electronics

IoT Dataset Validation Using Machine Learning Techniques for Traffic Anomaly Detection

Laura Vigoya * , Diego Fernandez , Victor Carneiro and Francisco J. Nóvoa

Centre for Information and Communications Technology Research (CITIC), Campus de Elviña s/n, 15071 A Coruña, Spain; dfernandezi@udc.es (D.F.); victor.carneiro@udc.es (V.C.); fjnovoa@udc.es (F.J.N.) * Correspondence: l.v.vigoya@udc.es; Tel.: +34-881-011-213

Abstract: With advancements in engineering and science, the application of smart systems is increasing, generating a faster growth of the IoT network traffic. The limitations due to IoT restricted power and computing devices also raise concerns about security vulnerabilities. Machine learning-based techniques have recently gained credibility in a successful application for the detection of network anomalies, including IoT networks. However, machine learning techniques cannot work without representative data. Given the scarcity of IoT datasets, the DAD emerged as an instrument for knowing the behavior of dedicated IoT-MQTT networks. This paper aims to validate the DAD dataset by applying Logistic Regression, Naive Bayes, Random Forest, AdaBoost, and Support Vector Machine to detect traffic anomalies in IoT. To obtain the best results, techniques for handling unbalanced data, feature selection, and grid search for hyperparameter optimization have been used. The experimental results show that the proposed dataset can achieve a high detection rate in all the experiments, providing the best mean accuracy of 0.99 for the tree-based models, with a low false-positive rate, ensuring effective anomaly detection

Keywords: IoT; sensors; dataset validation; machine learning; intrusion detection systems; analysis; metric; algorithm design

electronics

اعتبارسنجی مجموعه‌داده اینترنت اشیا با استفاده از روش‌های یادگیری ماشین برای تشخیص ناهنجاری ترافیک

لائورا ویگویا*، دیگو فرناندز، ویکتور کارنیرو و فرانسیسکو خ. نووآ

استناد: ویگویا، ل.؛ فرناندز، د.؛ کارنیرو، و.؛ نووآ، ف.خ. اعتبارسنجی مجموعه‌داده اینترنت اشیا با استفاده از روش‌های یادگیری ماشین برای تشخیص ناهنجاری ترافیک. الکترونیکس ۲۰۲۱، ۱۰، ۲۸۵۷.
https://doi.org/10.3390/electronics10222457
سردبیر علمی: دیمیترا ای. کاکلامانی
دریافت: ۲۹ ژوئن ۲۰۲۱
پذیرش: ۱۶ نوامبر ۲۰۲۱
انتشار: ۱۹ نوامبر ۲۰۲۱
یادداشت ناشر: انتشارات MDPI در مورد ادعاهای حوزه‌ای موجود در نقشه‌های منتشرشده و وابستگی‌های مؤسسه‌ای بی‌طرف باقی می‌ماند.

مرکز تحقیقات فناوری اطلاعات و ارتباطات(CITIC) ، کمپوس الوینیا، شماره‌ای، ۱۵۰۷۱ آ کورونیا، اسپانیا؛ dfernandezi@udc.es(د.ف.)؛ victor.carneiro@udc.es(و.ک.)؛ fjnovoa@udc.es ف.خ.ن(.)

- مکاتباتl.v.vigoya@udc.es :؛ تلفن: +۲۱۳-۰۱۱-۸۸۱-۳۴

چکیده: با پیشرفت‌های حاصل در مهندسی وعلوم، کاربرد سیستم‌های هوشمند در حال افزایش است و رشد سریع‌تری در ترافیک شبکه اینترنت اشیا (IoT) ایجاد می‌کند. محدودیت‌های ناشی از توان و منابع محاسباتی محدود دستگاه‌های IoT نیز دغدغه‌هایی در مورد آسیب‌پذیری‌های امنیتی ایجاد کرده است. روش‌های مبتنی بر یادگیری ماشین (ML) اخیراً در کاربردهای موفقیت‌آمیز برای تشخیص ناهنجاری‌های شبکه، از جمله شبکه‌هایIoT ، معتبر شده‌اند. با این حال، روش‌های یادگیری ماشین بدون داده‌های نماینده نمی‌توانند عمل کنند. با توجه به کمبود مجموعه‌داده‌های اختصاصیIoT ، مجموعه‌داده DAD به‌عنوان ابزاری برای شناخت رفتار شبکه‌های اختصاصی IoT-MQTT مطرح شده است. این مقاله با هدف اعتبارسنجی مجموعه‌داده DAD با به‌کارگیری رگرسیون لجستیک (Logistic Regression)، ناوِی بیز(Naive Bayes) ، جنگل تصادفی(Random Forest) ، آدابوست (AdaBoost) و ماشین بردار پشتیبان (Support Vector Machine) برای تشخیص ناهنجاری‌های ترافیکی در IoT ارائه شده است. برای به‌دست آوردن بهترین نتایج، از تکنیک‌های مدیریت داده‌های نامتوازن، انتخاب ویژگی و جستجوی شبکه‌ای (grid search) برای بهینه‌سازی هایپرپارامترها استفاده شده است. نتایج آزمایش‌ها نشان می‌دهد که مجموعه‌داده پیشنهادی می‌تواند نرخ تشخیص بالایی را در تمام آزمایش‌ها به‌دست آورد و بهترین دقت میانگین ۰.۹۹ را برای مدل‌های مبتنی بر درخت، همراه با نرخ کم خطای مثبت کاذب(false-positive rate)، فراهم آورد که این امر یک تشخیص مؤثر از ناهنجاری‌ها را تضمین می‌کند.

واژگان کلیدی: اینترنت اشیا(IoT) ؛ حسگرها؛ اعتبارسنجی مجموعه‌داده؛ یادگیری ماشین؛ سیستم‌های تشخیص نفوذ؛ تحلیل؛ معیار؛ طراحی الگوریتم

۱ .مقدمه

با توجه به رشد و پیاده‌سازی گسترده‌ی شبکه‌های اینترنت اشیا (IoT) در حوزه‌های مختلف، این فناوری نقش چشمگیری در فعالیت‌های روزمره‌ی ما ایفا می‌کند و به عنوان نیروی محرکه‌ی خانه‌های هوشمند، شهرهای هوشمند، سیستم‌های سلامت نوین و تولید پیشرفته تحول یافته است [۱]. با این حال، با پیشرفت‌های حاصل در مهندسی و علوم، ابعاد کاربردی سیستم‌های هوشمند در حال افزایش است که رشد سریع‌تری در ترافیک شبکه‌ی IoT ایجاد می‌کند و همین امر دغدغه‌هایی در مورد آسیب‌پذیری‌ها و محدودیت‌های امنیتی نیز به همراه دارد. اینترنت و کاربران آن از پیش تحت مداومت حملاتی هستند که می‌توانند به‌صورت ناهنجاری‌های ترافیکی خود را نشان دهند و تهدیدی برای IoT به‌عنوان مجموعه‌ای از دستگاه‌های محدود محسوب شوند. این واقعیت می‌تواند دیدگاه‌های متفاوتی را ایجاد کند که به تولید الگوهای مخرب جدید و خلاقانه منجر شود. چالش اصلی جلوگیری از گسترش این الگوها یا حداقل کاهش و محدود کردن تأثیر آن‌هاست [۲].

ناهنجاری‌های شبکه همیشه قابل دسته‌بندی به‌عنوان یک حمله نیستند و لزوماً عناصر مضر محسوب نمی‌شوند؛ با این وجود، بینش‌های مهمی را درباره‌ی رفتار ترافیک فراهم می‌کنند و می‌توانند در شناسایی اطلاعات مهم و حیاتی در کاربردهای مختلف کمک‌کننده باشند [۳]. یکی از راه‌های تشخیص تغییرات در رفتار شبکه، استفاده از سیستم‌های تشخیص نفوذ (IDSs) است که به کشف، تعیین و شناسایی استفاده‌های غیرمجاز، تکثیر، دستکاری و تخریب سیستم‌های اطلاعاتی کمک می‌کنند. روش‌های سنتی IDS به دلیل ویژگی‌های خاص IoT—مانند انرژی محدود، همه‌گیری (ubiquity)، ناهمگونی، پهنای‌باند محدود و اتصال‌پذیری جهانی—برای امنیت سیستم‌های IoT کمتر مؤثر یا ناکافی هستند [۴]. در واقع، IoT نیازمند استانداردها و پروتکل‌های ارتباطی تخصصی برای مقابله با چالش‌های ناشی از این ویژگی‌هاست. در لایه انتقال، پروتکل‌های TCP و UDP برای اکثر کاربردها غالب هستند. با این حال، این توابع باید به شیوه‌های استاندارد و قابل همکاری پیاده‌سازی شوند و بسته به نیازهای کاربردی IoT، توابع توزیع پیام‌های مختلفی مورد نیاز است. پروتکل MQTT اغلب برای انتشار و/یا اشتراک‌گذاری پیام‌های دریافتی از دستگاه‌هایی با پهنای‌باند پایین در شبکه‌های غیرقابل اعتماد به کار می‌رود. MQTT از طریق TCP برای ارسال و دریافت داده‌های لایه‌ی واسط (middleware) به مراکز IoT عمل می‌کند.

یک جنبه‌ی حیاتی این است که یادگیری ماشین بدون داده‌های نماینده نمی‌تواند عمل کند و صحت داده‌های جمع‌آوری‌شده از دستگاه‌ها برای ساخت سیستم‌های تصمیم‌گیری هوشمند و همچنین مدیریت مؤثر محیط‌های IoT ضروری است. روش‌های نظارت‌شده (که همچنین به‌عنوان روش‌های طبقه‌بندی شناخته می‌شوند) به یک مجموعه آموزشی برچسب‌دار نیاز دارند که شامل نمونه‌های عادی و ناهنجار باشد تا مدل پیش‌بینی‌کننده ساخته شود. از نظر نظری، روش‌های نظارت‌شده نرخ تشخیص بهتری را نسبت به روش‌های نیمه‌نظارت‌شده و بدون نظارت ارائه می‌دهند، زیرا به اطلاعات بیشتری دسترسی دارند. رایج‌ترین الگوریتم‌های نظارت‌شده، شبکه‌های عصبی (Neural Networks)، ماشین‌های بردار پشتیبان (Support Vector Machines)، k نزدیک‌ترین همسایه (k-Nearest Neighbors)، شبکه‌های بیزی (Bayesian Networks) و درخت‌های تصمیم (Decision Trees) هستند [۵].

۱.۱ .انگیزه

به دلیل تنظیمات درهم‌تنیده و وابسته‌ی اینترنت اشیا (IoT)، این محیط مستعد مسائل امنیتی و حریم خصوصی مختلفی است که ممکن است منجر به انجام وظایف غیرمجاز توسط کاربران مخرب از راه دور شود [۶] و در نتیجه، رفتار غیرعادی در شبکه ایجاد کند. در موارد دیگر، ترافیک ناهنجار در یک شبکه‌ی IoT ممکن است ناشی از پیکربندی اشتباه، نصب غیرمعمول یا خرابی سخت‌افزاری حسگرها باشد.

در نتیجه، نیاز به تحقیق درباره‌ی روش‌های تشخیص و پیشگیری خاص و طراحی و توسعه‌ی راه‌حل‌های امنیتی هوشمند برای محافظت از دستگاه‌های IoT آسیب‌پذیر و در برابر ناهنجاری‌های ناشی از دستگاه‌های IoT فشرده‌شده وجود دارد. ایجاد سیستم‌های تشخیص ناهنجاری (IDSs) نیازمند دانش نماینده از محیط است.

با توجه به کمبود مجموعه‌داده‌های IoT، مجموعه‌داده‌ی DAD [۷] به‌عنوان یک مکانیزم برای شناخت رفتار دستگاه‌های سبک‌وزن و اختصاصی در شبکه‌های IoT-MQTT مطرح شده است. مجموعه‌داده‌ی DAD [۷] یک مجموعه‌داده‌ی کامل و برچسب‌گذاری‌شده برای تشخیص ناهنجاری ترافیک در دنیای واقعی است که در شرایط مناسب، با حجم کافی ردپا (trace)، سناریوهای متنوع ناهنجاری و تحلیل پیشین ارائه شده و قرار است در الگوریتم‌های یادگیری ماشین (ML) به کار گرفته شود. با این حال، این مجموعه‌داده باید تحت کاربرد مدل‌های یادگیری ماشین قرار گیرد تا قابلیت استفاده‌ی آن برای هدفی که برای آن طراحی شده است، تأیید شود.

۱.۲. سهم‌های این پژوهش

این پژوهش با هدف نشان دادن این موضوع انجام شده است که مجموعه‌داده‌ی DAD را می‌توان برای تشخیص ناهنجاری‌ها در شبکه‌های ترافیکی MQTT-IoT به کار برد و برای این منظور از پنج روش رایج یادگیری سطحی (shallow learning) برای تشخیص ناهنجاری ترافیک در محیط‌های IoT استفاده شده است. سهم‌های اصلی این پژوهش عبارتند از:

- تحلیل و استخراج ویژگی‌ها برای شبکه‌های حسگر بی‌سیمIoT ؛
- پیاده‌سازی تکنیک‌های آماده‌سازی داده، شامل گروه‌بندی(binning) ، مدیریت داده‌های نامتوازن و استخراج ویژگی در محیط‌های واقعی؛
- به‌کارگیری الگوریتم‌های مختلف یادگیری ماشین برای اعتبارسنجی مجموعه‌داده‌ی انتخاب‌شده؛
- استفاده از معیارها و مقایسه‌ی عملکرد روش‌های مختلف یادگیری ماشین در تشخیص ناهنجاری‌های ترافیکی.

۱.۳. ساختار مقاله

این پژوهش به‌صورت زیر سازمان‌دهی شده است: مروری بر سیستم‌های تشخیص نفوذ (IDSs) مبتنی بر روش‌های یادگیری ماشین برای شبکه‌های اینترنت اشیا (IoT) در بخش ۲ ارائه می‌شود. بخش ۳ تعریف مختصری از الگوریتم‌های انتخاب‌شده را همراه با پارامترهای در نظر گرفته‌شده برای تنظیم هایپرپارامترها ارائه می‌دهد. بخش ۴ شرح مختصری از سناریوی مورد استفاده را ارائه می‌کند و داده‌های مربوط به مجموعه‌داده را توصیف، بررسی و کیفیت آن‌ها را تأیید می‌نماید. سپس، ارتقای کیفیت داده‌ها به سطح مورد نیاز توسط روش‌های یادگیری ماشین انتخاب‌شده ضروری است. بخش ۵ تکنیک‌ها و فرآیندهای لازم برای این‌که داده‌ها به‌صورت کامل برای الگوریتم‌های یادگیری ماشین قابل فهم باشند را معرفی می‌کند. بخش ۶ نتایج به‌دست‌آمده برای هر یک از مدل‌ها در مرحله آموزش را تحلیل می‌کند. در ادامه، بخش ۷ نتایج به‌دست‌آمده در مرحله آزمون را بررسی کرده و همچنین مقایسه‌ای میان مدل‌ها انجام می‌دهد. در نهایت، بخش ۸ نتیجه‌گیری‌های اصلی این پژوهش را ارائه می‌دهد.

۲. کارهای مرتبط

پیاده‌سازی نسبتاً ساده و نتایج حاصل از سیستم‌های تشخیص نفوذ (IDSs) مبتنی بر الگوریتم‌های یادگیری ماشین، باعث شده است که بسیاری از پژوهشگران این روش‌ها را به‌عنوان ابزاری برای توسعه پروژه‌های خود انتخاب کنند. با این حال، کاربرد IDS در محیط IoT به‌تازگی به‌صورت گسترده‌ای مورد توجه قرار گرفته است. این بخش مروری مختصر بر مهم‌ترین روش‌های یادگیری ماشین به‌کاررفته برای اعتبارسنجی مجموعه‌داده‌های IoT در سال‌های اخیر ارائه می‌دهد.

با توجه به کمبود اولیه مجموعه‌داده‌های عمومی اختصاصی IoT، کاربرد IDS در این محیط با تطبیق مجموعه‌داده‌های عمومی که توسط جامعه پذیرفته شده بودند، بر روی ردپاهای شبکه IoT آغاز شد. متداول‌ترین مجموعه‌داده به‌کاررفته در IoT، مجموعه‌داده UNSW-NB15 [۸] بوده است که پژوهشگران متعددی کاربردهای یادگیری ماشین خود را بر این مجموعه‌داده آزمایش کرده‌اند. کورونیوتیس و همکاران [۹] از چهار روش طبقه‌بندی، شامل درخت تصمیم (DT)، داده‌کاوی مبتنی بر قواعد انجمنی (ARM)، شبکه عصبی مصنوعی (ANN) و ناوِی بیز (NB) استفاده کردند تا توانایی الگوریتم‌ها در شناسایی بردارهای حمله را اعتبارسنجی نمایند. AD-IoT [۱۰] نیز بخشی از مجموعه‌داده UNSW-NB15 را انتخاب کرده و سیستمی برای تشخیص ناهنجاری جهت شناسایی سایبرحمله‌ها در گره‌های فوگ یک شهر هوشمند با استفاده از روش‌های گروهی (ensemble) مبتنی بر جنگل تصادفی (RF) و درخت اضافی (ET) پیشنهاد داده است. علاوه بر این، مجموعه‌داده‌های UNSW-NB15 و NIMS botnet حاوی داده‌های حسگرهای IoT شبیه‌سازی‌شده توسط مصطفی و همکاران [۱] برای پیاده‌سازی یک سیستم تشخیص نفوذ شبکه (NIDS) مبتنی بر الگوریتم یادگیری گروهی AdaBoost با به‌کارگیری سه روش یادگیری ماشین—DT، NB و ANN—مورد استفاده قرار گرفت. از سوی دیگر، پاجووه و همکاران [۱۱] از مجموعه‌داده NSL-KDD استفاده کرده و مدلی را پیشنهاد دادند که از یک ماژول طبقه‌بندی دو سطحی متشکل از ناوِی بیز (NB) و نسخه ضریب اطمینان (Certainty Factor) الگوریتم k نزدیک‌ترین همسایه (K-Nearest Neighbor) برای شناسایی فعالیت‌های مخرب نظیر حملات کاربر به ریشه (U2R) و راه‌دور به محلی (R2L) بهره می‌برد. با گسترش کاربرد محیط‌های IoT، تقاضا برای سیستم‌های تشخیص نفوذ تخصصی افزایش یافت و در نتیجه، نیاز به انتشار مجموعه‌داده‌های اختصاصی که امکان اعتبارسنجی سیستم‌ها را فراهم کند، بیشتر شد. سیستم کیتسون (Kitsune) [۱۲] یک سیستم تشخیص نفوذ شبکه است که از یک گروه از شبکه‌های عصبی به‌نام اتوانکُدِر (autoencoder) برای تمایز الگوهای ترافیک عادی و غیرعادی استفاده می‌کند و با به‌کارگیری مجموعه‌داده N-BaIoT [۱۳] توسعه یافته است. مجموعه‌داده N-BaIoT ترافیک را از دو شبکه جمع‌آوری می‌کند: شبکه نظارت‌برویدئویی با دوربین‌های IP که در آن هشت نوع مختلف حمله علیه در دسترس‌بودن و یکپارچگی پیوندهای ویدئویی انجام شده است؛ و شبکه IoT شامل سه رایانه شخصی و نُه دستگاه IoT که با بدافزار بات‌نت میرای (Mirai) آلوده شده‌اند. علاوه بر این، عباسی [۱۴] از این مجموعه‌داده برای طبقه‌بندی با استفاده از رگرسیون لجستیک (LR) و شبکه عصبی مصنوعی (ANN) استفاده کرده است. دوشی و همکاران [۱۵] با تکیه بر این واقعیت که ترافیک IoT اغلب متفاوت از دستگاه‌های دیگر متصل به اینترنت است، یک خط‌لوله یادگیری ماشین را توسعه داده‌اند که ویژگی‌هایی را طراحی می‌کند تا از رفتارهای شبکه‌ای خاص IoT بهره ببرد. این خط‌لوله برای اجرا روی میان‌جعبه‌های شبکه (مثل مسیریاب‌ها، فایروال‌ها یا سوئیچ‌های شبکه) طراحی شده تا ترافیک ناهنجار و دستگاه‌های مرتبط که ممکن است بخشی از یک بات‌نت فعال باشند را شناسایی کند. در این پژوهش، نویسندگان طیفی از رده‌بندان را برای تشخیص حمله مقایسه کرده‌اند که شامل الگوریتم k نزدیک‌ترین همسایه مبتنی بر درخت (KNN) KDTree، ماشین بردار پشتیبان با کرنل خطی (LSVM)، درخت تصمیم با معیار ناخالصی جینی (DT)، جنگل تصادفی با معیار ناخالصی جینی (RF) و شبکه عصبی (NN) می‌شود و اثربخشی جنگل تصادفی را نشان می‌دهند. حسن و همکاران [۱۷] با استناد به مجموعه‌داده خاص IoT ارائه‌شده توسط پاهل و همکاران [۱۶]، یک سیستم یادگیری ماشین هوشمند، امن و قابل اعتماد را پیشنهاد داده‌اند که از LR، SVM، DT، RF و ANN استفاده می‌کند. زیرساخت مبتنی بر IoT از چندین رده‌بند استفاده می‌کند که می‌تواند سیستم را هنگامی که در حالت غیرعادی قرار دارد تشخیص دهد و از آن محافظت نماید. بهترین عملکرد در دقت آموزش و آزمون با RF و ANN به‌دست آمده است. متأسفانه، این مجموعه‌داده تنها شامل یک روز از ترافیک است. کورونیوتیس و همکاران [۱۸] که پیش‌تر در زمینه کاربرد یادگیری ماشین در سیستم‌های تشخیص نفوذ فعالیت داشتند، مجموعه‌داده خاصی را برای بات‌نت در IoT با نام Bot-IoT طراحی کردند تا اثربخشی روش‌ها در این محیط را تأیید کنند. ساختار این مجموعه‌داده هم ترافیک عادی مرتبط با IoT و هم ترافیک شبکه‌های دیگر را دربرمی‌گیرد و همچنین انواع مختلف ترافیک حمله رایج در بات‌نت‌ها را شامل می‌شود. این مجموعه‌داده دارای دسته‌ها و زیردسته‌های حمله برای امکان طبقه‌بندی چندکلاسه است. آن‌ها از الگوریتم‌های ماشین بردار پشتیبان (SVM)، شبکه عصبی بازگشتی (RNN) و شبکه عصبی بازگشتی حافظه بلندمدت کوتاه‌مدت

(LSTM-RNN) برای مقایسه اعتبار و ارزیابی دقت مجموعه‌داده استفاده کرده‌اند. به‌طور مشابه، نویسندگان دیگری مانند سوسیلو و همکاران [۱۹] و السامیری [۲۰] آزمایش‌های خود را بر اساس مجموعه‌داده Bot-IoT انجام داده‌اند. سوسیلو سیستم تشخیص ناهنجاری را با به‌کارگیری RF، SVM، MLP و CNN ساخته است، در حالی که السامیری از NB، RF، MLP، درخت دیکوتومیزه‌کننده تکرارشونده ۳ (ID3)، AdaBoost، تحلیل تشخیصی درجه دوم (QDA) و KNN استفاده کرده است. آنتی و همکاران [۲۱] یک سیستم تشخیص نفوذ سه‌لایه (IDS) ارائه داده‌اند که مجموعه‌داده‌ای نماینده تولید می‌کند و از روش نظارت‌شده برای تشخیص طیفی از سایبرحمله‌های رایج مبتنی بر شبکه در شبکه‌های IoT استفاده می‌کند. برای اعتبارسنجی سیستم‌های تشخیص نفوذ بالقوه، آن‌ها NB، شبکه بیزی، J48، Zero R، OneR، رگرسیون لجستیک ساده، SVM، MLP و RF را در نظر گرفته‌اند. با این حال، مجموعه‌داده به‌کاررفته از پروتکل‌های خاص IoT استفاده نکرده و تنها شامل دو روز از ترافیک ناهنجار است.MedBIoT [۲۲] یک مجموعه‌داده برچسب‌گذاری‌شده رفتاری از IoT ارائه می‌دهد که حاوی ترافیک عادی و مخرب بات‌نت است. سناریوی پیشنهادی یک زیرساخت IoT با اندازه متوسط (۸۳ دستگاه IoT) است که از ترکیب دستگاه‌های واقعی و شبیه‌سازی‌شده رایج IoT و سه بدافزار بات‌نت برجسته (Mirai، BashLite و Torii) تشکیل شده است. مدل‌های طبقه‌بندی یادگیری ماشین (KNN، SVM، DT و RF) پیاده‌سازی شده‌اند تا قابلیت استفاده از مجموعه‌داده پیشنهادی تأیید شود. تامارایسلوی و همکاران [۲۳] از الگوریتم‌هایی مانند RF، NB، SVM و DT برای تشخیص ناهنجاری در شبکه‌های IoT استفاده کرده‌اند. برای انجام آزمایش‌های خود، از مجموعه‌داده IoT-23 [۲۴] بهره برده‌اند که مجموعه‌داده بزرگی از بیست‌وسه ردپای مختلف ترافیک شبکه IoT ارائه می‌دهد. این سناریوها به بیست ردپای شبکه از دستگاه‌های IoT آلوده و سه ردپای شبکه از ترافیک واقعی دستگاه‌های IoT تقسیم شده‌اند. ضبط‌های بدافزار برای بازه‌های طولانی اجرا شده‌اند و سناریوهای حمله متنوعی از جمله Mirai، Torii و Gagfyt را اجرا کرده‌اند. در [۲۵] کاربردی برای IIoT یافت شده است. الگوریتم‌های تشخیص ناهنجاری مبتنی بر یادگیری ماشین برای یافتن ترافیک مخرب در یک مجموعه‌داده تولیدشده مصنوعی از ارتباط Modbus/TCP در یک سناریوی صنعتی ساختگی به کار گرفته شده‌اند. این مجموعه‌داده شامل حملات نفوذ رایج مبتنی بر خانه و دفتر است که رفتار زمان‌بندی و نرخ بسته‌ها در واحد زمان را شبیه‌سازی می‌کند که این امر عامل تمایز خوبی برای حملات محسوب می‌شود. چهار الگوریتم یادگیری ماشین مختلف برای تشخیص ناهنجاری به کار گرفته شده‌اند: SVM، RF، KNN و خوشه‌بندی k-means. بهترین نتایج با ماشین بردار پشتیبان (SVM) به‌دست آمده است. لیو و همکاران [۲۶] یک مجموعه‌داده عمومی با استفاده از حسگرهای هوشمند در یک شبکه IoT ایجاد کرده‌اند. داده‌ها در محیط آزمایشگاه هوشمند و خانه هوشمند با استفاده از بوردهای حسگر Rainbow HAT نصب‌شده روی رزبری‌پای‌ها جمع‌آوری شده‌اند. این مجموعه‌داده حاوی پنج نوع سایبرحمله است: مسمومسازی پروتکل ARP (ARP Poisoning)، حمله سرکوب سرویس ARP (ARP DoS)، سیل UDP (UDP Flood)، حمله فورس‌بروت هیدرا با پروتکل Asterisk و SlowLoris. آن‌ها روشی ترکیبی پیشنهاد داده‌اند که یک مدل جاسازی‌شده برای انتخاب ویژگی را با یک شبکه عصبی کانولوشنی (CNN) برای طبقه‌بندی حمله ترکیب می‌کند. روش تشخیص نفوذ پیشنهادی دو مدل دارد: (الف) RCNN که در آن جنگل تصادفی با CNN ترکیب شده است؛ و (ب) XCNN که در آن XGBoost با CNN ترکیب شده است. یک تکنیک ترکیبی که تلفیقی از رایانش فوگ و رایانش ابری است، به‌کار گرفته شده است. همچنین، آن‌ها طبقه‌بندی‌ها را روی CCD-INID-V1 و دو مجموعه‌داده دیگر IoT، یعنی N_BaIoT [۱۳] و CIRA-CIC-DoHBrw-2020 (DoH20) [۲۷] بررسی کرده‌اند تا اثربخشی این مدل‌های امنیتی مبتنی بر یادگیری را بررسی کرده و کارایی روش پیشنهادی خود را با الگوریتم‌های سنتی یادگیری ماشین مانند KNN، NB، LR و SVM مقایسه نموده و نتایج مقایسه‌ای طبقه‌بندی ناهنجاری و چندکلاسه را ارائه دهند. واکاری و همکاران مجموعه‌داده MQTTset [۲۸] را ارائه داده‌اند که بر پروتکل MQTT متمرکز است و از دستگاه‌های IoT با ماهیت‌های مختلف (مانند حسگرهای دما، رطوبت، حرکت و غیره) تشکیل شده است تا محیط خانه/دفتر/ساختمان هوشمند را شبیه‌سازی کند. MQTTset ترافیک قانونی را با انواع مختلف ترافیک مخرب/حمله‌ای نظیر حملات سرکوب سرویس با سیل، سیل انتشار MQTT، SlowITe، داده‌های نامناسب و احراز هویت با فورس‌بروت ترکیب می‌کند تا زمینه‌های مختلفی از جمله اتوماسیون خانگی، نظارت بر زیرساخت‌های حیاتی یا

محیط‌های صنعتی را شبیه‌سازی نماید. بر اساس این سناریو، آن‌ها یک سیستم تشخیص نفوذ پیاده‌سازی کرده‌اند که از DT، RF، MLP، NN و ناوِی بیز گاوسی برای شناسایی رفتارهای مخرب و در نتیجه محافظت از سیستم در برابر حملات بهره می‌برد. بیشتر کارهای انجام‌شده در سیستم‌های تشخیص نفوذ شبکه (NIDS) به دنبال استخراج ویژگی‌ها و مجموعه‌ای از ترکیب الگوریتم‌ها هستند که نتایج را بهینه کنند. سرهان و همکاران [۲۹] در پژوهش خود تلاش کردند تا روش‌ها را استانداردسازی کنند تا بتوان آن‌ها را بر هر مجموعه‌داده‌ای اعمال نمود. برای آزمایش‌های خود، آن‌ها از شش مدل یادگیری ماشین استفاده کردند: شبکه پیشخور عمیق (DFF)، شبکه عصبی کانولوشنی (CNN)، شبکه عصبی بازگشتی (RNN)، درخت تصمیم (DT)، رگرسیون لجستیک (LR) و ناوِی بیز (NB). همچنین آن‌ها سه الگوریتم استخراج ویژگی—تجزیه و تحلیل مؤلفه‌های اصلی (PCA)، تحلیل تشخیصی خطی (LDA) و اتوانکُدِر (AE)—را بر روی سه مجموعه‌داده معیار [UNSW-NB15 ۸]، ToN-IoT [۳۰] و CSE-CIC-IDS2018 [۳۱] بررسی نمودند. نتایج به‌دست‌آمده نشان می‌دهند که روش استخراج ویژگی یا مدل یادگیری ماشین خاصی وجود ندارد که بتواند بهترین نتایج را برای تمام مجموعه‌داده‌ها کسب کند و انتخاب مجموعه‌داده به‌طور قابل‌توجهی عملکرد روش‌های به‌کاررفته را تحت تأثیر قرار می‌دهد. در ابتدا، روش‌های یادگیری ماشین بر مجموعه‌داده‌هایی اعمال شدند که متعلق به شبکه‌های اینترنت اشیا (IoT) نبودند و تنها با تعدیل ردپاها (traces) به این محیط، نشان داده شد که این روش‌ها می‌توانند برای تشخیص حملات در چنین سناریوهایی به‌کار گرفته شوند. ظهور مجموعه‌داده‌های اختصاصی IoT امکان مدل‌سازی محیط‌های واقعی را فراهم کرده و کاربرد بهینه الگوریتم‌های یادگیری ماشین را برای اعتبارسنجی مجموعه‌داده‌های موجود تضمین می‌کند. پس از مرور روش‌های اصلی به‌کاررفته برای تشخیص ناهنجاری ترافیک با استفاده از یادگیری ماشین در محیط‌های IoT، تصمیم گرفتیم تا پنج مدل مختلف یادگیری ماشین با ماهیت‌های متفاوت را برای اعتبارسنجی مجموعه‌داده DAD انتخاب کنیم: رگرسیون لجستیک (Logistic Regression)، ناوِی بیز (Naive Bayes)، جنگل تصادفی (Random Forest)، آدابوست (AdaBoost) و ماشین بردار پشتیبان (Support Vector Machine).

۳. روش‌های یادگیری ماشین

یادگیری ماشین به استخراج دانش از داده‌ها می‌پردازد. موفق‌ترین انواع الگوریتم‌های یادگیری ماشین، الگوریتم‌هایی هستند که با تعمیم از مثال‌های شناخته‌شده، فرآیندهای تصمیم‌گیری را خودکار می‌کنند. این روش به‌عنوان یادگیری نظارت‌شده شناخته می‌شود [۳۲]. از میان روش‌های متعدد یادگیری ماشین، برخی کمتر انعطاف‌پذیر یا محدودتر هستند؛ مدل‌های سطحی (shallow models) در این دسته قرار می‌گیرند، زیرا تنها می‌توانند طیف نسبتاً محدودی از اشکال را برای تخمین تولید کنند. به‌طور کلی، برای ساخت یک مدل طبقه‌بندی با دقت بالا، انتخاب یک الگوریتم یادگیری ماشین قدرتمند و همچنین تنظیم مناسب پارامترهای آن بسیار مهم است. بیشتر الگوریتم‌های یادگیری ماشین در صورتی که پارامترهایشان به‌درستی تنظیم شوند، نتایج بهینه‌ای حاصل می‌کنند. تنظیمات نامناسب پارامترها منجر به نتایج ضعیف طبقه‌بندی می‌شود. بهینه‌سازی پارامترها در صورت انجام دستی—به‌ویژه زمانی که الگوریتم یادگیری دارای پارامترهای زیادی باشد—می‌تواند بسیار زمان‌بر باشد. از این رو، جستجوی شبکه‌ای (grid search) در اصل یک جستجوی جامع بر پایه زیرمجموعه‌ای از فضای هایپرپارامترهاست. جستجوی شبکه‌ای با استفاده از تکنیک اعتبارسنجی متقاطع (CV) به‌عنوان معیار عملکرد، پارامترهای مدل را بهینه می‌کند. هدف، شناسایی ترکیب مناسبی از هایپرپارامترهاست تا رده‌بند بتواند داده‌های ناشناخته را به‌دقت پیش‌بینی کند. تکنیک اعتبارسنجی متقاطع می‌تواند از مشکل بیش‌برازش (overfitting) جلوگیری کند [۳۳]. همان‌طور که پیش‌تر اشاره شد، برای انجام آزمایش‌ها، پنج الگوریتم طبقه‌بندی نظارت‌شده و سطحی با ماهیت‌های مختلف انتخاب شدند: رگرسیون لجستیک، ناوِی بیز، جنگل تصادفی، آدابوست و ماشین بردار پشتیبان. در ادامه توضیح مختصری از هر الگوریتم ارائه می‌شود، همراه با شرح کوتاهی از هایپرپارامترهای در نظر گرفته‌شده برای تنظیم آن‌ها. اولین مدل یادگیری ماشین در نظر گرفته‌شده، رگرسیون لجستیک است که یک روش آماری برای تحلیل و پیش‌بینی یک خروجی دودویی محسوب می‌شود. این مدل یادگیری ماشین حالت خاصی از مجموعه‌ای از مدل‌های خطی تعمیم‌یافته است. هدف رگرسیون لجستیک، ایجاد بهترین مدل برازشی برای برقراری رابطه وابستگی بین

متغیر کلاس و ویژگی‌هاست [۳۴]. این مدل در واقع هایپرپارامترهای حیاتی زیادی برای تنظیم ندارد. با این حال، پارامترهای L1، L2 و C را می‌توان به‌عنوان تکنیک‌های نظم‌دهی (regularization) برای مقابله با بیش‌برازش تنظیم کرد. تکنیک نظم‌دهی L1 مربوط به مدل رگرسیون لاسو (Lasso Regression) است، در حالی که L2 بر مدل رگرسیون ریج (Ridge Regression) اعمال می‌شود. به‌طور خلاصه، نظم‌دهی L1 سعی می‌کند میانه داده‌ها را تخمین بزند، در حالی که نظم‌دهی L2 سعی در تخمین میانگین داده‌ها دارد. پارامتر C معکوس پارامتر نظم‌دهی است؛ این یک متغیر کنترلی است که با قرارگیری معکوس نسبت به ضریب لاندا (Lambda)، میزان قدرت نظم‌دهی را تنظیم می‌کند. مقادیر بالاتر C متناظر با نظم‌دهی کمتری هستند که از بیش‌برازش مدل جلوگیری می‌کند. مدل ناوِی بیز یک مدل احتمالی بیزی بسیار ساده‌شده است که بر پیش‌فرض استقلال قوی بین ویژگی‌ها عمل می‌کند. این بدین معناست که احتمال یک ویژگی بر احتمال دیگری تأثیر نمی‌گذارد. با فرض داشتن مجموعه‌ای از n ویژگی، رده‌بند ناوِی بیز $n!^2$ فرضیه استقلال ایجاد می‌کند. با این وجود، نتایج رده‌بند ناوِی بیز اغلب صحیح است [۳۵]. رده‌بندهای ناوِی بیز خانواده‌ای از رده‌بندها هستند که از نظر ساختاری بسیار شبیه به رگرسیون لجستیک و رده‌بند بردار پشتیبان خطی (Linear SVC) می‌باشند. با این حال، آن‌ها معمولاً در فرآیند آموزش حتی سریع‌تر عمل می‌کنند. قیمت پرداختی برای این کارایی این است که مدل‌های ناوِی بیز اغلب عملکرد تعمیم‌پذیری را ارائه می‌دهند که کمی ضعیف‌تر از رده‌بندهای خطی است. دلیل کارایی بالای مدل‌های ناوِی بیز این است که آن‌ها پارامترها را با بررسی هر ویژگی به‌صورت جداگانه یاد می‌گیرند و آمار ساده‌ای را برای هر کلاس از هر ویژگی جمع‌آوری می‌کنند. برای طبقه‌بندی داده‌های دودویی، از ناوِی بیز برنولی (Bernoulli Naive Bayes) استفاده می‌شود. این رده‌بند فرض می‌کند که داده‌ها دودویی هستند و تعداد دفعاتی را می‌شمارد که هر ویژگی در هر کلاس مقدار غیرصفر دارد. ناوِی بیز برنولی تنها یک پارامتر، یعنی α، دارد که پیچیدگی مدل را کنترل می‌کند. نحوه عملکرد α بدین صورت است که الگوریتم α نقطه داده مجازی به داده‌ها اضافه می‌کند که تمام ویژگی‌های آن‌ها مقادیر مثبت دارند. این امر منجر به «هموارسازی» (smoothing) آمار می‌شود. مقدار α بزرگ‌تر به معنای هموارسازی بیشتر و در نتیجه مدل‌هایی با پیچیدگی کمتر است. عملکرد الگوریتم نسبت به تنظیم α نسبتاً قوی (robust) است، به این معنا که تنظیم دقیق α برای دستیابی به عملکرد خوب ضروری نیست [۳۶]. از سوی دیگر، مدل‌های مبتنی بر درخت در کاربردهای خود در حوزه سیستم‌های تشخیص نفوذ مبتنی بر رفتار، عملکرد چشمگیری از خود نشان داده‌اند. جنگل تصادفی (Random Forest یا RF) چندین درخت تصمیم بدون هرس (pruning) می‌سازد که با روش نمونه‌برداری با جایگذاری (bootstrapping) از داده‌های آموزشی و انتخاب تصادفی زیرمجموعه‌ای از ویژگی‌ها القا می‌شوند. پیش‌بینی با تجمیع (aggregating) نتایج مجموعه انجام می‌شود (در طبقه‌بندی، رأی اکثریت و در رگرسیون، میانگین‌گیری). جنگل تصادفی عموماً بهبود قابل‌توجهی در عملکرد نسبت به یک درخت واحد از خود نشان می‌دهد و نرخ خطای پایینی همراه با مقاومت برجسته در برابر نویز فراهم می‌کند [۳۲]. پارامتر n_estimators تعداد درخت‌هایی است که در مدل RF ساخته می‌شوند. تعداد درخت‌های بیشتر، مدل‌هایی پایدارتر تولید می‌کند اما نیازمند حافظه بیشتر و زمان اجراطولانی‌تر است. پارامتر max_features تعداد ویژگی‌هایی است که هنگام جستجوی بهترین تقسیم (split) در نظر گرفته می‌شوند. در اینجا مقادیر sqrt (یعنی $\sqrt{}$(تعداد ویژگی‌ها)) و log2 (یعنی $\log_2$(تعداد ویژگی‌ها)) را به‌عنوان مقادیر ممکن برای این پارامتر در نظر گرفته‌ایم. آدابوست (AdaBoost مخفف Adaptive Boosting) یک الگوریتم نماینده از خانواده روش‌های تقویت‌کننده (boosting) است که ایده اصلی آن انتخاب و ترکیب گروهی از رده‌بندهای ضعیف برای ساخت یک رده‌بند قوی است. الگوریتم تقویت‌کننده به‌صورت یک رویکرد تکرارشونده برای تولید یک رده‌بند قوی به‌کار گرفته می‌شود که با ترکیب رده‌بندهای ضعیفی که تنها از حدس‌های تصادفی ساخته شده‌اند، خطای آموزشی بسیار کوچکی را به‌دست می‌آورد. در این روش، از تکنیک گروهی (ensemble) برای استخراج سوابق آموزشی با به‌روزرسانی مکرر توزیع نمونه‌های داده‌های قبلاً آموزش‌دیده استفاده می‌شود [۳۷]. در این پیاده‌سازی، رده‌بند درخت تصمیم (Decision Tree) به‌عنوان تخمین‌گر پایه (base estimator) مورد استفاده قرار گرفته است. برای تنظیم هایپرپارامتر، پارامتر n_estimators حداکثر تعداد تخمین‌گرهایی را تعیین می‌کند که در آن فرآیند تقویت متوقف می‌شود. در صورت برازش کامل (perfect fit)، فرآیند یادگیری زودتر متوقف می‌شود. پارامتر learning_rate برای کاهش سهم هر رده‌بند (یعنی وزنی که در هر تکرار تقویت به هر رده‌بند

اختصاص داده می‌شود) در نظر گرفته شده است. نرخ یادگیری (learning rate) بالاتر، سهم هر رده‌بند را افزایش می‌دهد. بین پارامترهای learning_rate و n_estimators یک رابطه مبادله (trade-off) وجود دارد.

جدیدترین روش نظارت‌شده در این حوزه، ماشین بردار پشتیبان (Support Vector Machine یا SVM) است. SVM داده‌ها را به فضاهای با ابعاد بالاتر تبدیل می‌کند و ابرصفحه‌ای (hyperplane) را پیدا می‌کند که به‌بهترین‌شکل داده‌ها را از هم جدا کند. یک ماشین بردار پشتیبان با یافتن بهترین ابرصفحه‌ای که تمام نقاط داده یک کلاس را از نقاط داده کلاس دیگر جدا می‌کند، داده‌ها را طبقه‌بندی می‌نماید. SVMها حول مفهوم «حاشیه» (margin) به دو سوی ابرصفحه‌ای که دو کلاس داده را از هم جدا می‌کند، می‌چرخند. حاشیه، عرض حداکثری نواری موازی با ابرصفحه است که هیچ نقطه داده‌ای در داخل آن وجود ندارد. حداکثر کردن این حاشیه و ایجاد بیشترین فاصله ممکن بین ابرصفحه جداکننده و نمونه‌های هر دو سوی آن، ثابت شده است که کران بالای خطای تعمیم‌پذیری مورد انتظار را کاهش می‌دهد [۳۸]. کرنل (kernel)، نحوه تصویرسازی متغیرهای ورودی را کنترل می‌کند. گزینه‌های متعددی وجود دارند، اما کرنل‌های خطی، چندجمله‌ای و RBF (تابع پایه شعاعی) رایج‌ترین آن‌ها هستند. هنگامی که یک رده‌بند بردار پشتیبان (SVC) برای طبقه‌بندی از کرنل خطی استفاده می‌کند، به‌عنوان یک SVC خطی (Linear SVC) تعریف می‌شود. مدل‌های خطی در فضاهای کم‌بعد می‌توانند بسیار محدودکننده باشند، زیرا خطوط و ابرصفحه‌ها انعطاف‌پذیری محدودی دارند. یکی از راه‌های افزایش انعطاف‌پذیری مدل خطی، افزودن ویژگی‌های بیشتر است. یک پارامتر حیاتی، پارامتر نظم‌دهی جریمه‌ای C است که می‌تواند طیف وسیعی از مقادیر را بپذیرد و تأثیر چشمگیری بر شکل نواحی نهایی هر کلاس دارد. استفاده از مقیاس لگاریتمی می‌تواند نقطه شروع مناسبی باشد. شدت نظم‌دهی (regularization) به‌صورت معکوس با C متناسب است؛ بنابراین C باید کاملاً مثبت باشد. همان‌گونه که در مدل‌های خطی دیده می‌شود، مقدار کوچک C به معنای مدلی بسیار محدود است که در آن هر نقطه داده تنها می‌تواند تأثیر بسیار محدودی داشته باشد [۳۶].

۴. مروری بر مجموعه‌داده

برای به‌کارگیری روش‌های یادگیری ماشین، وجود مجموعه‌داده‌ای با تعداد نمونه‌های قابل‌توجه، زمینه‌سازی شده و برچسب‌گذاری‌شده به‌درستی ضروری است. این بخش توضیح مختصری از سناریویی ارائه می‌دهد که در آن مجموعه‌داده انتخاب‌شده توسعه یافته است. سپس، تحلیل خلاصه‌ای از داده‌های مرتبط با رده‌بند ارائه می‌شود. تحلیل جامع‌تر این مجموعه‌داده در منبع [۳۹] قابل‌یافتن است.

۴.۱. توصیف سناریو

برای نزدیک‌کردن مجموعه‌داده به یک محیط واقعی، داده‌ها از حسگرهای دمای مرکز داده CITIC [۴۰] جمع‌آوری شده‌اند. این مجموعه‌داده که DAD نامیده می‌شود، هفت روز از وضعیت روزانه‌ای را نشان می‌دهد که در یک محیط واقعی رخ می‌دهد و دارای حجم کافی ردپا (trace) و استخراج ویژگی‌های مشخص است. حسگرها دمای مرکز داده را روی سه مؤلفه نظارت می‌کنند: رک‌ها (racks)، نوارهای برق (PDU) و دستگاه‌های سرمایشی (InRow). ما تنها از حسگرهای InRow استفاده کردیم، زیرا مقادیر حاصل از سایر حسگرها ثابت و بی‌معنی هستند. مهم‌ترین دستگاه‌های سیستم، واحدهای InRow (شماره‌های ۱۳، ۱۵، ۲۳ و ۲۵) هستند که مسئول سرمایش هوا در مرکز داده از طریق یک سیستم خنک‌کننده مایعی می‌باشند. این واحدهای InRow دارای چهار حسگر مرتبط هستند:

(i)دمای هوای ورودی واحد(TAS) ؛
(ii)دمای هوای بازگشتی واحد(TAR) ؛
(iii)دمای سیال ورودی واحد(TFEU) ؛
(iv) دمای سیال خروجی واحد (TFSU).

در مورد محیط مجازی‌سازی‌شده، پنج ماشین مجازی با سیستم‌عامل Ubuntu Server 18.04 ایجاد شد که هر کدام به شبکه داخلی اینترنت اشیا (IoT) متصل بودند، به‌طوری‌که ترافیک بین آن‌ها کاملاً ایزوله بود.

در هر یک از چهار واحد سرمایشی، چهار حسگر وجود دارد. هر گره مشتری (client node) فرآیندی را اجرا می‌کند که هر حسگر را با یک شناسه مرتبط با شناسه واحد سرمایشی (InRow) شبیه‌سازی می‌کند. به‌عنوان مثال، واحد سرمایشی ۱۳ برای TAS شناسه ۱۳۱، برای TAR شناسه ۱۳۲، برای TFEU شناسه ۱۳۳ و برای TFSU شناسه ۱۳۴ دارد. هنگامی که حسگر برای انتشار موضوع (topic) مربوطه به بروکر متصل می‌شود، پیامی با پروتکل MQTT و با شناسه گره خود به‌عنوان ClientId ارسال می‌کند. همه حسگرهای InRow دارای یک آدرس IP یکسان هستند، اما در هر روز، پورت‌های مورد استفاده برای انتقال پیام برای هر حسگر متفاوت است. به همین دلیل، در مجموع از ۱۱۲ پورت TCP استفاده می‌شود. هر یک از این حسگرها هر پنج دقیقه یک‌بار داده را به بروکر ارسال می‌کند، یعنی ۲۸۸ نمونه در ساعت و در مجموع ۴۰۳۲ نمونه در روز برای هر حسگر. این حسگرها ارتباطی مستقیم با یکدیگر ندارند. مجموعه‌داده شامل سناریوهای متنوعی از ناهنجاری است که در آن ترافیک غیرعادی از نظر آماری با ترافیک عادی متفاوت است و اکثر نمونه‌های ترافیک شبکه، عادی هستند. این مجموعه‌داده در سطح بسته (packet) برچسب‌گذاری شده است و نوع توکن (عادی یا ناهنجار) را مشخص می‌کند.

رفتار گره ناهنجار به یکی از روش‌های زیر تغییر یافته است:

- اعتراض :(Interception) حذف تصادفی برخی از بسته‌های ارسالی.
- تغییر :(Modification) تغییر دمای ارسالی بدون رعایت الگوی تعیین‌شده.
- تکثیر (Duplication): ارسال توکن‌هایی بیشتر از تعداد برنامه‌ریزی‌شده اولیه.

۴.۲. توصیف محتوا

داده‌های عددی به‌منظور درک مدل مفهومی مقادیر مشاهده‌شده تحلیل شده‌اند. ترافیک ناهنجار از واحد InRow 13 با آدرس IP منبع ۱۰.۶.۵۶.۴۱ ایجاد شده است و بسته‌های ناهنجار در روزهای خاصی از هفته (چهارشنبه، پنج‌شنبه، جمعه و شنبه) ارسال شده‌اند. مجموعه‌داده DAD در مجموع دارای ۱۰۱,۵۸۳ بسته است که ترافیک UDP و TCP غالب هستند. از این بسته‌ها، ۶۳.۳٪ متعلق به ترافیک MQTT هستند که ۱۶٪ از آن‌ها به‌عنوان ناهنجار علامت‌گذاری شده‌اند. تعداد بایت‌های منبع و مقصد، بسته‌های منبع و مقصد، بسته‌های TCP، UDP و MQTT در طول روزهای هفته یکنواخت است و به‌طور میانگین روزانه شامل ۱۴,۰۲۰ بسته TCP است که از این میان ۹,۲۶۷ بسته متعلق به MQTT هستند. ناهنجاری‌های ناشی از «اعتراض» در مجموعه‌داده برچسب‌گذاری نشده‌اند، زیرا این بسته‌ها حذف شده‌اند. بسته‌های مربوط به اعتراض وجود ندارند و به همین دلیل در آمار کلی به‌عنوان بسته‌های غیرعادی منعکس نمی‌شوند. به عبارت دیگر، درصد ترافیک ناهنجار، ناهنجاری‌های ناشی از اعتراض را در نظر نمی‌گیرد. هر حسگر هر ۵ دقیقه یک‌بار پیام به بروکر ارسال می‌کند. یک زمان انتظار بیکاری (idle timeout) برابر با ۳۰ ثانیه پیکربندی شده است، بنابراین اگر گره‌ای برای این مدت بسته‌ای ارسال نکند، بسته بعدی به‌عنوان بخشی از یک جریان (flow) جدید در نظر گرفته می‌شود. جریان‌ها به‌صورت یک‌سویه (unidirectional) تعریف شده‌اند. با توجه به نوع اتصال معمول در شبکه‌های MQTT، تغییر در پیام یکی از حسگرها بر کل جریان MQTT تأثیر می‌گذارد. در نتیجه، اگر بسته‌ای متعلق به جریانی باشد که حداقل یکی از بسته‌های آن به‌عنوان ناهنجار برچسب‌گذاری شده باشد، تمام بسته‌های آن جریان غیرعادی در نظر گرفته می‌شوند. مجموعه‌داده در مجموع دارای ۲۲۴ جریان اتصال TCP است که از این میان ۲۰۸ جریان عادی و ۱۶ جریان ناهنجار هستند. علاوه بر این، این مجموعه‌داده شامل ۶۷,۸۴۸ جریان MQTT است که از

آن‌ها ۵۴۴ جریان مربوط به ناهنجاری هستند. تعداد جریان‌ها در طول روزهای هفته یکنواخت است، اما در روز یکشنبه تعداد کمتری دیده می‌شود که به دلیل قطع اتصال‌هایی است که روز بعد انجام می‌شود. تمام ناهنجاری‌ها بر روی جریان‌های TCP اعمال شده‌اند.

۵. آماده‌سازی داده‌ها

به‌عنوان مرحله‌ای پیش از به‌کارگیری الگوریتم‌های یادگیری ماشین، لازم است داده‌ها، نمونه‌ها و ویژگی‌های مورد استفاده در تحلیل مشخص شوند. این وظیفه شامل ارتقای کیفیت داده‌ها به سطح مورد نیاز توسط روش‌های تحلیلی انتخاب‌شده است و در برگیرنده تبدیل‌های داده و ساخت ویژگی‌های جدید از یک یا چند ویژگی موجود برای اهداف پاک‌سازی و همچنین بررسی تأثیر احتمالی آن‌ها بر نتایج تحلیل می‌باشد.

شناخت پارامترهایی که الگوریتم‌های یادگیری ماشین برای اجرای بهینه نیاز دارند، امری حیاتی است. با توجه به ماهیت و نحوه عملکرد رده‌بند، ۱۴ ویژگی به‌عنوان مرتبط شناسایی و برای پردازش انتخاب شدند:

«ip.src.» ،«tcp.srcport.» ،«frame.len» ،«frame.time»،

«ip.dst» ،«tcp.dstport» ،«tcp.flags» ،«tcp.flags.ack»،

«tcp.flags.fin» ،«tcp.flags.res» ،«tcp.flags.reset»،

.««label ،«protocol» ،tcp.flags.syn»»

ویژگی‌های جدیدی برای بهبود تخمین مدل ایجاد شدند:

- Weekday(روز هفته): روز هفته استخراج‌شده از ویژگی.frame.time
- Protocol_Type(نوع پروتکل): مشخص می‌کند که آیا پروتکل مربوطه، MQTT-IoTاست یا خیر.
- ClientID (شناسه مشتری): ویژگی که بر اساس ip.src و tcp.srcport مشخص می‌کند که هر سنسور به کدام واحد InRow تعلق دارد.

برخی داده‌ها به‌طور ذاتی چرخه‌ای هستند؛ این موضوع در مورد ساعات، دقایق و ثانیه صدق می‌کند. اگر ساعت را به‌عنوان یک متغیر خطی در نظر بگیریم، رده‌بند متوجه نخواهد شد که ساعت ۲۳ قبل از ساعت ۰ قرار دارد. برای حل این مشکل، باید مقادیر چرخه‌ای به الگوریتم یادگیری ماشین ارائه شوند؛ این کار با جایگزینی ویژگی موجود با دو ویژگی جدید با استفاده از نمایش سینوسی انجام می‌شود. هر دو متغیر ضروری هستند، در غیر این‌صورت حرکت صحیح در طول زمان از بین می‌رود. این امر به دلیل تغییر مشتق توابع سینوس یا کسینوس در طول زمان است که در آن موقعیت (x,y) به‌صورت پیوسته در اطراف دایره واحد تغییر می‌کند.

در مورد ساعت‌شمار ۲۴ ساعته، این کار به‌ترتیب با تبدیل‌های سینوسی و کسینوسی مطابق معادله (۱) انجام می‌شود:

$$x = \sin(2 * \pi * hour / 24)$$
$$y = \cos(2 * \pi * hour / 24).$$

(۱)

فاصله بین دو نقطه در این فضای جدید، معادل اختلاف زمانی بین آن‌هاست که دقیقاً همان چیزی است که از یک چرخه ۲۴ ساعته انتظار داریم.

همان‌طور که پیش‌تر اشاره شد، در مجموعه‌داده از ۱۱۲ پورت TCP استفاده شده است. این امر در فرآیند نگاشت منجر به ساخت ۱۱۲ ویژگی می‌شود که اطلاعات تکراری ضمنی در ClientId را دارند. با توجه به این موضوع، ویژگی‌های tcp.srcport و tcp.dstport حذف شدند.

بهترین روش نمایش داده‌ها نه‌تنها به معنای داده‌ها، بلکه به نوع مدل مورد استفاده نیز بستگی دارد. مدل‌های خطی و مدل‌های مبتنی بر درخت (مانند درخت‌های تصمیم، درخت‌های تقویت‌شده گرادیانی و جنگل تصادفی)، دو خانواده بزرگ و بسیار رایج، خواص بسیار متفاوتی در برخورد با نمایش‌های مختلف ویژگی دارند. مدل‌های خطی تنها می‌توانند روابط خطی را مدل کنند که در مورد یک ویژگی منفرد به‌صورت خطوط ساده ظاهر می‌شوند. یکی از راه‌های افزایش توان مدل‌های خطی در مواجهه با داده‌های پیوسته، استفاده از روش گروه‌بندی (binning) یا گسسته‌سازی (discretization) است که در آن یک ویژگی پیوسته به چندین ویژگی گسسته تبدیل می‌شود. در اینجا، به‌عنوان فرآیندی مشترک برای تمام مدل‌ها، یک ویژگی ورودی پیوسته از مجموعه‌داده به یک ویژگی گسسته تبدیل می‌شود که نشان می‌دهد هر نمونه داده به کدام دسته (bin) تعلق دارد [۳۶]. ویژگی‌های گسسته در مجموعه‌داده باید به نمایش عددی تبدیل شوند. این فرآیند با استفاده از کدگذاری دودویی معمول انجام می‌شود، به‌طوری‌که هر متغیر گسسته با m مقدار ممکن، با $m-1$ متغیر مجازی (dummy variable) جایگزین می‌شود. یک متغیر مجازی برای یک دسته خاص مقدار ۱ و برای سایر دسته‌ها مقدار ۰ دارد [۴۱].

در مورد ویژگی برچسب (label)، کلاس عادی مقدار عددی ۰ و کلاس ناهنجار مقدار عددی ۱ دریافت می‌کند. ویژگی frame.time به یک متغیر تاریخ تبدیل می‌شود تا امکان استخراج اطلاعات مرتبط دیگر از آن فراهم شود. ویژگی d_flow نشان‌دهنده مدت‌زمان جریان (flow) است و frame.len_mean معادل میانگین حسابی طول فریم تمام بسته‌های متعلق به یک جریان است. تمام این ویژگی‌ها به فرم گسسته تبدیل شده‌اند.

ویژگی‌های نهایی به‌صورت زیر هستند:

[«frame.len» ،«hora_cos» ،«hora_sin» ،«Npackperflow» ،«d_flow»،
«tcp.flags.ack» ،«tcp.flags.fin» ،«tcp.flags.reset» ،«tcp.flags.syn»،
«frame.len_mean» ،«tcp.flags.res» ،«label_fl»،
«ip.src_10.6.56.1» ،«ip.src_10.6.56.34» ،«ip.src_10.6.56.36»،
«ip.src_10.6.56.41» ،«ip.src_10.6.56.50» ،«ip.dst_10.6.56.1»،
«ip.dst_10.6.56.34» ،«ip.dst_10.6.56.36» ،«ip.dst_10.6.56.41»،
«ip.dst_10.6.56.50» ،«ClientId_131» ،«ClientId_132» ،«ClientId_133»،
«ClientId_134» ،«ClientId_151» ،«ClientId_152» ،«ClientId_153»،
«ClientId_154»،
«ClientId_231» ،«ClientId_232» ،«ClientId_233» ،«ClientId_234»،
«ClientId_251» ،«ClientId_252» ،«ClientId_253» ،«ClientId_254»،
«ClientId_Broker» ،«protocol_TIP_MQTT» ،«protocol_TIP_TCP»،
«weekday_Friday» ،«weekday_Monday» ،«weekday_Saturday»،
«weekday_Sunday» ،«weekday_Thursday» ،«weekday_Tuesday»،
.[«weekday_Wednesday»

پس از نگاشت ویژگی‌های نمادین به مقادیر عددی، در صورت وجود واریانس قابل‌توجه، نیاز به مقیاس‌بندی ویژگی‌ها احساس شد. مقیاس‌بندی ویژگی‌ها از طریق نرمال‌سازی میانگین (mean normalization) انجام گرفت. از سوی دیگر، کاهش تعداد ویژگی‌ها می‌تواند اجرای الگوریتم‌های یادگیری ماشین را کارآمدتر (با پیچیدگی فضایی یا زمانی کمتر) و مؤثرتر کند. برخی الگوریتم‌های یادگیری ماشین ممکن است توسط ویژگی‌های ورودی نامرتبط گمراه شوند که این امر منجر به عملکرد پیش‌بینی ضعیف‌تر می‌شود. با توجه به همگن‌بودن پروتکل‌های MQTT و نحوه عملکرد الگوریتم‌های مختلف یادگیری ماشین، انتخاب ویژگی‌ها امکان حذف ویژگی‌های هم‌خطی (correlated) و بهبود کارایی محاسباتی الگوریتم‌ها را فراهم می‌آورد. تابع حذف بازگشتی ویژگی‌ها (Recursive Feature Elimination یا RFE) یک روش انتخاب معکوس از پیش‌بین‌کننده‌ها بر اساس رتبه‌بندی اهمیت آن‌ها پیاده‌سازی می‌کند. در این روش، پیش‌بین‌کننده‌ها رتبه‌بندی شده و کم‌اهمیت‌ترین آن‌ها قبل از مدل‌سازی به‌صورت متوالی حذف می‌شوند. هدف یافتن زیرمجموعه‌ای از پیش‌بین‌کننده‌هاست که بتوان از آن برای تولید یک مدل دقیق استفاده کرد [۴۲]. مدل انتخاب‌شده به‌عنوان تخمین‌گر RFE، یک رده‌بند درخت تصمیم با اعتبارسنجی متقاطع k-fold استراتیفاید با پنج تقسیم بود. پس از انتخاب مهم‌ترین پیش‌بین‌کننده‌ها توسط الگوریتم، دقت به‌دست‌آمده در تعدادهای مختلفی از پیش‌بین‌کننده‌های مورد استفاده برآورد شد. همان‌طور که در شکل ۱ مشاهده می‌شود، بهترین دقت با استفاده از ۱۲ ویژگی حاصل شد. ویژگی‌های انتخاب‌شده عبارت بودند از: hour_cos، hour_sin، d_flow، ip.src_10.6.56.41، ClientId_131، ClientId_132، ClientId_133، ClientId_134، weekday_Friday، weekday_Saturday، weekday_Thursday، weekday_Wednesday.

شکل ۱. دقت در برابر اهمیت متغیر.

در محیط تشخیص نفوذ شبکه، معمولاً با این وضعیت مواجه هستیم که یکی از کلاس‌ها بخش بسیار کوچکی از داده‌ها را تشکیل می‌دهد، در حالی‌که همین کلاس اقلیت، موارد مهمی برای تشخیص محسوب می‌شوند. هنگام یادگیری از داده‌های بسیار نامتوازن، احتمال این‌که نمونه انتخاب‌شده حاوی تعداد بسیار کم یا حتی فاقد نمونه‌های کلاس اقلیت باشد، به‌طور قابل‌توجهی بالاست؛ این امر منجر به ایجاد الگوریتمی با عملکرد ضعیف در پیش‌بینی کلاس اقلیت می‌شود [۴۳]. این مشکل باعث می‌شود که الگوریتم‌های یادگیری ماشین به‌راحتی دسته‌بندی‌های نادرستی انجام دهند. اعتبارسنجی متقاطع (cv) یک روش آماری برای ارزیابی عملکرد تعمیم‌پذیری است که در مقایسه با تقسیم ساده داده به مجموعه آموزش و آزمون، پایدارتر و جامع‌تر است. در اعتبارسنجی متقاطع، داده‌ها به‌صورت مکرر تقسیم شده و مدل‌های متعددی آموزش داده می‌شوند. اعتبارسنجی متقاطع k-fold استراتیفاید (stratified k-fold cross-validation) نوعی تغییریافته از روش k-fold است که چینش‌های استراتیفاید (stratified folds) تولید می‌کند. در اعتبارسنجی متقاطع استراتیفاید، چینش‌ها به گونه‌ای ساخته می‌شوند که درصد نمونه‌های هر کلاس در تمام چینش‌ها حفظ شود و رابطه تعادل‌یافته بین کلاس‌ها در داده‌ها حفظ گردد. معمولاً استفاده از اعتبارسنجی متقاطع k-fold استراتیفاید به‌جای اعتبارسنجی متقاطع k-fold ساده برای ارزیابی یک رده‌بند، ایده خوبی است، زیرا تخمین‌های قابل‌اطمینان‌تری از عملکرد تعمیم‌پذیری ارائه می‌دهد [۳۶]. رویکرد دیگری که امکان کاهش مشکل نامتوازن‌بودن داده را فراهم می‌کند، استفاده از تکنیک‌های نمونه‌برداری است. SMOTE تابعی برای مدیریت مسائل طبقه‌بندی نامتوازن است. این روش با کاهش تصادفی نمونه‌های کلاس اکثریت (down-sampling) همراه است و به‌جای نمونه‌برداری با جایگذاری از کلاس اقلیت، نمونه‌های مصنوعی از کلاس اقلیت ایجاد می‌کند تا تعداد آن را افزایش دهد [۴۳,۴۴]. در مجموعه‌داده DAD، ۱۰۸۸ بسته ناهنجار و ۹۶,۸۹۴ بسته عادی وجود دارد. این امر نسبت ۱:۸۹ را نشان می‌دهد که وضعیت آشکار نامتوازن‌بودن داده را بیان می‌کند. برای تضمین این‌که نمونه‌هایی از هر دو کلاس در نظر گرفته شوند، آزمایش‌ها با به‌کارگیری SMOTE در درون یک حلقه تو در توی اعتبارسنجی متقاطع k-fold استراتیفاید با k=5 انجام شد. به‌صورت پیش‌فرض، SMOTE از پنج همسایه (k-

neighbors) استفاده می‌کند. در ابتدا، کلاس اقلیت تا نسبتی حدود ۰.۴ افزایش نمونه‌برداری (over-sampled) شد. سپس، کلاس اکثریت تا نسبت ۰.۵ کاهش نمونه‌برداری (under-sampled) گردید، به‌طوری که هر چینش (fold) در تقسیم‌بندی اعتبارسنجی متقاطع، توزیع کلاسی یکسانی مطابق با پیکربندی SMOTE داشته باشد.

۶. آموزش مدل‌ها

در آزمایش‌ها، تمام مدل‌ها در چارچوب یک اعتبارسنجی متقاطع k-fold استراتیفاید تو در تو ارزیابی شدند؛ حلقه بیرونی با k=5 و حلقه درونی نیز با k=5، در حالی‌که SMOTE در حلقه درونی اعمال شد. در این بخش، روش‌های برازش یادگیری ماشین روی داده‌های باقی‌مانده از مجموعه آموزش k-fold cv تشریح می‌شود و مقدار «ناحیه زیر منحنی عملکرد دستگاه دریافت‌کننده» (ROC AUC) از نمره‌های پیش‌بینی محاسبه می‌گردد. نتایج ارائه‌شده متناظر با میانگین مقادیر به‌دست‌آمده از حلقه درونی هستند. برای رگرسیون لجستیک، از حل‌کننده liblinear استفاده شد. مجازات‌های L1 و L2 در نظر گرفته شدند و پارامتر C در مقادیر ۰.۰۰۱، ۰.۱، ۱، ۱۰۰، و ۱۰۰۰ تنظیم گردید. به دلیل ماهیت تصادفی الگوریتم، نتایج به‌دست‌آمده بسته به تکرارهای انجام‌شده متفاوت است. بهترین نتیجه به‌دست‌آمده، مقدار ۰.۹۷۹۰ با مجازات L2 و مقدار C برابر با ۱۰۰۰ است. کمترین مقدار، ۰.۹۶۷۶ با مجازات L1 و مقدار C برابر با ۰.۰۰۱ است. میانگین کلی در تمام تکرارها برابر با ۰.۹۷۵۶ و انحراف معیار آن ۲.۰۶×۱۰^{-۳} است. معمولاً مقادیر بالای C آزادی بیشتری به مدل می‌دهد، بنابراین انتظار عملکرد بهتری در مقادیر بالای C وجود دارد. با این حال، همان‌طور که در شکل ۲ مشاهده می‌شود، تغییر پارامتر C به‌تنهایی بدون انتخاب مناسب نوع مجازات، به‌صورت قابل‌توجهی عملکرد مدل را بهبود نمی‌بخشد.

شکل ۲. میانگین نمره رگرسیون لجستیک بر اساس تکرار.

جدول ۱ بهترین پارامترهای انتخاب‌شده برای بهترین نمره میانگین در هر یک از حلقه‌های درونی cv و همچنین انحراف معیار به‌دست‌آمده را نشان می‌دهد. پارامترهای تنظیم‌شده که در اکثر موارد بهترین نتایج را ارائه داده‌اند، C برابر با ۱ و مجازات L1 هستند. پارامترهای ارائه‌شده در جدول، همان پارامترهایی هستند که برای ارزیابی مدل در مجموعه آزمون انتخاب شده‌اند.

جدول ۱. بهترین مدل رگرسیون لجستیک بر اساس تکرار.

در میان مدل‌های ناوِی بیز که پیش‌تر ذکر شدند، ناوِی بیز برنولی (Bernoulli Naive Bayes) انتخاب شد. برای یافتن ترکیب بهینه هایپرپارامترها که تابع از دست‌داد (loss function) از پیش‌تعریف‌شده را کمینه کند، نتایج با تغییر پارامتر α در مقادیر ۰.۰۱، ۰.۱، ۰.۵، ۱ و ۱۰ محاسبه شدند. میانگین مقدار ROC AUC در تمام تکرارها برابر با ۰.۹۶۵۴ و انحراف معیار آن ۳.۲۳×۱۰^{-۳} است. کمترین مقدار ROC AUC برابر با ۰.۹۵۸۷ با =۰.۱α و بهترین مقدار ۰.۹۷۲۱ با =۰.۰۱α به‌دست آمد. شکل ۳ میانگین نتایج به‌دست‌آمده از تنظیم هایپرپارامترها را بر اساس تکرار نشان می‌دهد. همان‌طور که در مورد رگرسیون لجستیک مشاهده شد، ناوِی بیز برنولی نیز عملکرد متناسب با مقادیر α را در هر تکرار بهبود نمی‌بخشد.

شکل ۳. میانگین نمره ناوِی بیز بر اساس تکرار.

جدول ۲ پارامترهایی را ارائه می‌دهد که در هر تکرار درونی بهترین نمره را ارائه داده‌اند. این پارامترها در ارزیابی مدل مورد استفاده قرار خواهند گرفت. بهترین مقدار به‌دست‌آمده، ۰.۹۷۲۱ با =۱α و کمترین انحراف معیار برابر با ۳.۲۲×۱۰^{-۳} است.

جدول ۲. بهترین مدل ناوِی بیز بر اساس تکرار.

برای تنظیم هایپرپارامترهای جنگل تصادفی (Random Forest)، مهم‌ترین تنظیمات عبارتند از تعداد درخت‌ها در جنگل (n_estimators) و تعداد ویژگی‌هایی که برای تقسیم در هر گره برگ (leaf node) در نظر گرفته می‌شوند (max_features). در طول اجرا، ارزیابی با مقادیر ۱۰۰، ۲۰۰ و ۳۰۰ درخت و مقادیر sqrt و log2 برای max_features آزمایش شد. میانگین کلی به‌دست‌آمده برابر با ۰.۹۹۹۹ و انحراف معیار آن ۵.۵۷×۱۰$^{-6}$ بود که در برخی تکرارها به ROC AUC برابر با ۱ دست یافت. شکل ۴ میانگین نتایج ROC AUC را بر اساس تکرار نشان می‌دهد. نقاط بالا در نمودار نشان‌دهنده مقادیر ۱ در آن تکرارهاست. جنگل تصادفی بهترین عملکرد را در مرحله آموزش از خود نشان داده است.

شکل ۴. میانگین نمره جنگل تصادفی بر اساس تکرار.

جدول ۳ بهترین پارامترها را در هر تکرار درونی نشان می‌دهد. چندین پیکربندی مختلف وجود دارند که در تمام تکرارها بهترین پاسخ را ارائه می‌دهند. با این حال، تعداد درخت‌ها و ویژگی‌ها می‌توانند زمان و عملکرد مدل را در حین اجرا تحت تأثیر قرار دهند.

جدول ۳. بهترین مدل جنگل تصادفی بر اساس تکرار.

رده‌بند آدابوست (AdaBoost) به‌صورت پیش‌فرض از یک رده‌بند درخت تصمیم به‌عنوان تخمین‌گر پایه استفاده می‌کند که با max_depth=1 مقداردهی اولیه شده است. برای برازش یک مدل خوب، ممکن است افزایش تعداد درخت‌های رده‌بند مطلوب باشد، هرچند این امر ممکن است عملکرد سیستم را تحت تأثیر قرار دهد. در این مدل، تعداد تخمین‌گرها در مقادیر ۵۰۰، ۱۰۰۰ و ۲۰۰۰ و نرخ یادگیری (learning rate) در مقادیر ۰.۰۰۱، ۰.۰۱ و ۰.۱ تنظیم شد. میانگین کلی ROC AUC به‌دست‌آمده برابر با ۰.۹۹۶۹ و انحراف معیار آن ۱.۶۹×۱۰$^{-3}$ بود. در شکل ۵ می‌توان مشاهده کرد که افزایش پارامتر نرخ یادگیری، عملکرد مدل را نسبت به تعداد درخت‌های انتخاب‌شده بهبود می‌بخشد؛ با این حال، نتایج عالی را می‌توان حتی با سطوح متوسط نرخ یادگیری و تعداد کم درخت‌ها در رده‌بند به‌دست آورد.

شکل ۵. میانگین نمره آدابوست بر اساس تکرار.

جدول ۴ پارامترهایی را نشان می‌دهد که در پیاده‌سازی اعتبارسنجی متقاطع درونی، بهترین میانگین را به‌دست آورده‌اند. با انحراف‌های معیار بسیار کم و دستیابی به نمرات ۱ در برخی تکرارها، بهترین مدل‌ها با مقادیر بالای نرخ یادگیری و تعداد زیاد درخت‌ها یافت شدند.

جدول ۴. بهترین مدل آدابوست بر اساس تکرار.

در نهایت، برای ماشین بردار پشتیبان (SVM)، از رده‌بند بردار خطی استفاده شد، زیرا این مدل انعطاف‌پذیری بیشتری در انتخاب توابع جریمه و تابع زیان دارد و باید مقیاس‌پذیری بهتری نسبت به نمونه‌های زیاد داشته باشد. برای بهبود دقت مدل، پارامتر C تنظیم شد. پارامتر جریمه C، تعادل بین مرز تصمیم و جمله سوءطبقه‌بندی را کنترل می‌کند و در مقادیر ۰.۱، ۱، ۵ و ۱۰ ثابت شد. با در نظر گرفتن تمام آزمایش‌ها، میانگین نمره ROC AUC برابر با ۰.۹۷۵۹ و انحراف معیار آن ۱.۷۴×۱۰$^{-3}$ به‌دست آمد. شکل ۶ میانگین نمره به‌دست‌آمده در تکرارها را نشان می‌دهد و کاهش عملکرد مدل را در مقدار C=۱۰ برجسته می‌کند. بهترین میانگین‌های نمره مدل در مقادیر C برابر با ۱ و ۵ مشاهده شد که در جدول ۵ قابل‌مشاهده است.

۷. نتایج و مقایسه مدل‌ها

برای انجام پیش‌بینی روی داده‌ها، لازم است از داده‌هایی استفاده شود که در فرآیند آموزش مدل مورد استفاده قرار نگرفته‌اند. تخمین عملکرد مدل یادگیری ماشین بر مبنای مجموعه آزمون حاصل از تقسیم‌بندی اعتبارسنجی متقاطع انجام می‌شود. این بخش نتایج چهار معیار سنتی—دقت (accuracy)، صحت (precision)، فراخوانی (recall) و F1—را برای تمام مدل‌های یادگیری ماشین سطحی ارزیابی‌شده ارائه می‌دهد. علاوه بر این، مقایسه‌ای بین عملکرد الگوریتم‌ها انجام شده است. شکل a۷ میانگین مقادیر دقت را برای هر یک از مدل‌ها در تمام تکرارهای انجام‌شده نشان می‌دهد. در این شکل مشاهده می‌شود که کمترین عملکرد در این معیار مربوط به رده‌بند ناوِی بیز است. مدل‌های مبتنی بر درخت تصمیم بهترین نتایج را کسب کرده‌اند و مقادیری بسیار نزدیک به نتیجه بهینه (۱.۰) را به‌دست آورده‌اند. در معیار صحت، باز هم رده‌بندهای مبتنی بر درخت تصمیم در سه مورد از پنج تکرار انجام‌شده، مقادیری برابر با ۱ را کسب کرده‌اند که در شکل b۷ نشان داده شده است. در فراخوانی (شکل c۷)، بهترین نتایج متعلق به ناوِی بیز است، در حالی که آدابوست و جنگل تصادفی نیز نتایج بسیار رضایت‌بخشی داشته‌اند. از سوی دیگر، در معیار F1 (شکل d۷)، باز هم آدابوست و جنگل تصادفی بهترین نتایج را در تمام تکرارها کسب کرده‌اند و عملکردی برجسته در مقایسه با سایر رده‌بندها از خود نشان داده‌اند. جدول ۶ میانگین‌ها را برای تمام تکرارهای انجام‌شده برای تمام رده‌بندها همراه با انحراف معیار آن‌ها ارائه می‌دهد. رده‌بند ناوِی بیز کمترین مقادیر دقت را در میان تمام مدل‌ها دارد، اما تنها مدلی است که فراخوانی کامل (recall=1) را به‌دست آورده است. همان‌طور که انتظار می‌رفت، به دلیل شباهت در ساختار مدل‌ها، رده‌بندهای رگرسیون لجستیک و ماشین بردار پشتیبان نتایج بسیار مشابهی ارائه داده‌اند. هر دو رده‌بند دقتی حدود ۰.۹۶، صحتی در حدود ۰.۱ و F1 حدود ۰.۳ دارند. این نتایج، رده‌بندهای مبتنی بر درخت را به‌عنوان بهترین رده‌بندها برای این مجموعه‌داده تأیید می‌کنند؛ این مدل‌ها مقادیری تقریباً کامل (نزدیک به ۱.۰) را به‌دست آورده‌اند و همچنین کمترین انحراف معیار را نشان داده‌اند که این امر تضمین‌کننده پایداری رده‌بند است.

شکل ۷. معیارها بر اساس تکرار: (الف) دقت. (ب) صحت. (پ) فراخوانی. (ت) F1.

جدول ۶. بهترین مدل از نظر نمره بر اساس تکرار.

۸ .نتیجه‌گیری و کارهای آینده

برای کاهش ناهنجاری‌های ترافیکی، توسعه‌دهندگان نیازمند روش‌های جدیدی برای تشخیص دستگاه‌های IoT آسیب‌دیده هستند. روش‌های یادگیری ماشین اخیراً به‌دلیل کاربردهای موفق در تشخیص ناهنجاری‌های شبکه، از جمله شبکه‌های IoT، اعتبار قابل‌ملاحظه‌ای کسب کرده‌اند. با توجه به کمبود مجموعه‌داده‌های IoT، مجموعه‌داده DAD به‌عنوان ابزاری برای شناخت رفتار دستگاه‌های سبک‌وزن و اختصاصی در شبکه‌های IoT-MQTT ایجاد شده است. DAD یک مجموعه‌داده کامل و برچسب‌گذاری‌شده با تحلیل‌های پیشین است که قرار است برای تشخیص ناهنجاری‌های ترافیکی با استفاده از الگوریتم‌های یادگیری ماشین به‌کار گرفته شود. بنابراین، این پژوهش با هدف نشان دادن این موضوع انجام شده است که مجموعه‌داده DAD را می‌توان برای تشخیص ناهنجاری در شبکه‌های ترافیکی MQTT-IoT استفاده کرد. برای این منظور، داده‌ها باید به‌گونه‌ای پردازش شوند که برای رده‌بند کاملاً قابل‌فهم باشند. روش‌های پاک‌سازی و آماده‌سازی داده سعی می‌کنند نویز را هموار کرده و نقاط پرت را شناسایی نمایند. استفاده از ویژگی‌های چرخه‌ای، متغیرهایی یکتا برای رده‌بند ایجاد می‌کند، در حالی که تکنیک‌های گسسته‌سازی رتبه‌بندی مرتبه بالایی از مقادیر را فراهم می‌آورند که می‌توانند روابط بین مشاهدات را هموار کنند. الگوریتم RFE پیاده‌سازی شد تا وابستگی‌ها و هم‌خطی‌های بین ویژگی‌ها را حذف کرده و خطاهای محاسباتی و هزینه‌های محاسباتی را کاهش دهد. برای تضمین عملکرد بهینه مدل‌ها، تنظیم هایپرپارامترها اعمال شد که این امر مقادیر مناسبی را برای جلوگیری از بیش‌برازش (overfitting) تضمین می‌کند. علاوه بر این، استفاده از SMOTE در ترکیب با اعتبارسنجی متقاطع k-

fold استراتیفاید به‌عنوان یک تکنیک مدیریت داده برای داده‌های نامتوازن، امکان حل ساده ولی مؤثر مشکل وجود کلاس اقلیت در مجموعه‌داده را فراهم آورد و به‌طور چشمگیری عملکرد رده‌بند را بهبود بخشید. در مورد روش‌ها، ما پنج الگوریتم رایج یادگیری ماشین سطحی—رگرسیون لجستیک، ناوِی بیز، جنگل تصادفی و ماشین بردار پشتیبان خطی—را انتخاب کردیم. سه مورد از این الگوریتم‌ها—رگرسیون لجستیک، ناوِی بیز و SVM—بر توابع آماری خطی مبتنی هستند و همان‌طور که انتظار می‌رفت، نتایج بسیار مشابهی ارائه دادند. از سوی دیگر، رده‌بندهای مبتنی بر درخت—جنگل تصادفی و آدابوست—رفتار مشابهی از خود نشان دادند. در مرحله آموزش، تمام مدل‌ها عملکرد استثنایی داشتند؛ کمترین نمره متعلق به رده‌بند ناوِی بیز با میانگین ROC AUC برابر با ۰.۹۵۸۷ بود و بالاترین مقدار که برابر با ۱ بود، توسط جنگل تصادفی و آدابوست کسب شد. در مرحله آزمون، نتایج به‌دست‌آمده از رده‌بندهای مبتنی بر درخت، مقادیر عالی در معیارهای سنتی را نشان داد و عملکرد برجسته‌ای در کاربردهای سیستم‌های تشخیص نفوذ مبتنی بر رفتار ارائه کرد و بهبود چشمگیری در کارایی، نرخ خطای پایین و مقاومت در برابر نویز را به همراه داشت. مجموعه‌داده DAD به‌منظور تشخیص ناهنجاری در شبکه‌های ترافیکی MQTT-IoT ایجاد شده است. از این رو، این آزمایش‌ها تأیید می‌کنند که این مجموعه‌داده را می‌توان برای کاربرد تشخیص ناهنجاری ترافیک در IoT با به‌کارگیری تکنیک‌های بهینه‌سازی استفاده کرد. با توجه به جدید بودن این مجموعه‌داده، هنوز کار دیگری روی آن انجام نشده است. در نتیجه، ما اولین کاربرد الگوریتم‌های یادگیری ماشین را برای تشخیص ناهنجاری در شبکه‌های IoT بر اساس این مجموعه‌داده ارائه می‌دهیم. کارهای آینده می‌توانند شامل به‌کارگیری سایر انواع الگوریتم‌های یادگیری ماشین روی این مجموعه‌داده باشند تا با مقایسه نتایج به‌دست‌آمده، اثربخشی رده‌بندهای ارائه‌شده تأیید گردد. مشارکت نویسندگان: مفهوم‌پردازی، ل.و. و و.ک؛ روش‌شناسی، و.ک. و ف.خ.ن؛ نرم‌افزار، ل.و.، و.ک. و د.ف؛ اعتبارسنجی، و.ک.، د.ف. و ف.خ.ن؛ تحلیل رسمی، ل.و.، و.ک. و د.ف؛ پژوهش، ل.و.، و.ک. و د.ف.؛ منابع، ل.و.، و.ک.، د.ف. و ف.خ.ن؛ نگارش—پیش‌نویس اولیه، ل.و.، و.ک. و د.ف؛ نگارش—بازبینی و ویرایش، ل.و.، و.ک.، د.ف. و ف.خ.ن؛ نظارت، و.ک؛ مدیریت پروژه، ل.و. تمام نویسندگان نسخه منتشرشده مقاله را خوانده و با آن موافق هستند.

تأمین مالی: این پروژه توسط برنامه «تأیید اعتبار، ساختاردهی و بهبود واحدهای پژوهشی تثبیت‌شده و مراکز منحصربه‌فرد» (ED431G/01) که توسط آموزش حرفه‌ای شورای زانتا د گالیسیا و با بودجه اتحادیه اروپا (صندوق FEDER) و وزارت علوم و نوآوری اسپانیا از طریق پروژه PID2019-111388GB-I00 تأمین شده است، حمایت مالی شده است.

1. Moustafa, N.; Turnbull, B.; Choo, K.R. An Ensemble Intrusion Detection Technique Based on Proposed Statistical Flow Features for Protecting Network Traffic of Internet of Things. IEEE Internet Things J. 2019, 6, 4815–4830.
2. Roman, R.; Najera, P.; Lopez, J. Securing the Internet of Things. Computer 2011, 44, 51–58.
3. Agrawal, S.; Agrawal, J. Survey on Anomaly Detection using Data Mining Techniques. Procedia Comput. Sci. 2015, 60, 708–713.
4. Asharf, J.; Moustafa, N.; Khurshid, H.; Debie, E.; Haider, W.; Wahab, A. A Review of Intrusion Detection Systems Using Machine and Deep Learning in Internet of Things: Challenges, Solutions and Future Directions. Electronics 2020, 9, 1177.
5. Omar, S.; Ngadi, M.; Jebur, H.; Benqdara, S. Machine Learning Techniques for Anomaly Detection: An Overview. Int. J. Comput. Appl. 2013, 79, 33–41.
6. Wazid, M.; Das, A.K.; K, V.; Vasilakos, A. LAM-CIoT: Lightweight authentication mechanism in cloud-based IoT environment. J. Netw. Comput. Appl. 2019, 150, 102496.
7. DAD: Dataset for Anomaly Detection. Available online: https://github.com/dad-repository/dad (accessed on 30 March 2021).
8. Moustafa, N.; Slay, J. UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In Proceedings of the 2015 Military

Communications and Information Systems Conference (MilCIS), Canberra, Australia, 10–12 November 2015; pp. 1–6.
9. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Slay, J. Towards Developing Network Forensic Mechanism for Botnet Activities in the IoT Based on Machine Learning Techniques. In Proceedings of the International Conference on Mobile Networks and Management (MONAMI), Melbourne, Australia, 13–15 December 2017.
10. Alrashdi, I.; Alqazzaz, A.; Aloufi, E.; Alharthi, R.; Zohdy, M.; Ming, H. AD-IoT: Anomaly Detection of IoT Cyberattacks in Smart City Using Machine Learning. In Proceedings of the 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 7–9 January 2019; pp. 305–310.
11. Pajouh, H.H.; Javidan, R.; Khayami, R.; Dehghantanha, A.; Choo, K.K.R. A Two-Layer Dimension Reduction and Two-Tier Classification Model for Anomaly-Based Intrusion Detection in IoT Backbone Networks. IEEE Trans. Emerg. Top. Comput. 2019, 7, 314–323.
12. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection. arXiv 2018, arXiv:1802.09089.
13. Meidan, Y.; Bohadana, M.; Mathov, Y.; Mirsky, Y.; Shabtai, A.; Breitenbacher, D.; Elovici, Y. N-BaIoT—Network-Based Detection of IoT Botnet Attacks Using Deep Autoencoders. IEEE Pervasive Comput. 2018, 17, 12–22.
14. Abbasi, F.; Naderan, M.; Alavi, S.E. Anomaly detection in Internet of Things using feature selection and classification based on Logistic Regression and Artificial Neural Network on N-BaIoT dataset. In Proceedings of the 2021 5th International Conference on Internet of Things and Applications (IoT), Isfahan, Iran, 19–21 May 2021; pp. 1–7.
15. Doshi, R.; Apthorpe, N.; Feamster, N. Machine Learning DDoS Detection for Consumer Internet of Things Devices. In Proceedings of the 2018 IEEE Security and Privacy Workshops (SPW), Francisco, CA, USA, 24 May 2018; pp. 29–35.
16. Pahl, M.; Aubet, F. All Eyes on You: Distributed Multi-Dimensional IoT Microservice Anomaly Detection. In Proceedings of the 2018 14th International Conference on Network and Service Management (CNSM), Rome, Italy, 5–9 November 2018; pp. 72–80.
17. Hasan, M.; Islam, M.M.; Zarif, M.I.; Hashem, M.M. Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. Internet Things 2019, 7, 100059.
18. Koroniotis, N.; Moustafa, N.; Sitnikova, E.; Turnbull, B.P. Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. Future Gener. Comput. Syst. 2019, 100, 779–796.
19. Susilo, B.; Sari, R.F. Intrusion Detection in IoT Networks Using Deep Learning Algorithm. Information 2020, 11, 279.
20. Alsamiri, J.; Alsubhi, K. Internet of Things Cyber Attacks Detection using Machine Learning. Int. J. Adv. Comput. Sci. Appl. 2019, 10, 627–663.
21. Anthi, E.; Williams, L.; Słowińska, M.; Theodorakopoulos, G.; Burnap, P. A Supervised Intrusion Detection System for Smart Home IoT Devices. IEEE Internet Things J. 2019, 6, 9042–9053.
22. Guerra-Manzanares, A.; Medina-Galindo, J.; Bahsi, H.; Nõmm, S. MedBIoT: Generation of an IoT Botnet Dataset in a Medium-sized IoT Network. In Proceedings of the 6th International Conference on Information Systems Security and Privacy—ICISSP, INSTICC, SciTePress, Valletta, Malta, 25–27 February 2020; pp. 207–218.
23. Thamaraiselvi, D.; Mary, S. Attack and Anomaly Detection in IoT Networks using Machine Learning. Int. J. Comput. Sci. Mob. Comput. 2020, 9, 95–103.

24. Parmisano, A.; Garcia, S.; Erquiaga, M.J. Stratosphere Laboratory. A Labeled Dataset with Malicious and Benign IoT Network Traffic. Available online: https://www.stratosphereips.org/datasets-iot23 (accessed on 3 September 2020).
25. Anton, S.D.; Kanoor, S.; Fraunholz, D.; Schotten, H.D. Evaluation of Machine Learning-based Anomaly Detection Algorithms on an Industrial Modbus/TCP Data Set. In Proceedings of the 13th International Conference on Availability, Reliability and Security, Hamburg, Germany, 27–30 August 2018.
26. Liu, Z.; Thapa, N.; Shaver, A.; Roy, K.; Siddula, M.; Yuan, X.; Yu, A. Using Embedded Feature Selection and CNN for Classification on CCD-INID-V1—A New IoT Dataset. Sensors 2021, 21, 4834.
27. MontazeriShatoori, M.; Davidson, L.; Kaur, G.; Lashkari, A.H. Detection of DoH Tunnels using Time-series Classification of Encrypted Traffic. In Proceedings of the 2020 IEEE IEEE International Conference on Dependable Autonomic, & Secure Computing International Conference on Pervasive Intelligence & Computing International Conference Cloud Big Data Computing International Conference Cyber Science and Technology Congress (DASC/Picom/Cbdcom/Cyberscitech), Calgary, AB, Canada, 17–22 August 2020; pp. 63–70.
28. Vaccari, I.; Chiola, G.; Aiello, M.; Mongelli, M.; Cambiaso, E. MQTTset, a New Dataset for Machine Learning Techniques on MQTT. Sensors 2020, 20, 6578.
29. Sarhan, M.; Layeghy, S.; Moustafa, N.; Gallagher, M.; Portmann, M. Feature Extraction for Machine Learning-based Intrusion Detection in IoT Networks. arXiv 2021, arXiv:2108.12722.
30. Moustafa, N. New Generations of Internet of Things Datasets for Cybersecurity Applications based Machine Learning: TON_IoT Datasets. 2019. Available online: http://handle.unsw.edu.au/1959.4/resource/collection/resdatac_921/1 (accessed on 20 October 2021).
31. Sharafaldin, I.; Lashkari, A.H.; Ghorbani, A.A. Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018), Madeira, Portugal, 22–24 January 2018.
32. James, G.; Witten, D.; Hastie, T.; Tibshirani, R. An Introduction to Statistical Learning: With Applications in R; Springer Publishing Company: New York, NY, USA, 2014. Available online: https://www.statlearning.com/ (accessed on 15 November 2021).
33. Syarif, I.; Prugel-Bennett, A.; Wills, G. SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance. Telkomnika Telecommun. Comput. Electron. Control 2016, 14, 1502.
34. Ghosh, P.; Mitra, R. Proposed GA-BFSS and logistic regression based intrusion detection system. In Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), West Bengal, India, 7–8 February 2015; pp. 1–6.
35. Mukherjee, S.; Sharma, N. Intrusion Detection using Naive Bayes Classifier with Feature Reduction. Procedia Technol. 2012, 4, 119–128.
36. Muller, A.C.; Müller, A.C. Introduction to Machine Learning with Python: A Guide for Data Scientists, 1st ed.; O'Reilly Media: Sebastopol, CA, USA, 2016.
37. Mebawondu, O.J.; Alowolodu, O.D.; Adetunmbi, A.O.; Mebawondu, J.O. Optimizing the Classification of Network Intrusion Detection Using Ensembles of Decision Trees Algorithm. In Proceedings of the Third International Conference on Information and Communication

Technology and Applications (ICTA 2020), Minna, Nigeria, 24–27 November 2020; pp. 286–300.
38. Hamid, Y.; Sugumaran, M.; Balasaraswathi, V. IDS Using Machine Learning- Current State of Art and Future Directions. Br. J. Appl. Sci. Technol. 2016, 15, 1–22.
39. Vigoya, L.; Fernandez, D.; Carneiro, V.; Cacheda, F. Annotated Dataset for Anomaly Detection in a Data Center with IoT Sensors. Sensors 2020, 20, 3745.
40. Centro de Investigación en Tecnoloxías da Información e as Comunicacións de Galicia. Available online: https://www.citic-research.org/ (accessed on 30 January 2020).
41. Hasan, M.A.M.; Nasser, M.; Pal, B.; Ahmad, S. Support Vector Machine and Random Forest Modeling for Intrusion Detection System (IDS). J. Intell. Learn. Syst. Appl. 2014, 6, 45–52.
42. Recursive Feature Elimination with Cross-Validation. Available online: https://scikit-learn.org/stable/auto_examples/feature_selection/plot_rfe_with_cross_validation.html (accessed on 6 October 2021).
43. Chen, C.; Breiman, L. Using Random Forest to Learn Imbalanced Data; University of California: Berkeley, CA, USA, 2004.
44. Chawla, N.V.; Bowyer, K.W.; Hall, L.O.; Kegelmeyer, W.P. SMOTE: Synthetic Minority over-Sampling Technique. J. Artif. Int. Res. 2002, 16, 321–357.