ITRC
ICT Research Institute
(Iran Telecommunication Research Center)

Journal of Information and Communication Technology

Iranian Society of Information and Communication Technology

Volume 18, Issue 68-69, Spring and Summer 2026, pp. 115-129

A Hybrid Intrusion Detection System for RPL-Based IoT Networks Using Decision Tree and CNN-BiLSTM Models

Mohammad Fazeli[1*], Mohsen Raji [2], Mohammad Mojtaba Fazeli [3]

[1] Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[2] Department of Electrical and Computer Engineering, Shiraz University, Shiraz, Iran
[3] Department of Computer Engineering, Islamic Azad University, Sepidan Branch, Sepidan, Iran

Received: 19 January 2026, Revised: 02 June 2026, Accepted: 21 July 2026
Paper type: Research

Abstract

The Routing Protocol for Low-Power and Lossy Networks (RPL) is a widely adopted standard in Internet of Things (IoT) environments, enabling efficient and stable communication. However, its susceptibility to cyberattacks poses a significant threat. Traditional intrusion detection methods, while effective, are computationally complex and resource-intensive, rendering them unsuitable for resource-constrained IoT devices. This paper presents a lightweight, two-layer intrusion detection system (IDS) for RPL-based IoT networks. The first layer employs a decision tree model to identify potential threats, offering rapid analysis with minimal computational overhead. The second layer utilizes a hybrid Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BiLSTM) model to classify attack types. This combination facilitates the automatic extraction of deep features from network traffic data and the learning of complex temporal dependencies. Experimental results on the ROUT-4-2023 dataset, comprising four common RPL attacks (Blackhole, Flooding, Version Number, and Rank Attacks), demonstrate the proposed method's superior performance, achieving an overall accuracy of 97% with precision, recall, and F1-scores of approximately 96%. The proposed system achieves an attack detection time of 0.062 seconds. These findings confirm the efficacy and high speed of our approach, establishing it as a suitable solution for securing resource-constrained IoT networks.

Keywords: Intrusion Detection, Internet of Things (IoT), IoT Security, RPL Protocol, RPL attacks, Lightweight Detection Model, Layered Security Architecture, Attack Classification, Decision Tree (DT), CNN-BiLSTM.

[*] Corresponding Author's email: mohammad.fazeli@hafez.shirazu.ac.ir

دوفصلنامه
فناوری اطلاعات و ارتباطات ایران

سال هجدهم، شماره‌های ۶۸ و ۶۹ ، بهار و تابستان ۱۴۰۵، صفحه ۱۱۵ الی ۱۲۹

یک سامانه تشخیص نفوذ ترکیبی برای شبکه‌های اینترنت اشیای مبتنی بر RPL با بهره‌گیری از مدل‌های درخت تصمیم و CNN-BiLSTM

محمد فاضلی[1*]، محسن راجی[2]، محمد مجتبی فاضلی[3]

[1]دانشکده مهندسی برق و کامپیوتر، دانشگاه شیراز، شیراز، ایران

[2]دانشکده مهندسی برق و کامپیوتر، دانشگاه شیراز، شیراز، ایران

[3]دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد سپیدان، سپیدان، ایران

تاریخ دریافت: ۱۴۰۴/۱۰/۲۹ تاریخ بازبینی: ۱۴۰۵/۰۳/۱۲ تاریخ پذیرش: ۱۴۰۵/۰۴/۳۰

نوع مقاله: پژوهشی

چکیده

پروتکل مسیریابی برای شبکه‌های کم‌مصرف و پراتلاف (RPL:Routing Protocol for Low-Power and Lossy Networks) به‌عنوان استانداردی رایج در مسیریابی شبکه‌های اینترنت اشیاء شناخته شده و به طور گسترده ای در این شبکه‌ها استفاده می‌شود. با این حال، آسیب‌پذیری این پروتکل در برابر حملات سایبری، تهدیدی جدی محسوب می‌شود. روش‌های سنتی تشخیص نفوذ، به دلیل پیچیدگی محاسباتی و مصرف بالای منابع، دستگاه‌های اینترنت اشیاء مناسب نیستند. در این پژوهش، یک سامانه تشخیص نفوذ دو لایه‌ای سبک‌وزن برای شناسایی و طبقه‌بندی حملات در شبکه‌های اینترنت اشیاء مبتنی برRPL ارائه شده است. در لایه اول، از یک مدل درخت تصمیم برای شناسایی وقوع حمله استفاده می‌شود که ضمن سرعت بالا، نیاز به منابع محاسباتی اندکی دارد. در لایه دوم، برای طبقه‌بندی دقیق نوع حمله، از ترکیبی از شبکه عصبی کانولوشنی و حافظه بلند-کوتاه‌مدت دوطرفه (CNN-BiLSTM) بهره گرفته شده است. این ترکیب، توانایی استخراج خودکار ویژگی‌های عمیق از داده‌های ترافیک شبکه و یادگیری وابستگی‌های زمانی و الگوهای پیچیده حملات را دارد.نتایج آزمایش‌ها بر روی مجموعه داده ROUT-4-20223 با چهار حمله متداول RPL شامل سیاه‌چاله، سیل‌آسا، شماره‌نسخه و کاهش‌رتبه نشان می‌دهد که روش پیشنهادی با صحت کلی ۹۷٪ و مقادیر دقت، بازخوانی و امتیاز F1 حدود ۹۶٪، عملکرد بهتری نسبت به روش‌های موجود دارد. زمان شناسایی حمله نیز به حدود ۰,۰۶۲ ثانیه دست یافته است. نتایج به دست آمده، کارایی مناسب و سرعت بالای سامانه پیشنهادی را در محیط‌های اینترنت اشیاء مبتنی بر RPL نشان می‌دهد.

کلیدواژگان: تشخیص نفوذ، اینترنت اشیاء (IoT)، امنیت اینترنت اشیاء، مدل سبک وزن تشخیص نفوذ، معماری امنیتی لایه‌ای.

* رایانامهٔ نویسنده مسؤول: mohammad.fazeli@hafez.shirazu.ac.ir

۱- مقدمه

اینترنت اشیا (IoT)[1] به‌عنوان یکی از فناوری‌های تحول‌آفرین قرن حاضر، با اتصال میلیاردها دستگاه به شبکه‌های ارتباطی، امکان ایجاد سیستم‌های هوشمند در حوزه‌هایی نظیر خانه‌های هوشمند، حمل‌ونقل، کشاورزی دقیق و مراقبت‌های بهداشتی را فراهم کرده است. این فناوری با بهره‌گیری از دستگاه‌های متصل، داده‌های عظیمی تولید می‌کند که تحلیل آن‌ها می‌تواند به بهبود کیفیت زندگی و بهینه‌سازی فرایندها منجر شود [۱]. با این حال، محدودیت‌های منابع در دستگاه‌های IoT، از جمله توان پردازشی پایین، حافظه محدود و مصرف انرژی بالا، چالش‌های جدی را در طراحی و پیاده‌سازی این سیستم‌ها ایجاد کرده است.

یکی از پروتکل‌های کلیدی در شبکه‌های IoT، پروتکل مسیریابی RPL است که توسط گروه ویژه مهندسی اینترنت (IETF)[2] برای شبکه‌های کم‌مصرف و با تلفات بالا (LLN)[3] استاندارد شده است. این پروتکل مسیریابی با استفاده از ساختار گراف جهت‌دار بدون چرخه (DODAG)[4] و معیارهای بهینه‌سازی مانند مصرف انرژی یا تعداد گام‌ها، مسیریابی کارآمدی را در این شبکه‌ها فراهم می‌کند [۲]. بااین‌وجود، این پروتکل در برابر حملات سایبری نظیر حمله سیاه‌چاله، سیل‌آسا و دستکاری‌رتبه آسیب‌پذیر[5] است. این حملات می‌توانند عملکرد شبکه را مختل کرده و امنیت آن را به خطر اندازند. روش‌های سنتی مانند رمزنگاری سنگین یا سیستم‌های تشخیص نفوذ (IDS)[6] به دلیل نیاز به منابع محاسباتی بالا، برای محیط‌های محدود منابع IoT ناکارآمد هستند [۳].

برای غلبه بر این چالش‌ها، رویکردهای نوینی مانند یادگیری ماشین و یادگیری عمیق برای تشخیص نفوذ در شبکه‌های IoT پیشنهاد شده‌اند. به‌عنوان نمونه، از مدل‌های سنتی مانند ماشین بردار پشتیبان (SVM)[7]، درخت تصمیم (DT) و بیز ساده (NB)[8] و همچنین از مدل‌های یادگیری عمیق نظیر شبکه عصبی کانولوشنی (CNN)[9]، شبکه حافظه بلند-کوتاه‌مدت[10] (LSTM) در سیستم‌های تشخیص نفوذ در شبکه‌های اینترنت اشیا استفاده شده است. ازسوی‌دیگر، باتوجه‌به تفاوت بنیادی که پروتکل مسیریابی RPL دارد، برای شبکه‌های مبتنی بر این پروتکل مسیریابی سامانه‌های تشخیص نفوذ خاصی با بهره‌گیری از مدل‌های مختلف یادگیری ماشین و عمیق مانند جنگل تصادفی (RF)[11]، شبکه عصبی چندلایه (MLP)، ماشین بردار پشتیبان (SVM)، شبکه LSTM، و مدل ترکیبی CNN-LSTM ارائه شده است.

باوجود دستیابی به دقت نسبتاً مطلوب در شناسایی انواع حملات، بسیاری از روش‌های پیشین مبتنی بر مدل‌های عمیق یا ترکیبی، به دلیل پیچیدگی محاسباتی و نیاز بالای انرژی، برای استقرار در محیط‌های واقعی IoT با منابع محدود مناسب نیستند [۳]، [۸]، [۱۱].

در این پژوهش، یک سامانهٔ تشخیص نفوذ دولایه‌ای سبک‌وزن و کارآمد برای شناسایی و طبقه‌بندی حملات در شبکه‌های اینترنت اشیاء مبتنی بر RPL ارائه شده است. در لایهٔ نخست از سامانه پیشنهادی، به‌منظور عملکرد دقیق، سریع و بدون نیاز به منابع محاسباتی زیاد از یک مدل درخت تصمیم برای تشخیص وقوع حمله استفاده می‌شود. سپس در لایه دوم، ترکیبی از شبکه عصبی کانولوشنی و حافظه بلند - کوتاه‌مدت دوطرفه (CNN-BiLSTM) به کار گرفته می‌شود تا نوع حمله مشخص شود. علت انتخاب این مدل به دلیل توانایی بالای آن در استخراج خودکار ویژگی‌های عمیق از داده‌های ترافیک شبکه توسط لایه‌های CNN و یادگیری وابستگی‌های زمانی و الگوهای پیچیده حملات از طریق BiLSTM بوده، که همین امر موجب افزایش دقت طبقه‌بندی، کاهش نرخ هشدار کاذب و بهبود شناسایی حملات ناشناخته و پیچیده می‌شود. علاوه بر این، طراحی دومرحله‌ای این مدل باعث شده است تا با انتقال وظیفه غربالگری اولیه به یک لایه سبک، بار محاسباتی سیستم به‌شدت کاهش یابد. این رویکرد نه‌تنها باعث تسریع در فرایند آموزش نسبت به سایر مدل‌های پیچیده عمیق شده، بلکه نیاز به منابع حافظه را نیز تعدیل می‌کند تا سامانه به‌راحتی بر روی گره‌های محدود در شبکه‌های اینترنت اشیاء قابل‌اجرا باشد. عملکرد مدل پیشنهادی با استفاده از مجموعه‌داده ROUT-4-2023 ارزیابی می‌شود که شامل چهار نوع حمله متداول RPL از جمله سیاه‌چاله، سیل‌آسا، شماره نسخه و کاهش رتبه ارزیابی شده است. نتایج آزمایش‌ها نشان می‌دهد که روش پیشنهادی با دستیابی به صحت

[1] Internet of Things
[2] Internet Engineering Task Force
[3] Low-Power and Lossy Networks
[4] Destination Oriented Directed Acyclic Graph
[5] Rank Manipulation
[6] Intrusion Detection System
[7] Support Vector Machine
[8] Naive Bayes
[9] Convolutional Neural Network
[10] Long Short-Term Memory
[11] Random Forest

کلی ۹۷٪ و مقادیر دقت، بازخوانی و امتیاز F1 حدود ۹۶٪، عملکرد بهتری نسبت به روش‌های موجود دارد. همچنین، زمان شناسایی حمله به حدود ۰٫۰۶۲ ثانیه بهبودیافته است. نتایج به‌دست‌آمده، گویای کارایی مطلوب و سرعت پردازش فزاینده سامانه سلسله‌مراتبی پیشنهادی در محیط‌های اینترنت اشیاء مبتنی بر RPL است؛ درحالی‌که دقت بالای سیستم در مواجهه با دستگاه‌های محدود از نظر منابع، به‌خوبی حفظ شده است.

ادامهٔ این مقاله به شرح زیر سازماندهی شده است: بخش ۲ کارهای مرتبط با امنیت RPL و تشخیص حمله مبتنی بر یادگیری ماشین و یادگیری عمیق را مرور می‌کند. بخش ۳ مدل پیشنهادی و تنظیمات مربوطه و مجموعه‌داده استفاده شده را معرفی می‌کند، بخش ۴ نتایج تجربی را ارائه می‌دهد و عملکرد چارچوب سامانه را بر اساس مجموعه‌داده ROUT-4-2023 ارزیابی می‌کند. در نهایت، بخش ۵ این مقاله را به پایان می‌رساند و خلاصه نتیجه حاصل از این پژوهش را ارائه می‌دهد.

۲- کارهای مرتبط

در سال‌های اخیر، به دلیل گسترش دستگاه‌ها در شبکه‌های اینترنت اشیا (IoT) و وابستگی روزافزون آن‌ها به پروتکل‌های مسیریابی، تحقیقات گسترده‌ای در زمینه تشخیص نفوذ و شناسایی حملات در شبکه‌های مبتنی بر RPL انجام شده است. این مطالعات عمدتاً بر دو محور اصلی متمرکز بوده‌اند:

۱. مدل‌های یادگیری عمیق برای تشخیص نفوذ
۲. روش‌های یادگیری گروهی و بهینه‌سازی ترکیبی برای بهبود دقت و کارایی.

در ادامه به کارهای انجام‌گرفته در هر محور پرداخته می‌شود.

۲-۱- مدل‌های یادگیری عمیق برای تشخیص نفوذ

در سیستم‌های تشخیص نفوذ برای شبکه‌های IoT، مدل‌های یادگیری عمیق نتایج چشمگیری داشته‌اند.

در [۵]، از شبکه‌های حافظه بلند-کوتاه‌مدت (LSTM) برای شناسایی حملات محروم‌سازی از سرویس (DoS)[1] در محیط‌های IoT استفاده شده است. این مدل توانسته وابستگی‌های زمانی داده‌ها را به خوبی یاد بگیرد و دقت تشخیص بالاتری نسبت به روش‌های سنتی ارائه دهد، اما همچنان به منابع محاسباتی زیادی برای آموزش و استنتاج نیاز دارد.

در ادامه، [۶]، یک مدل شبکه عصبی پیچشی زمانی (TCNN[2]) همراه با مهندسی ویژگی پیشرفته را معرفی کرد که توانست ویژگی‌های حملات را با دقت بالاتری استخراج کند و عملکرد مناسبی ارائه دهد، اما همچنان نیاز به توان محاسباتی بالایی دارد.

همچنین در [۷]، یک مدل یادگیری عمیق فضایی-زمانی نیمه‌نظارتی [3] معرفی شده که نیاز به داده‌های برچسب‌گذاری‌شده را کاهش می‌دهد و تعمیم‌پذیری بهتری نسبت به مدل‌های نظارتی کامل ارائه می‌کند. با این حال، پیچیدگی بالای مدل، پیاده‌سازی آن در گره‌های کم‌منبع LLN را محدود می‌سازد.

به طور مشابه، در [۸]، ترکیبی از شبکه‌های کانولوشنی (CNN) و حافظه بلند-کوتاه‌مدت دوطرفه (BiLSTM) برای طبقه‌بندی حملات RPL در شبکه‌های هوشمند IoT پیشنهاد شد. این مدل با بهره‌گیری همزمان از ویژگی‌های مکانی و زمانی، دقتی بیش از ۹۰٪ به دست آورد اما دقت و زمان محاسبه همچنان چالش است.

۲-۲- مدل‌های یادگیری گروهی و ترکیبی برای تشخیص نفوذ

چندین پژوهش اخیر بر استفاده از مدل‌های یادگیری گروهی[4] تمرکز کرده‌اند تا دقت و کارایی تشخیص حملات RPL را افزایش دهند. در [۹]، یک مدل گروهی شامل درخت تصمیم، بیز ساده، شبکه عصبی مصنوعی و نزول گرادیان تصادفی معرفی شد که دقت حدود ۸۶٪ به دست آورد، اما دقت و زمان محاسبه همچنان چالش است.

در [۱۰]، یک چارچوب یادگیری گروهی مبتنی بر انتخاب ویژگی ژنتیکی پیشنهاد شد که با ترکیب طبقه‌بندی‌کننده‌هایی مانند جنگل تصادفی و AdaBoost[5] توانست نرخ تشخیص بالایی در شناسایی حملاتی مانند سیل DIS[6] و دستکاری نسخه RPL به دست آورد، اما پیچیدگی محاسباتی بالای آن، کاربرد عملی‌اش را در دستگاه‌های IoT با منابع محدود دشوار می‌کند.

در [۱۱]، یک مدل ترکیبی بهینه‌سازی‌شده مبتنی بر الگوریتم

[1] Denial of Service
[2] Temporal Convolutional neural network
[3] Semi-supervised Spatiotemporal DL
[4] Ensemble Learning
[5] Adaptive Boosting
[6] DODAG Information Solicitation

آموزش-یادگیری (TLBO[1]) و ماشین یادگیری شدید (ELM[2]) معرفی شد که با انتخاب بهینه ویژگی‌ها، سرعت آموزش و دقت بالایی را به دست آورد، اما اجرای آن روی گره‌های کم‌منبع مورد بررسی کامل قرار نگرفته است.

در نهایت، [۱۲]، یک سیستم تشخیص نفوذ مبتنی بر یادگیری گروهی برای شبکه‌های RPL ارائه می‌کند که با ترکیب چندین الگوریتم، نسبت به مدل‌های تک‌لایه‌ای، دقت بالاتری در شناسایی انواع حملات ارائه می‌دهد.

به‌طورکلی، هرچند مدل‌های ترکیبی و مبتنی بر یادگیری عمیق دقت و انعطاف‌پذیری بالایی در شناسایی نفوذ ارائه داده‌اند، اما بیشتر این روش‌ها محدودیت‌های منابع دستگاه‌های IoT را نادیده گرفته‌اند. با بررسی مطالعات فوق، مشخص است که تمرکز اصلی اکثر پژوهش‌ها (نظیر [۵] و [۸]) بر بهبود شاخص‌های صحت بوده و به مسئله پیچیدگی مدل کمتر پرداخته شده است. همچنین، مدل‌های گروهی [۹] اگرچه در شناسایی حملات موفق بوده‌اند، اما به دلیل نیاز به اجرای همزمان چندین مدل، بار پردازشی بالایی تحمیل می‌کنند. پژوهش حاضر، با هدف رفع این چالش و پر کردن این خلأ، یک چارچوب دولایه و سبک‌وزن ارائه می‌شود که با ترکیب هوشمند الگوریتم‌های کلاسیک و عمیق، توازنی مؤثر میان دقت تشخیص و کارایی محاسباتی برقرار می‌سازد. در لایه‌ی نخست، از یک درخت تصمیم سبک‌وزن برای شناسایی سریع ناهنجاری‌ها استفاده می‌شود و در لایه‌ی دوم، ترکیب شبکه عصبی کانولوشنی و حافظه بلند–کوتاه‌مدت دوطرفه (CNN-BiLSTM) به‌منظور طبقه‌بندی دقیق انواع حملات RPL به‌کار گرفته می‌شود. چارچوب پیشنهادی با حفظ دقت بالا و بهبود چشمگیر پیچیدگی محاسباتی، امکان پیاده‌سازی مقیاس‌پذیر و کارآمد در شبکه‌های IoT مبتنی بر این پروتکل مسیریابی را فراهم می‌سازد.

در مجموع، بررسی پیشینه پژوهش نشان می‌دهد که اگرچه روش‌های مبتنی بر یادگیری عمیق توانسته‌اند دقت بالایی در تشخیص نفوذ به دست آورند، اما اکثر این مدل‌ها با چالش جدیِ پیچیدگی محاسباتی بالا روبرو هستند که استفاده از آن‌ها را در گره‌های محدود منبع IoT با دشواری مواجه می‌کند. ازسوی‌دیگر، مدل‌های سبک‌وزن‌تر (مانند درخت تصمیم یا الگوریتم‌های کلاسیک) اگرچه سرعت بالایی دارند، اما در تفکیک حملات مشابه و پیچیده در پروتکل RPL کارایی لازم را ندارند.

به‌منظور تبیین جایگاه پژوهش حاضر و برجسته‌سازی نوآوری آن، جدول ۱ به مقایسه تحلیلی میان مدل پیشنهادی و برجسته‌ترین مطالعات اخیر می‌پردازد. این مقایسه نه‌تنها از منظر شاخص کمی (صحت)، بلکه از دیدگاه کیفی شامل مزایا و معایب عملیاتی انجام شده است. همان‌طور که در این جدول مشاهده می‌شود، ویژگی متمایز کار حاضر، بهره‌گیری از یک ساختار دولایه است؛ این معماری به مدل اجازه می‌دهد تا با استفاده از لایه اول (غربالگری سریع)، سربار محاسباتی را به‌شدت کاهش داده و تنها نمونه‌های مشکوک را برای تحلیل دقیق‌تر به لایه دوم (CNN-BiLSTM) ارسال کند. این رویکرد، خلأ موجود در کارهای پیشین (مانند مراجع [۸] و [۹]) را که میان «دقت» و «پیچیدگی» با تضاد روبرو بودند، به طور موثری پوشش داده است.

جدول ۱. تحلیل مقایسه‌ای چارچوب پیشنهادی و روش‌های پیشرفته بر اساس معیارهای عملکردی و قابلیت‌های معماری

مدل	صحت	مزایای اصلی	معایب و چالش‌ها	دولایه
TLBO-ElM [11]	۸۸,۲۱	سرعت آموزش بالا	عدم کارایی در شناسایی حملات پیچیده شبکه؛ دقت پایین در لایه‌های توزیع‌شده	✗
SS-DeepID [7]	۸۸,۳۳	کاهش نیاز به داده‌های برچسب‌گذاری شده (نیمه‌نظارتی)	پیچیدگی بالای مدل برای گره‌های سنسور؛ نرخ مثبت کاذب (FPR)بالا در ترافیک‌های مشابه.	✗
RNN-BPTT [6]	۸۷,۲۰	توانایی درک وابستگی‌های زمانی کوتاه.	مشکل محوشدگی گرادیان؛ مصرف انرژی بسیار زیاد در دستگاه‌های با منابع محدود.	✗
CNN-BiLSTM [8]	۹۱,۸۷	استخراج هم‌زمان ویژگی‌های مکانی و زمانی.	سربار محاسباتی بسیار سنگین برای پردازش تمام نمونه‌ها (حتی نمونه‌های عادی).	✗
DT-NB-ANN-SGD [9]	۸۶,۲۱	استفاده از دیتاست تخصصی RPL	نرخ فراخوانی (Recall) پایین؛ عدم توانایی در طبقه‌بندی دقیق حملات مشابه در.RPL	✗
Proposed	۹۷,۰۰	توازن بهینه میان سرعت و دقت؛ سبک‌وزن بودن برای لبه شبکه.	نیاز به بررسی در مواجهه با طیف وسیع‌تری از حملات ترکیبی (در دست توسعه).	✓

[1] Teaching learning-based optimization

[2] Extreme Learning Machine

۳- روش پیشنهادی دولایه‌ای برای تشخیص نفوذ در شبکه‌های اینترنت اشیا مبتنی بر RPL

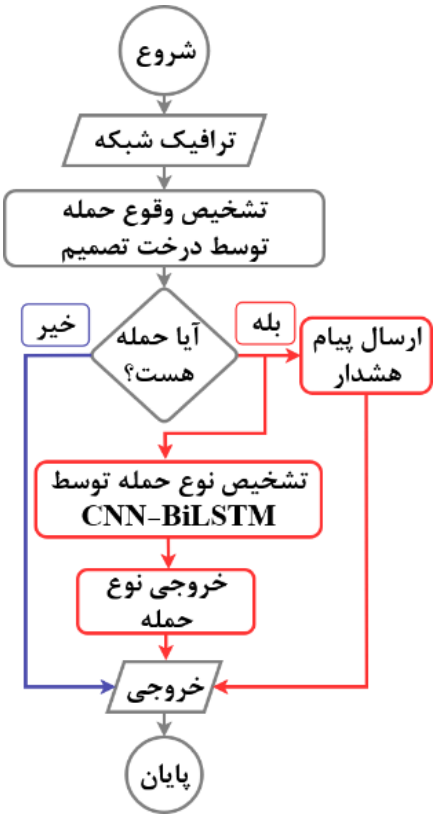
در این بخش، مدل تشخیص نفوذ دولایه‌ای و ترکیبی پیشنهادی برای شبکه‌های اینترنت اشیاء مبتنی بر پروتکل مسیریابی RPL با بهره‌گیری از یادگیری ماشین و یادگیری عمیق معرفی می‌شود. به این منظور، ابتدا دید کلی از این سیستم ارائه می‌گردد و سپس، توضیح مفصلی دربارهٔ تنظیمات آزمایشی و مجموعه‌داده مورد استفاده در این پژوهش ارائه خواهد شد. در ادامه نیز مراحل پیش‌پردازش داده‌ها، آموزش مدل و فرایند ارزیابی به‌تفصیل شرح داده می‌شوند.

۳-۱- نمای کلی از روش پیشنهادی

شکل ۱ نمای کلی روش پیشنهادی تشخیص نفوذ را نمایش می‌دهد. در این مدل، ابتدا ترافیک شبکه وارد یک مدل سبک‌وزن درخت تصمیم (DT) می‌شود که با سرعت بالا ترافیک ورودی را تجزیه و تحلیل کرده و نمونه‌های مشکوک را شناسایی و علامت‌گذاری می‌کند. سپس در صورت شناسایی رفتار غیرعادی، یک پیام هشدار به خروجی ارسال می‌شود. این لایه به‌عنوان یک فیلتر اولیه، نقش مؤثری در حذف بخش عمده‌ای از ترافیک‌های بی‌خطر با حداقل سربار محاسباتی ایفا می‌کند. در مرحله بعد، ترافیک علامت‌گذاری‌شده به لایه دوم ارسال می‌شود. این لایه از ترکیب یک شبکه عصبی کانولوشنی و حافظه بلند-کوتاه مدت دوطرفه (CNN-BiLSTM) تشکیل شده است که تجزیه و تحلیل عمیق‌تری انجام داده و نوع دقیق حمله RPL را شناسایی می‌نماید. در نهایت، خروجی سیستم شامل تشخیص نهایی و نوع حمله شناسایی‌شده خواهد بود. فلسفه بهره‌گیری از این توالی، بیشینه‌سازی بهره‌وری منابع در لایه‌های عملیاتی است. در واقع، مدل DT به عنوان یک سدِ واکنشیِ سریع، ترافیک متعارف را در همان مراحل اولیه عبور می‌دهد تا توان پردازشی لایه CNN-BiLSTM تنها برای تحلیل سناریوهای پیچیده و حملات احتمالی فعال شود.

۳-۲- تنظیمات آزمایشی

روش پیشنهادی در پایتون با استفاده از TensorFlow 2.10.1 پیاده‌سازی و بر روی یک رایانه شخصی مجهز به پردازنده گرافیکی NVIDIA GeForce GTX 1660 Ti با ۶ گیگابایت VRAM اجرا شد. این سیستم از CUDA 11. با درایور NVIDIA نسخه ۲۲,۰۶ برای فعال کردن شتاب‌دهی GPU استفاده کرد .محیط بر روی سیستم عامل ویندوز اجرا شد.



شکل ۱. روند کار مدل پیشنهادی

۳-۳- مجموعه‌داده (ROUT-4-2023)

ما برای آزمایش مدل پیشنهادی از مجموعه‌دادهٔ ارائه شده در [۴] استفاده کردیم که یک مجموعه ترافیک شبکه IoT با تمرکز بر حملات RPL است. این مجموعه‌داده‌ها از طریق شبیه‌سازی‌های Cooja از یک شبکه LoWPAN۶ تحت عملکرد عادی و چهار سناریوی حمله ایجاد شده است. جدول ۲ تعداد رکوردهای جریان مربوط به هر نوع حمله را در مجموعه‌دادهٔ [۴] خلاصه می‌کند.

جدول ۲. اندازه برچسب مجموعه‌داده ROUT-4-2023 [۴]

انواع حمله	تعداد ردیف‌ها
حمله سیاه‌چاله (Blackhole)	۴۰۴،۱۳۴
حمله شماره نسخه (DODAG)	۴۶۸،۰۶۰
حمله سیل‌آسا (Flooding)	۳۹۸،۷۸۲
حمله کاهش رتبه (Rank)	۳۶۸،۹۹۹

این مجموعه شامل حدود ۱٫۶۴ میلیون رکورد جریان است که حدود ۱٫۰۶ میلیون از آنها با برچسب «عادی» و ۰٫۵۸ میلیون با برچسب «حمله» مشخص شده‌اند. چهار کلاس حمله عبارت‌اند از :

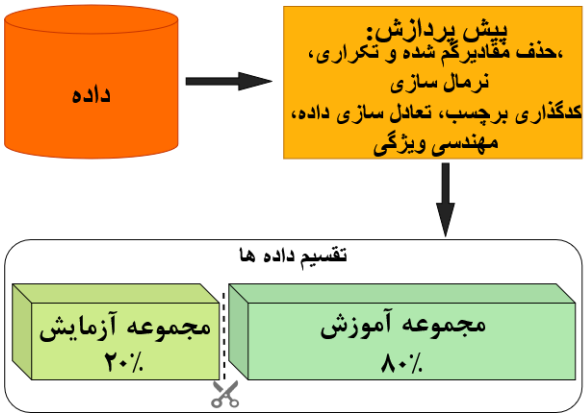
- سیاه‌چاله (ارسال انتخابی) - گره‌های مخرب تمام بسته‌ها را رها می‌کنند
- سیل‌آسا - گره‌ها شبکه را با پیام‌های کنترلی بیش از حد (مثلاً DIS) غرق می‌کنند تا آن را تحت‌الشعاع قرار دهند
- حمله شماره نسخه - DODAG یک مهاجم شماره نسخه RPL را جعل می‌کند تا به‌روزرسانی‌های مسیریابی را مختل کند
- رتبه کاهش‌یافته - یک گره رتبه‌ای غیرقانونی پایین را برای جذب و مسیریابی نادرست ترافیک تبلیغ می‌کند.
- باتوجه‌به تعداد مثال‌های موجود در این مجموعه‌داده، هر حمله به‌خوبی نمایش‌داده‌شده است، اگرچه ترافیک عادی کلی تقریباً دوبرابر حجم حملات است (≈۶۵٪ عادی در مقابل ۳۵٪ حمله)، که می‌تواند به‌عنوان نزدیکی این مجموعه‌داده به فعالیت عادی شبکه نیز در نظر گرفته شود. هر نمونه داده (یک بازه زمانی از ترافیک شبکه) توسط ۱۶ ویژگی توصیف می‌شود، همان‌طور که در جدول ۳ نشان‌داده‌شده است.

جدول ۳. ویژگی‌های مجموعه‌داده ROUT-4-2023 [۴]

نام ویژگی	توضیحات
TIME	زمان شبیه‌سازی
SOURCE	گره مبدأ IP آدرس
DESTINATION	گره مقصد IP آدرس
LENGTH	طول بسته
INFO	اطلاعات بسته
TR	نرخ ارسال (در هر ۱۰۰۰ میلی‌ثانیه)
RR	نرخ دریافت (در هر ۱۰۰۰ میلی‌ثانیه)
TAT	میانگین زمان ارسال
RAT	میانگین زمان دریافت
RAT	میانگین زمان دریافت
TPC	تعداد بسته‌های ارسالی (در هر ثانیه)
RPC	تعداد بسته‌های دریافتی (در هر ثانیه)
TTT	مجموع زمان ارسال
TRT	مجموع زمان دریافت
DAO	تعداد بسته‌های DAO
DIS	تعداد بسته‌های DIS
CATEGORY	نوع حمله یا وضعیت عادی
LABEL	برچسب عادی/مخرب

۳-۴- پیش پردازش داده

قبل از آموزش، مجموعه‌داده‌ها چندین مرحله پیش‌پردازش را طی می‌کنند تا از سازگاری و آمادگی مدل اطمینان حاصل شود. شکل ۲ این مراحل را نشان می‌دهد که شامل موارد زیر است:



شکل ۲. مرحله پیش‌پردازش داده

- **پاک‌سازی داده‌ها**: حذف مقادیر گمشده، ورودی‌های ناسازگار و رکوردهای تکراری از مجموعه‌داده‌ها با هدف افزایش کیفیت داده.
- **نرمال‌سازی**: مقیاس‌بندی ویژگی‌های عددی به یک محدوده یکنواخت برای اطمینان از آموزش پایدار.
- **کدگذاری برچسب**: تبدیل متغیرهای دسته‌بندی‌شده و برچسب‌های کلاس به قالب عددی.
- **متعادل‌سازی داده‌ها**: برای اصلاح عدم تعادل کلاس در مجموعه‌داده‌های نفوذ، از نسخهٔ بهبودیافته‌ای از روش SMOTE[1] استفاده شده است، **الگوریتم ۱**. در این الگوریتم، نمونه‌های کلاس اقلیت (M_T)، درصد نمونه‌برداری بیش از حد ($N\%$)، تعداد نزدیکترین همسایه (k) و پارامتر دانه‌بندی (α) به‌عنوان ورودی در نظر گرفته می‌شوند.

ابتدا مجموعهٔ اقلیت (S^*)، مکانیزم رتبه‌بندی ویژگی‌ها (y)، پارامتر دانه‌بندی (c) و مقدار بیشینه‌ی (r) مقداردهی می‌شوند. سپس برای هر نمونه‌ی اقلیت در مجموعه‌ی (M_T) فواصل وزنی میان نمونه‌ها با استفاده از عملگر IMOWAD (x_i, x_j, w) محاسبه می‌گردد تا همسایگان مناسب شناسایی شوند. با درون‌یابی بین نمونه‌ی انتخاب‌شده و همسایه‌ی آن، نمونه‌های مصنوعی جدید x'_k تولید شده و به مجموعه‌ی اقلیت افزوده می‌شوند. در صورت نیاز، برخی از نمونه‌های قدیمی برای حفظ تعادل حذف می‌گردند.

[1] Synthetic Minority Over-sampling Technique

الگوریتم ۱: اصلاح عدم تعادل کلاس SMOTE

Input ← minority instances (M_T); oversampling percentage (N%);
Value for nearest neighbors (k); granularity parameter (α)
Output ← synthetic data (S_D)
S^* ← the minority class instances (M_T)
y ← mechanism to perform feature ranking
c ← granularity parameter (y, α)
P ← maximum value of cas $'r'$
for i ∈ M_T do
  for each j ∈ M_T and j ≠ i do
    Calculate IMOWAD (x_i, x_j, w); distance measure operator
  end for
  for k ← 1 to N do
    x_k← selects random samples from M_T
    x'_k← perform interpolation to generate synthetic data
    T^*← synthetic data added to S^*
    S_D← remove specific sample x_k from P
  end for
end for

این فرایند تا زمانی ادامه می‌یابد که مجموعه‌داده مصنوعی S_D به تعادل کامل برسد. نتیجه‌ی نهایی، مجموعه‌ای از نمونه‌های مصنوعی واقع‌گرایانه است که باعث بهبود عملکرد مدل در مواجهه با داده‌های نامتوازن می‌شود.

- **تقسیم داده‌ها**: پس از پیش‌پردازش، مجموعه‌داده به دو بخش آموزشی و آزمایشی تقسیم می‌شود. در این پژوهش، به‌منظور حفظ وابستگی‌های زمانی و شبیه‌سازی دقیق شرایط واقعی شبکه، داده‌ها بدون درهم‌سازی[1] و به‌صورت زمانی[2] تقسیم شدند تا از نشت زمانی[3] جلوگیری شود. این امر ارزیابی منصفانه‌ای از قابلیت تعمیم مدل را تضمین می‌کند.

پس از اتمام مراحل پیش‌پردازش، برای بهبود هزینه‌ی محاسباتی و افزایش تعمیم‌پذیری، فرآیند مهندسی ویژگی دومرحله‌ای طراحی شده است که مرحله‌ی نخست آن در **الگوریتم ۲** خلاصه می‌شود.

در این مرحله، یک مدل درخت تصمیم اولیه روی مجموعهٔ آموزشی آموزش داده می‌شود و اهمیت هر ویژگی بر اساس میزان کاهش ناخالصی در گره‌های درخت محاسبه می‌گردد. برای هر گره تقسیم t که در آن ویژگی f به‌کاررفته است، کاهش ناخالصی طبق فرمول (۱) و در خط ۳ از الگوریتم ۲ تعریف می‌شود:

$$\Delta i_t = I_t - \frac{N_{left}}{N_t}.I_{left} - \frac{N_{right}}{N_t}.I_{right} \qquad (۱)$$

الگوریتم ۲: محاسبه اهمیت ویژگی و تولید لیست τ

Input ← Training set (X_{train}, Y_{train}), Testing set ,Decision Tree parameter ($max - depth$)
1: Train initial Decision Tree $\mathbf{DT_{model}}$ on all features,
2: Compute baseline accuracy baseline_acc
3: Compute feature importance for each feature:
Compute feature importance (f) **for** each feature :
$$\Delta i_t = I_t - \frac{N_{left}}{N_t}.I_{left} - \frac{N_{right}}{N_t}.I_{right}$$
$$FI(f) = \frac{\sum_{t\in(f)splits} w_t.\Delta i_t}{\sum_j \sum_{t\in splits(j)} w_t.\Delta i_t}$$
4: τ_list (sorted descending) ← Extract unique importance values
Output ← baseline_acc, FI(f) for all features, τ_list

دراین‌رابطه I_t میزان ناخالصی (مانند Gini یاEntropy) در گره t، N_t تعداد نمونه‌های آن گره، و I_{right}, I_{left} مقادیر ناخالصی در گره‌های فرزند چپ و راست هستند.

اهمیت کلی ویژگی f از تجمیع کاهش ناخالصی‌ها در تمام گره‌هایی که ویژگی در آن‌ها ظاهر شده است به دست می‌آید و طبق فرمول (۲) و در ادامه خط ۳ الگوریتم ۲ به‌صورت زیر محاسبه می‌شود:

$$\boldsymbol{FI(f)} = \frac{\sum_{\boldsymbol{t\in splits(f)}} \boldsymbol{w_t.\Delta i_t}}{\sum_{\boldsymbol{j}} \sum_{\boldsymbol{t\in splits(j)}} \boldsymbol{w_t.\Delta i_t}} \qquad (۲)$$

در اینجا $\mathbf{W_t}$ وزن گره متناسب با تعداد نمونه‌های آن و $\mathbf{splits(f)}$ مجموعه‌ی گره‌هایی است که در آن ویژگی f ظاهر شده است. پس از محاسبه‌ی اهمیت تمام ویژگی‌ها در **خط ۴** الگوریتم۲، مجموعه‌ای از مقادیر آستانه (τ) استخراج و به‌صورت نزولی مرتب می‌شود. این خروجی شامل دقت پایه، اهمیت ویژگی‌ها و لیست آستانه‌ها است و به‌عنوان ورودی الگوریتم ۳ مورد استفاده قرار می‌گیرد.

الگوریتم ۳ در این بخش که مرحله‌ی دوم مهندسی ویژگی است، مقدار آستانهٔ بهینه τ بر اساس کنترل افت دقت تعیین می‌شود. برای هر مقدار آستانه τ مجموعهٔ ویژگی‌های منتخب طبق فرمول (۳) و در خط ۳ الگوریتم به‌صورت زیر تعریف می‌شود:

$$\boldsymbol{S(\tau)} = \{\boldsymbol{f} \mid \boldsymbol{FI(f)} > \boldsymbol{\tau}\} \qquad (۳)$$

[1] Shuffle
[2] Temporal Split
[3] Temporal Leakage

الگوریتم ۳: انتخاب τ با توجه به δ

Input ← (X_{train}, Y_{train}),(X_{test}, Y_{test}),baseline_acc, τ_list, FI(f), allowed δ

1: best_tau ← None
2: best_acc ← 0
3: **for** each τ in τ_list:
  selected_features ← {f | FI(f) > τ}
  Train DT_model2 on selected features
  acc ← Accuracy on $(X_{test_{selected}}, Y_{test})$
  If (baseline_acc - acc) ≤ δ then:
    best_tau ← τ
    best_acc ← acc
    break
4: **If** best_tau = None then:
  best_tau ← 0
  best_acc ← baseline_acc

Output ← best_tau, best_acc, selected_features

مدل درخت تصمیم جدید با استفاده از این ویژگی‌ها آموزش داده می‌شود و دقت حاصل از اعتبارسنجی - Kبخشی[1] طبق فرمول (۴) محاسبه می‌گردد:

$$Acc(\tau) = \frac{1}{K_{CV}} \sum_{i=1}^{K_{CV}} Acc_i \qquad (۴)$$

اگر اختلاف دقت نسبت به مقدار پایه کمتر از حد مجاز δ باشد، آستانهٔ τ پذیرفته می‌شود. شرط پذیرش τ در خط ۳ الگوریتم طبق فرمول (۵) به‌صورت زیر تعریف می‌شود:

$$\boldsymbol{BaselineAcc} - \boldsymbol{Acc(\tau)} \leq \boldsymbol{\delta} \qquad (۵)$$

که در آن $\boldsymbol{BaselineAcc}$ همان دقت پایه است. بالاترین مقدار τ که این شرط را برآورده سازد، به‌عنوان τ بهینه انتخاب می‌شود. در صورتی که هیچ آستانه‌ای این شرط را برآورده نکند، مقدار صفر در نظر گرفته می‌شود (خط ۴). در نهایت، خروجی این مرحله شامل τ، مجموعه ویژگی‌های منتخب و دقت حاصل روی داده‌های آزمون است.

این طراحی دومرحله‌ای ضمن حفظ توضیح‌پذیری و کنترل افت دقت، باعث انتخاب مجموعه‌ای بهینه و فشرده از ویژگی‌ها می‌شود. چرا که مرحله اول اهمیت نسبی ویژگی‌ها را آشکار می‌سازد و مرحله دوم با کنترل افت دقت، مجموعه‌ای بهینه و کوچک‌تری از ویژگی‌ها را برای مدل انتخاب می‌کند. در نتیجه، پیچیدگی مدل کاهش‌یافته و کارایی آن در محیط‌های واقعی IoT به طور قابل‌توجهی بهبود می‌یابد.

۳-۵- لایه اول سیستم تشخیص نفوذ: درخت تصمیم

در لایهٔ نخست مدل پیشنهادی، از یک طبقه‌بندی‌کنندهٔ درخت تصمیم‌گیری دودویی بهره گرفته شده است که به طور خودکار و با سرعت بالا، نمونه‌های ترافیک به دو دستهٔ عادی و حمله تفکیک می‌شوند. در این پژوهش، از درخت تصمیم‌گیری استاندارد CART با معیار ناخالصی جینی بهره گرفته شده است که بر روی برچسب‌های دودویی کل مجموعهٔ آموزشی، آموزش‌داده‌شده است. این درخت دارای عمق متوسط (عمق ۱۴) بوده و به‌منظور جلوگیری از بیش‌برازش، هرس شده است. انتخاب درخت تصمیم برای این لایه بر اساس ویژگی‌های کلیدی آن از جمله توضیح‌پذیری بالا و هزینه‌ی محاسباتی پایین آن و امکان استقرار در دستگاه‌های لبه یا دروازه‌های اینترنت اشیا صورت گرفته است.

- ورودی: بردار ویژگی ۱۶ بعدی یک جریان شبکه.
- خروجی: یک برچسب دودویی (عادی یا حمله)

مزیت رویکرد دولایه از همین مرحله قابل‌مشاهده است؛ درخت تصمیم قادر است بخش عمده‌ای از ترافیک عادی را با سرعت زیاد و کمترین هزینهٔ محاسباتی فیلتر کرده و تنها جریان‌هایی را که به‌عنوان حمله شناسایی شده‌اند، به لایهٔ دوم ارسال کند. این فرایند منجر به کاهش بار محاسباتی مدل یادگیری عمیق در لایهٔ دوم می‌شود.

در طراحی این بخش، پارامترهای درخت تصمیم به‌گونه‌ای تنظیم شده‌اند که فراخوانی برای نمونه‌های حمله در سطح بالایی حفظ شود. به‌منظور اطمینان از شناسایی تمامی فعالیت‌های مخرب، وجود تعداد محدودی مثبت کاذب مجاز در نظر گرفته شده است.

پس از تصمیم‌گیری درخت، روند پردازش به‌صورت زیر ادامه می‌یابد:

- اگر نمونه در گروه «عادی» قرار گیرد، به‌عنوان بی‌خطر در نظر گرفته شده و پردازش بیشتری بر روی آن انجام نمی‌شود.
- اگر نمونه در گروه «حمله» قرار گیرد، یک پیام هشدار تولید می‌شود و ویژگی‌های همان نمونه یا دنباله‌ای از بردارهای ویژگی اخیر برای تشخیص نوع حمله به لایهٔ دوم منتقل می‌شوند.

این سازوکار موجب افزایش کارایی کلی سیستم می‌شود؛ زیرا تمرکز

[1] K-fold cross-validation

درخت تصمیم روی نمونه‌های مشکوک باعث صرفه‌جویی در منابع محاسباتی شده و نرخ منفی‌های کاذب را کاهش می‌دهد. حتی در شرایطی که درخت بیش از حد حساس عمل کند و برخی نمونه‌های عادی را به‌عنوان حمله برچسب‌گذاری کند، این موارد در لایهٔ دوم توسط طبقه‌بندی‌کنندهٔ دقیق‌تر بازشناسایی و اصلاح می‌شوند.

۳-۶- لایه دوم سیستم تشخیص نفوذ: طبقه‌بندی‌کننده ترکیبی CNN–BiLSTM

در لایهٔ دوم، از یک طبقه‌بندی‌کنندهٔ سبک‌وزن مبتنی بر ترکیب شبکهٔ عصبی کانولوشنی (CNN) و حافظهٔ طولانی‌مدت دوطرفه (BiLSTM) برای طبقه‌بندی ناهنجاری‌ها در ترافیک شبکه استفاده شده است. این مدل به‌صورت اختصاصی برای تمایز میان چهار نوع حملهٔ RPL و ترافیک عادی طراحی شده و با هدف استخراج الگوهای محلی و مدل‌سازی وابستگی‌های زمانی در هر دو جهت (گذشته و آینده) پیکربندی شده است.

۳-۶-۱- پیکربندی و نقش هر لایه

این مدل از چند بخش اصلی تشکیل شده است که هر یک وظیفهٔ مشخصی در فرایند یادگیری و طبقه‌بندی دارند:

- نمایش ورودی

دنباله کوتاه از بردارهای ویژگی اخیر: ورودی مدل، یک دنباله از بردارهای ویژگی اخیر است که به‌صورت (طول_دنباله × تعداد_ویژگی‌ها) شکل‌دهی شده‌اند. این ورودی به‌عنوان یک سیگنال ۱ بعدی به لایه Conv1D منتقل می‌شود.

- بخش کانولوشن (استخراج ویژگی‌های موضعی)

هدف این بخش، شناسایی الگوهای محلی و کوتاه‌مدت در داده‌های ترافیکی است.

در این بخش از ساختار زیر استفاده شده است:

  - Conv1D(128, kernel_size=3, activation='relu', padding='same')

این لایه الگوهای محلی کوتاه‌مدت مانند پیک‌ها و پرش‌های ناگهانی را استخراج می‌کند.

  - BatchNormalization()

این لایه به پایدارسازی و تسریع فرآیند آموزش کمک می‌کند.

  - Conv1D(64, kernel_size=3, activation='relu', padding='same')

این لایه وظیفه استخراج ویژگی‌های سطح بالاتر را دارد.

  - MaxPooling1D(pool_size=2)

با کاهش طول دنباله، این لایه الگوهای مهم را برجسته می‌کند.

  - Dropout(0.3)

برای جلوگیری از بیش‌برازش (Overfitting)در این مرحله استفاده می‌شود.

- بخش مدل‌سازی زمینه زمانی (BiLSTM)

این بخش به مدل‌سازی وابستگی‌های زمانی در هر دو جهت (پیش‌رو و پس‌رو) اختصاص‌یافته است.

پیکربندی شامل موارد زیر است:

  - Bidirectional(LSTM(64,activation='relu',return_sequences=True))

این لایه وابستگی‌های زمانی را در هر دو جهت (گذشته به آینده و آینده به گذشته) مدل می‌کند و دنباله را برای لایه بعدی حفظ می‌کند.

  - Bidirectional(LSTM(32,return_sequences=False))

این لایه زمینه زمانی را به یک بردار خلاصه فشرده می‌کند.

  - Dropout(0.3)

برای کاهش بیش‌برازش پس از مدل‌سازی بازگشتی به کار می‌رود.

- سر خروجی (طبقه‌بندی)

  - Dense(64) → ReLU → Dropout(0.3)

این لایه‌ها ویژگی‌های استخراج‌شده را برای تصمیم‌گیری نهایی ترکیب و آماده می‌کنند.

  - Dense(num_classes) with softmax

این لایه نهایی توزیع احتمالات نهایی را برای کلاس‌های مختلف مانند (نرمال، حمله سیاه‌چاله، حمله شماره نسخه، حمله سیل‌آسا، حمله کاهش رتبه) تولید می‌کند.

- دلایل طراحی معماری ترکیبی

  - لایه‌های کانولوشنی وظیفهٔ استخراج الگوهای محلی و گذرا (مانند پیک‌های ناگهانی در شمارش بسته‌ها) را بر عهده دارند و با فشرده‌سازی طول دنباله، هزینهٔ محاسباتی کاهش می‌یابد.
  - BiLSTM با مدل‌سازی وابستگی‌های زمانی در دو جهت، امکان تمایز میان نوسانات (نویز) گذرا از الگوی حمله را فراهم می‌کند.
  - انتخاب ساختار ترکیبی: CNN–BiLSTM مبتنی بر نیاز به یک طبقه‌بند سبک‌وزن، سریع و مقاوم در برابر نویز انجام شد. این ترکیب به کاهش تعداد پارامترها و هزینهٔ

محاسباتی نسبت به معماری‌های عمیق‌تر منجر شده و در نتیجه زمان آموزش کوتاه‌تر و سرعت استنتاج بالاتر حاصل می‌شود. ازاین‌رو، استفاده از این مدل در محیط‌های اینترنت اشیا (IoT) با منابع محاسباتی محدود، رویکردی کارآمد و بهینه محسوب می‌شود.

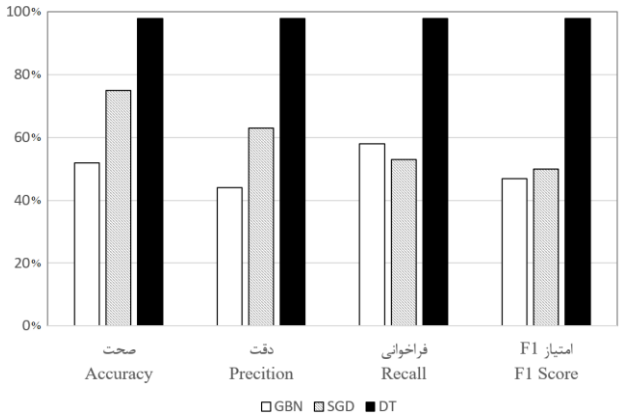
۴- نتایج تجربی

در این بخش، عملکرد مدل پیشنهادی دولایه شامل یک درخت تصمیم دودویی و مدل CNN–BiLSTM بر روی مجموعه‌داده‌ی ROUT-4-2023، که شامل ترافیک عادی و چهار نوع حمله‌ی RPL است، ارزیابی شده است. عملکرد مدل با استفاده از معیارهای استاندارد صحت، دقت تشخیص، فراخوانی و امتیاز F1 سنجیده شده است و تحلیل ماتریس درهم‌ریختگی برای بررسی دقیق عملکرد انجام شده است. علاوه بر این، زمان آموزش و سرعت استنتاج را برای نشان دادن مناسب بودن مدل برای برنامه‌های کاربردی اینترنت اشیا در زمان واقعی اندازه‌گیری شده است.

۴-۱- ارزیابی لایه اول: تشخیص نفوذ دودویی با استفاده از مدل‌های کلاسیک یادگیری ماشین

در گام نخست، هدف، تفکیک ترافیک عادی از ترافیک غیرعادی بوده است. برای این منظور، عملکرد سه الگوریتم کلاسیک یادگیری ماشین شامل بیز ساده گاوسی(GNB[1]) ، نزول گرادیان تصادفی (SGD[2]) و درخت تصمیم (DT) مقایسه شده است. همان‌طور که در شکل ۳ نشان‌داده‌شده است، مدل درخت تصمیم در تمامی معیارها عملکردی برتر نسبت به دو روش دیگر ارائه کرده است. به طور مشخص، صحت تشخیص DT بیش از ۹۸٪ بوده، درحالی‌که برای SGDو GNB به ترتیب ۷۵٪ و ۵۲٪ گزارش شده است.

جدول ۴ معیارهای عملکرد طبقه‌بندی دودویی مدل درخت تصمیم را نشان می‌دهد. همان‌طور که مشاهده می‌شود، الگوریتم درخت تصمیم توانایی قابل توجهی در شناسایی الگوهای پیچیده و غیرخطی ناهنجاری‌ها ارائه کرده است، قابلیتی که در مدل‌های خطی یا ساده‌تر مشاهده نمی‌شود. برتری عملکردی ذکر شده، انتخاب درخت تصمیم در لایه اول صرفاً مبتنی بر دقت بالاتر نیست؛ بلکه به دلیل سرعت تصمیم‌گیری، تفسیرپذیری نسبی و هزینه محاسباتی پایین نیز توجیه می‌شود. ازاین‌رو، DT به‌عنوان مکانیزم فیلترینگ سریع در لایه اول عمل می‌کند و با شناسایی سریع و قابل‌اعتماد ترافیک مشکوک، ضمن کاهش خطا، بار پردازشی تحمیل‌شده به لایه دوم را نیز محدود می‌سازد.



شکل ۳. نتایج پارامترهای ارزیابی برای سه مدل کلاس دودویی

GNB: بیز ساده گاوسی،SGD: نزول گرادیان تصادفی، DT: درخت تصمیم

جدول ۴. معیارهای عملکرد طبقه‌بندی دودویی مدل درخت تصمیم

	صحت	فراخوانی	امتیاز F1
نرمال (۰)	۱,۰۰	۰,۹۷	۰,۹۹
حمله (۱)	۰,۹	۱,۰۰	۰,۹۷
صحت			۰,۹۸
میانگین وزن‌دار	۰,۹۸	۰,۹۹	۰,۹۸
میانگین ماکرو	۰,۹۸	۰,۹۸	۰,۹۸

۴-۲- ارزیابی لایه دوم: طبقه‌بندی حمله چند کلاسه با استفاده از CNN-BiLSTM

در لایه دوم، تنها نمونه‌هایی که توسط لایه اول به‌عنوان «غیرعادی» شناسایی شده‌اند، پردازش شده‌اند و در چهار دسته حمله شامل سیاه‌چاله، سیل‌آسا، شماره نسخه و کاهش رتبه طبقه‌بندی شده‌اند.

در این معماری، CNN وظیفه استخراج الگوهای محلی را بر عهده دارد و BiLSTM وابستگی‌های ترتیبی و زمانی را مدل‌سازی می‌کند؛ ازاین‌رو لایه دوم نسبت به ناهنجاری‌هایی که در سطح توالی/زمان بروز می‌کنند حساس‌تر بوده و امکان تفکیک دقیق‌تر حملات هم‌خانواده را فراهم می‌سازد.

همچنین هشدارهای کاذب تولید شده در لایه اول بازشناسی و به رده ترافیک عادی تخصیص داده شده‌اند.

[1] Gaussian Naive Bayes

[2] Stochastic Gradient Descent

نتایج حاصل از ارزیابی عملکرد طبقه‌بندی چندکلاسه مدل DT+CNN–BiLSTM در پنج کلاس روی مجموعه داده در جدول ۵ ارائه شده است. همان‌طور که مشاهده می‌شود، مدل پیشنهادی توانسته است حملات سیل‌آسا و شماره نسخه را با صحت و یادآوری کامل شناسایی کند. همچنین، حملات کاهش رتبه و سیاه‌چاله با امتیاز F1 به ترتیب برابر ۰٫۹۴ و ۰٫۸۷ طبقه‌بندی شده‌اند، به‌طوری‌که تنها سردرگمی جزئی ناشی از شباهت‌های توپولوژیکی مشاهده می‌شود. علاوه بر این، برای ترافیک عادی، F1 برابر ۰٫۹۸ حفظ شده است، که نشان‌دهنده توانایی بازیابی مؤثر نمونه‌های خوش‌خیم طبقه‌بندی‌نشده از لایه اول است.

جدول ۵. عملکرد طبقه‌بندی چند کلاسه مدل DT+CNN-BiLSTM در پنج کلاس روی مجموعه‌داده

	صحت	فراخوانی	امتیاز F1
سیاه‌چاله	۰٫۸۶	۰٫۸۸	۰٫۸۷
شماره نسخه	۱٫۰۰	۱٫۰۰	۱٫۰۰
سیل‌آسا	۱٫۰۰	۱٫۰۰	۱٫۰۰
نرمال	۰٫۹۸	۰٫۹۸	۰٫۹۸
کاهش رتبه	۰٫۹۶	۰٫۹۳	۰٫۹۴

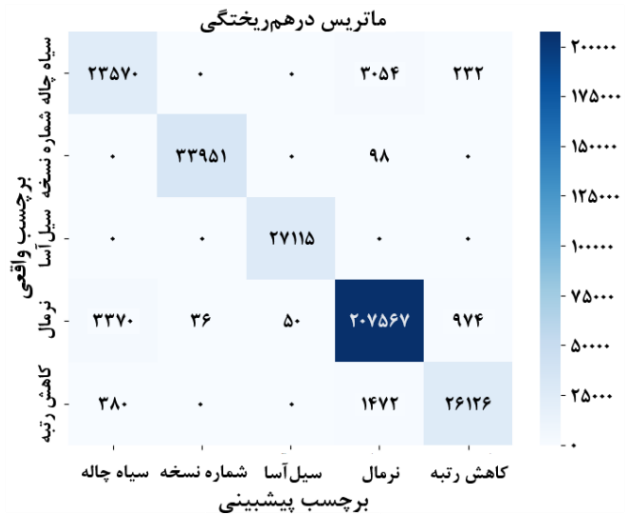
این نتایج بیانگر آن هستند که معماری CNN–BiLSTM در لایه دوم توانایی تحلیل عمیق و دقیق الگوهای حمله را ارائه کرده است و عملکرد مبتنی بر قواعد درخت تصمیم در لایه اول را تکمیل می‌کند. ترکیب این دو لایه یک خط لوله قوی برای تشخیص سریع ناهنجاری‌ها و طبقه‌بندی دقیق انواع حمله ایجاد می‌کند.

۴-۳- تحلیل ماتریس درهم‌ریختگی

به‌منظور درک دقیق‌تر از نحوه عملکرد مدل در فرایند طبقه‌بندی، ماتریس درهم‌ریختگی پیش‌بینی‌های آن بر روی داده‌های آزمایشی مورد بررسی قرار گرفته است (شکل ۴). این ماتریس بر اساس مقایسه برچسب‌های واقعی و مقادیر پیش‌بینی‌شده ساخته شده و عملکرد مدل را در تفکیک پنج کلاس زیر نشان می‌دهد:

- **کلاس صفر :**حمله سیاه‌چاله
- **کلاس ۱ :**حمله شماره نسخه
- **کلاس ۲ :**حمله سیل‌آسا
- **کلاس ۳ :**ترافیک عادی
- **کلاس ۴ :**حمله کاهش رتبه

همان‌طور که در شکل ۴ مشاهده می‌شود، بالاترین مقادیر در قطر اصلی ماتریس قرار دارند که نشان‌دهنده دقت بالای مدل در طبقه‌بندی صحیح نمونه‌ها است. به‌طور خاص:



شکل ۴. ماتریس درهم‌ریختگی مدل پیشنهادی روی داده‌های تست

- از مجموع نمونه‌های واقعی کلاس صفر سیاه‌چاله، تعداد ۲۳۵۷۰ مورد به‌درستی شناسایی شدند. تنها ۳۳۷۰ نمونه به‌اشتباه به‌عنوان نرمال و ۳۸۰ نمونه به‌عنوان کاهش رتبه طبقه‌بندی شدند.
- **کلاس ۱ شماره نسخه،** با دقت بسیار بالا شناسایی شده و تقریباً تمام نمونه‌ها ۳۳۹۵۱ به‌درستی طبقه‌بندی شدند. تنها ۳۶ نمونه به اشتباه در کلاس نرمال قرار گرفتند.
- **کلاس ۲ سیل‌آسا،** نیز عملکرد عالی داشته و تمامی۲۷۱۱۵ نمونه به‌طور صحیح شناسایی شده‌اند، بدون هیچ خطای قابل توجه.
- در **کلاس ۳ نرمال،** تعداد ۲۰۷۵۶۷ نمونه درست طبقه‌بندی شدند. با این حال ۳۰۵۴ نمونه به‌اشتباه به‌عنوان سیاه‌چاله، ۹۸ نمونه به‌عنوان شماره نسخه ،صفر نمونه به‌عنوان سیل‌آسا و ۱۴۷۲ نمونه به‌عنوان کاهش رتبه شناسایی شدند.
- **کلاس ۴ کاهش رتبه** ، نیز با موفقیت بالا تشخیص داده شد؛ ۲۶۱۲۶ نمونه به‌درستی شناسایی شدند، اما ۹۷۴ نمونه به‌اشتباه به‌عنوان نرمال و ۲۳۲ نمونه به‌عنوان سیاه‌چاله طبقه‌بندی شدند.

تحلیل ماتریس درهم‌ریختگی نشان می‌دهد که چالش‌برانگیزترین کلاس، حمله «سیاه‌چاله» است که در برخی موارد با ترافیک عادی اشتباه گرفته شده است. این امر ریشه در شباهت رفتاری میان این

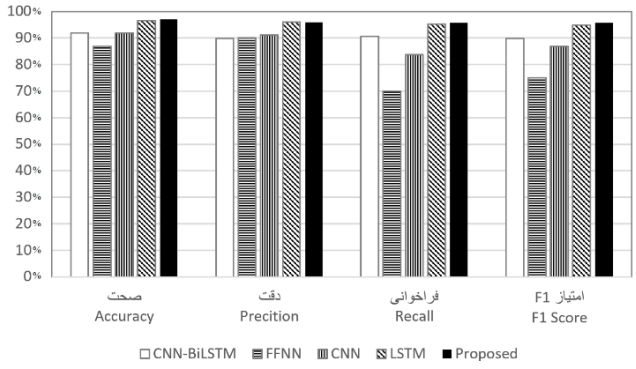
حمله و پدیده افت بسته[1] ناشی از نویز و نوسانات محیطی در شبکه‌های کم‌مصرف و پر اتلاف (LLNs) دارد. بااین‌حال، استفاده از لایه BiLSTM در مرحله دوم توانسته است با تحلیل جهت‌دارِ دنباله ترافیک و استخراج ویژگی‌های زمانیِ پیچیده، این خطا را نسبت به مدل‌های تک‌لایه‌ای سنتی به شکل معناداری کاهش دهد. در نهایت، طراحی دومرحله‌ای مدل با به‌حداقل‌رساندن خطاهای بحرانی به‌ویژه اشتباه‌گرفتن ترافیک عادی با حملات، کارایی خود را برای پیاده‌سازی در محیط‌های واقعی اینترنت اشیا به اثبات رسانده است. موضوعی که برای کاربرد در محیط‌های واقعی اینترنت اشیا اهمیت زیادی دارد.

۴-۴- مقایسه با سایر مدل‌ها

به‌منظور ارزیابی اثربخشی رویکرد پیشنهادی، یک تحلیل مقایسه‌ای جامع در برابر مجموعه‌ای از روش‌های پیشرفته مطرح در مقالات انجام شد. شکل ۵ عملکرد پنج مدلFFNN ،CNN ،LSTM، CNN-BiLSTM و مدل پیشنهادی DT + CNN-BiLSTM را در چهار معیار کلیدی ارزیابی شامل صحت[2]، دقت پیش‌بینی[3]، فراخوانی[4] و امتیاز F1[5] نشان می‌دهد. همان‌گونه که در شکل مشاهده می‌شود، مدل پیشنهادی ما در تمامی معیارهای ارزیابی، عملکردی برتر نسبت به سایر مدل‌های پایه ارائه می‌دهد.

درحالی‌که مدل‌های LSTM و CNN-BiLSTM نیز عملکرد قابل‌توجهی در طبقه‌بندی نشان می‌دهند، استفاده از معماری ترکیبی درخت تصمیم و CNN-BiLSTM منجر به دستیابی به تعادلی بهینه میان صحت و فراخوانی شده و در نتیجه بالاترین مقدار امتیاز F1 را نسبت به سایر رویکردها کسب کرده است.

جدول ۶ مقایسه جامعتری میان مدل پیشنهادی و چندین مدل مختلف مبتنی بر یادگیری عمیق را که بر روی مجموعه‌داده‌های مرتبط با شبکه و اینترنت اشیا ارزیابی شده‌اند، ارائه می‌دهد. این مقایسه شامل مدل‌هایی است که به‌طور خاص بر شناسایی حملات مبتنی بر RPL تمرکز داشته‌اند. نتایج نشان می‌دهد که مدل پیشنهادی در مقایسه با سایر روش‌ها از نظر صحت تشخیص، نرخ هشدار کاذب و توانایی شناسایی حملات پیچیده عملکرد بهتری ارائه می‌دهد و می‌تواند به‌عنوان یک چارچوب کارآمد و بهینه در حوزه تشخیص حملات در شبکه‌های اینترنت اشیا مورد استفاده قرارگیرد.



شکل ۵. مقایسه عملکرد مدل‌های پایه و پیشنهادی بر اساس چهار معیار کلیدی

این نتایج تأیید می‌کند که روش پیشنهادی به بهترین تعادل کلی میان صحت، یادآوری و کارایی محاسباتی دست می‌یابد، به‌ویژه در سناریوهای اینترنت اشیا مبتنی بر RPL طراحی سلسله‌مراتبی ارائه‌شده، نقش شبکه CNN-BiLSTM را ساده‌تر کرده و موجب می‌شود که این مدل با تمرکز بر نمونه‌های پیچیده‌تر، صحت کلی و یادآوری مختص هر کلاس را به طور محسوسی بهبود بخشد، بدون آن که هزینه محاسباتی اضافی بر سیستم تحمیل شود.

این استراتژی نه‌تنها یک رویکرد کارآمد و مقیاس‌پذیر برای امنیت اینترنت اشیا ارائه می‌دهد، بلکه یک الگوی کلی برای طراحی سیستم‌های تشخیص نفوذ محسوب می‌شود: انتقال تصمیمات دودویی سریع به طبقه‌بندی‌کننده‌های سبک و اختصاص مدل‌های پیچیده به تحلیل عمیق‌تر. در نتیجه، روش پیشنهادی نه‌تنها با روش‌های پیشرفته موجود قابل‌مقایسه است، بلکه در بسیاری از معیارها، به‌ویژه در شاخص‌های امتیاز F1 و استحکام عملکرد از آن‌ها پیشی می‌گیرد. این یافته‌ها پتانسیل بالای این رویکرد را به‌عنوان یک راه‌حل عملی، سبک و قابل‌استقرار برای شناسایی حملات در محیط‌های واقعی اینترنت اشیا برجسته می‌کند.

[1] Packet Drop
[2] Accuracy
[3] Precision
[4] Recall
[5] F1-Score

جدول ۶. مقایسه معیارهای عملکرد برای مدل‌های اخیر روی مجموعه‌داده‌های تشخیص نفوذ مبتنی بر اینترنت اشیا وRPL

مدل	دیتاست	حوزه	صحت	امتیاز F1	دقت	فراخوانی	دولایه
TLBO-ElM [11]	NSL-KDD	Network	88.21	-	-	-	✗
SS-DeepID [7]	CIC-IDS	Network	88.33	88.85	88.78	88.91	✗
RNN-BPTT [6]	BoT-IoT	IoT	87.20	-	-	-	✗
CNN-BiLSTM [8]	IoTID20	IoT	91.87	89.98	89.91	90.70	✗
CNN-BiLSTM [8]	N-BaIoT	IoT	90.12	89.97	89.87	90.21	✗
DT-NB-ANN-SGD [9]	ROUT-4-2023	RPL-based IoT	86.21	78.21	–	69.54	✗
FFNN	ROUT-4-2023	RPL-based IoT	87.00	75.00	90.00	70.00	✗
CNN	ROUT-4-2023	RPL-based IoT	92.00	87.00	91.20	83.80	✗
LSTM	ROUT-4-2023	RPL-based IoT	97.00	95.35	95.50	95.20	✗
Proposed	ROUT-4-2023	RPL-based IoT	97.00	95.80	96.00	95.90	✓

۴-۵- زمان آموزش و کارایی

یکی از انگیزه‌های کلیدی پشت معماری دولایه پیشنهادی، بهبود کارایی محاسباتی در کنار حفظ دقت تشخیص است. همان‌گونه که در جدول ۷ نشان داده شده است، مدل پیشنهادی برای آموزش درخت تصمیم روی کل مجموعه‌داده به حدود ۸٫۲ ثانیه و برای آموزش مدل CNN–BiLST روی زیرمجموعه برچسب‌گذاری‌شده‌ی حمله (حدود ۲۵٪ از داده‌ها) به حدود ۱۴۵ دقیقه نیاز دارد که در مجموع تقریباً ۱۴۶ دقیقه می‌شود. نتایج جدول ۷ نشان می‌دهد که اگرچه فاز دوم مدل از نظر آموزش پرهزینه‌تر از مدل‌های سبک‌وزن است، این هزینه تنها بر زیرمجموعه‌ای از داده‌ها اعمال می‌شود و همین موضوع کارایی کل سامانه را حفظ می‌کند. در واقع، ساختار دولایه باعث می‌شود تصمیم‌گیری پرهزینه فقط در موارد لازم فعال شود و بدین‌ترتیب میان دقت بالا و مقیاس‌پذیری عملی توازن برقرار گردد. در زمان آزمایش نیز درخت تصمیم بخش عمده‌ای از ترافیک را در زمانی بسیار کوتاه غربال می‌کند و تنها نمونه‌های مشکوک برای تحلیل عمیق‌تر به لایه دوم منتقل می‌شوند؛ سازوکاری که توان عملیاتی و مقیاس‌پذیری سیستم را حتی در شرایط حمله بالا حفظ می‌کند. افزون بر این، سبک‌وزن بودن نسبی مدل، شامل چند کیلوبایت برای درخت تصمیم و چند مگابایت برای–CNN BiLSTM، استقرار آن را در محیط‌های اینترنت اشیا با محدودیت منابع عملی‌تر می‌سازد. بنابراین، مزیت روش پیشنهادی تنها به بهبود معیارهای عملکرد محدود نمی‌شود، بلکه در سطح معماری نیز راهکاری متوازن برای دستیابی هم‌زمان به دقت، سرعت و قابلیت استقرار ارائه می‌دهد.

جدول ۷. مقایسه زمان آموزش و آزمایش مدل‌های پیشین و پیشنهادی

مدل	دوره‌ها	میانگین زمان در هر دوره (ثانیه)	کل زمان آموزش (دقیقه)	زمان آزمایش (ثانیه)
CNN-BiLSTM [8]	۶۰۰	–	–	–
CNN	۵۰	۸۶٫۷۴	۶۳٫۶۸	۶۲٫۳
LSTM	۲۵	۲۲۳٫۸	۹۳٫۲۵	۱۲۹٫۱۲
روش پیشنهادی (تشخیص وقوع حمله)	–	–	۰٫۰۸	۰٫۹۱۶
روش پیشنهادی (تشخیص نوع حمله)	۲۰	۴۲۲٫۸۲	۱۴۵٫۲	۱۰۷

۵- نتیجه‌گیری

در این مقاله، یک چارچوب تشخیص نفوذ دولایه‌ای سبک‌وزن برای شبکه‌های اینترنت اشیاء مبتنی بر RPL ارائه شد. معماری پیشنهادی با تفکیک فرایند تشخیص به دو مرحلهٔ غربالگری سریع (توسط درخت تصمیم) و طبقه‌بندی عمیق توسط CNN-BiLSTM، توازن بهینه‌ای میان دقت و پیچیدگی محاسباتی برقرار کرده است. ارزیابی‌های انجام‌شده روی مجموعه‌داده ROUT-4-2023 نشان داد که صحت کلی مدل به ۹۷٪ و شاخص F1 به ۹۵٫۸٪ رسیده است، درحالی‌که زمان شناسایی ۰٫۰۶۲ ثانیه‌ای، کارایی آن را برای محیط‌های با منابع محدود و کاربردهای بی‌درنگ[1] اثبات می‌کند. با

[1] Real-time

این حال، این پژوهش از منظر اعتبار بیرونی[1] با محدودیت‌هایی مواجه است؛ از جمله تمرکز بر داده‌های آفلاین که ممکن است تمامی چالش‌های پویای ترافیک واقعی شبکه را پوشش ندهد. همچنین، پیاده‌سازی عملیاتی در گره‌های لبه با محدودیت‌های شدید حافظه، همچنان یک چالش استقراری[2] تلقی می‌شود. بر این اساس، در کارهای آتی، پیاده‌سازی مدل بر روی سخت‌افزارهای واقعی لبه[3] و ارزیابی تاب‌آوری سامانه در مواجهه با حملات نوظهور و ترکیبی[4] پیشنهاد می‌شود. علاوه بر این، بررسی معماری‌های یادگیری عمیقِ جایگزین با ساختاری فشرده‌تر[5] جهت بهبود هرچه بیشتر توازن میان سربار محاسباتی و دقت مدل، از اهداف توسعه‌ای این تحقیق خواهد بود.

سپاسگزاری

نویسندگان این مقاله بر خود لازم می‌دانند از حمایت‌های مادی و معنوی **پژوهشگاه ارتباطات و فناوری اطلاعات** در اجرای این پژوهش صمیمانه قدردانی نمایند. همچنین از راهنمایی‌های ارزشمند جناب آقای **دکتر امیر منصور یادگاری** به عنوان راهبر پژوهش این طرح، کمال تشکر و سپاس را دارند.

مراجع

[1] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, “Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications,” IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2347–2376, 2015.

[2] Alfriehat N., Anbar M., Aladaileh M., Hasbullah I., et al., “RPL-based attack detection approaches in IoT networks: review and taxonomy,” Artificial Intelligence Review, vol. 57, art. 248, Aug. 2024

[3] T. Winter et al., RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks, RFC 6550, IETF, 2012

[4] M. Emeç and M. H. Özcanhan, “ROUT-4-2023: RPL based routing attack dataset for IoT,” IEEE Dataport, doi: 10.21227/3mbe-5j70, 2023.

[5] K. O. A. Alimi, K. Ouahada, A. M. M. Abu-Mahfouz, S. Rimer, and O. A. Alimi, “Refined LSTM Based Intrusion Detection for Denial-of-Service Attack in Internet of Things,” Journal of Sensor and Actuator Networks (JSAN), vol. 11, no. 3, article 32, July 2022. DOI: 10.3390/jsan11030032

[6] Abdelouahid Derhab, Arwa Aldweesh, Ahmed Z Emam, and Farrukh Aslam Khan. 2020. Intrusion detection system for internet of things based on temporal convolution neural network and efficient feature engineering. Wireless Communications and Mobile Computing 2020 (2020), 1–16.

[7] Mohamed Abdel-Basset, Hossam Hawash, Ripon K Chakrabortty, and Michael J Ryan. 2021. Semi-supervised spatiotemporal deep learning for intrusions detection in IoT networks. IEEE Internet of Things Journal 8, 15 (2021), 12251–12265.

[8] Y. Guan, M. Noferesti, and N. Ezzati-Jivan, “CNN-BiLSTM-Based Classification of RPL Attacks in IoT Smart Grid Networks (Industry Track),” in Proc. ACM/IFIP/USENIX Middleware Conf., Bologna, Italy, Dec. 2023

[9] A. Etheridge and V. Anu, RPL Attack Detection in IoT Environments: An Ensemble Approach, in Proc. IEMTRONICS 2024 – International IoT, Electronics and Mechatronics Conference, Lecture Notes in Electrical Engineering, vol. 1228, Springer, 2025, pp. 113–122

[10] M. Osman, J. He, N. Zhu, and F. M. M. Mokbal, “An ensemble learning framework for the detection of RPL attacks in IoT networks based on the genetic feature selection approach,” Ad Hoc Netw., vol. 152, 103331, Oct. 2023

[11] Mustafa Qahatan Alsudani, Salah H. Abbdal Reflish, Kohbalan Moorthy, and Myasar Mundher Adnan. 2023. A new hybrid teaching learning based Optimization -Extreme learning Machine model based Intrusion-Detection system. Materials Today: Proceedings 80 (2023),

[12] A. Chougule, R. Mane, K. Bhattacharjee, and S. Mehta, “Ensemble Learning Based Intrusion Detection System for RPL-Based IoT Networks,” in Demystifying AI and ML for Cyber-Threat Intelligence, A. Chougule and R. Mane, Eds. Cham: Springer, 2025, pp. 27–37, doi: 10.1007/978-3-031-90723-4_3.

[1] External Validity

[2] Deployment Challenge

[3] Edge

[4] Hybrid Attacks

[5] Compressed