focusing either on kernel-level telemetry and in-kernel mitigation using eBPF, or on ML–based detection applied to network or flow-level data. Prior work typically deploys these techniques as standalone intrusion detection or mitigation systems, without integrating them into centralized security monitoring frameworks. In contrast, the system proposed in this paper introduces a unified, modular architecture that forwards fine-grained eBPF telemetry to a SIEM platform, where ML–based analytics are applied for DDoS detection. By embedding eBPF-driven observability within a SIEM-centric monitoring pipeline, the proposed approach enables centralized analysis and correlation while preserving low-overhead data collection in a multi-VM environment.

III. SYSTEM MODEL

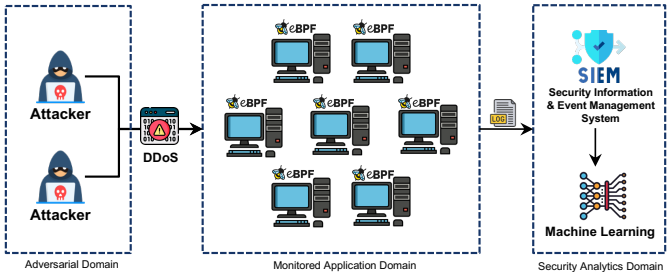


Fig. 1. General system architecture of the proposed eBPF–SIEM framework for DDoS detection.

Figure 1 illustrates the high-level architecture of the proposed DDoS detection framework. The system is composed

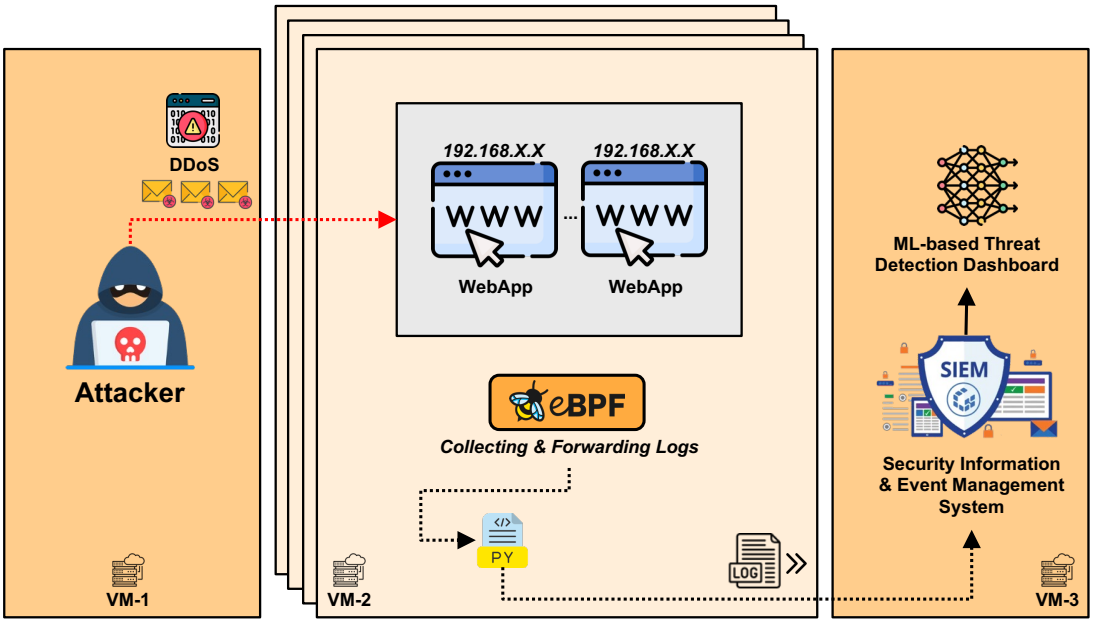


Fig. 2. Detailed system model of the proposed SIEM-based DDoS detection framework

of three logical domains: (i) the Adversarial Domain, (ii) the Monitored Application Domain, and (iii) the Security Analytics Domain. An external adversarial node generates malicious traffic targeting one or more protected service nodes.

Within the monitored environment, multiple application nodes (e.g., VMs or cloud instances) independently execute kernel-level eBPF telemetry programs. Each node locally captures fine-grained network and system events using XDP-attached eBPF hooks, enabling low-overhead observability without requiring modifications to hosted applications.

The collected telemetry from all monitored nodes is forwarded in near real-time to a centralized SIEM platform in a many-to-one aggregation model. The SIEM aggregates, normalizes, and correlates telemetry streams across nodes to provide unified visibility of distributed traffic behavior. An ML model integrated within the SIEM performs behavioral analysis and anomaly detection to identify potential DDoS activity across the monitored infrastructure.

This modular and horizontally scalable design separates attack generation, per-node telemetry collection, and centralized detection logic, improving system robustness, cross-node visibility, and deployment flexibility in multi-VM and cloud environments.

IV. EXPERIMENTAL SETUP AND METHODOLOGY

The experimental testbed is implemented using virtualized infrastructure to emulate a controlled and reproducible network environment for evaluating SYN-based DDoS detection in web service workloads. Specifically, multiple VMs are deployed to simulate the logical components of the proposed detection framework, including attack generation, application hosting, telemetry collection, and security analytics. This virtualization-based design enables safe experimentation, isolation of system functions, and realistic modeling of distributed network behavior without impacting external systems. Each VM performs a distinct role within the attack observation and detection pipeline, as illustrated in Figure 2. The methodology proceeds in a step-by-step manner as follows.

Algorithm 1 SYN-Flood Attack Generation

Require: Target IP address T, target port P
Ensure: Generation of SYN-flood traffic toward (T, P)

```
Initialize (T, P) from user input or default values
Define attack duration range D = {d | 2 ≤ d ≤ 20, d ∈ ℤ}
while attack execution is active do
    Randomly sample attack duration d_i ∼ D
    Generate a high-rate TCP SYN packet stream S_i targeting (T, P)
    Transmit S_i continuously over time interval [t_i, t_i + d_i]
    Terminate transmission at t_i + d_i
end while
End attack execution
```

A. VM-1 – Adversarial Node and Attack Generation

`VM-1` operates as the threat actor machine and is isolated from the monitored web service environment, interacting only through the network services exposed by `VM-2`. This design emulates an external adversary attempting to disrupt or degrade web-based services through a network-layer attack. `VM-1` generates attack traffic to support comparative analysis of detection performance under malicious and normal conditions, following the SYN-flood attack generation process described in Algorithm 1.

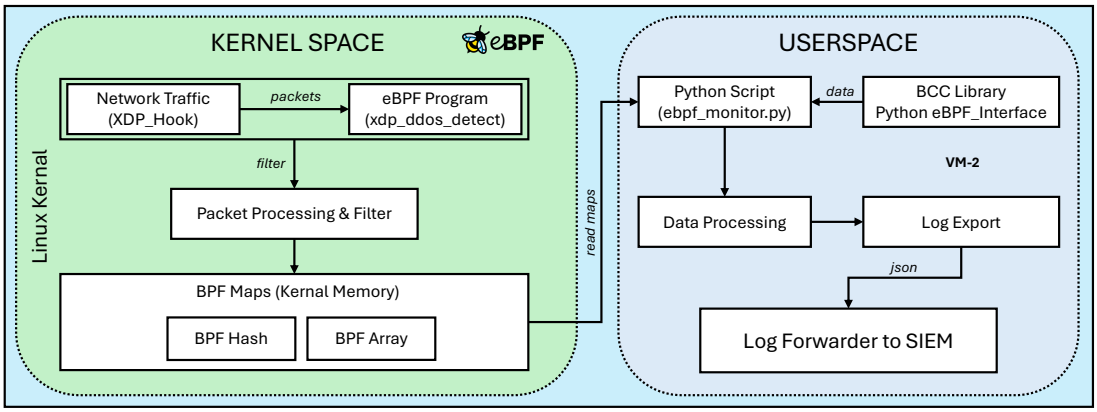


Fig. 3. Network logs Collection using eBPF from Kernel Space to Userspace in VM−2

B. VM-2 – Application Environment

VM−2 represents the monitored application domain. In Figure 2, each layer superimposed represents an independent node that can execute its own kernel-level eBPF telemetry pipeline. The layered depiction illustrates the horizontal scalability of the monitoring architecture. Each monitored node may host one or more application services (e.g., web servers or other networked workloads). The specific application (NGINX in our experimental setup) is used only as a representative workload and is not required by the proposed architecture. The monitoring mechanism operates independently of application logic.

To monitor incoming traffic, VM−2 integrates a kernel-level eBPF–based monitoring mechanism. Figure 3 illustrates the end-to-end eBPF monitoring workflow in VM−2. eBPF programs are attached to the Linux kernel network stack to capture fine-grained packets and connection-level events associated with incoming requests to the NGINX service via an eXpress Data Path (XDP) hook, processed by an eBPF program, and stored in BPF maps.

A lightweight data processing pipeline implemented in Python3 is used to extract relevant features from the eBPF maps using the BPF Compiler Collection (BCC) library, structure the collected telemetry, and forward the resulting logs to the SIEM platform hosted on VM−3 for further analysis and DDoS detection. This separation of kernel-level telemetry collection enables real-time visibility into traffic behavior, which reduces the computational burden on the monitored service.

C. VM-3 – ML-enhanced Security Information and Event Management System (SIEM)

The telemetry logs from VM−2 are forwarded to the SIEM hosted at VM−3, normalized into structured event fields. They are correlated over time windows and stored for downstream analytics. Although the methodology remains SIEM-agnostic in principle, a Splunk Enterprise [15] instance is configured to collect, store the logs at specific indexes, and analyze security telemetry from a virtualized environment. Splunk uses SPL query language to search, filter, and analyze machine data. Splunk served as the centralized analytics and decision-making component, responsible for detecting anomalies and triggering mitigation actions. This ensures that no detection logic is executed locally on VM−2, preserving experimental neutrality and maintaining separation between workload generation and threat analysis.

A script based on the Python3 module linked with eBPF is used to collect and forward telemetry to Splunk through the HTTP Event Collector (HEC), where it is stored in the index “ebpf-index” in Splunk.

A Splunk dashboard is designed as a generic telemetry analytics tool to monitor real-time telemetry logs interface and is not specific to a single attack type. It supports ingestion, visualization, and analysis of structured eBPF-derived events from multiple monitored nodes. To perform attack detection we used the Splunk Artificial Intelligence (AI) Toolkit [16].

This AI toolkit enabled seamless integration of ML models with live telemetry data, supporting automated inference, alerting, and visualization within the Splunk environment. A pre-trained RF model was deployed (discussed in Section V-A1) on the incoming telemetry to identify malicious activity. The RF classifier was selected due to its strong performance on tabular network telemetry, robustness to feature variance, and ability to model non-linear decision boundaries [17], which are required for distinguishing between normal and malicious traffic.

The ingested telemetry from VM−2 is continuously retrieved in real-time from the corresponding SIEM indexes and provided as input to the trained RF model for classification. The deployed model is tightly integrated into the Splunk dashboard, enabling live analysis of incoming network traffic as it is ingested into the system. When malicious activity is identified, the classification results and associated alerts are immediately visualized within the dashboard interface, providing the Security Operations Center (SOC) team with real-time situational awareness of a potential DDoS attack.

D. Experimental Setup

The system configurations of the three VMs, including operating systems and deployed tools, are summarized in Table I. All VMs are deployed on the same local network to ensure consistent communication latency and stable network behavior. The experimental environment supports an end-to-end attack detection workflow, from attack initiation to kernel-level telemetry collection and automated SIEM-based analysis, enabling accurate observation of traffic behavior and real-time security monitoring.

TABLE I
SYSTEM CONFIGURATIONS

VM-1 (Attacker)	VM-2 (eBPF)	VM-3 (SIEM)
Ubuntu v24.04.3 LTS	Ubuntu v24.04.3 LTS	Ubuntu v24.04.3 LTS
hping3 v3.a2.ds2	eBPF Tool v7.4.0	Splunk Enterprise v10.0.2
-	Python3 v3.14.2	Splunk AI Toolkit v5.6.4

E. Attack Scenario: External SYN-Flood DDoS Evaluation

The attacker located on `VM-1` launches a high-rate SYN-flood attack using the `hping3` [18] traffic generator, targeting the exposed NGINX service running on `VM-2`. The attack is conducted entirely through the network interface, without compromising the host or deploying additional services. It represents a conventional volumetric DDoS scenario in which the adversary interacts solely via the service exposure layer.

During the attack, eBPF probes on `VM-2` capture packet-level and connection-related events associated with incoming SYN requests, including high connection initiation rates and incomplete Transmission Control Protocol (TCP) handshakes. The collected telemetry is forwarded in near real-time to `VM-3`, where it is analyzed by the SIEM and ML–based detection components. The experiment is conducted in a fully controlled environment, and no systems outside the testbed are targeted. The objective of this scenario is to reproduce the traffic characteristics of SYN-flood attacks and evaluate the effectiveness of the proposed eBPF–SIEM detection pipeline.

This attack scenario is selected due to the prevalence of SYN-flood attacks as a fundamental and widely observed form of network-layer DDoS in real-world environments. SYN floods exploit the TCP connection establishment mechanism, making them particularly effective at exhausting server-side resources while generating traffic patterns that can closely resemble legitimate connection attempts. Evaluating the proposed system under this scenario allows assessment of its ability to distinguish malicious volumetric behavior from normal traffic using kernel-level telemetry. Furthermore, this scenario provides a clear and interpretable baseline for validating the effectiveness of the eBPF–SIEM detection pipeline before extending the approach to more complex or multi-vector DDoS attacks.

TABLE II
DATASET

Labels	Records	Training Records	Testing Records
Normal	50,000	35,000	15,000
SYN-DDoS	50,000	35,000	15,000

V. EXPERIMENTAL RESULTS

A. Performance Evaluation

Figure 4 illustrates the distribution of packet rates observed for normal traffic and SYN-flood DDoS traffic, derived from eBPF telemetry and analyzed within the SIEM. The packet rate is shown on a logarithmic scale to accommodate the wide dynamic range between normal and attack traffic. As shown in the figure, normal traffic exhibits consistently low packet rates with limited variance, reflecting stable and legitimate connection behavior. In contrast, SYN-flood traffic exhibits a substantially higher packet rate, spanning several orders of magnitude, indicating the sustained volumetric nature of the attack. The clear separation between normal and malicious traffic distributions highlights the effectiveness of packet rate as a discriminative feature for detecting SYN floods.

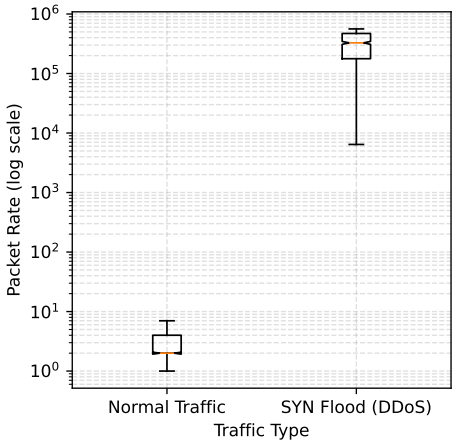


Fig. 4. Distribution of packet rates for normal traffic and SYN-flood DDoS traffic on a logarithmic scale, derived from eBPF telemetry and analyzed within the SIEM.

1) Dataset & Classification Performance: The RF classifier was trained using a custom, labeled dataset generated within the experimental testbed, as shown in Table II by injecting controlled SYN-flood DDoS traffic alongside normal traffic. The dataset consists of a total of 100,000 records, with an equal class distribution of 50% normal samples and 50% SYN-flood attack samples, ensuring balanced learning and unbiased classification. Each record is characterized by features including packet_rate (packets/s) and byte_size (bytes/s), which effectively capture the traffic intensity and connection behavior associated with volumetric SYN-flood attacks, along with a corresponding class label. The dataset was partitioned using a 70:30 split, with 70% of the samples used for training

the model and the remaining 30% reserved for testing and performance evaluation.

The classification performance of the RF model is summarized in Table III, which reports the results obtained on the held-out testing dataset. The evaluation is based on standard classification metrics, including accuracy, precision, recall, and F1-score, computed separately for normal traffic and SYN-flood attack classes.

Accuracy represents the overall proportion of correctly classified samples, while precision measures the fraction of correctly identified positive samples among all samples predicted as positive. Recall quantifies the ability of the model to correctly identify actual positive samples, and the F1-score provides a balanced measure by combining precision and recall into a single metric. These metrics collectively provide a comprehensive assessment of the model's classification effectiveness.

The trained model achieves perfect classification performance for both classes on the test dataset, indicating clear separability between normal and SYN-flood traffic patterns. This result demonstrates that the selected features, derived from eBPF telemetry, effectively capture the distinguishing characteristics of volumetric SYN-flood attacks and enable reliable detection when analyzed using the proposed SIEM-based ML pipeline.

TABLE III
MODEL PERFORMANCE EVALUATION

Labels	Accuracy	Precision	Recall	F1-Score
Normal	99.72%	99.81%	99.74%	99.78%
SYN-DDoS	99.68%	99.74%	99.71%	99.72%

VI. CONCLUSION

Our work presented a practical and efficient DDoS detection framework that integrates kernel-level observability with centralized security analytics in a virtualized environment. By leveraging eBPF for low-overhead, fine-grained telemetry collection and forwarding this data to a SIEM platform enhanced with ML–based detection, the proposed system enables timely and accurate identification of SYN-flood DDoS attacks without requiring application modifications or specialized hardware. The experimental evaluation demonstrated that kernel-level features extracted via eBPF effectively capture the traffic characteristics of volumetric SYN-flood attacks, while the SIEM-integrated RF ML-classifier provides reliable real-time classification and visualization of attack events. The separation of attack generation, monitored services, telemetry collection, and analytics across multiple VMs further enhances system robustness and reflects realistic deployment scenarios. The results confirm that the proposed eBPF–SIEM pipeline achieves effective attack visibility and detection while maintaining minimal performance impact on the protected web service. Future work will focus on extending the framework to support multi-vector DDoS attacks, incorporating adaptive learning models.

ACKNOWLEDGMENT

This work was supported in part by the European Union—NextGenerationEU through the Italian Ministry of University and Research Mission 4, Component 2, Investment 1.3, CUP E83C22004640001, partnership on "Telecommunications of the Future" (PE00000001 - program "RESTART") and by Ministry of Enterprises and Made in Italy through EDIHAMO European Digital Innovation hub CUP B17H22004170001.

REFERENCES

[1] J. Mirkovic and P. Reiher, "A taxonomy of ddos attack and ddos defense mechanisms," *ACM SIGCOMM Computer Communication Review*, vol. 34, no. 2, pp. 39–53, 2004.

[2] M. Aamir and M. A. Zaidi, "Ddos attack and defense: Review of some traditional and current techniques," *arXiv preprint arXiv:1401.6317*, 2014.

[3] Z. Ali, W. Tiberti, A. Marotta, and D. Cassioli, "Empowering network security: Bert transformer learning approach and mlp for intrusion detection in imbalanced network traffic," *IEEE Access*, vol. 12, pp. 137 618–137 633, 2024.

[4] V. Jyothi, X. Wang, S. K. Addepalli, and R. Karri, "Brain: Behavior based adaptive intrusion detection in networks: Using hardware performance counters to detect ddos attacks," in *2016 29th international conference on VLSI design and 2016 15th international conference on embedded systems (VLSID)*. IEEE, 2016, pp. 587–588.

[5] N. Agrawal and S. Tapaswi, "Defense mechanisms against ddos attacks in a cloud computing environment: State-of-the-art and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3769–3795, 2019.

[6] H. Sharaf, I. Ahmad, and T. Dimitriou, "Extended berkeley packet filter: An application perspective," *IEEE Access*, vol. 10, pp. 126 370–126 393, 2022.

[7] G. González-Granadillo, S. González-Zarzosa, and R. Diaz, "Security information and event management (siem): analysis, trends, and usage in critical infrastructures," *Sensors*, vol. 21, no. 14, p. 4759, 2021.

[8] P. Bojović, I. Bašičević, S. Ocovaj, and M. Popović, "A practical approach to detection of distributed denial-of-service attacks using a hybrid detection method," *Computers & Electrical Engineering*, vol. 73, pp. 84–96, 2019.

[9] O. M. A. Ali, R. A. Hamaamin, B. J. Youns, and S. W. Kareem, "Innovative machine learning strategies for ddos detection: A review," *UHD Journal of Science and Technology*, vol. 8, no. 2, pp. 38–49, 2024.

[10] S.-R. Chen, S.-J. Chen, and W.-B. Hsieh, "Enhancing machine learning-based ddos detection through hyperparameter optimization," *Electronics*, vol. 14, no. 16, p. 3319, 2025.

[11] S. Berríos, S. Garcia, P. Hermosilla, and H. Allende-Cid, "A machine-learning-based approach for the detection and mitigation of distributed denial-of-service attacks in internet of things environments," *Applied Sciences*, vol. 15, no. 11, p. 6012, 2025.

[12] A. Tolay, "ebpf-based real-time ddos mitigation for iot edge devices," *arXiv preprint arXiv:2508.00851*, 2025.

[13] S.-Y. Wang and J.-C. Chang, "Design and implementation of an intrusion detection system by using extended bpf in the linux kernel," *Journal of Network and Computer Applications*, vol. 198, p. 103283, 2022.

[14] N. Anand, S. MA, P. K. Aakula, R. B. Ponnuru, R. Patan, and C. R. P. Reddy, "Enhancing intrusion detection against denial of service and distributed denial of service attacks: Leveraging extended berkeley packet filter and machine learning algorithms," *IET Communications*, vol. 19, no. 1, p. e12879, 2025.

[15] Splunk LLC, "Splunk enterprise," https://splunk.com/en_us/products/splunk-enterprise.html, 2026, accessed: 2026-01-13.

[16] Splunk Inc., "Splunk ai toolkit," https://www.splunk.com/en_us/products/ai-toolkit.html, 2025, accessed: 2026-01-29.

[17] S. J. Rigatti, "Random forest," *Journal of insurance medicine*, vol. 47, no. 1, pp. 31–39, 2017.

[18] Kali Linux / OffSec Services Limited, "hping3 — kali linux tools," https://www.kali.org/tools/hping3/, May 2024, updated: 2024-05-23; Accessed: 2025-11-11.