یک سامانهٔ تشخیص نفوذ برای شهرهای هوشمندبا استفاده از شبکهٔ عصبی و الگوریتم اره‌ماهی

زهرا سرحدی[1]*، مهدی خزاعی‌پور[2]

دانشجوی دکترا، گروه مهندسی کامپیوتر، دانشکدهٔ فنی‌ومهندسی، واحد بیرجند، دانشگاه آزاد اسلامی، بیرجند، ایران[1*]

استادیار، گروه مهندسی کامپیوتر، دانشکدهٔ فنی‌ومهندسی، واحد بیرجند، دانشگاه آزاد اسلامی، بیرجند، ایران[2]

چکیده

اینترنت اشیا (IoT) شبکه‌ای گسترده از اشیای هوشمند متصل به اینترنت است که در شهرهای هوشمند برای یک‌پارچه‌سازی سامانه‌هایی مانند حمل‌ونقل، برق و بهداشت کاربرد دارد. یکی از چالش‌های مهم در شبکه‌های IoT حملات سایبری است که موجب اختلال در سرویس‌ها می‌شود. برای مقابله با این تهدیدات استفاده از سامانه‌های تشخیص نفوذ مبتنی بر یادگیری ماشین ضروری است. در این مقاله روشی ترکیبی برای تشخیص حملات به شهرهای هوشمند ارائه شده‌است که شامل سه مرحله است: ۱) متعادل‌سازی داده‌ها با تئوری بازی و شبکهٔ GAN، ۲) انتخاب ویژگی با الگوریتم بهینه‌سازی اره‌ماهی، ۳) تنظیم پارامترهای ماشین بردار پشتیبان (SVM) با الگوریتم‌های محاسبات ریاضی. شبکهٔ عصبی چندلایه برای تحلیل ویژگی‌ها و SVM برای طبقه‌بندی ترافیک استفاده شده‌اند. نتایج آزمایش‌ها روی مجموعه‌دادهٔ NSL-KDD در نرم‌افزار MATLAB، دقت ۹۹.۱۲ درصد، حساسیت ۹۸.۹۲ درصد و صحت ۹۸.۹۶ درصد را نشان می‌دهد. روش پیشنهادی نسبت به الگوریتم‌های گرگ خاکستری و ژنتیک عملکرد دقیق‌تری دارد.

واژگان کلیدی: اینترنت اشیا، شهرهای هوشمند، سامانهٔ تشخیص نفوذ، یادگیری ماشین، الگوریتم اره‌ماهی ، انتخاب ویژگی.

A detection system for smart cities Using a neural network and Sailfish Optimizer algorithm

Zahra Sarhadhi[1*], Mehdi khazaiepoor[2]

PhD student, Computer Engineering Department, Technical and Engineering Faculty, Birjand Branch, Islamic Azad University, Birjand, Iran[1*]

Assistant Professor, Department of Computer Engineering, Technical and Engineering Faculty, Birjand Branch, Islamic Azad University, Birjand, Iran[2]

Abstract

"The Internet of Things" is an extensive network of intelligent objects that has a large number of objects connected to the Internet. One of the applications of the IOT network is in smart cities. In smart cities, all parts of the city, such as the transportation system, electricity network, health network, etc., are interconnected with IOT support. One of the critical challenges of the IOT network is the occurrence of attacks against this network, which causes the network services to be disrupted. Intrusion detection systems are used to detect attacks on the IOT. The role of an IoT network intrusion detection system is to analyze the network traffic, detect abnormal traffic, and send necessary warning to the firewall. One of the methods of detecting attacks on the IOT and smart cities is to use machine learning methods such as support vector machines(SVM). One method to reduce the error of the support vector machine in detecting attacks on the IOT network and the smart city is to use feature selection methods and optimize its parameters. By selecting the feature and optimizing the parameters of the support vector machine, the attack detection error will be reduced. In this article, an intrusion detection method with an artificial

* نویسندهٔ عهده‌دار مکاتبات

* Corresponding author

• تاریخ ارسال مقاله: ۱۴۰۲/۲/۷ • تاریخ پذیرش: ۱۴۰۴/۴/۲۹ • تاریخ انتشار: ۱۴۰۴/۶/۲۲ • نوع مطالعه: پژوهشی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

neural network and a swordfish optimization algorithm is presented to detect attacks on the smart city. The proposed method includes three different phases: data set balancing with game theory and GAN network, feature selection with Sailfish Optimizer algorithm, and optimization of SVM parameters with Archimedes optimization algorithm (AOA) algorithm. The role of a multilayer neural network in the proposed method of evaluating feature vectors and the role of the support vector machine is to classify network traffic into two categories: attack and normal. The evaluation and tests performed in MATLAB software and on the NSL-KDD data set show that the accuracy, sensitivity, and precision of the proposed method are 99.12%, 98.92%, and 98.96%, respectively, and the support vector machine with Gaussian kernel seems to be more accurate. The results of the experiments showed that the proposed method is more accurate than meta-heuristic algorithms, such as gray wolf optimization and genetic algorithms in detecting attacks on the smart city.

Keywords: Internet of Things, Smart Cities, Intrusion Detection System, Machine Learning, Sailfish Optimizer Algorithm, Feature Selection.

۱-مقدمه

امنیت فناوری‌های جدید از جمله شبکهٔ اینترنت اشیا[1] [۱] و شهرهای هوشمند[2] [۲] یک موضوع مداوم در حال توسعه و در حال ظهور است. پیشرفت بیشتر فناوری‌ها منجر به آسیب‌پذیری بیشتر و افزایش تهدید حملات به شبکه می‌شود و باعث می‌شود نوع حملات نیز پیشرفته شود. روش‌های امنیتی سنتی برای برقراری امنیت شبکه دیگر معتبر نیستند؛ زیرا حملات هوشمندانه‌تری در حال انجام است. این امر باعث شده‌است تا پژوهش‌گران جنبهٔ دیگری برای محافظت از سامانه‌ها در برابر حملات استفاده کنند؛ زیرا در این حالت داده‌ها کمابیش از هر دستگاه تولید می‌شود و هر دستگاه متصل به شبکهٔ اینترنت اشیا می‌تواند نقش یک حمله‌کننده به شبکه را داشته باشد. مطالعات نشان می‌دهد که فناوری‌های اخیر داده‌های عظیمی از طیف گسترده‌ای از منابع را تولید می‌کنند؛ برای مثال تلفن‌های هوشمند که منبع داده‌های چندجریانی را از مجموعه‌های حس‌گر خود مانند شتاب‌سنج، ژیروسکوپ و سامانهٔ موقعیت‌یابی جهانی ارائه می‌کنند[۳].

در سال‌های اخیر، با توسعهٔ مداوم فناوری ارتباطات شبکه، اینترنت اشیا، محاسبات ابری و سایر فناوری‌ها این فناوری‌های مبتنی بر شبکه به شدت در جامعهٔ مدرن ریشه پیدا کرده‌اند. هم‌زمان با رشد و توسعهٔ شبکهٔ اینترنت اشیا، وضعیت امنیتی در فضای سایبری به طور فزاینده‌ای پیچیده می‌شود. امنیت سایبری شرکت‌ها، دولت‌ها و افراد دائم در معرض تهدیدات مختلف امنیت سایبری قرار دارند. بر اساس «گزارش ربات بد ۲۰۲۱» منتشرشده به‌وسیلهٔ Imperva در سال ۲۰۲۱، تنها ۵۹.۲ درصد از کل ترافیک شبکه را ترافیک انسانی تشکیل می‌دهد. در میان ترافیک تولیدشده به وسیلهٔ ماشین‌ها، ترافیک مخرب ۲۵.۶ درصد از کل ترافیک شبکه را تشکیل می‌دهد[۴]. در بسیاری از حوادث امنیتی شبکه، نفوذهای مخرب سهم زیادی را به خود اختصاص می‌دهند که تهدیدات بزرگی هم به کاربران شبکه و هم به شرکت‌ها تحمیل می‌کند؛ سامانه‌های تشخیص نفوذ[3] در حال حاضر بخش مهمی از حفاظت از امنیت میزبان‌ها و سامانه‌ها هستند. وظیفهٔ مهم یک سامانهٔ تشخیص نفوذ در شبکه تجزیه‌وتحلیل ترافیک شبکه و شناسایی حملات است و این حملات را می‌تواند به یک دیوار آتش گزارش دهد[۵].

اینترنت اشیا شبکه‌ای بزرگ از اشیای متنوع و ناهمگن است و در این شبکه حملات می‌تواند بسیار پیچیده باشد؛ لذا برای طراحی یک سامانهٔ تشخیص نفوذ چالش‌های مختلفی وجود دارد؛ برای نمونه در این شبکه گره‌های متصل به شبکه می‌توانند نقش حمله‌کننده داشته باشند، بدون آن‌که برای این هدف طراحی شده باشند. در اینترنت اشیا ارتشِ بات‌نت‌ها[4] [۶] نقش مخربی علیه سرویس‌های ارائه‌دهندهٔ خدمات دارد. در اینترنت اشیا و شهرهای هوشمند انواع ویروس و بدافزار[5] [۷] می‌تواند اشیای هوشمند را آلوده کند و این اشیای هوشمند نقش حمله‌کننده را برعهده خواهند داشت. یکی از انواع حملاتی که می‌تواند در اینترنت اشیا، آسیب زیادی به زیرساخت‌ها وارد کند، حملات رد سرویسِ خدمات توزیع‌شده[6] است. در این نوع از حملات تعدادی زیادی بات‌نت مشارکت دارند و هر بات‌نت یک ترافیک کاذب را برای یک سرویس‌دهنده ارسال می‌کند. در این نوع از حملات هدف آن است که سرویس‌دهنده دچار اختلال شود و نتواند به کاربران ارائهٔ خدمات کند. دوربین‌های هوشمند در شهرهای هوشمند به‌دلیل پیکربندی ضعیف امنیتی می‌توانند گزینه‌ای مناسبی برای هکرها باشند تا از این طریق حملات خود را عملی کنند[۸].

در شبکهٔ اینترنت اشیا و شهرهای هوشمند اشیای ناهمگن زیادی وجود دارد و این دستگاه‌ها توسط سازندگان مختلفی تولید می‌شوند که برای کاهش هزینه‌ها تلاش دارند تا نکات امنیتی را رعایت نکنند. با وجود این چالش‌ها برای ارتقای امنیت شبکهٔ اینترنت اشیا و

[1] Internet of Things
[2] Smart cities
[3] Intrusion detection systems (IDS)
[4] Botnet
[5] Malware
[6] Distributed denial-of-service attack (DDoS attack)

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

شهرهای هوشمند نیاز به سامانه‌های امنیتی هوشمند و دقیقی است که بتواند جلوی حملات را بگیرید و دارای هوشمندی بالایی باشد. یک روش کارآمد برای تشخیص حملات به شبکهٔ اینترنت اشیا و شهرهای هوشمند، استفاده از روش‌های یادگیری ماشین و یادگیری عمیق است. از جمله روش‌های یادگیری که برای تشخیص حملات به‌کار گرفته شده‌است می‌توان به شبکهٔ عصبی مصنوعی[1] [۹]، ماشین بردار پشتیبان[2] [۱۰] و جنگل تصادفی[3] [۱۱] اشاره کرد؛ هر کدام از این روش‌ها تلاش دارند تا الگوی حملات به شبکه را تشخیص دهند و حملات مخرب را به دیوار آتش گزارش دهند. یکی از چالش‌های مهم روش‌های یادگیری ماشین آن است که روی همهٔ ویژگی‌ها آموزش داده می‌شوند و این موضوع باعث می‌شود که یادگیری روی ویژگی‌های مهم متمرکز نشود. یک روش برای آن‌که خطای خروجی روش‌های یادگیری ماشین در تشخیص حملات کاهش داده شود، استفاده از انتخاب ویژگی[4] است[۱۲]. انتخاب ویژگی باعث می‌شود که ورودی‌های یادگیری ماشین نیز کاهش داده شود و این موضوع باعث می‌شود که زمان یادگیری نیز کاهش یابد. برای تشخیص نفوذ به شبکه در اینترنت اشیا تاکنون از چند روش انتخاب ویژگی استفاده شده‌است که نمونهٔ آن الگوریتم ژنتیک[5] [۱۳] و الگوریتم بهینه‌سازی ذرات[6] [۱۴] است. در این پژوهش برای تشخیص حملات به شبکهٔ اینترنت اشیا از یک روش دو مرحله‌ای استفاده می‌شود. در روش پیشنهادی در ابتدا مرحلهٔ انتخاب ویژگی با الگوریتم اره‌ماهی [7] [۱۵] انجام می‌شود؛ سپس از طبقه‌بندی‌کنندهٔ شبکهٔ عصبی برای تحلیل ترافیک شبکه استفاده می‌شود. سامانهٔ پیشنهادی مورد نظر از این جهت مزیت دارد که مفاهیم هوش گروهی را با شبکهٔ عصبی ترکیب می‌کند تا تحلیل ترافیک با دقت بیشتری انجام شود. سهم ما نویسندگان در این مقاله در موارد زیر خلاصه شده‌است:

- ارائهٔ یک نسخهٔ دودویی از الگوریتم اره‌ماهی برای انتخاب ویژگی
- متعادل‌سازی مجموعه‌داده با تئوری بازی و یادگیری عمیق مبتنی بر روش GAN
- بهینه‌سازی فراپارامترهای SVM با الگوریتم بهینه‌سازی محاسبات ریاضی
- تلفیق شبکهٔ عصبی مصنوعی و الگوریتم اره‌ماهی در انتخاب ویژگی

هدف اصلی ما از این پژوهش ارائهٔ یک سامانهٔ تشخیص نفوذ هوشمند برای تشخیص حملات در اینترنت اشیا است. این مقاله دارای چند بخش مختلف است در بخش دوم پیشینهٔ پژوهش ارائه می‌شود و در بخش سوم نیز روش پیشنهادی برای تشخیص حملات به اینترنت اشیا با الگوریتم اره‌ماهی و شبکهٔ عصبی مصنوعی ارائه می‌شود. در بخش چهارم روش پیشنهادی تجزیه‌وتحلیل و نتایج آن بیان می‌شود. در بخش پنجم نیز نتیجه‌گیری مقاله و پیشنهادهای آینده ارائه می‌شود.

۲- پیشینهٔ پژوهش

در این بخش ادبیات موضوعی و مروری بر پیشینهٔ پژوهش و کارهای مرتبط در حوزهٔ اینترنت اشیا، شهرهای هوشمند و سامانه‌های تشخیص بررسی خواهد شد.

۲-۱-اینترنت اشیا

اینترنت اشیا به‌طور گسترده به‌عنوان یکی از رایج‌ترین انقلاب‌های فناوری در دو دههٔ گذشته در نظر گرفته می‌شود. دستگاه‌های اینترنت اشیا بیشتر به‌عنوان دستگاه‌های محاسباتی با قابلیت‌های سنجش، قدرت محاسباتی داخلی و شبکه‌ای مجهز به اینترنت برای برقراری ارتباط با یکدیگر تلقی می‌شوند. این حوزهٔ فناوری، یکی از حوزه‌هایی است که به‌سرعت در حال رشد است و به ارتباطات ماشین به ماشین متکی است و از پشتهٔ پروتکل اینترنت[8] برای برقراری ارتباط سرتاسری[9] استفاده می‌کند. در ساده‌ترین عبارت، اینترنت اشیا به‌عنوان شبکه‌ای متشکل از میلیاردها دستگاه تلقی می‌شود که می‌تواند اطلاعات را حس کرده و فعال کند و به یک سامانهٔ متمرکز منتقل کند. امروزه دستگاه‌ها و برنامه‌های IoT در حوزه‌های مختلفی مانند ترابری، خرده‌فروشی، مراقبت‌های بهداشتی، شبکهٔ شهر هوشمند، حمل‌ونقل هوشمند و مدیریت بلایا مستقر و استفاده شده‌اند.

با وجود پیشرفت‌های فناوری در این حوزه‌های فردی، ناهمگونی دستگاه‌های اینترنت اشیا و وجود چالش‌های استانداردسازی اشیا هنوز مورد توجه قرار نگرفته است. با وجود فقدان استانداردها، سامانه‌های اینترنت اشیا در حال حاضر با سرعت خیره‌کننده‌ای در حال گسترش‌اند. سامانه‌های اینترنت اشیا اکنون از جمعیت جهان فراتر رفته و پیش‌بینی می‌شود تا سال ۲۰۲۵ به هشتاد میلیارد دستگاه برسد. به‌منظور اطمینان از قابلیت همکاری، مقیاس‌پذیری و قابلیت اطمینان سامانه‌های اینترنت اشیا، بررسی دقیق‌تری از پشتهٔ فناوری مورد نیاز است. برآوردها نشان می‌دهد که در سال ۲۰۲۵ تعداد کل نصب

[1] Artificial neural network
[2] Support vector machines (SVMs)
[3] Random forest
[4] Feature selection
[5] Genetic algorithm
[6] Particle swarm optimization algorithm
[7] Sailfish optimization algorithm
[8] Internet Protocol Stack
[9] end-to-end communication

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

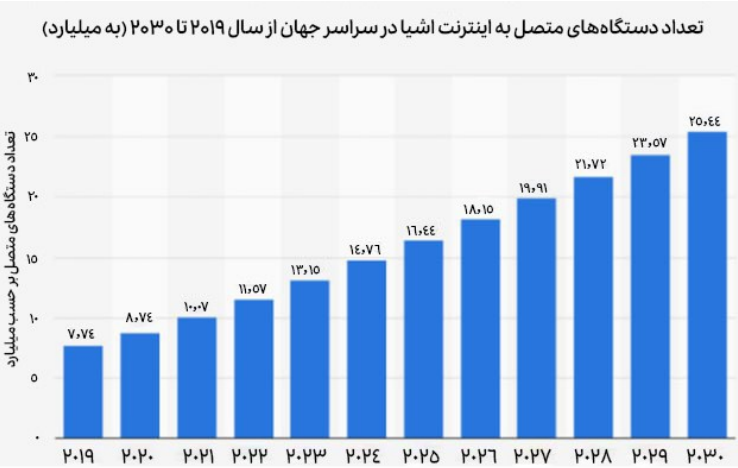
دستگاه‌های اینترنت اشیا در سراسر جهان از ۷۵ میلیارد نفر فراتر رود که این موضوع در شکل (۱)، نمایش داده شده‌است[۱۶]. پژوهش‌گران در حال حاضر در حال بررسی روش آدرس‌دهی IPv6 برای شبکه‌های بزرگ‌ترند که پیش‌بینی می‌شود به استاندارد نشان‌دهی برای دستگاه‌های IoT تبدیل شود.

مدل‌های مرجع اینترنت اشیا گامی روبه‌جلو در درک دسترسی به منابع، فناوری‌ها و هم‌گرایی لایه‌های تجاری به‌منظور اطمینان از مقیاس‌پذیری بازار است. به‌طور مشابه، نقش عناصر مختلف فعال در این لایه‌های اینترنت اشیا باید بررسی شود.

اینترنت مدرن ترکیبی پیچیده از گره‌های اینترنت، دستگاه‌های اینترنت اشیا و اشیای هوشمند است. شبکه‌های مجهز به اینترنت نیاز به پیاده‌سازی یک پشته IP برای ارتباط بین شبکه‌های اشیا دارند. با گسترش شبکه‌های اینترنتی، شرکت‌ها و همچنین جوامع پژوهشی در حال سرمایه‌گذاری در شبکه‌های IP انعطاف‌پذیر و مقیاس‌پذیر برای آینده‌اند.

در حال حاضر، شبکه‌های مجهز به اینترنت هم از نظر پیاده‌سازی فنی و هم در نیازهای برنامه‌های پایانی بسیار متفاوت‌اند. یک گره محاسباتی معمولی (مانند رایانه‌های شخصی، لپ‌تاپ‌ها، تلفن‌های همراه و تبلت‌ها) کل پشتهٔ TCP/IP را بر اساس مدل اتصال سامانه‌های باز پیاده‌سازی می‌کند؛ با این حال، به‌دلیل منابع محدود موجود در دستگاه‌های اینترنت اشیا، بر حسب معمول یک پشته IP سبک‌وزن پیاده‌سازی می‌شود. در دسترس‌بودن منابع داخلی و مصرف انرژی دستگاه در درجهٔ نخست اجرای پروتکل‌ها و استانداردهای مناسب در دستگاه‌های اینترنت اشیا را تنظیم می‌کند؛ بنابراین، در اصل بررسی معماری‌ها و پلتفرم‌های IoT برای درک نقش و رفتار در هر لایه از پشتهٔ فناوری مهم است. برای اینترنت اشیای چندمعماری در نظر گرفته شد که سه مدل مطرح آن در شکل (۲) به نمایش گذاشته شده‌است[۱۶].

در این معماری‌ها که به معماری سه‌لایه و پنج‌لایه معروف‌اند، لایه‌های حس‌گر، لایهٔ شبکه، لایهٔ کاربردی، لایهٔ میان‌افزار و لایهٔ تجاری وجود دارد. در اینترنت اشیا، لایهٔ حس‌گر اطلاعات را به‌وسیلهٔ حس‌گرها گردآوری کرده و بهکمک لایهٔ شبکه می‌توان این اطلاعات را برای لایهٔ کاربردی ارسال کرد و آن‌ها را پردازش کرد. اطلاعات در لایهٔ کاربردی می‌تواند در سامانه‌های ابری پردازش و تحلیل شود و الگوی مفید آن‌ها استخراج شود.



(شکل-۱): افزایش تعداد اشیای هوشمند بین سال‌های ۲۰۱۹ تا ۲۰۳۰[۱۶]

(Figure-1):Increasing the number of smart objects between 2019 and 2030[16]

۲-۲-حملات به شبکهٔ اینترنت اشیا

طی چند سال گذشته استفاده از دستگاه‌های هوشمند اینترنت اشیا مانند دوربین‌های هوشمند، تلویزیون‌های هوشمند، پوشیدنی‌های هوشمند، اسباب‌بازی‌های هوشمند، لامپ‌های هوشمند و غیره به‌طور تصاعدی در زندگی روزمرهٔ ما در حال افزایش بوده‌است؛ درنتیجه، این پیشرفت در فناوری محاسباتی، به دستگاه‌های هوشمند این قابلیت را داده‌است که بدون دخالت انسان، به‌صورت خودکار با یکدیگر در تعامل باشند؛ با وجود این‌که دستگاه‌های اینترنت اشیا در بسیاری از زمینه‌ها به ما



(شکل-۲): سه معماری مطرح در اینترنت اشیا[۱۶]

(Figure -2): three proposed architectures in the Internet of Things[16]

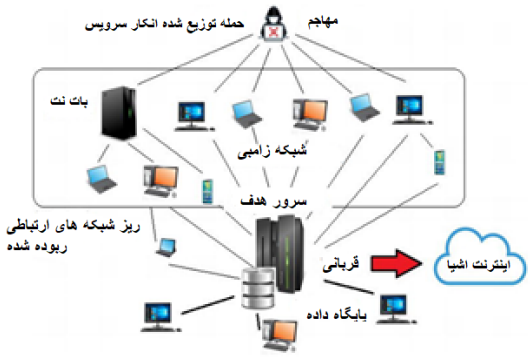
[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

کمک می‌کنند، ویژگی‌های امنیتی ناچیز یا بسیار محدودی دارند؛ علاوه‌براین، بسیاری از دستگاه‌های اینترنت اشیا دارای یک کلید ثابت یا نام کاربری و رمز عبور پیش‌فرض با کد سخت هستند که کاربر نمی‌تواند آن‌ها را تغییر دهد. این تله‌های امنیتی، بهره‌برداری از این دستگاه‌های ناامنِ اینترنت اشیا و کنترل آن‌ها را برای هکرها آسان می‌کند. روندهای اخیر نشان می‌دهد که با افزایش سریع تعداد دستگاه‌های ناامنِ اینترنت اشیا، حملات سایبری روزبه‌روز در حال افزایش است[۱۷].

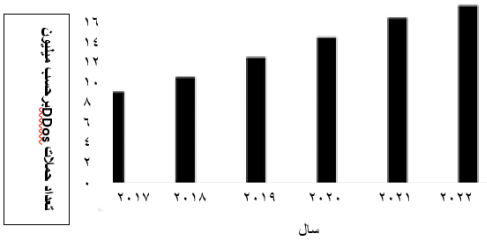
در میان حملات سایبری اخیر، حملات بات‌نت و انکار سرویس توزیع‌شده (DDoS) شایع‌ترین حملاتی هستند که هم از نظر فراوانی و هم از نظر بزرگی در دههٔ گذشته افزایش یافته‌اند. حملهٔ بات‌نت یک حملهٔ سایبری است که در آن مهاجم ابتدا یک شبکه را پویش می‌کند تا دستگاه‌های دارای امنیت ضعیف یا آسیب‌پذیر را جست‌وجو کند، پس از تجزیه‌وتحلیل اطلاعات پویش، مهاجم دستگاه‌های آسیب‌پذیر را هدف قرار می‌دهد تا یک برنامهٔ ربات را از طریق بدافزار در آن‌ها نصب کند. برنامهٔ ربات نصب‌شده دستگاه‌های آلوده را به یک سرور مرکزی یا یک شبکهٔ همتا متصل می‌کند که از آنجا دستورهای بیشتر به آن‌ها ارسال می‌شود تا فعالیت‌های مخرب مختلفی مانند ارسال هرزنامه‌ها، DDoS و غیره را از تعداد زیادی دستگاه انجام دهند. هنگامی که یک دستگاه اینترنت اشیا آلوده و بخشی از یک بات‌نت می‌شود، مهاجم از دستگاه آلوده برای انجام حملات DDoS استفاده می‌کند. حملهٔ بات‌نت نه‌تنها تهدیدی جدی برای دستگاه‌های ناامن اینترنت اشیا، بلکه تهدیدی حیاتی برای کل اینترنت است. با ظهور حملهٔ بات‌نت Mirai در سال ۲۰۱۶، حملات بات‌نت IoT به‌طور مداوم در حال افزایش است. پس از افشای عمومی کد منبع بات‌نت Mirai، بسیاری از انواع بات‌نت Mirai و تقلیدکنندگان آن تکامل یافته‌اند. گونه‌های جدیدی از بات‌نت Mirai و نسخه‌های تقلیدشده از آن با استفاده از کدهای مخرب، میلیون‌ها دستگاه اینترنت اشیا را آلوده کرده‌اند و حملات DDoS بزرگ و فاجعه‌باری علیه GitHub، AWS و غیره را در چند سال گذشته انجام داده‌اند[۱۸]. در شکل (۲)، یک سناریو از حملات علیه شبکهٔ اینترنت اشیا و شهرهای هوشمند نشان داده شده‌است که در آن هکر می‌تواند در بخش‌های مختلف شبکهٔ اینترنت اشیا نفوذ کند و سرویس‌های شبکه را مورد حمله قرار دهد[۱۹].

در این نوع از حملات تلاش می‌شود تا یک هکر تعداد زیادی سامانه را به بدافزار آلوده کرده و هر کدام از آن‌ها نقش یک بات‌نت را بازی کنند. در این حالت مشاهده می‌شود که بات‌نت‌ها با هم هماهنگ بوده و حملاتی منسجم در شبکه علیه یک سرویس، اجرایی می‌کنند. در این نوع حملات سرویس‌دهنده دیگر نمی‌تواند به ارائهٔ خدمات به کاربرانِ قانونی بپردازد. ارزیابی‌ها مطابق گزارش سیسکو نشان می‌دهد حملات DDoS تا سال ۲۰۲۲ رشد قابل توجهی خواهد داشت که یکی از دلایل اصلی این حملات وجود گره‌های آلوده به بدافزار است. طبق برآوردهای سیکسو تا سال ۲۰۲۲، میزان حملات DDoS در مقایسه با سال ۲۰۱۷ حدود دو برابر می‌شود که در نمودار شکل (۴) این موضوع نشان داده شده‌است. افزایش حجم و ترافیک حملات DDoS به‌سرعت در حال افزایش است و برآوردها نشان می‌دهد که حجم ترافیک این حملات تا ۱.۷ ترابایت بر ثانیه نیز خواهد بود. در مقایسه بین سال‌های ۲۰۱۷ و ۲۰۱۸، تعداد حملات علیه سامانه‌های امنیتی مانند سامانهٔ تشخیص حملات و دیوار آتش به‌ترتیب از ۱۶ به ۳۱ درصد افزایش داشته است[۲۰].



(شکل-۳): حملات رد سرویس خدمات توزیع‌شده در اینترنت اشیا[۱۹]

(Figure-3): denial of service attacks of distributed [19] services in the Internet of Thing



(شکل-۴): رشد حملات رد سرویس خدمات توزیع‌شده در سال‌های اخیر[۲۰]

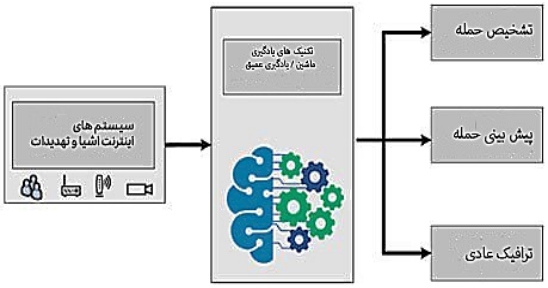
(Figure -4): The growth of distributed service denial of service attacks in recent year[20]

۲-۳-سامانهٔ تشخیص نفوذ

سامانهٔ تشخیص نفوذ به شبکه یک وظیفهٔ مهم دارد و آن تجزیه‌وتحلیل اطلاعات و ترافیک شبکه برای تشخیص حملات است. بیشتر سامانه‌های تشخیص نفوذ ساختار مشترکی دارند که شامل یک ماژول جمع‌آوری داده‌ها و ترافیک است که به‌احتمال زیاد حاوی شواهدی از یک

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

حمله است. ماژول تجزیه‌وتحلیل حملات وظیفه دارد ترافیک شناسایی‌شده را بررسی کرده و در صورت تشخیص الگوی یک حمله، نتایج را از طریق سازوکار گزارش‌دهی ارسال کند. در ماژول جمع‌آوری داده‌ها، ترافیک ورودی هر بخش از اینترنت اشیا را می‌توان جمع‌آوری و بررسی کرد تا ترافیک عادی از ناهنجاری یا حمله شناخته شود. ماژول تجزیه‌وتحلیل را می‌توان با استفاده از روش‌های مختلف پیاده‌سازی کرد؛ با این حال، روش‌های مبتنی بر یادگیری ماشین و یادگیری عمیق از بهترین روش‌ها برای تحلیل ترافیک شبکه است. در شکل (۵) یک سامانهٔ تشخیص نفوذ به شبکه با روش‌های یادگیری ماشین و یادگیری عمیق نشان داده شده‌است.



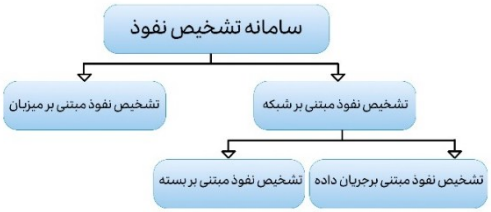
(شکل-۵): نقش روش‌های یادگیری ماشین و یادگیری عمیق برای تحلیل ترافیک اینترنت اشیا[۲۱]

(Figure-5): The role of machine learning and deep learning methods for Internet of Things traffic analysis[21]

در این شکل، مشاهده می‌شود که ورودی‌های مرتبط با روش‌های یادگیری ماشین و یادگیری عمیق درواقع ترافیک شبکه بوده و این اطلاعات برای آموزش و ارزیابی شبکهٔ عصبی مصنوعی استفاده می‌شود. در اینجا ترافیک شبکه می‌تواند در دستهٔ حمله یا نرمال و یا مشکوک قرار داده شود.

۲-۴-مرور مطالعات

سامانهٔ تشخیص نفوذ نرم‌افزار یا سرویسی است که فعالیت‌های غیرعادی را در یک شبکه یا سامانه کنترل یا شناسایی می‌کند. برای پیش‌بینی ناهنجاری‌ها در شبکه‌های اینترنت اشیا، به طور معمول از روش‌های متنوع مانند یادگیری ماشین، روش‌های اکتشافی یا روش‌های فهرست سیاه استفاده می‌شود. شکل (۶)، نشان می‌دهد که سامانه‌های تشخیص نفوذ بیشتر به دو نوع سامانه‌های تشخیص نفوذ مبتنی بر میزبان و مبتنی بر شبکه تقسیم می‌شوند. یک سامانهٔ تشخیص نفوذ مبتنی بر میزبان، یک دستگاه یا میزبان واحد را بر اساس اطلاعات دستگاه، مانند گزارش‌های سامانه، نظارت و ایمن می‌کند[۲۷]. در مقابل، یک سامانهٔ تشخیص نفوذ مبتنی بر شبکه، جریان داده‌ها را در یک شبکه با دسترسی و تجزیه‌وتحلیل داده‌ها اندازه‌گیری می‌کند. سامانه‌های تشخیص نفوذ مبتنی بر شبکه را می‌توان بیشتر به سامانه‌های مبتنی بر بسته و جریان تقسیم کرد. اطلاعات بستهٔ شبکه مانند بار یا هدر، برای سامانه‌های تشخیص نفوذ مبتنی بر بسته استفاده می‌شود؛ درنتیجه، بیشتر به آن سامانه‌های تشخیص نفوذ سنتی می‌گویند. در مقابل، سامانه‌های تشخیص نفوذ مبتنی بر جریان، ناهنجاری‌های درون یک شبکه را بر اساس ویژگی‌های جریان شبکه، مانند نرخ داده و بایت‌ها تحلیل و نظارت می‌کند؛ بنابراین، آن را به‌عنوان تحلیل رفتار شبکه نیز می‌شناسند. طبقه‌بندی فعالیت‌های مخرب یا غیرعادی شبکه با استفاده از مدل‌های یادگیری ماشین نظارت‌شده یا بدون نظارت امکان‌پذیر است[۲۷].



(شکل-۶): دسته‌بندی انواع سامانه‌های تشخیص نفوذ[۲۷]

(Figure-6): Classification of types of intrusion detection systems [27]

در ادامهٔ این بخش تعدادی از مطالعات در زمینهٔ سامانه‌های تشخیص نفوذ به شبکه مرور و بررسی می‌شود.

در [۲۸] با هدف مقابله با حملات سایبری روشی برای تشخیص و پیش‌بینی نفوذ در شبکه ارائه می‌شود که بر پایهٔ الگوریتم متاهیوریستیک گرگ خاکستری بهبودیافته است. ارزیابی با مجموعه‌دادهٔ NSL-KDD نشان داد که مدل پیشنهادی دقتی برابر با ۹۷.۱۴ درصد و ۹۸.۹۷ درصد در شناسایی حملات دارد که نسبت به روش‌های سنتی برتری محسوسی دارد؛ از جمله دستاوردها می‌توان به افزایش دقت، بهبود توازن تشخیص و کاهش چالش‌های طبقه‌بندی اشاره کرد.

در [۲۹] با توجه به افزایش پیچیدگی و تنوع حملات سایبری، مدلی هوشمند برای تشخیص نفوذ در شبکه ارائه شده‌است. با استفاده از روش PCA حجم داده‌ها کاهش یافته و سپس شبکهٔ عصبی CNN برای شناسایی و طبقه‌بندی حملات به‌کار رفته‌است؛ این ترکیب باعث بهبود دقت، افزایش سرعت و کاهش نرخ خطا نسبت به روش‌های پیشین شده‌است.

در [۳۰] یک مدل انتخاب ویژگی مؤثر با استفاده از الگوریتم‌های فرابتکاری ترکیبی برای تشخیص نفوذ اینترنت اشیا ارائه شده‌است. این پژوهش یک روش انتخاب ویژگی جدید با استفاده از الگوریتم بهینه‌سازی پرندگان و گوریل ارائه داده‌است. روش آن‌ها روی چهار مجموعه‌دادهٔ NSL-KDD،

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

CICIDS-2017، UNSW-NB15 و BoT-IoT پیاده‌سازی شده‌است. نتایج آزمایش‌ها نشان داد سامانهٔ تشخیص نفوذ آن‌ها از روش‌های بهینه‌سازی ذرات و کلونی زنبور عسل در مرحلهٔ انتخاب ویژگی دقیق‌تر است.

در [۳۱] یک مدل تشخیص نفوذ انتخاب ویژگی مبتنی بر یادگیری تقویتی عمیق ارائه شده‌است. این پژوهش یک مدل تشخیص نفوذ شبکه را بر اساس استخراج ویژگی RFE و یادگیری تقویت عمیق ارائه می‌کند. سامانهٔ تشخیص نفوذ پیشنهادی زیرمجموعهٔ بهینهٔ ویژگی‌ها را با استفاده از روش انتخاب ویژگی RFE پالایه می‌کند. در این مدل، ویژگی‌های منتخب با روش RFEبه شبکهٔ عصبی داده شده و با استفاده از مجموعه‌دادهٔ CSE-CIC-IDS2018 برای آموزش سامانهٔ تشخیص نفوذ استفاده شده‌اند. نتایج تجربی نشان می‌دهد که مدل پیشنهادی می‌تواند زیرمجموعهٔ بهینهٔ ویژگی‌ها را انتخاب کند، حدود هشتاددرصد از ویژگی‌های اضافی را حذف کند و ویژگی‌های انتخاب‌شده را از طریق مدل یادگیری یاد بگیرد تا عملکرد سامانهٔ تشخیص نفوذ را برای شناسایی حملهٔ شبکه افزایش دهد.

در [۳۲] یک مدل یادگیری عمیق برای طراحی سامانه‌های تشخیص نفوذ شبکه ارائه شده‌است. برای تشخیص نفوذ در این مطالعه یک NIDS مبتنی بر ناهنجاری با استفاده از یک مدل یادگیری عمیق انباشته-LSTM با یک تکنیک پیش‌پردازش جدید پیشنهاد شده‌است که ویژگی‌های بدون زمینه به آن داده می‌شود و مجموعه‌دادهٔ CICIDS2017 به دست می‌آید؛ این سامانه همچنین می‌تواند در محیط‌های مختلف بدون ازدست‌دادن دقت خود به‌دلیل مبتنی بر ویژگی‌های بدون متن اعمال شود.

در [۳۳] برای حل مشکل عدم تعادل طبقه در سامانه‌های تشخیص نفوذ شبکه از نمونه‌گیری مجدد داده‌ها و یادگیری عمیق استفاده شده‌است. در این پژوهش یک روش نمونه‌گیری مجدد داده‌ها در ترکیب با مدل‌های مختلف یادگیری عمیق برای کاهش مشکل عدم تعادل طبقه پیشنهاد شده‌است. مدل پیشنهادی بر روی مجموعه‌دادهٔ معیار NSL-KDD با استفاده از معیارهای دقت ارزیابی می‌شود. نتایج تجربی نشان می‌دهند که در طبقه‌بندی دودویی، روش پیشنهادی عملکرد سامانهٔ تشخیص نفوذ را بهبود می‌بخشد.

در [۳۴] یک رویکرد یادگیری عمیق فرااِبتکاری ترکیبی با مجموعه‌ای از شبکه‌های عصبی مکرر ارائه شده‌است. انواع مختلف حملات در سامانه‌های اینترنت اشیا با استفاده از مدل‌های یادگیری عمیق مانند LSTM، GRU و RNN شناسایی شده‌اند. انتخاب ویژگی‌ها با استفاده از بهینه‌سازی هریس هاک و جهش مشتق کسری، همان طور که در این پژوهش به‌کاررفته‌است، انجام می‌شود. برای ارزیابی رویکرد پیشنهادی، از مجموعه‌داده‌های در دسترس عموم استفاده شد و تحلیل تجربی نشان داد که رویکرد پیشنهادی از نظر دقت و کارایی به خوبی نسبت به سایر روش‌های مرتبط کار می‌کند.

در [۳۵] یک رویکرد سامانهٔ تشخیص نفوذ قابل اعتماد با استفاده از شبکهٔ عصبی پیچیدهٔ عمیق ارائه شده‌است. تازگی این مطالعه در توسعهٔ یک سامانهٔ تشخیص نفوذ قابل اعتماد است که از پتانسیل DCNN برای تجزیه‌وتحلیل ترافیک شبکه استفاده می‌کند. چهارچوب پیشنهادی با چهار مجموعه‌دادهٔ سامانهٔ تشخیص نفوذ در دسترس عموم، یعنی ISCX-IDS 2012، DDoS، CICIDS2017 و CICIDS2018 ارزیابی می‌شود. نتایج آن‌ها اثربخشی مدل بهینه‌سازی‌شدهٔ DCNN را در بهبود عملکرد و دقت تشخیص نفوذ نشان می‌دهد.

در [۳۶] ایمن‌سازی سامانه‌های اینترنت اشیا و SDN با استفاده از تشخیص نفوذ خودکار مبتنی بر یادگیری عمیق ارائه شده‌است. این پژوهش یک سامانهٔ تشخیص نفوذ خودکار دوسطحی امن را بر اساس یک شبکهٔ بهبودیافتهٔ LSTM پیشنهاد می‌کند. سامانهٔ پیشنهادی بین ترافیک حمله و ترافیک خوش‌خیم تفاوت قائل می‌شود، دستهٔ حمله را شناسایی و نوع حملهٔ فرعی را با عملکرد بالا تعریف می‌کند. سامانهٔ پیشنهادی با دو مجموعه‌دادهٔ واقعی جدید (ToN-IoT) و (InSDN) آموزش داده شده و عملکرد آن مورد ارزیابی قرار گرفته است. نتایج تجربی نشان می‌دهد که سامانهٔ پیشنهادی در تشخیص بسیاری از انواع حملات از سایرین بهتر عمل می‌کند. دقت ۹۶.۳۵ درصد، نرخ تشخیص ۹۶ درصد و دقت ۹۸.۴ درصد را برای مجموعه‌دادهٔ ToN-IoT به‌دست می‌آورد.

در [۳۷] سامانهٔ تشخیص نفوذ بر پایهٔ LSTM و GRU عمیق با شبکهٔ ایمن برای تشخیص نفوذ پیشرفته در محیط‌های IoT ارائه شده‌است. ارزیابی‌ها نشان می‌دهد سامانهٔ تشخیص نفوذ پیشنهادی نسبت به LSTM و GRU دقت بیشتری ارائه می‌کند و از طرفی توانایی تشخیص حملات ترکیبی و پیچیده را دارد.

در [۳۸] یک سامانهٔ تشخیص نفوذ بر اساس شبکهٔ عصبی عمیق برای تشخیص حملات ارائه شده‌است. هدف از این پژوهش طراحی یک مدل شبکهٔ عصبی عمیق برای تشخیص نفوذ است و روش‌هایی مانند SMOTE و نمونه‌گیری تصادفی برای رسیدگی به عدم تعادل داده‌ها در مجموعه‌دادهٔ CICIDS 2017 استفاده شده‌است. نتایج نشان می‌دهد که مدل یادگیری عمیق در پیش‌بینی حملات با مجموعه‌داده‌های CICIDS 2017 عالی بود و نرخ ازدست‌دادن ۰.۰۱۰۲ است.

در [۳۹] یک سامانهٔ تشخیص نفوذ بر پایهٔ یادگیری عمیق با رویکرد هوش مصنوعی برای تشخیص حملهٔ نفوذ

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

در شبکهٔ اینترنت اشیا ارائه شده‌است. در این مطالعه تکنیک بهینه‌سازی سلسله‌مراتب شرکتی[1] برای انتخاب ویژگی‌های مهم از پایگاه‌های داده استفاده می‌شود. راه‌حل پیشنهادی مبتنی بر بهینه‌سازی عقاب طلایی سیستم فازی چندلایه پرسپترون رابط فازی حریم خصوصی و امنیت را در زیرساخت شبکه حرفه‌ای بهبود می‌بخشد. آن‌ها روش خود را روی دو مجموعه‌دادهٔ NSL-KDD و UNSW-NB15 اجرا و بررسی کرده‌اند و نتایج نشان داده‌است که دقت آن از الگوریتم ژنتیک و الگوریتم ذرات در تشخیص نفوذ بیشتر است.

(جدول-۱): مرور کارهای مرتبط با تشخیص حملات
(Table-1): A Review of Related Works on Attack Detection

معایب	مزایا	روش	پژوهش
عدم آزمون روی داده‌های دنیای واقعی و جدی	بهبود سرعت طبقه‌بندی دقت بالا کاهش نرخ خطای منفی کاذب (FN) و مثبت کاذب (FP)	الگوریتم بهبودیافتهٔ گرگ خاکستری همراه با انتخاب ویژگی و Random Fores	[28]
مدل پیچیده	کاهش ابعاد داده ورودی اندک دقت بالا	ترکیبی از الگوریتم PCA + شبکهٔ عصبی پیچشی (CNN) + یادگیری عمیق	[29]
افزایش زمان سربار انتخاب ویژگی	از روش‌های بهینه‌سازی ذرات و کلونی زنبور عسل دقیق‌تر است	ترکیب الگوریتم بهینه‌ساز پردگان و گوریل	[30]
عدم انتخاب ویژگی هوشمندانه	هشتاد درصد از ویژگی هل اضافی را حذف می‌کند	تقویتی و انتخاب ویژگی انتخاب ویژگی مبتنی بر یادگیری تقویتی عمیق	[31]
عدم متعادل‌سازی مجموعه‌داده	دقت بالا	مدل‌گیری عمیق انباشته:LSTM	[32]
عدم انتخاب ویژگی هوشمندانه	متعادل‌سازی مجموعه‌داده و افزایش دقت یادگیری	نمونه‌گیری مجدد داده‌ها با یادگیری عمیق	[33]
پیچیدگی مدل قابل توجه است	دقت بالا	مدل‌هایی مانندLSTM،GRU ، RNN	[34]
عدم تعادل مجموعه‌داده و عدم کاهش ابعاد ورودی بدون انتخاب ویژگی هوشمندانه	دقت بیشتر از CNN	شبکهٔ عصبی پیچیدهٔ عمیق	[35]
عدم بهینه‌سازی پارامترها	دقت تشخیص ۹۶ درصد و دقت ۹۸.۴درصد	تشخیص نفوذ خودکار مبتنی بر SDN با یادگیری عمیق	[36]
عدم تشخیص حملات در زمان واقعی	دقت GRU و LSTM در تشخیص حملات ترکیبی	تشخیص حملات مبتنی بر یادگیری عمیق با پایهGRU ، LSTM	[37]
متعادل‌سازی به روش SMOTE ضعیف است	نرخ ازدست‌دادن اندک در حدود ۰.۰۱۰۲ است	متعادل‌سازی ترافیک شبکه با روش SMOTE	[38]
قطعیت کم در روش‌های فازی	دقت بیشتر از الگوریتم GA ,PSO	تکنیک بهینه‌سازی سلسله‌مراتب شرکتی و یادگیری عمیق فازی	[39]
عدم تعادل ویژگی در مجموعه‌داده	دقت مدل پیش‌بینی ۹۸.۷۶ درصد	یادگیری عمیق با استفاده از استراتژی بهینه‌سازی آشفته	[40]

[1] Corporate Hierarchy optimization (CHO)

در [۴۰] یک سامانهٔ تشخیص نفوذ شبکه مبتنی بر یادگیری عمیق با استفاده از استراتژی بهینه‌سازی آشفته[2] ارائه شده‌است. پس از پیش‌پردازش، مجموعه‌داده‌های نامتعادل با استفاده از رویکرد نمونه‌گیری مصنوعی توسعه‌یافته متعادل می‌شوند و پس از متعادل‌سازی، ویژگی‌های مجموعه‌داده با استفاده از تجزیه‌وتحلیل مؤلفهٔ اصلی به‌کمک هسته برداشته می‌شوند. ویژگی‌های بهینه به‌وسیلهٔ الگوریتم بهینه‌سازی انتخاب می‌شوند؛ پس از استخراج همهٔ ویژگی‌های مورد نیاز، حملات به‌وسیلهٔ مدل Dugat-LSTM طبقه‌بندی می‌شوند. در این پژوهش از مجموعه‌دادهٔ TON-IOT برای ارزیابی استفاده می‌شود. نتایج آزمایش‌ها نشان می‌دهد که مدل پیشنهادی در مجموعه‌دادهٔ TON-IOT به دقت ۹۸.۷۶ درصد دست یافته است. در جدول (۱)، کارهای مرتبط با تشخیص حملات با مزایا و معایب آن‌ها بررسی شده‌است.

بررسی کارهای مرتبط در زمینهٔ تشخیص نفوذ به شبکه نشان می‌دهد که این روش‌ها از چالش‌های زیر رنج می‌برند:

- در برخی از سامانه‌های تشخیص نفوذ از سازوکار امضا و فهرست سیاه استفاده شده‌است، اما این روش‌ها به حافظهٔ زیادی نیاز دارند و از طرفی زمان جست‌وجو در آن‌ها قابل توجه است و توانایی تشخیص حملات جدید را ندارند.
- روش‌های اکتشافی یکی دیگر از روش‌های توسعهٔ سامانه‌های تشخیص نفوذ است، اما این روش فاقد سازوکار یادگیری و تشخیص حملات پیچیده است و از طرفی کیفیت تشخیص حملات به کیفیت تابع اکتشافی بستگی دارد.
- روش‌های یادگیری ماشین و یادگیری عمیق توانایی تشخیص حملات پیچیده را دارند، اما این روش‌ها تعادل در مجموعه‌داده، انتخاب ویژگی و بهینه‌سازی پارامترهای مدل‌های یادگیری را ندارند.

برای تشخیص نفوذ دقیق در این مقاله و رفع این چالش‌ها یک سامانهٔ تلفیقی هوش گروهی و یادگیری ماشین ارائه می‌شود. برای رفع چالش عدم تعادل در مجموعه‌داده‌های تشخیص نفوذ از تئوری بازی و شبکهٔ عصبی GAN استفاده می‌شود و در مرحلهٔ بعدی برای کاهش ابعاد ترافیک و انتخاب ویژگی از الگوریتم دودویی‌شدهٔ ارهماهی استفاده می‌شود. در مرحلهٔ سوم از ماشین بردار پشتیبان برای تشخیص و طبقه‌بندی ترافیک شبکه استفاده شده و پارامترهای ماشین بردار پشتیبان را می‌توان با الگوریتم‌های جدید مانند بهینه‌سازی محاسبات ریاضی[3] [۴۱] بهبود داد. دلیل استفاده از الگوریتم AOA

[2] Chaotic Honey Badger
[3] Archimedes optimization algorithm

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

آن است که پیچیدگی بالایی ندارد و اگر برای بهینه‌سازی پارامترهای SVM استفاده شود آنگاه زمان یادگیری آن را افزایش قابل توجه نمی‌دهد. دلیل استفاده از الگوریتم بهینه‌سازی اره‌ماهی در مرحلهٔ انتخاب ویژگی به شرح زیر خلاصه می‌شود:

- مسئلهٔ انتخاب ویژگی بر خلاف بهینه‌سازی پارامترهای SVM یک مسئله با ابعاد بسیار بیشتر است و از این جهت نیاز به الگوریتم‌های هوشمندتر به ویژه با رویکرد هوش گروهی دارد.
- الگوریتم بهینه‌سازی اره‌ماهی با تقسیم جمعیت راه‌حل‌ها به دو دسته اره‌ماهی و ساردین می‌تواند بین جست‌وجوی اکتشافی یا سراسری و جست‌وجوی بهره‌برداری یا محلی توازن ایجاد کند.

الگوریتم اره‌ماهی از الگوریتم‌های فراابتکاری رایج مانند بهینه‌سازی ذرات، الگوریتم خفاش و کرم‌شب‌تاب دارای دقت بیشتری برای یافتن جواب بهینه است.

۳- روش پیشنهادی

روش پیشنهادی برای تشخیص حملات به شبکهٔ یک سامانهٔ تشخیص نفوذ بر پایهٔ تئوری بازی، یادگیری عمیق، هوش گروهی و یادگیری ماشین است. در روش پیشنهادی چند مرحلهٔ اصلی زیر برای تشخیص ارائه می‌شود که به شرح زیر است:

- متعادل‌سازی ترافیک شبکه با استفاده از تئوری بازی و یادگیری عمیق مبتنی شبکهٔ GAN
- ارائهٔ یک نسخهٔ دودویی و انتخاب ویژگی از الگوریتم اره‌ماهی
- بهینه‌سازی پارامترهای SVM با استفاده از الگوریتم AOA
- یادگیری سامانهٔ تشخیص نفوذ و ارزیابی آن با نمونه‌های آزمون

در این بخش روش پیشنهادی برای تشخیص حملات به شبکه ارائه و معرفی می‌شود و در ابتدا چهارچوب روش پیشنهادی ارائه می‌شود و در ادامه تلاش می‌شود تا روابط و معادلات مرتبط توضیح داده شود.

۳-۱-مراحل روش پیشنهادی

مراحل روش پیشنهادی برای تشخیص حملات به اینترنت اشیا و شهر هوشمند مطابق شکل (۶)، نمایش داده شده‌است که به شرح زیر است:

۱) ترافیک شبکه در لایهٔ ادراک و حس‌گر تولید شده و این ترافیک برای گره‌های مه ارسال می‌شود. در روش پیشنهادی هر گره مه نقش یک سامانهٔ تشخیص نفوذ به شبکه را بر عهده دارد.

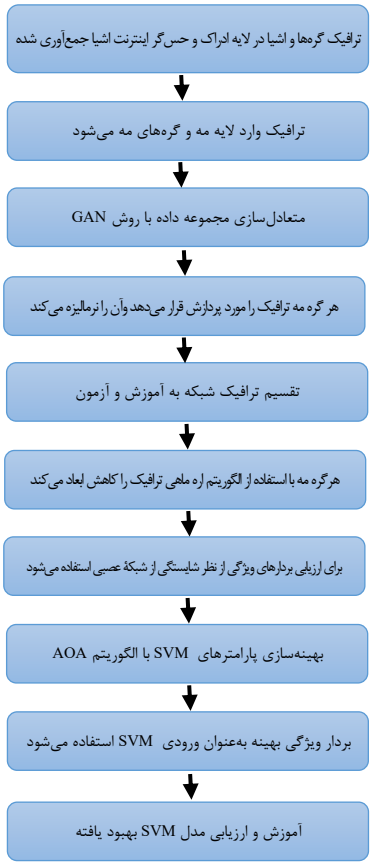
۲) برای متعادل‌سازی مجموعه‌داده از تئوری بازی و شبکهٔ عصبی یادگیری عمیق GAN استفاده می‌شود و هدف آن است که تعداد نمونه‌های حمله برابر تعداد نمونهٔ عادی باشد. هدف از متعادل‌سازی آن است که خطای طبقه‌بندی مدل یادگیری یا SVM کاهش داده شود.

۳) ترافیک شبکه در گره‌های مه به دو دستهٔ آموزشی و آزمون تقسیم شده و از ترافیک آموزشی برای آموزش روش پیشنهادی استفاده می‌شود و برای ارزیابی از روش ارزیابی متقاطع با kfold=10 استفاده می‌شود.

۴) هر گره مه با استفاده از الگوریتم اره‌ماهی انتخاب ویژگی کرده و ویژگی‌های مهم را انتخاب کرده و از آن‌ها به‌عنوان ورودی‌های SVM استفاده می‌شوند.

۵) ماشین بردار پشتیبان با بردار ویژگی بهینهٔ ترافیک شبکه آموزش داده می‌شود و سپس خطای آن با استفاده از داده‌های آزمون ارزیابی قرار می‌شود. در این مرحله برای کاهش خطای بیشتر مدل SVM از الگوریتم AOA به‌عنوان یک روش فراابتکاری بر پایهٔ عملیات ریاضی استفاده می‌شود. هدف آن است که پارامترهای مانند نرخ خطا و جریمه در SVM با روش فراابتکاری AOA بهینه‌سازی شود.

روش پیشنهادی یک سامانهٔ تشخیص نفوذ در معماری توزیع‌شده مه محاسباتی است و برای تشخیص حملات به شبکه دارای چهار مرحلهٔ اصلی زیر است:

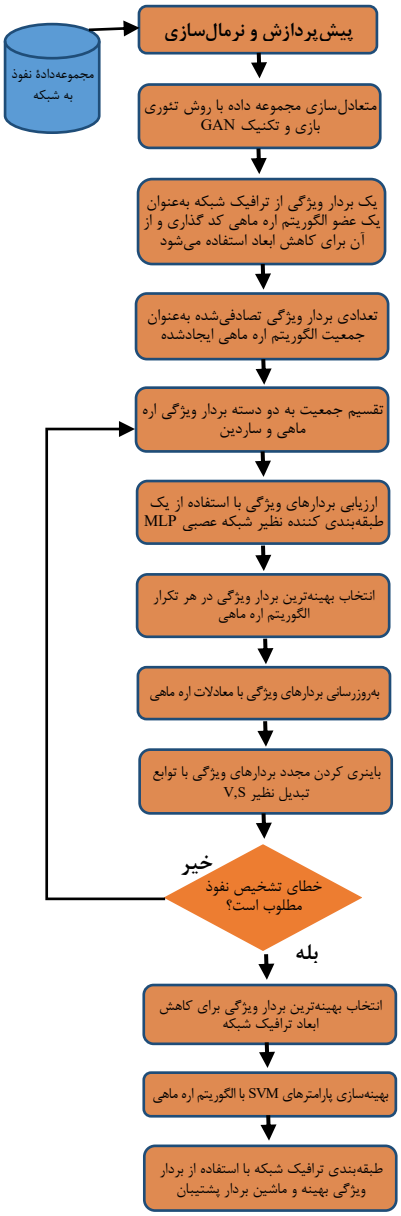


(شکل-۷): مراحل تشخیص حملات در شهرهای هوشمند
(Figure-7): Steps to detect attacks in smart cities

- متعادل‌سازی مجموعه‌داده با روش GAN
- انتخاب ویژگی در گره‌های مه با استفاده از الگوریتم اره‌ماهی

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

- بهینه‌سازی پارامترهای SVM با الگوریتم فرابتکاری AOA
- یادگیری ماشین بردار پشتیبان بر اساس ویژگی‌های مهم انتخاب‌شده



(شکل-۸): چهارچوب روش پیشنهادی برای تشخیص حملات
(Figure-8): Framework of the proposed method for detecting attacks

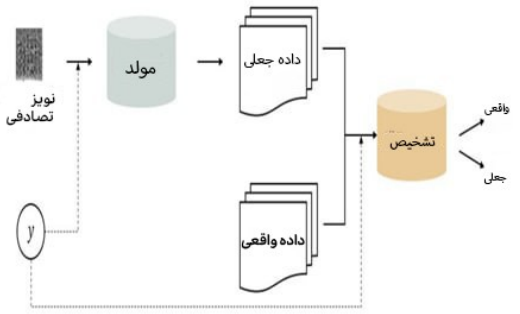
۳-۲-چهارچوب روش پیشنهادی

چهارچوب روش پیشنهادی در شکل (۸) برای تشخیص حملات نمایش داده شده‌است و مراحل آن به شرح زیر است:

۱) در ابتدا مجموعه‌ای از ترافیک شبکه به‌عنوان ورودی و مجموعه‌داده در نظر گرفته می‌شود.
۲) ترافیک شبکه با استفاده از تئوری بازی و روش GAN متعادل‌سازی می‌شود.
۳) ترافیک شبکه پیش‌پردازش و نرمال‌سازی می‌شود تا بتوان از ترافیک و نمونه‌های آن برای آموزش و یادگیری استفاده کرد.
۴) یک بردار ویژگی از ویژگی‌های ترافیک شبکه در نظر گرفته شده و این بردار ویژگی به‌عنوان یک عضو الگوریتم اره‌ماهی کدگذاری می‌شود.
۵) در تکرار نخست تعدادی بردار ویژگی تصادفی تعیین شده و هر کدام از این بردارهای دودویی یک عضو الگوریتم اره‌ماهی است.
۶) هر بردار ویژگی و الگوی آن روی مجموعه‌دادهٔ نفوذ به شبکه اعمال شده و مجموعه‌داده دچار کاهش ابعاد می‌شود.
۷) شبکهٔ عصبی مصنوعی به وسیلهٔ هر کدام از بردارهای ویژگی آموزش داده می‌شود.
۸) خطای طبقه‌بندی ترافیک شبکه در مرحلهٔ آموزش و تعداد ویژگی انتخاب‌شده در هر بردار ویژگی اگر کمینه‌تر باشد آن بردار ویژگی بهینه‌تر است.
۹) اگر خطای تشخیص حملات و نفوذ به شبکه کمینه و مطلوب نباشد مراحل بالا مجدد تکرار می‌شود تا بردارهای ویژگی به‌روزرسانی شوند.
۱۰)اگر خطای تشخیص حملات و نفوذ به شبکه مطلوب باشد آنگاه هر گره مه از بردار ویژگی بهینه در تکرار آخر برای کاهش‌دادن ورودی ماشین بردار پشتیبان در تشخیص حملات به شبکه استفاده می‌کند.
۱۱) پارامترهای SVM با استفاده از الگوریتم AOA بهینه‌سازی می‌شود.
۱۲)مدل SVM بهبودیافته با استفادهٔ بردار ویژگی بهینه آموزش و برای تشخیص حملات به‌کار گرفته می‌شود.

۳-۳-متعادل‌سازی مجموعه‌داده

عدم تعادل در یک مجموعه‌داده زمانی اتفاق می‌افتد که تعداد نمونه‌های یک طبقه مانند طبقه حملات از طبقه عادی کمتر باشد. عدم تعادل مجموعه‌داده باعث می‌شود تا دقت مدل‌های یادگیری عمیق و ماشین کاهش داده شود. یک روش برای متعادل‌سازی مجموعه‌دادهٔ تشخیص نفوذ که تعداد نمونه‌های حملات آن در کمترین حد است ایجاد نمونه‌های مصنوعی بر اساس نمونه‌های واقعی و اضافه‌کردن آن‌ها به مجموعه‌داده است. برای متعادل‌سازی می‌توان از تئوری بازی مبتنی بر یادگیری GAN[1] مطابق شکل (۹) استفاده کرد[۴۲].



(شکل-۹): عملکرد روش GAN در تولید نمونه‌های مصنوعی برای متعادل‌سازی مجموعه‌داده[۴۲]
(Figure-9): The performance of the GAN technique in generating synthetic samples for the balancing of the dataset[42]

[1] Generative adversarial network (GAN)

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

GAN یک مدل یادگیری بدون نظارت مبتنی بر یادگیری عمیق است که داده‌های جعلی شبیه به داده‌های واقعی را با قراردادن یک شبکهٔ عصبی (مولد، G) در مقابل دیگری (تمایزگر، D) تولید می‌کند. G با هدف تولید داده‌های جعلی شبیه داده‌های واقعی آموزش داده می‌شود، در حالی که D برای تعیین اینکه داده‌های ایجادشده به وسیلهٔ G در واقع جعلی‌اند آموزش داده می‌شود؛ به عبارت دیگر، G و D به روش خصمانه یاد می‌گیرند. شکل(۹)، ساختار GAN را نشان می‌دهد و نحوهٔ یادگیری G و D را توضیح می‌دهد. ابتدا زمانی که G یک بردار نوفهٔ تصادفی را به‌عنوان ورودی دریافت می‌کند، داده‌ها تولید می‌شوند.

هنگامی که داده‌های تولیدشده و واقعی در اختیار D قرار می‌گیرد، واقعی یا جعلی‌بودن آن‌ها را مشخص می‌کند. G و D برای این نتیجه رقابت می‌کنند و از آن درس می‌گیرند؛ به عبارت دیگر، هدف G به حداکثر رساندن احتمالی است که D داده‌های تولیدشده را واقعی تعیین می‌کند؛ در حالی که هدف D این است که احتمال تشخیص داده‌های تولیدشده را به‌عنوان جعلی به بیشترین حد برساند. معادلهٔ (۱) تابع هدف GAN را برای این فرایند یادگیری نشان می‌دهد:

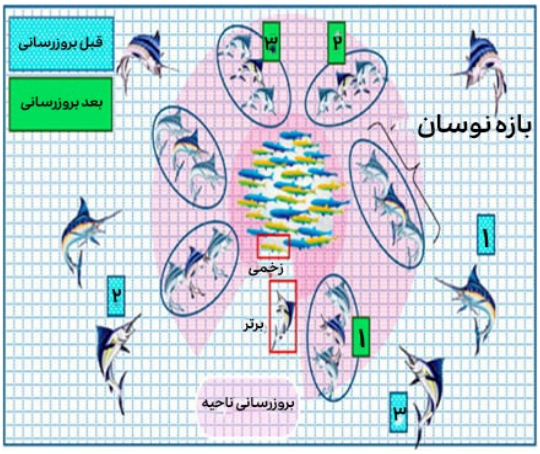
(۱)

$$\min_G \max_D V(D,G) = E_{x\sim p_{\text{data}}(x)}[log\ D(x)] + E_{z\sim p_z(z)}[log\ (1 - D(G(z)))].$$

در معادلهٔ $pdata(x)$ و $pz(z)$ به‌ترتیب به داده‌های واقعی و جعلی اشاره دارند. D دادهٔ x را به‌عنوان مقدار ورودی دادهٔ واقعی دریافت می‌کند و احتمال واقعی‌بودن داده را خروجی می‌دهد، $D(x)$. G یک بردار نوفهٔ تصادفی z را به‌عنوان مقدار ورودی می‌گیرد و داده‌های جعلی تولید می‌کند (z). از آنجایی که هدف D تشخیص مؤثر بین داده‌های جعلی تولیدشده و داده‌های واقعی است، GAN باید یاد بگیرد که $D(x)$ برابر یک و $D(G(z))$ برابر صفر باشد. از آنجایی که هدف D تشخیص مؤثر بین داده‌های جعلی تولیدشده و داده‌های واقعی است GAN باید یاد بگیرد که $D(x)$ برابر یک و $D(G(z))$ برابر صفر باشد. به عبارت دیگر، تابع هدف معادله به حداکثر رساندن از منظر D و کمینه سازی از دیدگاه G است.

۳-۴-انتخاب ویژگی پیشنهادی

الگوریتم اره‌ماهی از رفتار اره‌ماهی ان و شکار ساردین‌ها الگوبرداری شده‌است. در این الگوریتم راه‌حل‌های مسئله می‌توانند ساردین یا اره‌ماهی باشند. در الگوریتم نصف جمعیت به‌صورت ساردین است و نصف دیگر راه‌حل‌ها به‌صورت اره‌ماهی است. ساردین‌ها را می‌توان جواب‌های شایسته در نظر گرفت و اره‌ماهی ها را می‌توان جواب‌های غیر شایسته فرض کرد؛ اره‌ماهی ان برای یافتن غذا به سمت ساردین‌ها حمله می‌کنند و در مرتبهٔ نخست سعی می‌کنند یک ماهی را زخمی کنند؛ سپس ماهی زخمی را مورد حمله و شکار قرار دهند. رفتار این الگوریتم را می‌توان در شکل (۱۰) مشاهده کرد.



(شکل-۱۰): حملهٔ اره‌ماهی ان به سمت ساردین‌ها[۱۵]
(Figure-10): Sawfish attack towards sardines [15]

در این مسئله هر بردار ویژگی یک ساردین یا اره‌ماهی است و از طرفی نصف جمعیت بردارهای ویژگی اره‌ماهی و نصف دیگر ساردین هستند. در این الگوریتم مشاهده می‌شود که ساردین‌ها از دست اره‌ماهی ان فرار کرده و نوعی جست‌وجوی سراسری یا اکتشافی را انجام می‌دهند و در مقابل حملهٔ اره‌ماهی ان به سمت طعمه یا ساردین زخمی نوعی جست‌وجوی محلی است. علت استفاده از الگوریتم اره‌ماهی در انتخاب ویژگی به شرح زیر است:

در ابتدا تعدادی بردار ویژگی به‌صورت اعضای الگوریتم بهینه‌سازی اره‌ماهی به‌عنوان یک الگوریتم فراابتکاری مبتنی بر رفتار گروهی و دسته‌جمعی اره‌ماهی ان و ساردین‌ها در نظر گرفته شده و در اینجا یک بردار ویژگی نظیر $F_1^i, F_2^i, \dots, F_k^i$ نشان‌دهندهٔ عضو i الگوریتم اره‌ماهی است و در اینجا k برابر تعداد ویژگی‌های ممکن در مجموعه‌داده است. می‌توان یک جمعیت اولیه از بردارهای ویژگی که مقدار آن‌ها صفر و یک است به‌عنوان اره‌ماهی و ساردین در نظر گرفت که در رابطه‌های (۱) و (۲) به‌ترتیب بردارهای اره‌ماهی و ساردین نمایش داده شده‌است:

$$SF = \begin{bmatrix} SF_{1,1} & SF_{12} & \dots & SF_{1,k} \\ SF_{2,1} & SF_{22} & \dots & SF_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ SF_{m,1} & SF_{m,2} & \dots & SF_{m,k} \end{bmatrix} \quad (۱)$$

در این رابطه SF جمعیت اولیه از بردارهای ویژگی از نوع اره‌ماهی و به تعداد m است که هر کدام از آن‌ها دارای k ویژگی است. در رابطهٔ (۲) نیز بردارهای ویژگی از نوع

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

ساردین نمایش داده شده‌است و به‌طور معمول نصف جمعیت از این نوع بردارهای ویژگی است و تعداد این نوع بردارهای ویژگی برابر $\boldsymbol{n}$ عدد است:

$$S = \begin{bmatrix} S_{1,1} & S_{12} & \cdots & S_{1,k} \\ S_{2,1} & S_{22} & \cdots & S_{2,k} \\ \vdots & \vdots & \vdots & \vdots \\ S_{n.1} & S_{n.2} & \cdots & S_{n.k} \end{bmatrix} \quad (۲)$$

برای ارزیابی بردارهای ویژگی از نوع ارەماهی یا ساردین نیاز به یک تابع هدف مناسب است که این تابع در رابطهٔ (۳)، ارائه شده‌است:

$$CostFS = \alpha.\gamma_R(D) + \beta\frac{|R|}{|N|} \quad (۳)$$

یک بردار ویژگی نیاز به ارزیابی دقیق دارد تا مشخص شود که تا چه اندازه شایستگی دارد. برای ارزیابی بردارهای ویژگی دو شاخصهٔ مهم زیر در نظر گرفته می‌شود:

- یک بردار ویژگی زمانی شایستگی دارد که خطای تشخیص حملات توسط آن کمینه باشد. به عبارت بهتر هر بردار ویژگی که بتواند ورودی‌های SVM را به‌گونه‌ای کاهش ابعاد دهد که خطای طبقه‌بندی ترافیک عادی و نرمال توسط آن کمینه شود دارای شایستگی بیشتری است.
- یک بردار ویژگی هر چقدر باعث شود که فضای ویژگی دچار کاهش ابعاد شود و ورودی‌های SVM را بیشتر کاهش دهد آنگاه شایستگی بیشتری دارد.

می‌توان مطابق [۴۳] خطای تشخیص حملات و کاهش ابعاد را هم‌زمان برای ارزیابی بردار ویژگی در نظر گرفت و به ضرایبی وزنی بین صفر و یک آن‌ها را به‌عنوان یک تابع هدف ارائه داد. برای سنجش خطای تشخیص حملات به‌وسیلهٔ یک بردار ویژگی مانند FS_i می‌توان از یک طبقه‌بندی‌کنندهٔ ساده مانند شبکهٔ عصبی مصنوعی چندلایه استفاده کرد؛ در این حالت بردار ویژگی FS_i روی مجموعه‌داده نگاشت داده می‌شود به گونه‌ای که اگر مؤلفه j ام بردار ویژگی FS_i دارای مقدار یک است از ویژگی j ام به‌عنوان ورودی شبکهٔ عصبی استفاده می‌شود. فرض کنید بردار ویژگی FS_i دارای $|FS_i|$ ویژگی باشد که انتخاب شده‌اند(مقدار این ویژگی‌ها در بردار ویژگی برابر یک است)؛ از این جهت تعداد ورودی‌های شبکهٔ عصبی برای این بردار ویژگی برابر $|FS_i|$ خواهد بود. برای ارزیابی بردارهای ویژگی از نوع ارەماهی یا ساردین نیاز به یک تابع هدف مناسب است که این تابع در رابطهٔ (۴) ارائه شده‌است[۴۳]:

$$Cost(FS_i) = \alpha.E(FS_i) + \beta\frac{|FS_i|}{|Dim|} \quad (۴)$$

در این رابطه $E(FS_i)$ خطای طبقه‌بندی ترافیک حملات از ترافیک عادی به وسیلهٔ بردار ویژگی FS_i است. $|FS_i|$ نیز برابر تعداد ویژگی‌های انتخاب‌شده در بردار ویژگی است که اندازهٔ آن برابر $|Dim|$ است و همچنین α یک عدد تصادفی است که مقدار آن بین صفر و یک است و $\beta = 1 - \alpha$ در نظر گرفته می‌شود. ضرایب وزنی باعث می‌شود تا overfitting در مرحلهٔ انتخاب ویژگی تا حد زیادی تعدیل شود؛ از طرفی متعادل‌سازی مجموعه‌داده با روش GAN آن این چالش را در روش پیشنهادی کم‌رنگ می‌کند.

هر بردار ویژگی که این تابع را کمینه کند اگر از نوع ساردین باشد به‌عنوان ساردین زخمی و اگر از نوع ارەماهی باشد آنگاه به‌عنوان بهینه‌ترین ارەماهی برای حمله به دستهٔ ماهیان ساردین در نظر گرفته می‌شود. می‌توان بردارهای ارەماهی و ساردین که همگی از نوع بردارهای ویژگی‌اند را با این تابع هدف ارزیابی کرد و مقدار ارزیابی آن‌ها را در رابطه‌های (۵) و (۶) نمایش داد:

(۵)

$$SF_{fitness} = \begin{bmatrix} CostFS(SF_{1,1} & SF_{12} & \cdots & SF_{1,k}) \\ CostFS(SF_{2,1} & SF_{22} & \cdots & SF_{2,k}) \\ \vdots & \vdots & \vdots & \vdots \\ CostFS(SF_{m,1} & SF_{m,2} & \cdots & SF_{m,k}) \end{bmatrix} = \begin{bmatrix} F_{SF_1} \\ F_{SF_2} \\ \vdots \\ F_{SF_m} \end{bmatrix}$$

(۶)

$$S_{fitness} = \begin{bmatrix} CostFS(S_{1,1} & S_{12} & \cdots & S_{1,k}) \\ CostFS(S_{2,1} & S_{22} & \cdots & S_{2,k}) \\ \vdots & \vdots & \vdots & \vdots \\ CostFS(S_{n,1} & S_{n,2} & \cdots & S_{n,k}) \end{bmatrix} = \begin{bmatrix} F_{S_1} \\ F_{S_2} \\ \vdots \\ F_{S_n} \end{bmatrix}$$

در روش پیشنهادی، بردار ویژگی از نوع ساردین که بهینه‌ترین بردار ویژگی از نوع ساردین‌هاست و بهینه‌ترین بردار ویژگی ارەماهی ان می‌تواند برای انتخاب مکان حملهٔ بردارهای ویژگی از نوع ارەماهی و به مانند رابطهٔ (۷) در نظر گرفته شود:

(۷)

$$F^i_{new_{SF}} = \mathrm{F}^i_{elite_{SF}} - \lambda_i\left(rand(0,1).\left(\frac{F^i_{elite_{SF}} + F^i_{injured_S}}{2}\right) - F^i_{old_{SF}}\right)$$

در این رابطه، $\mathrm{F}^i_{new_SF}$ بردار ویژگی به‌روزشدهٔ یک ارەماهی، $F^i_{elite_SF}$ بردار ویژگی بهینه‌ترین ارەماهی، $F^i_{injured_S}$ بهینه‌ترین بردار ویژگی از نوع ساردین، $F^i_{old_SF}$ موقعیت قبلی یک بردار ویژگی از نوع ارەماهی،

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

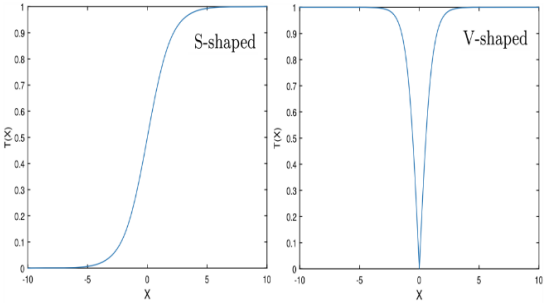
$rand(0,1)$ عدد تصادفی یک‌نواخت بین صفر و یک، λ_i ضریب هم‌گرایی در تکرار i-ام است که مرتب کاهش خواهد یافت. هر بردار ویژگی از نوع ساردین نیز می‌تواند تحت تأثیر فرار به‌روزرسانی شود و برای این منظور از رابطهٔ (۸) استفاده می‌شود:

$$F^i_{new_S} = rand(0,1).(F^i_{elite_SF} - F^i_{old_S} + AP) \quad (۸)$$

در این رابطه $F^i_{new_S}$ و $F^i_{old_S}$ به‌ترتیب موقعیت جدید یک بردار ویژگی از نوع ساردین و موقعیت پیشین آن بردار ویژگی برای تشخیص حملات به اینترنت اشیا است. در این رابطه، AP را توان حملهٔ اره‌ماهی ان می‌گویند که می‌توان آن را از رابطهٔ (۹) محاسبه کرد:

$$AP = A.(1 - 2.Itr.\epsilon) \quad (۹)$$

A و ϵ پارامترهای ثابت و اولیهٔ توان حمله برای به‌روزرسانی بردارهای ویژگی از نوع ساردین در نظر گرفته می‌شوند. هر بردار ویژگی از نوع ساردین یا اره‌ماهی به‌ترتیب تحت تأثیر مکانیزم فرار و حمله به‌روزرسانی شده و موقعیت آن‌ها دچار تغییر می‌شود، اما این تغییرات می‌تواند این بردارهای ویژگی را از حالت دودویی خارج و آن‌ها را تبدیل به فضای پیوسته که به کمک توابع تبدیل یا انتقال، بردارهای ویژگی را مجدد دودویی و صفر و یک کند. در شکل (۱۱)، دو تابع کاربردی نوع V و S که برای تبدیل فضای ویژگی به‌کار گرفته می‌شوند نمایش داده شده‌است.



(شکل-۱۱): به‌ترتیب از راست به چپ، توابع تبدیل انتخاب ویژگی از نوع V و S[۱۶]

(Figure-11): right to the left, the transformation functions of feature selection from type of V and S [16]

این دو تابع درواقع دارای بردی محدود به بازه ۱+و ۰ است و می‌تواند اعداد پیوستهٔ بردارهای ویژگی را به این بازه نگاشت دهند. نقش تابع S و V آن است که مقادیر بردارهای ویژگی بعد از به‌روزرسانی به‌وسیلهٔ الگوریتم ماهی در بازه صفر تا یک نرمالیزه شوند؛ به عبارت بهتر بردارهای ویژگی در الگوریتم اره‌ماهی شامل مقادیر دودویی صفر و یک هستند که به‌ترتیب نشان‌دهندهٔ عدم انتخاب و انتخاب ویژگی‌اند؛ از آن‌جا که به‌روزرسانی‌ها ممکن است این مقادیر را از حالت دودویی خارج کنند، توابع و V برای نرمال‌سازی و بازگرداندن آن‌ها به بازه [۰،۱] به‌کار می‌روند که بردارهای ویژگی از حالت صفر و یک خارج شوند و مقادیری غیر از صفر یا یک داشته باشند؛ برای مثال عدد هشت در یک بردار ویژگی بی‌معنی است؛ لذا باید نخست عدد هشت را به بازه [۱، ۰] نرمالیزه کرد و سپس آن را به عدد ۰ یا یک نگاشت داد؛ برای مثال اگر عدد هشت به عدد ۰.۴ نرمال شود؛ چون به صفر نزدیک‌تر است می‌توان آن را صفر در نظر گرفت (ویژگی انتخاب نشده‌است) و اگر عدد هشت به ۰.۷ نرمالیزه شود می‌توان آن را به یک تبدیل کرد (ویژگی انتخاب شده)؛ زیرا ۰.۷ به یک نزدیک‌تر است تا صفر. بعد از نرمال‌سازی مقادیر بردارهای ویژگی با استفاده از توابع انتقالی S وV، می‌توان با اعمال آستانهٔ ۰.۵ و بر اساس روابط (۱۰) و (۱۱)، مقادیر هر عنصر در بردار را دوباره به حالت صفر یا یک تبدیل کرد[۱۶]:

$$X_i^j(t+1) = \begin{cases} 0 & rand < T\left(X_i^j(t+1)\right) \\ 1 & otherwise \end{cases} \quad (۱۰)$$

$$X_i^j(t+1) = \begin{cases} \neg X_i^j(t) & rand < T\left(X_i^j(t+1)\right) \\ X_i^j(t) & otherwise \end{cases} \quad (۱۱)$$

در اینجا، تابع نگاشت و تبدیل می‌تواند از نوع S یا V باشد و FS_i یک بردار ویژگی است.

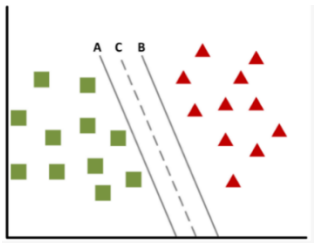
۵-۳-طبقه‌بندی با ماشین بردار پشتیبان

روش ماشین بردار پشتیبان نیز یک روش یادگیری ماشین است که برای طبقه‌بندی استفاده می‌شود. در این روش برای یادگیری یک خط جداکننده باید به‌گونه‌ای بین دو طبقه داده‌ها قرار داده شود که کمترین خطا در طبقه‌بندی به‌وجود آید. در شکل (۱۱) در این روش خط C به‌عنوان خط طبقه‌بندی‌کننده به‌کار گرفته می‌شود و خط A و B نیز ناحیه و مرز این خط را نشان می‌دهند و هدف آن است که عرض نوار بیشینه شود و به شرط این‌که تعداد کمتری از نمونه‌ها به‌اشتباه در دو طبقه قرار داده شوند.

در شکل (۱۲) در این روش خط C به‌عنوان خط طبقه‌بندی‌کننده به‌کار گرفته می‌شود و خط A و B نیز ناحیه و مرز این خط را نشان می‌دهند و هدف آن است که عرض نوار بیشینه شود، به شرط این‌که تعداد کمتری از نمونه‌ها به‌اشتباه در دو طبقه قرار داده شوند.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]



(شکل-۱۲): روش طبقه‌بندی ماشین بردار پشتیبان[۲۶]
(Figure-12): Support vector machine classification technique [۲۶]

طبقه‌بندی‌کنندهٔ SVM نمونه‌ها را به یک فضای مشخصه ابعادی بزرگ (از طریق یک هسته) تبدیل و مکان مناسب ابرصفحهٔ[1] مرزی را تعیین که داده‌های آموزشی را تقسیم می‌کند. ایجاد ابرصفحه در SVM باید از قانون رابطهٔ (۱۲) طبقه‌بندی پیروی کند:

$$f(x) = (W, x) + b \qquad (۱۲)$$

که در آن w بردار نرمال و b یک اصطلاح بایاس است. SVM ابرصفحه را برای یافتن یک طبقه‌بندی خطی با بهینه‌سازی قاعدهٔ f تنظیم می‌کند. این قانون طبقه‌بندی می‌تواند برای اختصاص یک برچسب به نمونهٔ آزمایشی x استفاده شود. اگر نتیجهٔ $f(x)$ کمتر از صفر باشد، به‌عنوان نفوذ طبقه‌بندی می‌شود، در غیر این صورت به‌عنوان نرمال طبقه‌بندی می‌شود. نتیجهٔ f(x) می‌تواند شرایط طبقه‌بندی را روشن کند به‌گونه‌ای که در طبقه نرمال مثبت و در طبقه نفوذ منفی در نظر گرفته می‌شود. سامانهٔ تشخیص نفوذ SVM یک طبقه را می‌توان به‌صورت نگاشت داده‌ها در بردار ویژگی H با استفاده از یک تابع هستهٔ مناسب بیان کرد؛ سپس تلاش می‌کند تا بردارهای نگاشت‌شده را از مبد با یک حاشیه مشخص جدا کند، رابطهٔ (۱۳).

$$f(x) = \begin{cases} +1, & \text{if } x \in \text{ Normal} \\ -1, & \text{if } x \in \text{ Intrusion} \end{cases} \qquad (۱۳)$$

در مرحلهٔ دوم از یک طبقه SVM، فرض کنید $x1$، $x2$، ... xl نمونه‌های آموزشی متعلق به یک طبقه X باشند، که در آن X زیرمجموعه‌ای از مجموعه‌داده است. فرض کنید $\Phi: X \to H$ یک نقشهٔ هسته باشد که نمونه‌های آموزشی را به فضای دیگری تبدیل می‌کند؛ سپس برای جداکردن مجموعه‌داده از مبدأ باید مسئلهٔ برنامه‌نویسی درجه دوم رابطهٔ (۱۴) را حل کند:

(۱۴)

$$\min \frac{1}{2} \| w \|^2 + \frac{1}{Vl} \sum_{i=1}^{l} \xi_i - p$$
$$(w \times \Phi(xi)) \geq \rho - \xi ii = 1,2,\dots,l \xi i \geq 0$$

که شرایط آن در معادلهٔ (۱۵)، تعریف شده‌است.

(۱۵)

[1] Hyperplane

اگر w و ρ در این مسئله حل شوند، تابع تصمیم‌گیری به‌صورت معادلهٔ(۱۶) ارائه می‌شود:

$$f(x) = \text{sign}\,((w \times \Phi(x)) - \rho) \qquad (۱۶)$$

برای بهبود عملکرد ماشین بردار پشتیبان می‌توان پارامترهای آن مانند نرخ خطا و ضریب جریمه را به‌عنوان یک عضو جمعیت الگوریتم فراابتکاری در نظر گرفت و سپس در ادامه با این الگوریتم فراابتکاری، پارامترهای SVM را بهینه‌سازی کرد. برای حل این مسئله از الگوریتم AOA استفاده می‌شود که بر اساس چهار عمل‌گر ضرب، تقسیم، جمع و ضرب به دنبال بهینه‌سازی جواب‌ها و پارامترهای SVM است. در الگوریتم AOA در هر تکرار پارامترهای MOA و MOP به‌ترتیب مانند معادلهٔ (۱۷) و (۱۸) به‌روزرسانی شوند:

(۱۷)

$$MOA(C_Iter) = Min + C_Iter \times \left(\frac{Max - Min}{M_Iter}\right)$$

(۱۸)

$$MOP(C_Iter) = 1 - \frac{C_Iter^{\frac{1}{\alpha}}}{M_Iter^{\frac{1}{\alpha}}}$$

Max و Min به‌ترتیب بیشینه و کمینهٔ پارامترهای MOA و MOP است. C_Iter شمارندهٔ تکرار الگوریتم AOA است و M_Iter بیشترین تکرار این الگوریتم است. ضریب MOP بر حسب پارامتر α تعیین و ضریب آلفا در حدود پنج تنظیم می‌شود. برای انجام جست‌وجوی اکتشافی از معادلهٔ (۱۹) استفاده می‌شود. یک عدد تصادفی بین صفر و یک برای هر راه‌حل تولید می‌شود و اگر این عدد تصادفی کمتر از ۰.۵ باشد از عمل‌گر تقسیم و در غیراین‌صورت از عمل‌گر ضرب برای به‌روزرسانی پارامترهای SVM استفاده می‌شود:

(۱۹)

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times ((UB_j - LB_j) \times \mu + LB_j) & rand > 0.5 \\ best(x_j) \times MOP \times \left((UB_j - LB_j) \times \mu + LB_j\right) & rand \leq 0.5 \end{cases}$$

$best(x_j)$ بهینه‌ترین راه‌حل یا بهینه‌ترین پارامترهای SVM است. UB_j و LB_j به‌ترتیب محدود بالا و پایین بعد j –ام یک مجموعه پارامتر SVM است. μ یک پارامتر و ثابت عددی است. ε یک عدد بسیار کوچک برای جلوگیری از تقسیم بر صفر است. برای به‌روزرسانی پارامترهای SVM از معادلهٔ (۲۰)، استفاده می‌شود.

(۲۰)

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best(x_j) - MOP \times ((UB_j - LB_j) \times \mu + LB_j) & rand > 0.5 \\ best(x_j) + MOP \times ((UB_j - LB_j) \times \mu + LB_j) & rand \leq 0.5 \end{cases}$$

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

۴- تجزیه‌وتحلیل

در این بخش به تجزیه‌وتحلیل روش پیشنهادی برای تشخیص حملات پرداخته شده‌است و روش پیشنهادی مورد ارزیابی قرار می‌گیرد.

۴-۱-مجموعه‌داده

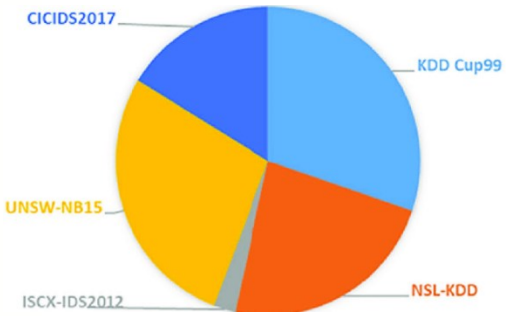
مجموعه‌دادهٔ NSL-KDD به‌عنوان یکی از بهترین مجموعه‌داده‌های مرتبط با تشخیص نفوذ به شبکه در نظر گرفته می‌شود. این مجموعه‌داده در ابتدا توسط استولفا[1] و همکاران با آنالیز اطلاعات ردوبدل‌شده در شبکهٔ نیروی هوایی آمریکا در یکی از پایگاه‌های آن گردآوری شده‌است. این مجموعه‌داده اعتبار بالایی برای سنجش دقت الگوریتم‌های تشخیص نفوذ به شبکه و اینترنت اشیا دارد؛ زیرا تنوعات گسترده‌ای را شامل می‌شود؛ درکل می‌توان گفت که مجموعه‌دادهٔ مورد نظر اعتبار و محبوبیت بالایی دارد و در بسیاری از پژوهش‌ها مورد استفاده قرار می‌گیرد. این مجموعه‌داده سه ویژگی ممتاز زیر را در خود نهفته دارد:

- تلفیقی از ترافیک عادی و مشکوک است که می‌توان با تجزیه‌وتحلیل ویژگی‌های آن الگوهای ترافیک شبکه را شناسایی و از آن برای طبقه‌بندی ترافیک شبکه استفاده کرد.
- در این مجموعه‌داده ترافیک مبتنی بر زمان در شبکه نیز مورد توجه قرار گرفته‌است؛ به‌گونه‌ای که در این مجموعه‌داده، داده‌ها به‌مرور زمان گردآوری شده‌اند و تنها مختص یک لحظه یا یک برش زمانی نیستند.
- در این مجموعه‌داده ترافیک مبتنی بر میزبان نیز مورد توجه قرار گرفته به این صورت که ترافیک شبکه تنها بر روی یک میزبان گردآوری نشده‌است؛ بلکه در میزبان‌های مختلف گردآوری شده‌است.

مجموعه‌دادهٔ تشخیص نفوذ به شبکه یا KDD، ۴۱ ویژگی مختلف و متنوع دارد که به درکل ۳۴ عدد از آن‌ها عددی و مابقی از نوع غیرعددی است. در این مجموعه‌داده مشاهده شده که سه نوع ویژگی کلی به‌کار گرفته شده که در زیر به آن‌ها اشاره شده‌است:

- ویژگی‌های اساسی[2]: این گروه از ویژگی‌ها در واقع تمام ویژگی‌هایی را که می‌تواند از یک ارتباط و ترافیک TCP-IP[3] استخراج شوند شامل می‌شود. شناسایی این مجموعه ویژگی‌ها نیاز به زمان کمابیش طولانی دارد؛ زیرا برخی از این ویژگی‌ها به‌مرور زمان در شبکه تشخیص داده شده و گردآوری می‌شوند.
- ویژگی‌های محتوایی: حملات از نوع U2R و R2L برخلاف حملات نوع DOS و Prob فاقد هر گونه الگوی ترتیبی و منظم هستند؛ زیرا به این دلیل که حملات DOS و Prob در یک مقطع زمانی کوتاه شروع و بعد از مدتی به پایان می‌رسند و این در حالی است که حملات نوع U2R و R2L در بخش دادهٔ بسته ارسالی به شبکه جاسازی می‌شوند و به‌طور عادی تنها یک ترافیک را درگیر می‌کنند، اما حملات نوع DOS و Prob تعداد بیشتری ترافیک را شامل می‌شوند.
- ویژگی‌های ترافیکی: این ویژگی‌ها تنها ترافیک را در زمانی فعلی یا در یک بازه کوتاه مورد توجه قرار می‌دهند و به دو نوع ویژگی‌های ترافیک میزبان و ویژگی‌های ترافیک سرویس دسته‌بندی می‌شوند که به‌ترتیب ناظر بر ترافیک وارد شده به یک میزبان و یا یک سرویس عملیاتی‌اند.

در این مجموعه‌داده به‌طورکلی چهار نوع حمله تعریف شده و هر حمله نیز خود دارای تعداد زیادی زیرحمله است. در حملات نوع پویشی مهاجم اطلاعاتی را به‌کمک پویش[4] سرویس‌ها یا پورت‌های رایانه به‌دست می‌آورد و بر اساس این اطلاعات سعی می‌کند به شبکه نفوذ و در آن اختلال ایجاد کند. در این نوع از حملات مهاجم می‌تواند برای نفوذ به شبکه از روش‌های مهندسی اجتماعی[5] نیز استفاده تا بتواند از اطلاعات افراد و کاربران درون شبکه برای نفوذ به شبکه استفاده کند. مطالعات مطابق شکل (۱۳)، نشان می‌دهد که در بین مجموعه‌داده‌های مختلف دو مجموعه‌دادهٔ KDD و NSL-KDD دارای بیشترین سهم در ارزیابی سامانه‌های تشخیص نفوذ به شبکه است. در شکل (۱۴)، چند رکورد از مجموعه‌دادهٔ NSL-KDD برای تشخیص حملات نمایش داده شده‌است و مشاهده می‌شود که ۴۱ ویژگی از نوع ورودی است و ویژگی آخر نیز می‌تواند حمله یا عادی‌بودن ترافیک را نشان دهد.



(شکل-۱۳): سهم مجموعه‌داده NSK-KDD در میان سایر مجموعه‌داده‌های تشخیص نفوذ [۲۲]

(Figure-13): Share of NSK-KDD dataset among other intrusion detection dataset[22]

[1] Stolfa
[2] Basic features
[3] Transmission Control Protocol-Internet Protocol
[4] Scan
[5] Social engineering

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]



(شکل-۱۴): ویژگی‌های مجموعه‌دادهٔ NSK-KDD در تشخیص حملات به شبکه [۲۳]
(Figure-14): Features of the NSK-KDD dataset in detecting network attacks[23]

۴-۲-شاخص‌های ارزیابی

برای ارزیابی سامانهٔ تشخیص نفوذ پیشنهادی از شاخص‌های طبقه‌بندی مطرح مانند دقت[۱]، حساسیت[۲] و صحت[۳] استفاده می‌شود که ضابطهٔ آن‌ها به‌ترتیب در رابطه‌های (۲۱، ۲۲ و ۲۳) فرموله شده‌است. برای محاسبهٔ این شاخص‌ها نیاز است که از شاخص‌های اولیهٔ مثبت واقعی[4]، منفی واقعی[5]، مثبت کاذب[6] و منفی کاذب[7] استفاده شود:

(۲۱)
$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

(۲۲)
$$Sensitivity = \frac{TP}{TP+FN}$$

(۲۳)
$$Precision = \frac{TP}{TP + FP}$$

در این معادلات TP نشان می‌دهد ترافیک از نوع حمله است و روش پیشنهادی به‌درستی آن را از نوع حمله تشخیص داده‌است. در اینجا TN نشان می‌دهد ترافیک نرمال و روش پیشنهادی آن را به‌درستی از نوع نرمال تشخیص داده‌است و FP نشان می‌دهد ترافیک به‌غلط حمله تشخیص داده شده و FN نیز نشان می‌دهد ترافیک به‌اشتباه نرمال تشخیص داده شده‌است.

[1] Accuracy
[2] Sensitivity
[3] Precision
[4] True positive(TP)
[5] True negative(TN)
[6] False positive(FP)
[7] False negative(FN)

۴-۳-آزمایش‌ها و بحث

در پیاده‌سازی‌ها هفتاد درصد از نمونه‌ها آموزشی و سی درصد از نمونه‌ها از نوع آزمون در نظر گرفته می‌شوند. اعتبارسنجی متقاطع روشی است که در یادگیری ماشین برای ارزیابی عملکرد یک مدل بر روی داده‌های دیده نشده استفاده می‌شود. این شامل تقسیم داده‌های موجود به چند زیرمجموعه، استفاده از یکی از این نمونه‌ها به‌عنوان مجموعهٔ اعتبارسنجی و آموزش مدل بر روی نمونه‌های باقی‌مانده است؛ این فرایند چندین بار تکرار می‌شود و هر بار از یک fold متفاوت به‌عنوان مجموعهٔ اعتبارسنجی استفاده می‌شود؛ درنهایت نتایج حاصل از هر مرحله اعتبارسنجی برای تولید تخمین قوی‌تری از عملکرد مدل میانگین می‌شود. اعتبارسنجی متقاطع مرحلهٔ مهمی در فرایند یادگیری ماشین است و به اطمینان از این‌که مدل انتخاب‌شده برای استقرار قوی است و به‌خوبی به داده‌های جدید تعمیم می‌یابد کمک می‌کند. در روش پیشنهادی از اعتبارسنجی متقاطع با k=10 استفاده می‌شود. تعداد رکوردهای به‌کاررفته برای ارزیابی آزمایش‌ها پنجاه‌هزار ترافیک تنظیم شده‌است که توسط روش GAN این مجموعه‌داده متعادل‌سازی شده و ۲۵ هزار نمونه از نوع حمله و ۲۵ هزار دیگر از نوع ترافیک عادی است. محدودهٔ نرمال‌سازی‌شده در بازه [۱، ۰] است و اندازهٔ جمعیت و تعداد تکرار الگوریتم اره‌ماهی به‌ترتیب برابر بیست و پنجاه است. تعداد آزمایش‌ها برای محاسبهٔ میانگین شاخص‌ها برابر سی آزمایش است. تعداد لایه‌های پنهان در شبکهٔ عصبی برابر دو و تعداد نورون‌های هر لایه برابر ۴۱ عدد و این تعداد برابر تعداد ویژگی‌های اولیهٔ مجموعه‌دادهٔ NSL-KDD است. برای طبقه‌بندی نهایی از ماشین بردار پشتیبان با کرنل خطی[8]، بازگشتی[9]، چند

[8] Linear
[9] RBF

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

جمله‌ای[1] و گوسین[2] استفاده می‌شود. در جدول (۲)، متوسط شاخص‌های دقت، حساسیت و صحت روش پیشنهادی به‌ازای کرنل‌های مقایسه‌شده است. در نمودار شکل (۱۵)، شاخص دقت، حساسیت و صحت روش پیشنهادی به‌ازای کرنل‌های مختلف نمایش داده شده‌است. آزمایش‌ها در محیط برنامه‌نویسی متلب نشان می‌هد، دقت، حساسیت و صحت روش پیشنهادی به‌ازای کرنل گوسین بیشترین مقدار ممکن است و دارای مقادیر ۹۹.۱۲ درصد، ۹۸.۹۲ درصد و ۹۸.۹۶ درصد است.

آزمایش‌ها نشان می‌دهد کمترین دقت، حساسیت و صحت روش پیشنهادی به‌ازای کرنل خطی است. اگر از ماشین بردار پشتیبان استفاده نشود و برای طبقه‌بندی از شبکهٔ عصبی نیز استفاده شود، آن‌گاه شاخص دقت، حساسیت و صحت به‌ترتیب تا حدود ۹۸.۸۳ درصد، ۹۸.۲۴ درصد و ۹۸.۱۶ درصد کاهش خواهد یافت. علت آن‌که روش پیشنهادی به‌ازای کرنل گوسین بهتر عمل کرده و خطای تشخیص نفوذ را بیشتر کاهش داده آن است که مسئلهٔ تشخیص نفوذ یک مسئلهٔ پیچیده است و فضای ویژگی آن خطی نیست؛ لذا برای طبقه‌بندی و تشخیص نفوذ در کرنل گوسین که غیرخطی است، بهتر جواب می‌دهد. برای آزمایش‌ها و مقایسه‌ها می‌توان نتایج پژوهش را با مطالعه در پژوهش [۲۴] که سال ۲۰۲۱ انجام شده‌است، مورد ارزیابی قرار داد. در نمودار شکل (۱۶)، روش پیشنهادی در تشخیص حملات با درخت تصمیم‌گیری J48، جنگل تصادفی RF و شبکهٔ عصبی چند لایه MLP مورد مقایسه قرار گرفته و ارزیابی شده‌است. ارزیابی‌ها نشان می‌دهد دقت روش پیشنهادی به‌دلیل انتخاب ویژگی دارای دقتی برابر ۹۹.۱۲ درصد و این در حالی است که دقت روش درخت تصمیم‌گیری J48، جنگل تصادفی RF و شبکهٔ عصبی چندلایهٔ MLP به‌ترتیب برابر ۸۱.۰۵ درصد، ۸۰.۶۷ درصد و ۷۷.۴۱ درصد است و روش پیشنهادی نسبت به این روش‌ها دارای دقت بیشتری است؛ به عبارت بهتر آزمایش‌ها نشان داد، ماشین بردار پشتیبان در ترکیب با اره‌ماهی نسبت به این روش‌ها در تشخیص حملات دقت بیشتری دارد.

برای ارزیابی منصفانه می‌توان روش پیشنهادی را با نتایج مطالعهٔ [۲۵] که دارای سازوکار انتخاب ویژگی هستند، مطابق نمودار شکل (۱۷)، با هم مقایسه کرد.

آزمایش‌ها نشان می‌دهد کمترین دقت، حساسیت و صحت روش پیشنهادی به‌ازای کرنل خطی است. اگر از ماشین بردار پشتیبان استفاده نشود و برای طبقه‌بندی از شبکهٔ عصبی نیز استفاده شود، آن‌گاه شاخص دقت، حساسیت و صحت به‌ترتیب تا حدود ۹۸.۸۳ درصد، ۹۸.۲۴ درصد و ۹۸.۱۶ درصد کاهش خواهد یافت. علت آن‌که روش پیشنهادی به‌ازای کرنل گوسین بهتر عمل کرده و خطای تشخیص نفوذ را بیشتر کاهش داده آن است که مسئلهٔ تشخیص نفوذ یک مسئلهٔ پیچیده است و فضای ویژگی آن خطی نیست؛ لذا برای طبقه‌بندی و تشخیص نفوذ در کرنل گوسین که غیرخطی است، بهتر جواب می‌دهد. برای آزمایش‌ها و مقایسه‌ها می‌توان نتایج پژوهش را با مطالعه در پژوهش [۲۴] که سال ۲۰۲۱ انجام شده‌است مورد ارزیابی قرار داد. در نمودار شکل (۱۶)، روش پیشنهادی در تشخیص حملات با درخت تصمیم‌گیری J48، جنگل تصادفی RF و شبکهٔ عصبی چندلایه MLP مورد مقایسه قرار گرفته و ارزیابی شده‌است. ارزیابی‌ها نشان می‌دهد دقت روش پیشنهادی به‌دلیل انتخاب ویژگی دارای دقتی برابر ۹۹.۱۲ درصد و این در حالی است که دقت روش درخت تصمیم‌گیری J48، جنگل تصادفی RF و شبکهٔ عصبی چندلایهٔ MLP به‌ترتیب برابر ۸۱.۰۵ درصد، ۸۰.۶۷ درصد و ۷۷.۴۱ درصد است و روش پیشنهادی نسبت به این روش‌ها دارای دقت بیشتری است؛ به عبارت بهتر آزمایش‌ها نشان داد، ماشین بردار پشتیبان در ترکیب با اره‌ماهی نسبت به این روش‌ها در تشخیص حملات دقت بیشتری دارد.

برای ارزیابی منصفانه می‌توان روش پیشنهادی را با نتایج مطالعهٔ [۲۵] که دارای سازوکار انتخاب ویژگی هستند مطابق نمودار شکل (۱۸)، با هم مقایسه کرد.

در این نمودار، روش پیشنهادی با روش‌های یادگیری عمیق مانند AE-CNN، LSTM، LSTM-RNN، GA-ELM، CNN-LSTM و ELM در شاخص دقت مقایسه شده‌است. آزمایش‌ها نشان می‌دهند دقت AE-CNN، LSTM، LSTM-RNN، GA-ELM، CNN-LSTM و ELM برای تشخیص حملات به‌ترتیب برابر ۹۳.۹۹ درصد، ۹۴.۱۱ درصد، ۹۶.۹۳ درصد، ۹۸.۹۰ درصد، ۹۹.۷۰ درصد، ۹۸.۹۴ درصد است و دقت روش پیشنهادی با کرنل گوسین برای تشخیص حملات برابر ۹۹.۱۲ درصد است. علت آن‌که روش پیشنهادی نسبت به روش‌های یادگیری عمیق در تشخیص حملات دارای دقت بیشتری است آن است که روش پیشنهادی در مرحلهٔ نخست مجموعه‌داده را با روش GAN متعادل‌سازی کرده و این عامل دقت مدل SVM را افزایش می‌دهد و از طرفی با الگوریتم AOA پارامترهای SVM بهینه‌سازی شده و ورودی‌های SVM نیز با الگوریتم اره‌ماهی بهینه شده‌است تا یادگیری روی ویژگی‌های اساسی انجام شود که در شکل (۱۹) قابل مشاهده است.

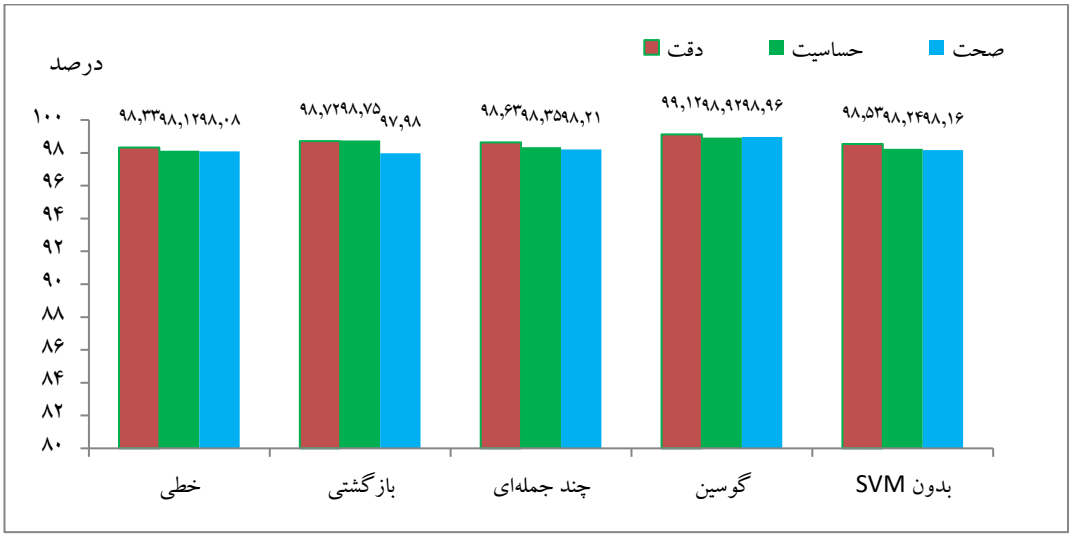
[1] Polynomial
[2] Gaussian

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]
[DOI: 10.61882/jsdp.22.2.43]

(جدول-۲): شاخص دقت، حساسیت و صحت به‌ازای کرنل‌های مختلف

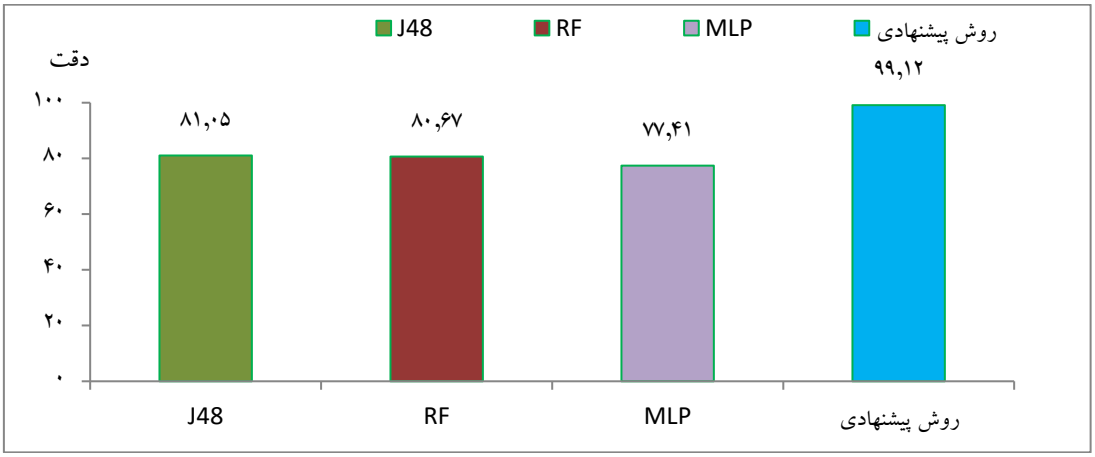
(Table -2): Accuracy, sensitivity and accuracy index for different kernels

نوع کرنل	دقت	حساسیت	صحت
خطی	۹۸.۳۳	۹۸.۱۲	۹۸.۰۸
بازگشتی	۹۸.۷۲	۹۸.۷۵	۹۷.۹۸
چند جمله‌ای	۹۸.۶۳	۹۸.۳۵	۹۸.۲۱
گوسین	**۹۹.۱۲**	**۹۸.۹۲**	**۹۸.۹۶**
SVMبدون	۹۸.۵۳	۹۸.۲۴	۹۸.۱۶



(شکل-۱۵): ارزیابی شاخص دقت، حساسیت و صحت روش پیشنهادی با کرنل‌های مختلف

(Figure-15): Evaluation of the index of accuracy, sensitivity and accuracy of the proposed method with different kernels

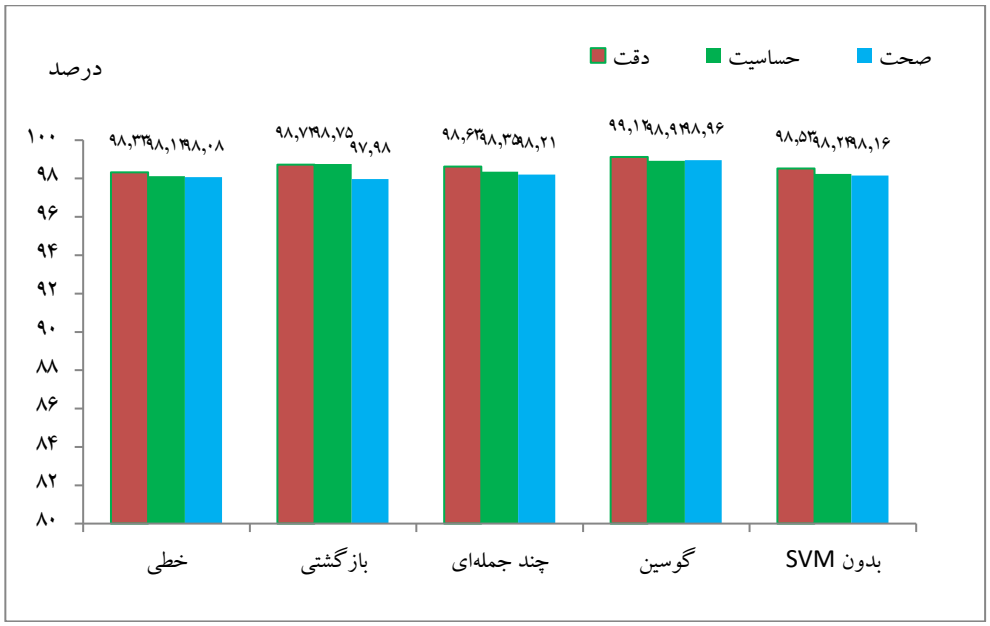


(شکل-۱۶): مقایسهٔ دقت روش پیشنهادی با سه روش یادگیری ماشین

(Figure-16): Comparison of the accuracy of the proposed method with three machine learning methods

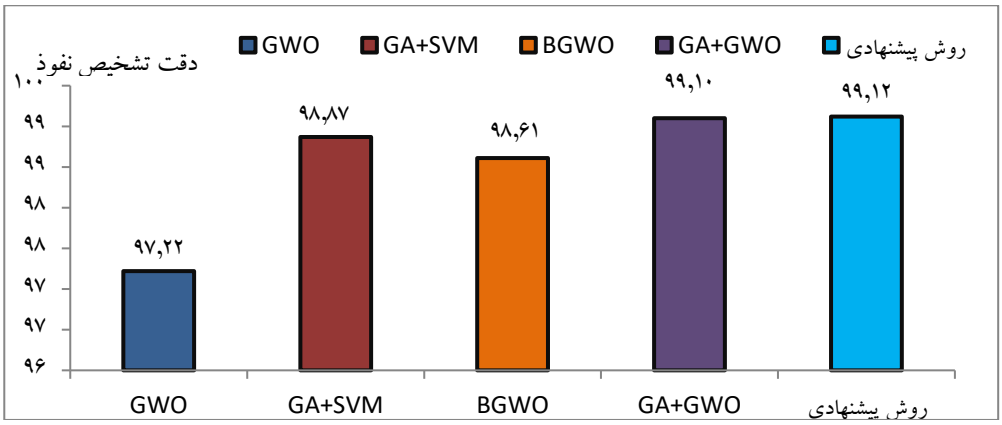
[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]



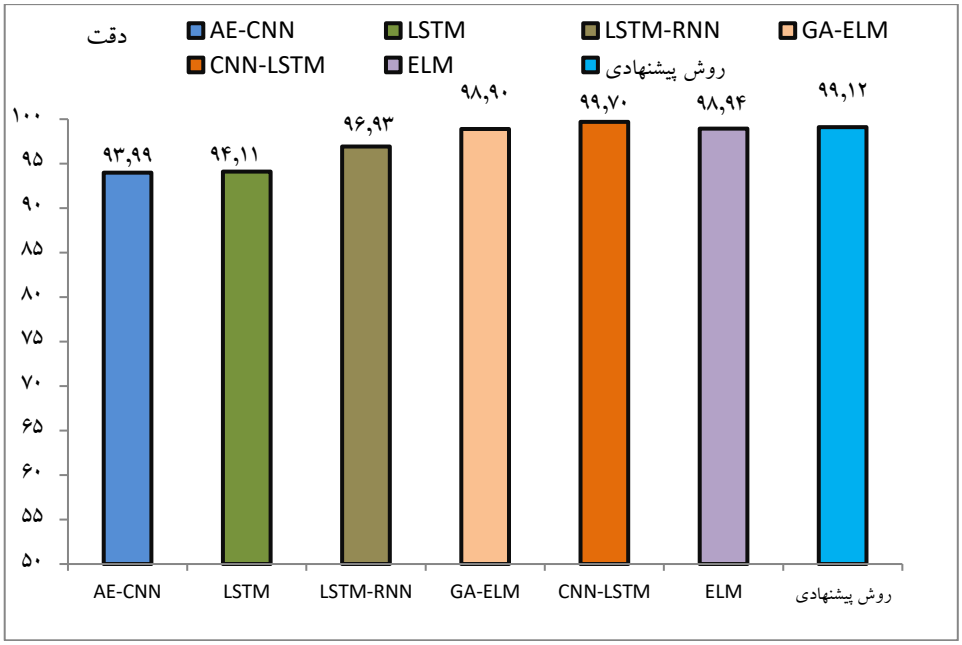
(شکل-۱۷): ارزیابی شاخص دقت، حساسیت و صحت روش پیشنهادی با کرنل‌های مختلف

(Figure-17): Evaluation of the index of accuracy, sensitivity and accuracy of the proposed method with different kernels



(شکل-۱۸): مقایسهٔ دقت تشخیص حملات در روش پیشنهادی با روش‌های انتخاب ویژگی

(Figure-18): Comparison of attack detection accuracy in the proposed method with feature selection methods



(شکل-۱۹): مقایسهٔ شاخص دقت روش پیشنهادی با چند روش یادگیری عمیق

(Figure-19): Comparison of the accuracy index of the proposed method with several deep learning methods

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

۵- نتیجه‌گیری و کارهای آینده

اینترنت اشیا در عصر مدرن ظهور و توسعهٔ فناوری‌های فرایند کسب‌وکار جدید را از طریق شبکه‌ای از رایانه‌ها و دستگاه‌هایی که قادر به برقراری ارتباط و تعامل با یکدیگرند، ممکن کرده‌است. در آنجا که تعداد حملات سایبری به سامانه‌های اینترنت اشیا به‌سرعت و به‌طور گسترده افزایش می‌یابد، افراد و کسب‌وکارها با طیف گسترده‌ای از چالش‌های مرتبط با اعتبار، تأمین مالی و عملیات تجاری مواجه‌اند. حملات به شبکهٔ اینترنت اشیا بسیار پیچیده شده‌است و می‌تواند سرویس‌های کاربردی شبکه را با اختلال مواجه سازد. در این شبکه، در بیشتر موارد تعداد زیادی شیء هوشمند به بدافزار آلوده می‌شود و هر کدام از آن‌ها علیه سرویس‌های شبکه اقدام به حمله می‌کنند. برای تشخیص حملات به شبکهٔ اینترنت اشیا نیاز به سامانه‌های هوشمند تشخیص نفوذ به شبکه است. سامانه‌های تشخیص نفوذ یکی از مؤثرترین روش‌ها برای محافظت از سامانه‌های IoT در برابر طیف وسیعی از حملات هستند.

امروزه تروش‌های یادگیری ماشین به‌طور گسترده برای بهبود عملکرد سامانهٔ تشخیص نفوذ استفاده شده‌است، اما هنوز نتوانسته‌اند به سطح مطلوبی از دقت دست یابند. برای اینکه سامانهٔ تشخیص نفوذ مؤثر باشد، نه‌تنها باید بتواند بین ترافیک قانونی و مخرب در طول فرایند تجزیه‌وتحلیل ترافیک شبکه تمایز قائل شود، بلکه باید قادر باشد تشخیص دهد که چه نوع حمله‌ای در سامانهٔ محافظت‌شده در حال انجام است. در این پژوهش برای تشخیص حملات به‌صورت دقیق و هوشمند یک سامانهٔ تشخیص نفوذ با هوش گروهی ارهماهی و ماشین بردار پشتیبان ارائه شده‌است. نقش الگوریتم ارهماهی در روش پیشنهادی، انتخاب ویژگی‌های مهم برای به‌کارگیری در ماشین بردار پشتیبان برای مقابله با نفوذ است. در روش پیشنهادی برای افزایش دقت ماشین بردار پشتیبان مجموعه‌داده با روش یادگیری عمیق GAN متعادل‌سازی می‌شود و سپس برای افزایش دقت مدل، پارامترهای SVM به‌وسیلهٔ الگوریتم AOA بهینه‌سازی می‌شود. آزمایش‌ها نشان داد پیشنهادی با کرنل گوسین دارای کارایی بیشتری و دقت روش دقت، حساسیت و صحت به‌ترتیب برابر ۹۹.۱۲ درصد، ۹۸.۹۲ درصد و ۹۸.۹۶ درصد است. روش پیشنهادی در تشخیص حملات از درخت تصمیم‌گیری، جنگل تصادفی و شبکهٔ عصبی چندلایه دقت بیشتری دارد. روش پیشنهادی در شاخص دقت نسبت به روش‌های یادگیری عمیق AE-CNN، CNN-LSTM، GA-ELM، LSTM-RNN، LSTM و ELM دقت بیشتری دارد. روش پیشنهادی نسبت به چند روش انتخاب ویژگی مانند الگوریتم ژنتیک و الگوریتم گرگ خاکستری دارای دقت بیشتری است. استفاده از شبکهٔ عصبی LSTM و CNN به جای شبکهٔ MLP برای تشخیص حملات از پژوهش‌ها و کارهای آیندهٔ ما است.

۶-مراجع 6-Reference

[1] Z. Abou El Houda, B. Brik, and L. Khoukhi, "Why Should I Trust Your IDS?": An Explainable Deep Learning Framework for Intrusion Detection Systems in Internet of Things Networks," *IEEE Open Journal of the Communications Society*, vol. 3, pp. 1164–1176, 2022.

[2] M. M. Rashid et al., "Adversarial Training for Deep Learning-based Cyberattack Detection in IoT-based Smart City Applications," *Computers & Security*, p. 102783, 2022.

[3] N. Al-Taleb and N. A. Saqib, "Towards a Hybrid Machine Learning Model for Intelligent Cyber Threat Identification in Smart City Environments," *Applied Sciences*, vol. 12, no. 4, p. 1863, 2022.

[4] R. Zhao, Y. Mu, L. Zou, and X. Wen, "A Hybrid Intrusion Detection System Based on Feature Selection and Weighted Stacking Classifier," *IEEE Access*, vol. 10, pp. 71414–71426, 2022.

[5] E. Mahdavi, A. Fanian, A. Mirzaei, and Z. Taghiyarrenani, "ITL-IDS: Incremental Transfer Learning for Intrusion Detection Systems," *Knowledge-Based Systems*, vol. 253, p. 109542, 2022.

[6] A. K. Zamani and A. Chapnevis, "BotNet Intrusion Detection System in Internet of Things with Developed Deep Learning," *arXiv preprint*, arXiv:2207.04503, 2022.

[7] A. A. R. Melvin et al., "Dynamic malware attack dataset leveraging virtual machine monitor audit data for the detection of intrusions in cloud," *Trans. Emerging Telecommunications Technologies*, vol. 33, no. 4, e4287, 2022.

[8] K. Malik et al., "Lightweight Internet of Things Botnet Detection Using One-Class Classification," *Sensors*, vol. 22, no. 10, p. 3646, 2022.

[9] D. B. Mandru et al., "Assessing deep neural network and shallow for network intrusion detection systems in cyber security," in *Computer Networks and Inventive Communication Technologies*, Springer, Singapore, 2022, pp. 703–713.

[10] Z. Rustama and N. P. A. A. Ariantari, "Comparison between Support Vector

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

Machine and Fuzzy Kernel C-Means as Classifiers for Intrusion Detection System using Chi-Square Feature Selection," in *AIP Conf. Proc.*, vol. 20214, no. 2018, 2023.

[11] T. Wu et al., "Intrusion detection system combined enhanced random forest with SMOTE algorithm," *EURASIP J. Adv. Signal Process.*, vol. 2022, no. 1, pp. 1–20, 2022.

[12] M. Jeyaselvi et al., "A highly secured intrusion detection system for IoT using EXPSO-STFA feature selection for LAANN to detect attacks," *Cluster Computing*, pp. 1–16, 2022.

[13] D. Aksu and M. A. Aydin, "MGA-IDS: Optimal feature subset selection for anomaly detection framework on in-vehicle networks-CAN bus based on genetic algorithm and intrusion detection approach," *Computers & Security*, vol. 118, p. 102717, 2022.

[14] M. Ajdani, A. Noori, and H. Ghaffary, "Providing a Consistent Method to Model the Behavior and Modelling Intrusion Detection Using A Hybrid Particle Swarm Optimization-Logistic Regression Algorithm," *Security and Communication Networks*, 2022.

[15] S. Shadravan, H. R. Naji, and V. K. Bardsiri, "The Sailfish Optimizer: A novel nature-inspired metaheuristic algorithm for solving constrained engineering optimization problems," *Eng. Appl. Artif. Intell.*, vol. 80, pp. 20–34, 2019.

[16] O. Ali et al., "A Comprehensive Review of Internet of Things: Technology Stack, Middlewares, and Fog/Edge Computing Interface," *Sensors*, vol. 22, no. 3, p. 995, 2022.

[17] F. Hussain et al., "A Two-Fold Machine Learning Approach to Prevent and Detect IoT Botnet Attacks," *IEEE Access*, vol. 9, pp. 163412–163430, 2021.

[18] S. M. Sajjad et al., "Detection and Blockchain-Based Collaborative Mitigation of Internet of Things Botnets," *Wireless Communications and Mobile Computing*, 2022.

[19] J. E. M. Díaz, "Internet of things and distributed denial of service as risk factors in information security," in *Bioethics in Medicine and Society*, IntechOpen, 2020.

[20] R. Vishwakarma and A. K. Jain, "A survey of DDoS attacking techniques and defence mechanisms in the IoT network," *Telecommunication Systems*, vol. 73, no. 1, pp. 3–25, 2020.

[21] J. Asharf et al., "A review of intrusion detection systems using machine and deep learning in internet of things: Challenges, solutions and future directions," *Electronics*, vol. 9, no. 7, p. 1177, 2020.

[22] Z. K. Maseer et al., "Benchmarking of machine learning for anomaly based intrusion detection systems in the CICIDS2017 dataset," *IEEE Access*, vol. 9, pp. 22351–22370, 2021.

[23] M. Almiani et al., "Deep recurrent neural network for IoT intrusion detection system," *Simulation Modelling Practice and Theory*, vol. 101, p. 102031, 2020.

[24] A. O. Alzahrani and M. J. Alenazi, "Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks," *Future Internet*, vol. 13, no. 5, p. 111, 2021.

[25] A. Davahli, M. Shamsi, and G. Abaei, "Hybridizing genetic algorithm and grey wolf optimizer to advance an intelligent and lightweight intrusion detection system for IoT wireless networks," *J. Ambient Intell. Humaniz. Comput.*, vol. 11, no. 11, pp. 5581–5609, 2020.

[26] R. Yao et al., "Intrusion detection system in the advanced metering infrastructure: a cross-layer feature-fusion CNN-LSTM-based approach," *Sensors*, vol. 21, no. 2, p. 626, 2021.

[27] E. Altulaihan, M. A. Almaiah, and A. Aljughaiman, "Anomaly Detection IDS for Detecting DoS Attacks in IoT Networks Based on Machine Learning Algorithms," *Sensors*, vol. 24, no. 2, p. 713, 2024.

[28] محمدی، شهریار، خلعتبری، احمد، باباگلی، مهدی، «ارائه یک مدل فراابتکاری تشخیص نفوذ به کمک انتخاب ویژگی مبتنی بر بهینه سازی گرگ خاکستری بهبودیافته و جنگل تصادفی»، *فصلنامهٔ پردازش علائم و داده‌ها*، دورهٔ ۲۰، شمارهٔ ۱، صص ۱۳۳-۱۴۴، ۱۴۰۲.

[28] Sh. Mohammadi, A. Khalatbari, and M. Babagoli, "Proposing a Meta-heuristic Model of Intrusion Detection Using feature Selection Based on Improved Gray Wolf Optimization and Random Forest," *Signal Data Processing*, pp. 133–144, 2023.

[29] تیموری، احمد، دی پیر، محمود، «سامانه دو سطحی تشخیص نفوذ برای شبکه اینترنت اشیا مبتنی بر یادگیری عمیق»، *فصلنامهٔ پردازش علائم و داده‌ها*، دورهٔ ۲۱، شمارهٔ ۳، صص۳-۲۲، ۱۴۰۱.

[29] A. Teymoori and M., Deypir "Two-level intrusion detection system for Internet of Things network based on deep learning," *Signal Data Process.*, vol. 3, no. 1, pp. 3–22, 2024.

[30] S. S. Kareem et al., "An effective feature selection model using hybrid metaheuristic algorithms for iot intrusion detection," *Sensors*, vol. 22, no. 4, p. 1396, 2022.

[31] K. Ren, Y. Zeng, Z. Cao, and Y. Zhang, "ID-RDRL: a deep reinforcement learning-based feature selection intrusion detection model," *Scientific Reports*, vol. 12, no. 1, p. 15370, 2022.

[32] J. Figueiredo, C. Serrão, and A. M. de Almeida, "Deep learning model transposition for network intrusion detection systems," *Electronics*, vol. 12, no. 2, p. 293, 2023.

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]

[33] A. Abdelkhalek and M. Mashaly, "Addressing the class imbalance problem in network intrusion detection systems using data resampling and deep learning," *The Journal of Supercomputing*, vol. 79, no. 10, pp. 10611–10644, 2023.
[34] P. Sanju, "Enhancing intrusion detection in IoT systems: A hybrid metaheuristics-deep learning approach with ensemble of recurrent neural networks," *Journal of Engineering Research*, vol. 11, no. 4, pp. 356–361, 2023.
[35] V. Hnamte and J. Hussain, "Dependable intrusion detection system using deep convolutional neural network: A novel framework and performance evaluation approach," *Telematics and Informatics Reports*, vol. 11, p. 100077, 2023.
[36] R. A. Elsayed, R. A. Hamada, M. I. Abdalla, and S. A. Elsaid, "Securing IoT and SDN systems using deep-learning based automatic intrusion detection," *Ain Shams Engineering Journal*, vol. 14, no. 10, p. 102211, 2023.
[37] M. Nanjappan et al., "DeepLG SecNet: utilizing deep LSTM and GRU with secure network for enhanced intrusion detection in IoT environments," *Cluster Computing*, pp. 1–13, 2024.
[38] E. Osa, P. E. Orukpe, and U. Iruansi, "Design and implementation of a deep neural network approach for intrusion detection systems," *e-Prime – Advances in Electrical Engineering, Electronics and Energy*, vol. 7, p. 100434, 2024.
[39] S. S. Shankar et al., "A novel optimization based deep learning with artificial intelligence approach to detect intrusion attack in network system," *Education and Information Technologies*, vol. 29, no. 4, pp. 3859–3883, 2024.
[40] R. Devendiran and A. V. Turukmane, "Dugat-LSTM: Deep learning based network intrusion detection system using chaotic optimization strategy," *Expert Systems with Applications*, vol. 245, p. 123027, 2024.
[41] F. A. Hashim et al., "Archimedes optimization algorithm: a new metaheuristic algorithm for solving optimization problems," *Applied Intelligence*, vol. 51, pp. 1531–1551, 2021.
[42] G. Eom and H. Byeon, "Searching for Optimal Oversampling to Process Imbalanced Data: Generative Adversarial Networks and Synthetic Minority Over-Sampling Technique," *Mathematics*, vol. 11, no. 16, p. 3605, 2023.
[43] J. Tanha and Z. Zarei, "The Bombus-terrestris bee optimization algorithm for feature selection," *Applied Intelligence*, vol. 53, no. 1, pp. 470–490, 2023.



زهرا سرحدی تحصیلات کارشناسی مهندسی کامپیوتر نرم‌افزار را در دانشگاه سیستان‌وبلوچستان و کارشناسی‌ارشد مهندسی کامپیوتر نرم‌افزار را در دانشگاه آزاد اسلامی واحد زاهدان به پایان رساند. وی هم اکنون دانشجوی دکترای مهندسی کامپیوتر نرم‌افزار دانشگاه آزاد واحد بیرجند است. زمینهٔ پژوهشی ایشان شبکهٔ حس‌گر بیسیم، یادگیری عمیق، شبکهٔ عصبی و هوش مصنوعی و الگوریتم‌های فراابتکاری است.
نشانی رایانامهٔ ایشان عبارت‌است از:
Z.sarhadi@iau.ac.ir



مهدی خزاعی پور تحصیلات کارشناسی مهندسی کامپیوتر را در دانشگاه آزاد اسلامی مشهد و کارشناسی‌ارشد را در دانشگاه آزاد اسلامی واحد علوم‌وتحقیقات تهران و دکترا را در دانشگاه آزاد کرمان به پایان رسانده‌است. ایشان عضو هیأت علمی دانشگاه آزاد اسلامی واحد بیرجند و مدیرگروه دانشکدهٔ کامپیوتر این دانشگاه هستند. زمینهٔ پژوهشی ایشان شبکهٔ عصبی، هوش مصنوعی و یادگیری عمیق است.
نشانی رایانامهٔ ایشان عبارت‌است از:
Mkhazaiepoor@iau.ir

[Downloaded from jsdp.rcisp.ac.ir on 2026-11-25]

[DOI: 10.61882/jsdp.22.2.43]