Machine Learning-based Intrusion Detection and Prevention using Cross-layer Features in Internet of Things (IoT) Networks

Noor Hafsa, Hadeel Alzoubi, and Sajida Imran

***Abstract*—The IoT has emerged as a significant target for cyber-attacks, particularly with a focus on the routing protocol for low-power and lossy networks (RPL) within Wireless Sensor Networks (WSNs). These attacks can disrupt network topologies and compromise data transmission. Early detection of routing attacks is crucial, particularly in resource-constrained RPL networks. This study employed a simulated dataset encompassing Hello Flood, Version Number, and Worst Parent attacks to develop a robust detection model for resource-based routing attacks in IoT networks. In this research, a novel cross-layer feature analysis was conducted, identifying 12 key features crucial for distinguishing between normal and malicious nodes within the network out of the 29 features examined. Various machine learning algorithms, including random forest, CatBoost, and extreme gradient boosting, were evaluated for precise classification. The optimized CatBoost model, a gradient-boosting decision tree (DT) algorithm, demonstrated outstanding performance with a 99% of detection rate, 0.8% of false positive rate, 98% of sensitivity, and 98% of positive predictive values on an independent test dataset. Furthermore, an advanced intrusion prevention algorithm leveraging cross-layer feature-induced intrusion detection was introduced to effectively combat prevalent routing attacks. This study significantly contributes to enhancing cybersecurity in IoT networks, particularly in smart cities, by offering robust intrusion detection and prevention mechanisms.**

***Index Terms*—Cross-layer, cyber attacks, intrusion detection, intrusion prevention, IoT, machine learning, RPL**

I. Introduction

THE IoT interlinks devices enabling communication, data collection, processing, and targeted information sharing. This advancement has revolutionized automation in smart homes and transformed sectors such as education, healthcare, agriculture, and industries. Projections suggest that by 2025, the IoT sector will burgeon to encompass 22 billion smart devices [1]. The IoT framework comprises three key layers: perception, network, and application [2], each presenting distinctive security challenges and vulnerabilities. This research, in particular, delves into addressing security concerns, risks, and threats inherent in the network layer.

The routing protocol for low power lossy network (RPL) is a specialized network layer routing protocol designed for IoT technologies, compatible with IPv6 headers and the IEEE 802.15.4 standard [3], [4]. Despite its utility, RPL is vulnerable to a range of internal and external attacks, including Selective Forwarding, Hello Flooding, Clone ID, Sinkhole, Black hole, Rank Attacks, and more, which can trigger data packet delays, losses, intensified power consumption, and security risks [5], [6]. The Internet Engineering Task Force (IETF) standardized 6LoWPAN as the benchmark for small, low-powered IoT devices utilizing wireless personal area networks (WPAN) [7]. This network layer protocol facilitates Internet connectivity via open standards, overcoming initial challenges associated with IPv6 implementation due to energy constraints. Noteworthy for its contributions, 6LoWPAN significantly advances IoT technology by reinforcing mobility, scalability, and addressing security threats such as confidentiality breaches, fragmentation issues, and ensuring data authentication through encryption mechanisms like IPsec. Unauthorized manipulation of fragmented data could lead to illicit access and potential security breaches.

IoT devices pose significant security challenges due to their diverse nature and limitations in processing power, memory, and power resources. The deployment of encryption and authentication on these devices is hindered by their low processing power, which struggles to meet the computational demands of these security measures. Limited memory on IoT devices constrains security data storage and the execution of complex security algorithms. Additionally, the restricted power capacity of IoT devices limits their ability to support energy-intensive security functions, impeding continuous security operations. Moreover. the heterogeneity of IoT devices, characterized by varied hardware and software configurations, complicates the adoption of standardized security solutions, making it challenging to develop a cohesive security strategy. Insecure communication channels and limited bandwidth further exacerbate security risks in IoT environments, with wireless networks, commonly used by IoT devices, being particularly vulnerable to data interception and malware insertion. The constrained bandwidth of IoT communication channels restricts secure data transmission, hindering the implementation of bandwidth-intensive secure communication technologies. Lastly, the inherent limitations in power and processing capacities of IoT devices make them challenging to secure effectively, with

Manuscript received May 7, 2024; revised February 4, 2025; approved for publication by Xie, Huiqiang, Division 4 Editor, July 2, 2025.

The authors extend their appreciation to the Deanship for Scientific Research in King Faisal University for funding this research work through Ambitious Track Research Project Number 250483.

N. Hafsa and H. Alzoubi are with the Department of Computer Science, College of Computer Science and Information Technology King Faisal University Al Ahsa, KSA, email: {nhafsa, hmalzoubi}@kfu.edu.sa.

S. Imran is with the Department of Computer Science School of Engineering Computing and Mathematics University of Chichester West Wussex, UK, email: s.imran@chi.ac.uk.

N. Hafsa is the corresponding author.

Digital Object Identifier: 10.23919/JCN.2025.000053

1229-2370/25/$10.00 © 2025 KICS

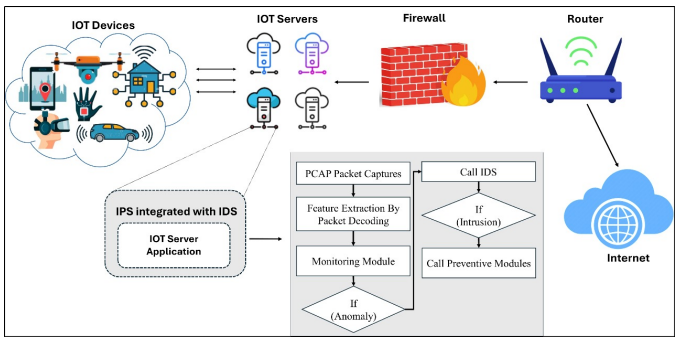

Fig. 1. The system architecture of the proposed Intrusion Prevention System with an integrated IDS module.

authentication and encryption often proving impractical due to resource constraints. The development of lightweight security mechanisms tailored for resource-constrained devices necessitates extensive research and innovation. As compared to alternative protocols like ZigBee and BLE, RPL confronts distinct obstacles such as topological instability, energy efficiency considerations, and scalability limitations, making it prone to external intrusions. In contrast, ZigBee encounters challenges related to mesh networking, interference mitigation, and accommodating diverse applications, while BLE grapples with issues like establishing connections, managing data throughput constraints, and addressing security vulnerabilities like eavesdropping and man-in-the-middle attacks.

This article presents a machine learning (ML) based intrusion detection solution for RPL by leveraging network features across data, link, and topology layers. We investigated three RPL attacks - Hello Flood, Version Number, and Worst-Parent Attack, conducting a thorough analysis to assess the predictive capabilities of cross-layer network features for intrusion detection through big data analytics. This research resulted in the creation of robust detection models for routing attacks in IoT networks. Various ML models including decision-tree-based gradient boosting techniques like random forest (RF), CatBoost model, and extreme gradient boosting, were employed. The primary aim is to pinpoint the most significant layer features in the routing attack dataset by identifying key cross-layer attributes that aid in classifying network nodes as either benign or malicious.

A system architecture illustrating the proposed intrusion prevention system operations with an integrated intrusion detection system (IDS) in the IoT network is presented in ROC curve1. The system commences by extracting cross-layer features from captured network packets, activating a monitoring module, and engaging the ML-based intrusion detection method as needed. It promptly issues alert messages upon intrusion detection and implements preventive measures to mitigate cybersecurity attacks within the IoT network.

The research contributions of the current study are summarized as follows.

1) We developed an accurate IDS that uses high-performing ML models to detect various network intrusions including Hello Flood, Version Number, and Worst Parent attacks in large-scale RPL-based IoT networks.
2) As far as the authors are aware, this is the first cross-layer RPL-based networks' intrusion detection system analysis to investigate the effect of cross-layer features derived from link, data, and topology layer on intrusion detection.
3) We proposed an intrusion prevention algorithm exploiting cross-layer feature-based IDS to mitigate the three intrusion incidents.

The rest of the paper is organized as follows; The preliminaries of RPL are covered in Section II. The relevant studies are addressed in Section III. The suggested IDS approach is described in section IV. Sections V and VI present a comprehensive description of the IDS performances, accompanied by a detailed analysis of the experimental findings. Finally, Sections VII and VIII presents Future Work and Conclusion.

II. Background

A. Routing Protocol for Low Power and Lossy Networks (RPL)

Contemporary IoT devices often cannot utilize traditional routing protocols like RIP, DSR, OSPF due to their limited power capabilities and the characteristics of lossy networks. Instead, they depend on the RPL protocol, tailored for routing in resource-constrained smart devices. Initially employing a directed acyclic graph (DAG), RPL encountered routing loop issues, which leads to the introduction of the destination-oriented DAG (DODAG). RPL facilitates three traffic types: point-to-multipoint (P2MP), multipoint-to-point (MP2P), and point-to-point (P2P). Each node's position in the DODAG graph and its distance from the root node and neighbors are identified using rank numbers. Fig. 2 illustrates the routing process in RPL, with the rank distinguishing each node's unique position and path from the low power and lossy networks (LLN) border routers (LBR). Alongside node categorization, various main control messages such as the DODAG information object (DIO) message, DODAG information solicitation (DIS) message, destination advertisement object (DAO), destination advertisement object acknowledgement (DAO-ACK) message, and consistency check (CC) message play crucial roles in the protocol.

When a node joins an RPL instance as a host, it utilizes a pre-shared authentication key. However, for a node looking to participate in the RPL as a router, an additional key must be acquired from the key authority. Regarding traffic management, DODAG route formation involves two types: upward route and downward route. In the former, DIO and DIS messages establish an upward route for MP2P traffic, conveying crucial details like version, Instance ID, timer, and object function (OF) through grounded nodes to calculate rank. The DIO message circulates among nodes seeking to join the DODAG, while the DIS message is multicast to neighboring nodes for floating nodes to select a preferred parent. In the latter method, DAO messages employ downward routes tailored for P2MP and P2P traffic. The neighbor discovery protocol aids route formation by reconstructing a comprehensive graph representation using nodes' information to detect potential attacks.

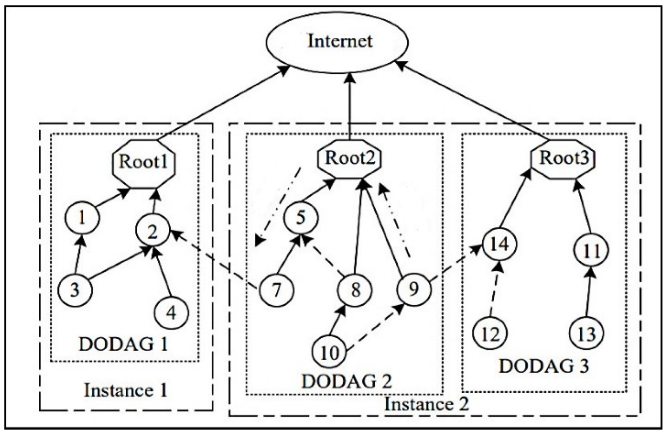

Fig. 2. RPL upward DODAG routings using two RPL instances are illustrated. Each root in the DODAG graph represents a LBR and the network nodes with associated ranks are shown using circles.

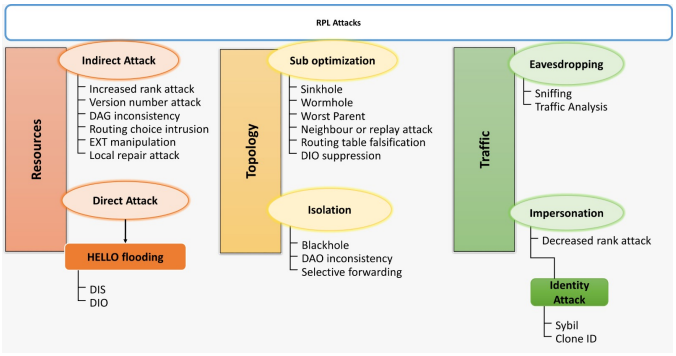

Fig. 3. Taxonomy of cyber-attacks specific to RPL protocol.

B. RPL Attacks

RPL is naturally susceptible to a multitude of attacks due to its reliance on the IPv6 open stack and substantial use of wireless media for node communications. The taxonomy of RPL attacks is illustrated in Fig. 3, as presented in [5]. Resource-based direct attacks within RPL-based IoT networks target specific elements to disrupt operations or compromise security by aiming at nodes or communication channels. Conversely, resource-based indirect attacks subtly manipulate resources to cause broader disruptions by exploiting vulnerabilities or altering routing paths, leading to issues like data loss or unauthorized access. These attacks threaten the integrity of RPL-based IoT networks, demanding robust security measures. The present study examined three resource-based direct and indirect attacks including Hello flood, Version Number and Worst Parent. These attacks pose significant challenges to cybersecurity by exhausting resources, complicating detection, and necessitating sophisticated prevention techniques. Advanced strategies are crucial to safeguard network security and reliability amidst these challenges. The following paragraphs delved deep into these three cyberattacks.

1) Hello Flood : A Hello Flood attack is a type of denial of service (DoS) attack that targets wireless networks and other networked systems. The Hello Flood attack begins when a malicious node within the network starts sending many HELLO packets. These packets are typically used in routing protocols to establish and maintain network connections between nodes. The malicious node floods the network with these HELLO packets. These packets are sent to a destination that does not exist or to multiple destinations, overwhelming the network infrastructure. The excessive number of HELLO packets consumes significant network resources, including bandwidth and processing power. Legitimate nodes in the network are forced to process these packets, which can lead to resource exhaustion. As a result of the resource exhaustion, the network's performance degrades significantly. Legitimate communication between nodes becomes difficult or impossible, leading to a denial of service for legitimate users. Hello Flood attack, is particularly relevant in IoT networks due to the high density of devices and the limited resources of sensor nodes. It is a common and effective method for disrupting network communication.

2) Version number : Version number attacks (VNAs), also known as protocol downgrade attacks, manipulate SSL/TLS compatibility to coerce a connection into using an older, less secure protocol. By intercepting and modifying ClientHello messages, attackers deceive servers into accepting outdated SSL/TLS versions. This allows attackers to exploit vulnerabilities like POODLE, compromising the connection's security and enabling eavesdropping, data tampering, and man-in-the-middle attacks. A VNA in RPL-based networks includes a malicious node changing the DIO control message's DODAG Version Number. Inconsistencies in the network might cause misrouting and other difficulties. VNA can lead to routing inconsistencies, packet loss, and overall network instability, as nodes may not be able to correctly interpret the routing information due to the manipulated version numbers.

3) Worst Parent Attack: The Worst Parent attack targets the routing protocol by causing a node to select a suboptimal parent node for routing. This is achieved by manipulating the metrics used to evaluate parent nodes, such as link quality or distance. This attack can result in inefficient routing paths, increased latency, and potential data loss, as packets may be routed through less reliable or slower paths than necessary which directly impacts the efficiency and reliability of routing crucial for the proper functioning of IoT networks.

III. Related Works

Conventional intrusion detection methods are limited by their inefficiency in processing extensive data, which results in restricted actionable insights. To enhance security in IoT networks, more sophisticated and flexible mechanisms such as ML and deep learning (DL) techniques are essential. ML excels in handling intricate scenarios by deriving insights from datasets, whereas DL provides superior accuracy while requiring extended training periods and substantial data volumes. These computational approaches effectively integrate intelligence into the IoT landscape. However, the challenge involves crafting ML and DL algorithms tailored for resource-constrained IoT networks. Deploying these algorithms across various nodes enables the identification of malicious activities, thereby evolving IoT security into an intelligence-driven monitoring system.

In [8], a cross-layer IDS for RPL-based IoTs was introduced. The method utilized a neural network to detect Version Number, Worst Parent, and Hello Flood attacks in both binary and multi-class classification scenarios by leveraging information from the routing and link layers across various IoT levels. The authors introduced new attributes from the connection layer to enhance intrusion detection. These features were categorized into data, topology, and link-layer attributes. The binary classification model exhibited a detection rate (DR) of 97.11% and a false positive rate (FPR) of 0.34% with a 10-fold cross-validated training, and a DR of 96.88% and an FPR of 0.13% with a 60% split. The multi-class model achieved a high DR of 97.52%. Notably, the inclusion of link-layer features reduced the FPR and improved VNA detection. A study by Fu et al. proposed a novel approach for detecting intelligent attacks using and long short-term memory (LSTM)-recurrent neural networks (RNN) [9]. This end-to-end framework seamlessly integrates data preprocessing, feature extraction, training, and detection to achieve superior detection rates. The model effectively distinguishes between attack and normal data, demonstrating cutting-edge performance with reduced time consumption compared to existing models. The authors in [10] addresse IoT routing attacks in healthcare settings through a framework employing convolutional neural networks (CNN) to minimize power consumption. Utilizing CNN, the study effectively identifies and forecasts different IoT routing attacks that influence power consumption, demonstrating accuracy improvements and decreased energy consumption. The study detailed in [11] introduces machine learning tools for a Cross-layer Hybrid IDS to detect IoT DoS attacks. Attack routes need not be compressed as machine learning algorithms can learn effectively from extensive datasets, especially when datasets exhibit high variances. To avoid overfitting or underfitting, the authors recommend combining multiple ML tools to develop a robust attack detection model. In a separate study [12], researchers explored RPL attacks using Google AutoML and Microsoft Azure ML for training their ML model, leveraging a variety of ML techniques. The investigation employed 2-class decision forest (DF) and 2-class support vector machines (SVM) classification algorithms. Furthermore, a Gated Recurrent Unit network model-based deep learning approach was devised in [13] to predict and prevent HF attacks on the RPL protocol in IoT networks.

In [14], a detailed review and taxonomy of detecting version number attacks (VNA) in LLNs for IoT routing is outlined. The study evaluates various defense strategies against VNAs in RPL-based IoT networks, encompassing secure protocol-based mechanisms, lightweight and AI-based methods, distributed monitoring, and intrusion detection systems. The VeRA system proposed employs hash chains to verify nodes with rank or version number alterations, enhancing VNA detection efficiency. By utilizing a hash-based authentication approach and multilayer encryption chains, VeRA ensures fast and secure validation of nodes' rankings and version numbers, solidifying its reliability in detecting VNAs within RPL-based IoT networks. The work provides a comprehensive analysis of VNA detection techniques in RPL-based IoT networks, evaluating each method based on accuracy, energy consumption, detection rate, and false positives, while also distinguishing its contributions and relevance compared to existing studies.

In their work [15], G. Sharma et al. illustrated the significant impact of version attacks on various metrics, showing a marked decrease in packet delivery ratio (PDR), particularly noticeable in scenarios involving mobile nodes that enhance PDR due to their mobility. Addressing threats like wormhole, black hole, and sinkhole attacks, researchers in [10] proposed robust authentication methods for H-IoT applications, analyzing attacks that compromise security and affect power consumption. The study aimed to pinpoint IoT routing vulnerabilities affecting smart device power usage. Introducing a deep learning-based machine learning algorithm in [16], a scalable solution was proposed to detect IoT routing attacks such as rank manipulation, hello-flood, and version number alterations with exceptional accuracy and sensitivity. Node power states, energy consumption variations, and model performance were evaluated against SVM and logistic regression methods.

The assessment of the RPL protocol under the influence of misbehaving nodes, detailed in [17], encompassed factors like overhead, convergence time, energy usage, parent changes, and network longevity. Results from simulations highlighted that rank attacks had a more severe impact in dynamic settings than in static environments. Furthermore, the study outlined in [18] introduces the Secured-RPL routing protocol to detect and prevent sinkhole attacks aiming to divert network traffic through a malicious node strategically positioned near the base station. These nodes intercept or manipulate incoming packets. The study in [19] investigates how applying feature selection techniques can improve IDS for IoT networks. The study addresses the security issues such as resource constraints and the heterogeneous nature of devices. By selecting the most relevant features from large, high-dimensional datasets, the proposed approach aims to reduce computational complexity while enhancing detection accuracy. Odeh et al. [20] proposed a state-of-the-art IDS tailored for IoT environments. The study primarily applies CNNs, along with AutoEncoders and LSTM networks. Key steps include extensive data preprocessing (cleaning, normalization, and feature selection) to enhance model efficiency. Notably, the CNN model achieved near-perfect results in binary classification (with accuracy, precision, recall, and F1 scores all around 99.89%), demonstrating its ability to differentiate between benign and malicious traffic.

The study by Selem at el. [21] introduces an ensemble learning approach merging deep neural networks (DNNs) and CNNs to increase the IDS accuracy and resilience in IoT networks. Evaluation conducted using the Edge-IIoTset dataset demonstrated the model's efficacy in enhancing security by leveraging DNNs' pattern recognition and CNNs' spatial feature identification. By combining these architectures, the IDS effectively detects and responds to various intrusion attempts, aligning with recent trends in IDS research that emphasize ensemble learning for improved detection in dynamic IoT threat environments.

Lastly, Table I offers a comparative overview of IDS performances utilizing diverse ML and neural network techniques as observed in different research studies.

Our comprehensive literature review underscores a critical

TABLE I
PERFORMANCE COMPARISONS AMONG DIFFERENT ML TECHNIQUES FOR RPL ATTACK DETECTION.

Reference	Year	ML Algorithm used	Accuracy
Selem et al. [21]	2025	DNN + CNN	98.2%
Odeh et al. [20]	2024	CNN, AutoEncoders, LSTM	99.89%
Almohaimeed et al. [19]	2024	Feature selection techniques	96.4%
Paul et al. [11]	2023	SVM, K-nearest neighbor, DT classifier, RF, and gradient boosting	95–96%
Ioulianou et al. [12]	2022	2-class DF, 2-class SVM	92.2%
Cakir et al. [13]	2020	RNN	99.52%
Canbalaban et al. [8]	2020	Neural networks	96%
Kamel et al. [10]	2020	Dynamic algorithm CNN	94%
Fu et al. [9]	2018	LSTM-RNNs	97.52%

gap in understanding how diverse layer features influence the detection of significant intrusions. Notably, there is a lack of research dedicated to identifying key cross-layer features, with existing studies primarily focusing on feature selection techniques applied to network characteristics rather than those spanning multiple layers within IoT networks. While current research tends to emphasize optimizing machine learning models or accuracy over pinpointing impactful feature combinations, our study aims to bridge this research gap by demonstrating that only a specific set of cross-layer features can proficiently detect intrusions. Utilizing these pivotal features can enable an IDS system to effectively mitigate the detrimental impacts of intrusions on network resources.

IV. MATERIALS AND METHODS

A. Dataset Description

In this study, we utilized an experimental dataset extracted from [8], which involved three different attack scenarios in a simulated large RPL-based network using Cooja. The simulation was conducted for 60 minutes in a 250 $\times$ 250 square meter area, with a total of 50 nodes including a sink node. Radio communication in the network followed a unit disc graph model, with a transmission range of 50 meters and an interference range of 100 meters. The initial positioning of the nodes in the simulation area was random. The MAC protocol used was IEEE 802.15.4, and the objective function for the routing protocol was minimum rank with hysteresis objective function (MRHOF). Each node was scheduled to send one user datagram protocol (UDP) packet every 60 seconds throughout the simulation time. The dataset consisted of 44,400 packet transmissions, incorporating 29 cross-layer features.

1) Cross-layer Features and Class Labels: The cross-layer features for the network nodes include the Data, Link, and Topology layer-specific attributes. The Data layer features include DATA packets related information such as packet length, number of transmitted packets, and the interval between packet transmissions. On the other hand, the Link layer features incorporate the number of dropped packets during the transmission due to collisions, queueing, neighbor allocation, and packeting. The features from the Topology layer comprise critical information regarding routing control messages called DIO, DAO, and DIS, during the attack simulation. These cross-layer features are detailed described in Table II. The network packets are labeled into two classes, benign and malicious with '0' and '1' labels, respectively. Using these class labels, 14,400 packets are labeled as benign, and 30,000 packets are classified as malicious. In the current study, the samples with malicious class are considered as positive samples and the benign samples are considered as negative. Therefore, the ratio between positive and negative samples is 2:1.

B. Dataset Preprocessing

As a pre-processing step, we performed feature normalization using a Z-score standardization technique. The aim was to achieve a consistent numerical range for different types of features, each with its varying ranges. This involved transforming each feature value by subtracting the mean and scaling it to unit variance. By applying the formula $x' = x - \mu/\sigma$, where x' represents the transformed normalized value, x is the original value, μ is the mean of the feature column, and σ is the standard deviation of the feature column, we obtained a normalized range for each feature column with a mean of zero (0) and a standard deviation of one (1). A positive standardized z-score signifies that the value is above average, while a negative score indicates that the corresponding value is below average. Feature normalization serves as a crucial pre-processing step in the application of machine learning algorithms.

C. Feature Importance Analysis

Prior to applying any machine learning algorithms, it is crucial to identify the features that have the most predictive power or contribute the most to predicting the output. These important features play a significant role in forecasting the outcome through the hypothesis function. On the other hand, less relevant features can be eliminated in subsequent stages such as model training and testing due to their lower impact on the outcome. This simplifies the model and accelerates the processes of training and prediction.

In the current study, we selected the most relevant features for the binary classification problem using the recursive feature elimination by cross-validation (RFECV) method. The RFECV algorithm identifies important features by gradually eliminating lesser relevant or unnecessary features through cross-validation. It chooses the optimal subset of features based on the cross-validation score of the external estimator model. In our work, we used the RF classifier as the external predictor in a 5-fold cross-validation framework within the RFECV algorithm. Fig. 4 illustrates the RFECV-generated CV scores for the features utilized in model training. The graph demonstrates the changes in the CV scores across various splits. The optimal number of features is determined to be 12 by the RFECV algorithm.

Fig. 5 displays the ascending order ranking of the 12 crucial features sourced from the Topology and Data layers.

TABLE II
DESCRIPTION OF DATASET FEATURES. NOTE THAT THE 'NA' IN THE COLUMN INDICATES NOT AVAILABLE, WHEREAS '×' REFERS TO NON-CALCULATED VALUES.

Feature name	Description	Min	Max	Average	Difference	Layer
DIO message count	Describes the number of DIO messages sent. DIO messages multicast to help other nodes discover the RPL instance to join.			NA		Topology
DIS message count	Describes the number of DIS messages sent. It requests for DIO neighbor discovery in the RPL instance.			NA		Topology
DAO message count	Describes the number of DAO messages sent. DAO messages construct the paths for message flows from child nodes to the parent node or to the root node.			NA		Topology
Version number	The version number is carried in a DIO message. It is used as an indicator for global repair operations. The DODA/G root node is the only node that can change the version number.	√	√	×	√	Topology
Rank	The node's rank value is a numerical representation of where it stands in relation to the root node. A lower rank value indicates that the node is closer to the sink node, whereas a higher rank value indicates the reverse.	√	√	√	×	Topology
Data length	The number of bytes in data packet	√	√	√	×	Data
Interval between DATA messages	The time elapsed between transmission of two data packets	√	√	√	×	Data
Interval between DIO messages	The time elapsed between transmission of two DIO messages	√	√	√	×	Topology
Interval between DIS messages	The time elapsed between transmission of two DIS messages	√	√	√	×	Topology
Interval between DAO messages	The time elapsed between transmission of two DAO messages	√	√	√	×	Topology
Number of dropped packets	due to collision/neighbour allocation/queueing/packeting			NA		Link
Number of DATA messages	Describes the number of DATA messages sent.			NA		Data

D. ML Models

In the current research, we employed six supervised classification models for the binary classification problem using the twelve most relevant features. These models include the RF, extreme gradient boosting (XgBoost), CatBoost, mulitlayer perceptron (MLP), K-nearest neighbor (KNN), and DT. These algorithms are widely recognized as state-of-the-art machine learning techniques that excel in handling non-linear classification learning problems. Below, we provide a brief description of each ML algorithm.

1) RF: The RF is a classification algorithm developed based on the random forest ensemble technique. It acts as a meta-estimator by combining multiple randomly drawn decision trees from the data and providing the output of a new classification as the majority vote. As an ensemble classifier, the RFR produces a stronger model by combining multiple decision tree models. It utilizes bootstrapping to randomly sample subsets of the dataset over a number of iterations using the required number of features to build each decision tree estimator [22].

2) CatBoost: CatBoost is a decision tree-based gradient-boosting model for supervised classification problems. It follows the boosting principle, sequentially fitting multiple weak models that are slightly better estimators than random choices. During the fitting process, the models continuously learn from the mistakes of previous models, and the process continues until a certain evaluation metric, such as accuracy or log-loss, is no longer improved. The minimization of the evaluation metric is performed using a gradient descent optimization algorithm, giving the technique the name "gradient boosting" [23].

3) XgBoost: The XgBoost is an efficient implementation of a gradient-boosting algorithm for classification-based predictive modeling. Like generic boosting, XgBoosting adds decision trees one at a time to an ensemble machine learning model

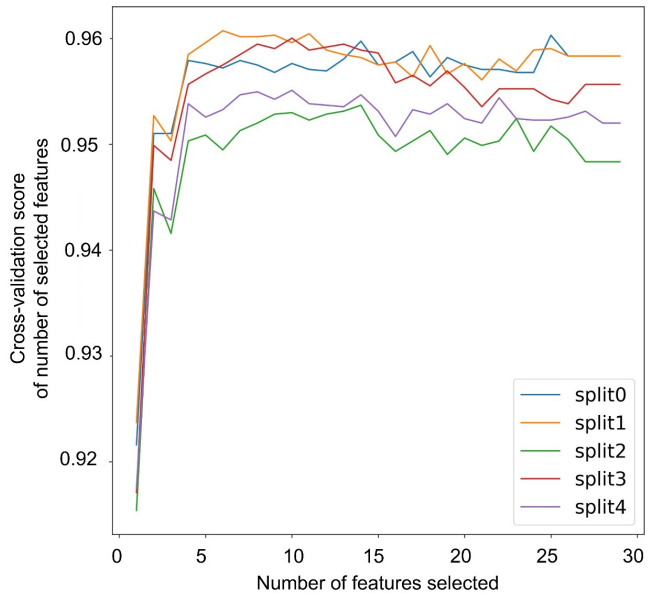

Fig. 4. RFECV-reported cross-validation (CV) scores are shown during the feature selection process in a 5-fold CV framework.

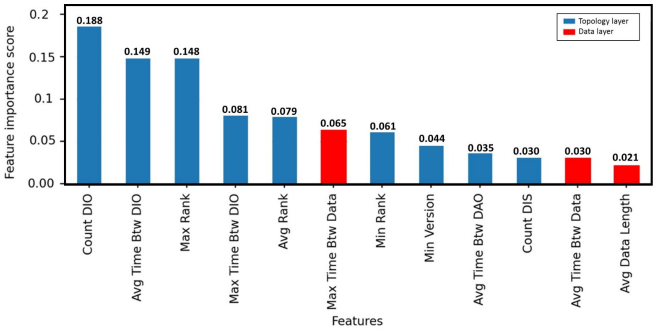

Fig. 5. RFECV-identified 12 most important cross-layer features for the binary class Intrusion detection and corresponding importance scores.

and corrects the classification errors made by previous models. It is designed to be computationally efficient and offers faster execution and enhanced prediction performances [24].

4) MLP: MLP implements a multi-layer perceptron algorithm that is trained using backpropagation in a neural network architecture consisting of several non-linear hidden layers, input, and output layers. The MLPR model is able to learn non-linear function approximations for any classification problem from the training observations [25].

5) KNN: The KNN is a non-parametric classification method that can be used for classification modeling. In general, the KNNR algorithm uses the concept of "feature proximity" to estimate the class label of any new observation. In this technique, a new observation is assigned a class label based on the majority class of its nearest neighbors in the training set. The K-value in the KNNR model indicates the number of neighbors that need to be considered. The proximity between features is usually calculated using distance methods such as Euclidean and Manhattan distance [26].

6) DT: DT Classification is the simplest classification technique that builds the classification model in the form of a tree structure. It splits the dataset into smaller subsets while developing an incremental associated decision tree. Finally, a single tree is produced with decision and leaf nodes. The decision nodes represent the feature attributes used to build the model, whereas the leaf nodes represent the class labels. The root node represents the best feature predictor and serves as the topmost decision node in the tree [26].

E. IDS Model Optimization and Training

The training process of the IDS was initiated by selecting suitable ML algorithms for the binary classification problem at hand. It involved several steps, starting with splitting the datasets into training and hold-out testing portions. The hyperparameters of the ML model were then tuned to the training dataset using an appropriate technique. Once the optimal hyperparameters were determined, the model was fitted using these parameters on the training data. Subsequently, the models were evaluated on the hold-out test data using appropriate evaluation metrics. It is crucial to highlight that the model's hyperparameter tuning, fitting, and testing, was conducted using the most significant cross-layer features identified during the feature importance analysis phase. In the present study, the dataset was divided into 80% for training and 20% for independent testing using a random sampling technique. Within the training phase, the hyperparameters of the ML algorithms were fine-tuned using a k-fold cross-validation framework. The goal was to find the optimal hyperparameter values that would yield the lowest estimation error for the current classification problem. In this approach, the training data points were divided into k-folds, with the ML model being trained on $(k-1)$ folds while validating the model with the remaining fold. This process was repeated k times until each fold served as a validation set for model evaluation. For this study, a cross-validated grid-search method was employed to tune the hyperparameters of the classification models on the training dataset. This method involved attempting various combinations of specified hyperparameters and their corresponding values, measuring the accuracy of the model for each combination on the validation dataset. The combination of hyperparameters that yielded the best performance, indicated by the highest accuracy, was reported as the optimal hyperparameters for a given ML model. Finally, the ML model was trained using the optimal hyperparameters on the entire training dataset consisting of 35,520 packet data. After the training, the model was evaluated on a hold-out dataset comprised of 8,880 network packets. The complete IDS development process using the "CatBoost" as the reference model is depicted in Fig. 6.

F. IDS Model Evaluation

To evaluate the performance of intrusion detection model, we chose a set of evaluation metrics suitable to assess binary class classification performances. The metrics are described below.

1) Confusion Matrix (CM): A CM is a tabular representation providing a detailed breakdown of the model's predicted labels compared to the actual ground truth labels across positive and negative classes. In a binary classification scenario, the

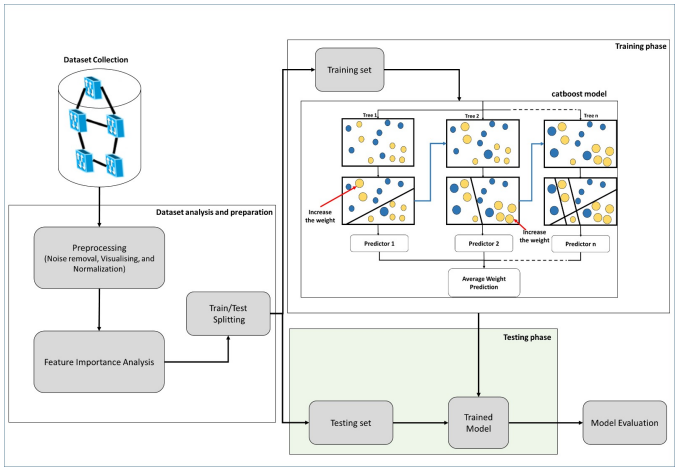

Fig. 6. The IDS development process is explained, which consists of three steps. 1) Dataset analysis and preparation; 2) Training; 3) Testing. Step-1 involves dataset preprocessing, feature importance analysis, and splitting into training and testing sets. Step-2 focuses on training the IDS with the CatBoost model, while the final Step evaluates the IDS model using an independent test dataset.

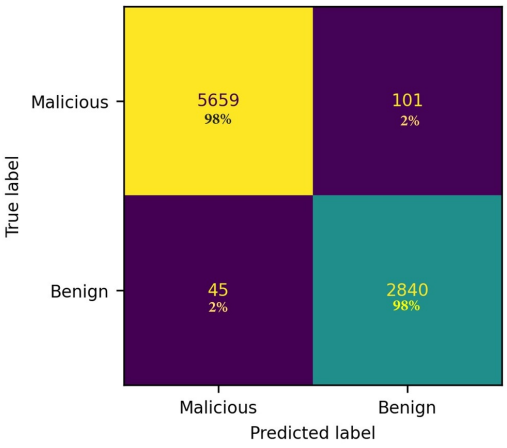

Fig. 7. A CM representing the CatBoost model performance in malicious vs. benign class classification task. In this representation, 'Malicious' represents the positive class, while 'Benign' indicating the negative class.

CM consists of four elements, namely true positive (TP), false positive (FP), true negative (TN), and false negative (FN). TP indicates the instances where the model correctly predicts the positive class, whereas TN refers to the instances where the model correctly predicts the negative class. On the other hand, FP specifies the cases where the model incorrectly predicts the positive class, while FN represents the cases where the model incorrectly predicts the negative class. Fig. 7 demonstrates a CM representing the binary classification performances of the CatBoost model.

2) Positive Predictive Value (PPV): The PPV is a way to measure the accuracy of the prediction model. It calculates the percentage of positive predictions that are truly positive. This metric is also known as Precision. The mathematical formula for PPV is shown in (1).

$$\text{PPV} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{1}$$

3) Negative Predictive Value (NPV): The NPV is a technique to quantify the accuracy of the prediction model. It calculates the percentage of negative predictions that are truly negative. Eq. (2) shows the formula for NPV.

$$\text{NPV} = \frac{\text{TN}}{\text{FN} + \text{TN}}. \tag{2}$$

4) Specificity: Specificity is a measure that calculates the model's ability to correctly identify the negative samples as negative. It is the ratio between truly predicted negative samples and the total number of negative samples. The Specificity formula is described in (3)

$$\text{Specificity} = \frac{TN}{TN + FP}. \tag{3}$$

5) False Positive Rate (FPR): FPR is a measure that calculates the proportion of the positive samples incorrectly predicted as positive. It is calculated as the ratio between the number of FP cases and the total number of actual negative samples. The mathematical equation for FPR is specified in (4).

$$\text{FPR} = \frac{\text{FP}}{\text{FP} + \text{TN}}. \tag{4}$$

The lower the FPR, the higher the performance of the classification model.

6) Detection Rate (DR): DR is also expressed as a TP rate (TPR). This metric measures the proportion of the positive samples correctly predicted as positive. The DR is also expressed as sensitivity or Recall. It is calculated as the ratio between the truly predicted positive samples and the total number of positive samples. Eq. (5) demonstrates the DR formula.

$$\text{DR} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{5}$$

As the DR increases, the performance of the classification model improves.

7) Receiver Operating Characteristics (ROC) – Area Under Curve (AUC): The ROC curve and AUC are ML metrics for binary classification models, with ROC-AUC offering an overall performance assessment across classification thresholds. The ROC curve graphically represents the trade-off between TP and FP rates, illustrating model performance at different thresholds based on TPR and FPR metrics. Fig. 8 illustrates the ROC curves and AUC scores for all the ML binary classification models developed in the current study. For an intrusion detection model, it is crucial to achieve a higher DR while maintaining a lower FPR. This indicates the model's effectiveness in accurately detecting malicious nodes and distinguishing between benign and malicious nodes during the attack.

V. Results

A. Benign vs. Malicious Node Classification Results

The binary classification results of six ML algorithms using individual layer features for the independent test data set are

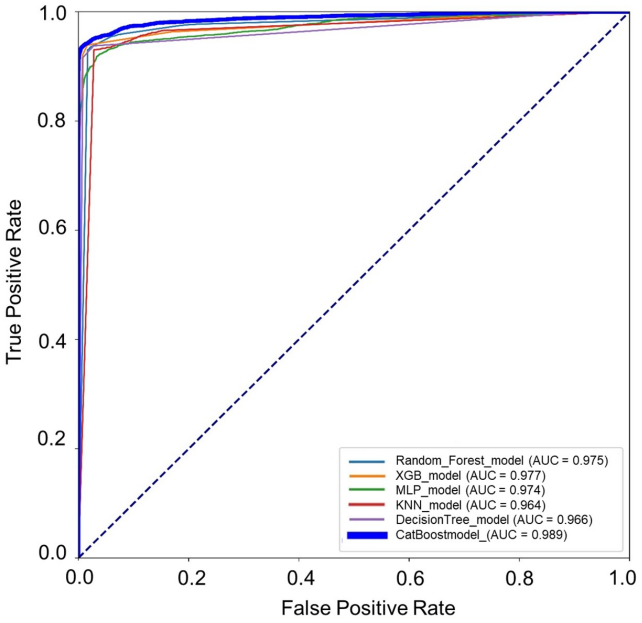

Fig. 8. The ROC curves for the ML models showcasing the AUC scores for the binary classification models for the intrusion detection.

presented in Table III. Table IV displays an in-depth performance analysis of the CatBoost model, incorporating four supplementary metrics: PPV, NPV, Sensitivity, and Specificity. Table V provides a performance analysis of the CatBoost model across two distinct network settings, while Table VI offers a comparative performance evaluation between the current study and a related study that utilized the same dataset.

B. An algorithmic framework for Intrusion detection and prevention in IoT network:

We proposed a centralized intrusion detection and prevention Algorithm 1 for an RPL-based IoT network utilizing the most significant cross-layer feature-based IDS in the present study, for instance, maximum rank, maximum time between DIO messages, average time between DIO messages, average rank of the nodes, and minimum rank of the nodes.

VI. Discussion

A. Individual and combined-layer feature effects in intrusion detection

We conducted an assessment to evaluate the accuracy of intrusion detection using individual layer features. Our objective was to assess the effectiveness of each layer's features in identifying malicious nodes. We carried out a total of seven experiments, exploring different combinations of features from single-layer, all-layer, and pairwise-layer sets. The results of these experiments are summarized in Table III, which presents the evaluation metrics of DR and FPR for all six ML models. In the experiments focusing on individual layers, we observed that the topology layer features yielded the highest DR ($\geq$99%) and lowest FPR ($<$1%). These topological features include important information such as the

Algorithm 1 Intrusion detection and prevention in IoT using cross-layer features

Input: Network topology with nodes description
Output: Intrusion detection and warning messages
Collect_network_thresholds()
Update_network_params()
if (maxRank $>$ maxRankThreshold) **or** (minRank $<$ minRankThreshold) **then**
  IS_ATTACK = run_intrusion_detection_model()
  if IS_ATTACK **then**
    Trigger_warning('Worst parent attack')
    Isolate_block_suspicious_nodes(Node ID)
  end if
end if
if (maxTimeM $<$ maxTimeMThreshold) **or** (avgTimeM $<$ avgTimeMThreshold) **or** (countM $>$ countMThreshold) **then**
  IS_ATTACK = run_intrusion_detection_model()
  if IS_ATTACK **then**
    Trigger_warning('Hello Flood attack')
    Traffic_flow_rate_limiting()
  end if
end if
if (avgRank $>$ avgRankThreshold) **or** (rootMaxVersion $>$ rootMaxVersionThreshold) **then**
  IS_ATTACK = run_intrusion_detection_model()
  if IS_ATTACK **then**
    Trigger_warning('Version Number Attack')
    Authenticate_validate_nodes()
  end if
end if

Update_network_params()

maxRank = 0
maxTimeM = 0
avgTimeBetweenM = 0
avgRank = 0
minRank = 0
totalNodes = 0
countM = 0
rootMaxVersion = 0
for each node $i = 1$ to n **do**
  Retrieve rank$_i$, timeM$_i$
  Update totalNodes
  if rank$_i$ $>$ maxRank **then**
    maxRank = rank$_i$
  end if
  if timeM$_i$ $>$ maxTimeM **then**
    maxTimeM = timeM$_i$
  end if
end for
avgRank = avgRank / totalNodes
avgTimeM = avgTimeM / totalNodes
countM = total M messages transmitted and received
return

Collect_network_thresholds()

return maxRankThreshold, minRankThreshold, rootMaxVersion

**M = DAO/DIO/DATA Messages*

TABLE III
BINARY CLASSIFICATION RESULTS OF SIX ML MODELS USING VARIOUS COMBINATIONS OF LAYER FEATURES ON THE INDEPENDENT TEST DATA. THE EVALUATION METRICS 'DR' AND 'FPR' ARE REPORTED IN DIFFERENT COLUMNS.

MODEL	DATA LAYER		LINK LAYER		TOPOLOGY LAYER		All LAYERS		DATA +LINK		LINK +TOPOLOGY		DATA +TOPOLOGY		IMPORTANT FEATURES	
	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR	DR	FPR
RF	0.853	0.505	1	1	0.986	0.034	0.991	0.040	0.842	0.373	0.986	0.035	0.992	0.043	0.991	0.021
CatBoost	0.911	0.536	1	1	0.983	0.017	0.983	0.016	0.840	0.282	0.984	0.023	0.984	0.021	0.982	0.008
MLP	0.935	0.590	1	1	0.977	0.019	0.979	0.031	0.846	0.306	0.987	0.040	0.981	0.027	0.977	0.023
KNN	0.868	0.511	1	1	0.982	0.037	0.966	0.029	0.845	0.376	0.981	0.039	0.971	0.034	0.972	0.012
DT	0.895	0.520	1	1	0.977	0.018	0.972	0.020	0.789	0.216	0.977	0.026	0.972	0.029	0.978	0.020
XGBoost	0.901	0.536	1	1	0.984	0.025	0.988	0.031	0.835	0.287	0.985	0.023	0.991	0.032	0.987	0.013

TABLE IV
COMPREHENSIVE BINARY CLASSIFICATION RESULTS FROM THE CATBOOST CLASSIFICATION MODEL ON THE INDEPENDENT TEST DATA.

Features / Metrics	Data Layer	Link Layer	Topology Layer	All layers	Data +Link	Link + Topology	Data + Topology	Important Features
PPV	0.772	0.674	0.991	0.992	0.856	0.989	0.987	0.992
NPV	0.723	0.586	0.966	0.968	0.693	0.968	0.968	0.964
Sensitivity	0.911	1	0.983	0.983	0.841	0.984	0.984	0.982
Specificity	0.464	0	0.982	0.984	0.718	0.978	0.976	0.984

TABLE V
A COMPARATIVE ANALYSIS OF BINARY CLASSIFICATION RESULTS BY THE CATBOOST CLASSIFICATION MODEL ON THE SMALL AND LARGE NETWORK DATASETS.

METRICS / NETWORK SIZE	DR	FPR	specificity	Sensitivity	PPV	NPV
Small (125 × 125) meter^2	0.965	0.013	0.987	0.965	0.994	0.931
Large (250 × 250) meter^2	0.982	0.008	0.984	0.982	0.992	0.964

TABLE VI
COMPARATIVE ANALYSIS OF THE BINARY CLASSIFICATION RESULTS.

Model	**All Layer**		**Topology**		**Link + Topology**	
	DR	FPR	DR	FPR	DR	FPR
Canbalaban et al. 2020 [8]	0.931	NA	0.971	0.61	0.969	0.13
Current study	**0.982**	**0.008**	**0.983**	**0.017**	**0.984**	**0.023**

number of transmitted DIO, DIS, and DAO messages, message transmission frequency, DIO message version number, and node ranking within the network. All these features contain crucial information related to the intrusion attacks analysed in our study. Following the topological features, the data layer features exhibited the next best performance in identifying malicious nodes. These features provide information about DATA packets, such as packet length, transmission interval, and the number of packet transmissions.

It is worth noting that the link layer features, which include details about dropped packets due to collisions, neighbour allocation, queuing, and packeting, may not be particularly relevant to the intrusion attacks in the network. In few instances, using these features yield extreme prediction by predicting all benign nodes as malicious. Significantly, the ML models exhibited a 100% FPR when exclusively utilizing these link-level attributes as predictive features, and this pattern was consistent even with the best-performing model, CatBoost. The lower performance observed in the link layer features can be attributed to the fact that, in most cases, packet drops in the routing layer are primarily caused by link layer issues in the normal (non-attack) scenario. Only VNAs have the potential to introduce some packet losses [8]. As a result, while link layer features offer valuable diagnostic information for identifying packet losses, they are not reliable indicators for detecting intrusion attacks such as Hello Flood and Worst Parent attacks. This limitation leads to ambiguous predictions, where benign nodes may be incorrectly classified as malicious. Our findings highlight the potency of topological features in detecting intrusion attacks, followed by data layer features. The link layer features, while less relevant to intrusion attacks, offer important diagnostic insights.

B. Identification of important cross-layer features and their contributions

We conducted a comprehensive feature importance study to not only identify significant features that effectively distinguish between malicious and benign nodes but also to reduce the feature dimension in the intrusion detection problem. The analysis revealed the identification of 12 important features, with nine (9) belonging to the topology layer and the remaining three derived from the data layer, as shown

in Fig. 5. This result highlights the significance of topology-related features, as they contain crucial information relevant to intrusion attacks. Specifically, these features capture important indicators related to network routing and control packets transmitted within the network. For instance, the number of DIS and DIO messages serves as valuable indicators for detecting intrusion attacks. For example, a sudden surge in DIS messages could potentially signify the occurrence of a Hello Flood attack within the network [8]. On the other hand, the version number and rank information of the routing nodes offers valuable insights for detecting specific types of attacks. The version number can be utilized to identify VNAs, while the rank information can be instrumental in detecting the Worst Parent attacks.

The 12 significant features were subsequently utilized in the model hyperparameter optimization, training and evaluation of six ML models for the binary classification task, and the results are presented in the last column of Table III. Notably, the trends observed in terms of the DR metric were like those obtained using topological-layer features. However, when examining the FPR, most of the ML models demonstrated a decrease in this metric. This decrease indicates an improved ability of the models to avoid mistakenly identifying benign nodes as malicious (class label '1'). The classification performance achieved using these important features is comparable to models that employ topology layer features either individually or in combination with other layer features. Importantly, this improved performance is accomplished with a significantly lower number of features, reducing the dimension from the original 29 features to 12. This feature dimension reduction contributes to the cost-effectiveness of the present intrusion detection model. Intrusion detection models aim to strike a balance between achieving a high DR to accurately identify genuine threats and maintaining a low FPR to minimize false alarms. Our proposed cross-layer feature-based intrusion detection model, which exhibits both high DR and low FPR, signifies enhanced reliability, trustworthiness, effectiveness, and efficiency in the context of RPL-based IoT networks.

C. CatBoost-based IDS model excels over other models

Among the six ML classification models examined, the CatBoost model exhibited the most outstanding performance when considering both DR and FPR evaluation metrics. Gradient-boosting based decision trees, such as the CatBoost model, are widely recognized as high-performing ML models that have consistently proven effective in several recent intrusion detection studies [27]–[29]. It is important to highlight that our experiment also encompassed other DL models, including CNNs and bi-directional LSTM-RNNs, but none surpassed the results obtained by the CatBoost model. One of the key aspects of CatBoost learning contributing to its superior performance is gradient regularization, a technique used to prevent the model from overfitting by penalizing large gradients during the training process. This regularization technique helps to smooth the learning process and improve the model's generalization over the unseen data. As previously mentioned, an effective IDS requires a high DR while maintaining a low FPR. In a comparative analysis with other machine learning models, as illustrated in Table III, the CatBoost model consistently exhibited outstanding performance across various feature combinations from different layers in terms of both DR and FPR. While models like the DT-based bagging algorithm RF showcased superior DR across nearly all feature combinations, they tended to exhibit comparatively lower performance in FPR. Conversely, the XGBoost model, a gradient-boosting counterpart, displayed performance levels close to CatBoost in terms of DR but demonstrated lower trends in FPR. CatBoost's optimized learning technique, which dynamically adjusts the learning rate, facilitates more efficient learning and improved generalization, potentially resulting in decreased FPRs. For a comprehensive understanding of the CatBoost model's performance, we meticulously analyzed its results using possible combinations of features, including the 12 significant cross-layer features, which are presented in Table IV. Additionally, we employed four additional evaluation metrics—PPV, NPV, Sensitivity, and Specificity. It is worth noting that higher values for these performance indicators indicate superior model performance. Impressively, the CatBoost model consistently delivered exceptional results, achieving scores exceeding 95% across all evaluation metrics. Once again, these outcomes demonstrate that the CatBoost model's performance, when utilizing important features, is comparable to that achieved using topological-layer features in any combination. However, it is noteworthy that the former approach employs a smaller subset of the original feature set, highlighting its efficiency and effectiveness.

In addition, we examined the CM for the classification outcomes generated by the CatBoost model, as depicted in Fig. 7. In The matrix provides insights into the TP, FP, FN, and TN prediction values arranged in successive columns, which correspond to 98%, 2%, 2%, and 98% of values, respectively. These results showcase that the CatBoost model correctly identified malicious (positive class) samples as malicious in 98% of cases while achieving a 98% accuracy rate in recognizing benign (negative class) samples. A considerably low FP of 2% indicates the model's ability to effectively avoid misclassifying benign samples as malicious. Moreover, only 2% of the malicious samples were mistakenly predicted as benign by the CatBoost model in the present study. Additionally, the ROC curve displaying AUC scores in Fig. 8 highlights CatBoost's exceptional performance compared to other ML models. The CatBoost achieves an AUC score of 98.9% unequivocally establishing the model as the most effective classifier for the current problem. A recent review of CatBoost model performance in predictive analytics highlights its superior performance, particularly in generating the lowest FPRs compared to other Gradient Boosting-based decision tree models in loss prediction tasks within power distribution networks [23]. Notably, CatBoost's capability to facilitate optimal feature interactions, known as "feature combinations," stands out as a key strength. This ability allows the model to select the most effective feature combinations during training, positioning CatBoost as one of the most effective models for minimizing FP predictions [30].

Regarding generalizability, we assessed the performance of

the trained CatBoost model on a separate simulated dataset, designed as an independent test set. This dataset was generated within a network simulation area measuring 125 × 125 square meters, with network configurations identical to the original dataset. Referred to as 'Small', in contrast to the original 'Large' dataset, Table V presents a comparative analysis of CatBoost's performance across both datasets. Notably, CatBoost exhibits consistent performance trends on the 'Small' dataset, showcasing an excellent DR of 96.5% and a FPR of 1.3%. Other evaluation metrics also exhibit close similarity to the original dataset's performance. This experiment showcases the CatBoost model's ability to generalize effectively across diverse network settings and simulation scenarios.

D. Intrusion prevention utilizing cross-layer features

The algorithm depicted in Algorithm 1, outlines an approach for intrusion detection and prevention in IoT networks using important cross-layer features. It is important to note that the proposed intrusion prevention system (IPS) serves as a network behavior analysis (NBA) system, detecting unusual traffic patterns and potential intrusion attacks by employing an anomaly-based detection technique. By comparing sampled network traffic against a baseline threshold, the IPS triggers IDS mechanisms and swiftly invokes preventive actions when network activity surpasses these parameters, aiming to maintain network performance and provide real-time responses to threats. The algorithm initiates by gathering adaptive thresholds specific to network parameters like max/minRankthreshold, max/avgTimeDIOthreshold, countDIOthreshold, and more. Subsequently, it fetches the most recent metrics for network nodes, computed iteratively across the network's nodes. These metrics are then compared against predefined threshold values by the monitoring module. If the network parameters surpass these thresholds, the intrusion detection model i.e., IDS (specifically, the CatBoost binary classifier) is activated with 12 cross-layer features. If the IDS identifies certain network nodes as malicious, it marks the IS_ATTACK flag as TRUE. Should the IDS indicate a potential attack, be it a Hello Flood, Worst Parent, or VNA, a corresponding alert is triggered, prompting the implementation of appropriate preventive measures. The algorithm's use of cross-layer features is a notable strength. By analyzing key metrics such as node rank, intervals between DIO messages, and root version, the algorithm integrates data from various layers of the network stack. The algorithm then strategically responds to dynamic network conditions based on real-time metrics, ensuring proactive and adaptive security protocols. For instance, it identifies Worst Parent attacks by comparing the rank differentials among nodes, promptly activating the intrusion detection model when irregularities are detected. To preemptively thwart potential threats, the algorithm isolates or blocks nodes displaying anomalous behavior. In the case of Hello Flood attacks, the algorithm scrutinizes the intervals between DIO messages and the message frequencies, promptly engaging the intrusion detection model to assess network conditions. Upon detection, the algorithm conducts a thorough traffic analysis and initiates rate limiting mechanisms to control network congestion effectively. Moreover, the algorithm distinguishes VNAs by cross-referencing average rank data with the root node's version number, promptly triggering the intrusion detection module when inconsistencies arise. As a preventive security measure, the algorithm enforces stringent authentication and validation protocols across network nodes to enhance overall network integrity.

The proposed IPS can be integrated into the Security Layer of an RPL-based IoT network, where it can actively monitor real-time traffic, analyze patterns, and detect anomalies to identify security breaches in the network routing level. The IDS in the prevention framework strengthens overall network security by working alongside firewalls, encryption protocols, and access controls utilized in the IoT network. Positioned in the Security Layer, the prevention algorithm conducts real-time monitoring and threat detection, safeguarding IoT devices and data from cyber threats. This placement facilitates early risk identification and mitigation, fortifying the security and resilience of the IoT network. On the other hand, by continuously monitoring the network and collecting updated values for relevant metrics, the algorithm ensures that the intrusion detection and prevention system remains adaptive and responsive to evolving network conditions. Periodically updating the thresholds based on observed behaviour and changing security requirements enables the system to effectively detect and prevent new types of intrusions. Hence, the proposed intrusion prevention algorithm offers a comprehensive approach to enhance network security through utilizing important cross-layer features and incorporating regular monitoring and threshold updates.

E. Comparison with existing approach

We conducted a comparative analysis between our proposed detection model, and the original dataset study conducted by Canbalaban et al. [8]. The aim of this comparative analysis is to assess the effectiveness of the models in detecting anomalies within the network and involves only those model evaluations that utilize the current attack dataset for both model training and validation. In the original dataset study, the authors presented a neural network-based classification framework specifically designed to address the multi-class intrusion detection problem. The proposed neural network architecture consisted of four hidden layers utilizing the rectified linear unit (ReLU) activation function. Our comparative analysis focused solely on binary classification, as the multi-class class labels were unavailable in our current study. The results of this comparison can be found in Table VI. We evaluated the performance of the models using two key performance metrics: DR and FPR. The comparison was made among models utilizing topology layer features, topology+link layer features, and all layer features. Interestingly, our CB model outperformed the NN model in both detecting malicious nodes and accurately identifying normal nodes without falsely predicting those as attacker nodes. This trend was consistent across all feature combinations. Notably, the improvement was particularly significant in terms of lowering the FPR metric, showing a remarkable 97% and 82% enhancement over the NN model's performance when using topology and topology+link layer features, respectively

These examples showcase the CB model's effectiveness in preventing the misclassification of normal nodes as malicious. Furthermore, the incorporation of routing layer features in the CB model resulted in an improved and consistent DR output. However, in the case of the NN model, the DR dropped when Data and Link-layer features were added to the topology layer features.

VII. Limitations and Future Research

One limitation of the study is the absence of multiclass attack labels in the dataset, restricting detailed data point classification. This constraint can be mitigated by crafting tailored simulation scenarios within RPL networks to generate attack-specific cross-layer feature datasets. Additionally, future research might explore incorporating AutoML techniques to boost IoT intrusion detection system adaptability to evolving threats and dynamic network behavior. AutoML simplifies automatic model selection dynamically, real-time hyperparameter tuning, and feature engineering for developing robust and flexible IDSs. Another avenue for research involves empirical studies in real-world RPL-based IoT networks to validate the proposed system's effectiveness and viability. Field trials offer insights into system performance and functionality under varying network conditions, highlighting challenges like extreme weather impacting network reliability. Further research could investigate adapting the system to decentralized IoT, addressing communication challenges, scalability, consensus, and security using technologies like distributed ledgers, peer-to-peer communication, consensus algorithms, homomorphic encryption, and edge computing for data protection.

VIII. Conclusion

This study presents an in-depth ML-based analysis focusing on identifying key cross-layer features crucial for detecting resource-based intrusions within RPL-based networks. Our research identified a set of 12 significant features integrating topology and data layer attributes. Among various ML models tested for intrusion detection, the CatBoost model demonstrated superior performance, achieving exceptional results with a DR of 99%, Sensitivity of 98%, PPV of 98%, and a remarkably low FPR of less than 1% when exploiting these critical cross-layer features. Moreover, our research delineates an intelligent intrusion prevention algorithm that employs cross-layer feature-based IDS to effectively pinpoint malicious nodes whenever network metrics display aberrant behaviors in RPL-based IoT networks. This algorithm presents a comprehensive strategy for strengthening network security, providing effective measures to prevent intrusions, and fortifying overall network protection. The proposed IPS can be easily deployed in smart cities or healthcare IoT networks by tailoring prevention system's configuration to the unique requirements and constraints such as network size, device heterogeneity, data sensitivity, regulatory compliance of the host environment.

References

[1] S. C. Mukhopadhyay and N. K. Suryadevara, "Internet of things: Challenges and opportunities," *Internet of Things*, pp. 1–17, 2014.

[2] H. A. Khattak, M. A. Shah, S. Khan, I. Ali, and M. Imran, "Perception layer security in Internet of things," *Future Gener. Comput. Syst.*, vol. 100, pp. 144–164, 2019.

[3] M. A. Burhanuddin *et al.*, "A review on security challenges and features in wireless sensor networks: IoT perspective," *J. Telecommun. Electron. Comput. Eng. JTEC*, vol. 10, no. 1–7, pp. 17–21, 2018.

[4] X. Liu, Z. Sheng, C. Yin, F. Ali, and D. Roggen, "Performance analysis of routing protocol for low power and lossy networks (RPL) in large scale networks," *IEEE Internet Things J.*, vol. 4, no. 6, pp. 2172–2185, 2017.

[5] A. Mayzaud, R. Badonnel, and I. Chrisment, "A taxonomy of attacks in RPL-based Internet of things," *Int. J. Netw. Secur.*, vol. 18, no. 3, pp. 459–473, 2016.

[6] I. Mishra, "A detailed classification of routing attacks against RPL in Internet of things," vol. 3, no. 1, pp. 692–703, 2017.

[7] A. Zolfaghari *et al.*, "A multi-mode WPAN (Bluetooth, BLE, IEEE 802.15.4) SoC for low-power and IoT applications," *Symposium on VLSI Circuits*, pp. C74–C75, 2017.

[8] E. Canbalaban and S. Sen, "A cross-layer intrusion detection system for RPL-based Internet of things," in *Proc. ADHOC-NOW*, 2020.

[9] Y. Fu *et al.*, "An intelligent network attack detection method based on RNN," in *Proc. IEEE DSC*, 2018.

[10] S. O. M. Kamel and S. A. Elhamayed, "Mitigating the impact of IoT routing attacks on power consumption in IoT healthcare environment using CNN," *Int. J. Comput. Netw. Inf. Secur.*, vol. 14, no. 4, pp. 11–29, 2020.

[11] A. Paul, S. Sinha, and S. Mishra, "Machine learning based hybrid intrusion detection system for detecting cross-layer DOS attacks in IoT," 2023.

[12] P. P. Ioulianou, V. G. Vassilakis, and S. F. Shahandashti, "Ml-based detection of rank and blackhole attacks in RPL networks," in *Proc. IEEE CSNDSP*, 2022.

[13] S. Cakir, S. Toklu, and N. Yalcin, "RPL attack detection and prevention in the Internet of things networks using a GRU based deep learning," *IEEE Access*, vol. 8, pp. 183678–183689, 2020.

[14] N. A. Alfriehat *et al.*, "Detecting version number attacks in low power and lossy networks for IoT routing," *IEEE Access*, vo. 12, pp. 31136–31158, 2024.

[15] G. Sharma, J. Grover, and A. Verma, "Performance evaluation of mobile RPL-based IoT networks under version number attack," *Comput. Commun.*, vol. 197, pp. 12–22, 2023.

[16] F. Y. Yousef, D. Ünal, and E. Gül, "Deep learning for detection of routing attacks in the Internet of things," *Int. J. Comput. Intell. Syst.*, vol. 12, no. 1, pp. 39–58, 2018.

[17] S. Ibrahimy, H. Lamaazi, and N. Benamar, "RPL assessment using the rank attack in static and mobile environments," in *Proc. IEEE 3ICT*, 2020.

[18] A. Jamil, M. Q. Ali, and M. E. A. Alkhalec, "Sinkhole attack detection and avoidance mechanism for RPL in wireless sensor networks," *Ann. Emerg. Technol. Comput.*, vol. 5, no. 5, pp. 94–101, 2021.

[19] M. Almohaimeed and F. Albalwy, "Enhancing IoT network security using feature selection for intrusion detection systems," *Appl. Sci.*, vol. 14, no. 24, pp. 2076–3417, 2024.

[20] A. Odeh and A. A. Taleb, "Robust network security: A deep learning approach to intrusion detection in IoT," *Comput. Mater. Cont.*, vol. 81, no. 3, pp. 4149–4169, 2024.

[21] M. Selem, F. Jemili, and O. Korbaa, "Deep learning for intrusion detection in IoT networks," *Peer-to-Peer Netw. Appl.*, vol. 18, no. 2, p. 22, 2025.

[22] V. Svetnik *et al.*, "Random forest: A classification and regression tool for compound classification and QSAR modeling," *J. Chem. Inf. Comput. Sci.*, vol. 43, no. 6, pp. 1947–1958, 2003.

[23] J. T. Hancock and T. M. Khoshgoftaar, "Catboost for big data: An interdisciplinary review," *J. Big Data*, vol. 7, no. 1, p. 94, 2020.

[24] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proc. ACM SIGKDD*, 2016.

[25] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5–6, pp. 183–197, 1991.

[26] T. Hastie, R. Tibshirani, and J. H. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics. New York, NY: Springer, 2009.

[27] A. Jumabek, S. Yang, and Y. Noh, "Catboost-based network intrusion detection on imbalanced cic-ids-2018 dataset," *J. Korean Institute Commun. Info. Sci.*, vol. 46, no. 12, pp. 2191–2197, 2021.
[28] O. Almomani *et al.*, "Reconnaissance attack detection via boosting machine learning classifiers," in *Proc. ICSIC*, 2022.
[29] R. Bolleddula *et al.*, "Routing attack detection using ensemble artificial intelligence model for iIoT," in *Proc. IEEE INCOFT*, 2023.
[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A. Dorogush, and A. Gulin, "Catboost: Ubiased boosting with categorical features," in *Proc. NeurIPS*, 2018.

Noor Hafsa earned her B.Sc. and M.Sc. from the University of Dhaka, Bangladesh, and her Ph.D. in Computing Science from the University of Alberta, Canada. She previously served as a Research Scientist in the Department of Electrical and Computer Engineering at Texas A&M University at Qatar (TAMUQ). Currently an Assistant Professor at King Faisal University's College of Computer Science and Information Technology, her research applies machine learning to diverse domains including natural language processing, image processing, cybersecurity, computational biology, and engineering.

Hadeel AlZoubi is an Assistant Professor at the College of Computer Science and Information Technology, King Faisal University. She earned her Ph.D. in 2018 from the University of the West of Scotland, with a specialization in Artificial Intelligence. Her research interests include the application of Machine Learning, Electronic Health Records, Computer Vision, and Natural Language Processing.

Sajida Imran received her Ph.D. in Computer Engineering from Ajou University, Republic of Korea, in 2018. She is currently serving as a Senior Lecturer in the Department of Computer Science at the University of Chichester, United Kingdom. Dr. Imran has authored a number of publications in international journals and conferences and has contributed to several funded research projects. Her research interests include wireless technologies for object detection and tracking with a focus on security and fault tolerance, as well as applications of the Internet of Things (IoT) using machine learning techniques.