دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۳، جلد ۲۲، شماره ۲ (۴۵-۵۵)
https://monadi.isc.org.ir

ارائه‌شده در بیست و یکمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه تربیت مدرس و پژوهشگاه ارتباطات و فناوری اطلاعات، ۲۵ و ۲۶ مهرماه ۱۴۰۳

پیشگیری و تشخیص حملات بات‌نت در اینترنت اشیا با استفاده از روش های یادگیری ترکیبی★

فاطمه پیش داد* و رضا ابراهیمی آتانی

گروه مهندسی کامپیوتر، دانشکده فنی، دانشگاه گیلان، رشت، ایران

اطلاعات مقاله

کلمات کلیدی:
اینترنت اشیا
بات‌نت
تشخیص ناهنجاری
یادگیری ترکیبی
انتخاب ویژگی

نوع مقاله: پژوهشی

چکیده

با پیشرفت و توسعه کاربردهای فناوری اینترنت اشیا، نیاز به اقدامات امن‌سازی زیرساخت در این حوزه به دلیل محدودیت‌های منابع محاسباتی و ذخیره‌سازی اطلاعات، اهمیت ویژه‌ای پیدا کرده است. یکی از مهم‌ترین چالش‌های امنیتی، بات‌نت‌ها هستند که با آلوده‌سازی گره‌های محاسباتی این فناوری، شبکه به مجموعه‌ای از ماشین‌های آلوده تحت کنترل مهاجمان تبدیل می‌شود. این مقاله یک سامانه تشخیص ناهنجاری مبتنی بر یادگیری ترکیبی را برای پیشگیری و شناسایی حملات بات‌نت اینترنت اشیا در مرحله اولیه پویش و حملات DDoS پیشنهاد می‌دهد. این سامانه از انتخاب ویژگی و تنظیم هایپرپارامترهای بهینه برای هر طبقه‌بندی‌کننده استفاده کرده تا دقت مدل را افزایش داده و از بیش‌برازش جلوگیری نماید. داده‌های مورد استفاده در این مقاله از مجموعه داده BoT-IoT گرفته شده که فعالیت‌های مربوط به مراحل مختلف چرخه حیات بات‌نت را پوشش می‌دهند. برای انتخاب ویژگی‌ها و طبقه‌بندی، از دو الگوریتم یادگیری ترکیبی LightGBM و Random Forest استفاده شده و بهینه‌سازی هایپرپارامترها با روش TPE انجام گرفته است. نتایج نشان می‌دهد که سیستم پیشنهادی در مجموعه داده BoT-IoT با دقت ۹۹٫۰۲٪ برای LightGBM و ۹۹٫۹۹٪ برای Random Forest عملکرد بسیار خوبی دارد. این نتایج پتانسیل بالای سیستم در بهبود امنیت دستگاه‌های اینترنت اشیا و امنیت کاربران را نشان می‌دهد.

© ۱۴۰۳ انجمن رمز ایران

۱ مقدمه

امروزه اینترنت اشیا[1] نقش بسیار جدی در مدیریت و کنترل زیرساخت‌های حیاتی زندگی بشر را ایفا می کند. خانه های هوشمند، حمل و نقل هوشمند، بیمارستان هوشمند و نیروگاه‌های هوشمند نمونه‌ای از کاربرد اینترنت اشیا در زندگی روزمره و صنعت هستند. با توجه به استفاده زیاد از دستگاه‌های اینترنت اشیا در زندگی روزمره، این دستگاه‌ها دارای مشکلات امنیتی زیادی هستند و با توجه به محدودیت‌های محاسباتی، انرژی و ذخیره سازی که دارند، شناسایی و مقابله با تهدیدات با چالش‌های زیادی مواجه شده است [۱]. شبکه‌های بات، مجموعه‌ای از گره‌های آلوده در مجموعه گره‌های اینترنت اشیا هستند که توسط مهاجمان از مسیر کانال فرمان و کنترل[2] سعی در به اختیار گرفتن شبکه و اثرگذاری مخربانه در عملکرد اینترنت اشیا دارند. این بات‌نت‌ها به مجرمان سایبری اجازه می‌دهند تا حملات DDoS[3]، سرقت اطلاعات حساس یا انتشار هرزنامه‌های مخرب را انجام دهند. حمله‌های بات‌نت نه تنها تهدیدی برای دستگاه‌های اینترنت اشیا ناامن هستند، بلکه برای

★از کمیته علمی بیست و یکمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.

*نویسنده مسئول.
آدرس‌های رایانامه: pishdadfateme@webmail.guilan.ac.ir (فاطمه پیش داد)، rebrahimi@guilan.ac.ir (رضا ابراهیمی آتانی)

© ۱۴۰۳ تمامی حقوق متعلق به انجمن رمز ایران است.

[1] Internet of things

[2] Command and control

[3] Distributed Denial of Service

شکل ۱. چرخه حیات باتنت [۴]

کل اینترنت نیز تهدیدی جدی به حساب می‌آیند [۲]. پس از ظهور حمله باتنت Mirai در سال ۲۰۱۶، حملات باتنت اینترنت اشیا به طور پیوسته در حال افزایش بوده‌اند. انتشار عمومی کد منبع باتنت Mirai منجر به ظهور بسیاری از نمونه‌های مشابه این بدافزار و تقلیدکنندگان آن شده که میلیون‌ها گره اینترنت اشیا را آلوده کرده و حملات DDoS بزرگی را در دنیا به وجود آورده‌اند [۳]. به طور کلی چرخه حیات یک باتنت شامل چهار مرحله است(شکل ۱): شکل‌گیری، فرمان و کنترل، حمله و پس از حمله. در مرحله شکل‌گیری، مهاجمان ابتدا شبکه‌ها را برای یافتن دستگاه‌های اینترنت اشیا آسیب‌پذیر اسکن می‌کنند و سپس با نصب نرم‌افزار مخرب روی این دستگاه‌ها، باتنت را شکل می‌دهند. مرحله فرمان و کنترل به مهاجمان امکان می‌دهد تا با باتها ارتباط مداوم داشته و آن‌ها را برای اقدامات آینده کنترل کنند. در مرحله حمله، مهاجمان با استفاده از باتها عملیات مخرب را انجام می‌دهند. مرحله پس از حمله نیز به انتشار بدافزار و گسترش باتنت اختصاص دارد [۲].

برای مقابله با باتنت‌ها، ایجاد سیستم‌های نظارت امنیتی به منظور شناسایی فعالیت‌های مخرب ضروری است. سازمان‌هایی که میزبان دستگاه‌های مختلف اینترنت اشیا هستند، به شناسایی دستگاه‌های آلوده شده توسط بدافزار باتنت علاقه‌مندند و تلاش می‌کنند تا فعالیت‌های مخرب را در فازهای اولیه یعنی پیش از حمله شناسایی کنند. از سوی دیگر، سازمان‌هایی که مورد حمله باتها قرار می‌گیرند، بر جلوگیری از ترافیک مخرب در فازهای حمله و پس از حمله تمرکز دارند. بنابراین، توسعه یک سیستم نظارت جامع که کل چرخه حیات باتنت را پوشش دهد، بسیار مهم است و نیازمند درک عمیقی از اقدامات مخرب در هر مرحله است [۲، ۴].

تحقیقات زیادی در زمینه تشخیص ناهنجاری‌های شبکه و نفوذ در اینترنت اشیا انجام شده است. سیستم‌های مبتنی بر امضا، توانایی شناسایی ترافیک مخرب شبکه را دارند اما در شناسایی حملات ناشناخته یا جدید محدودیت دارند. در مقابل، راه‌حل‌های مبتنی بر رفتار یا ناهنجاری، با تحلیل الگوهای رفتاری مخرب، می‌توانند حملات جدید را شناسایی کنند. یادگیری ماشین به عنوان یک راه‌حل کارآمد برای شناسایی انواع حملات جدید بدون نیاز به امضا شناخته می‌شود [۴].

رویکردهای یادگیری ماشین برای تشخیص حملات اینترنت اشیا متنوع‌اند، اما باید محدودیت‌های محاسباتی و پیچیدگی‌های شبکه‌های اینترنت اشیا را مد نظر قرار دهند. انتخاب ویژگی، به عنوان یک گام مهم در فرآیند یادگیری ماشین، می‌تواند عملکرد مدل‌های تشخیص نفوذ را بهبود بخشد و نیاز به منابع محاسباتی را کاهش دهد. این امر به ویژه در محیط‌های اینترنت اشیا که دارای منابع محدود هستند، اهمیت بیشتری دارد [۵].

روش‌های سنتی یادگیری ماشین ممکن است در مواجهه با داده‌های پیچیده مانند داده‌های نامتوازن یا داده‌های دارای نویز، عملکرد مطلوبی نداشته باشند. یادگیری ترکیبی [1] به عنوان یکی از روش‌های مؤثر برای افزایش دقت و انعطاف‌پذیری مدل‌های یادگیری ماشین شناخته می‌شود. این روش با ادغام ویژگی‌های مثبت چندین مدل یادگیری ماشین، توانایی تعمیم و مدیریت الگوهای پیچیده داده‌ها را بهبود می‌بخشد [۶].

در این مقاله، سیستمی برای تشخیص ناهنجاری در شبکه‌های اینترنت اشیا با استفاده از روش یادگیری ترکیبی معرفی شده است. استفاده از مجموعه داده Bot-IoT که شامل نمونه‌هایی از حملات پیشرفته‌ای همچون اسکن و DDoS است، به این سیستم امکان می‌دهد تا کاربردی‌تر و دقیق‌تر در محیط‌های واقعی عمل کند. برای طبقه‌بندی و انتخاب‌ویژگی‌ها، از دو الگوریتم قدرتمند یادگیری ترکیبی LightGBM و Random forest بهره برده شده است که کارایی و دقت سیستم را به شدت افزایش می‌دهند. در نهایت، از یکی از تکنیک‌های بهینه‌سازی بیزی به نام TPE [2] برای تنظیم مقادیر هایپرپارامترهای مدل‌ها استفاده شده است. تنظیم هایپرپارامترها نقش مهمی در بهینه‌سازی مدل‌های یادگیری ماشین برای دستیابی به عملکرد بهتر دارد. این فرآیند شامل انتخاب بهترین ترکیب از مقادیر هایپرپارامترها است که منجر به بالاترین دقت یا کمترین خطا می‌شود. همچنین، برای رفع چالش بزرگ عدم توازن داده‌ها از روش SMOTE [3] استفاده شده است. این روش با افزایش نمونه‌های کلاس اقلیت، به طور موثری مشکل عدم توازن داده‌ها را رفع می‌کند و به دقت بالاتر در تشخیص حملات منجر می‌شود.

ساختار مقاله به شرح زیر است: بخش ۲ به بررسی پیشینه تحقیق مرتبط با باتنت‌ها و روش‌های تشخیص آنها می‌پردازد. در بخش ۳، مجموعه داده و الگوریتم‌های استفاده شده مورد بررسی قرار می گیرد. در بخش ۴، روش پیشنهادی برای پیشگیری و تشخیص حملات باتنت در اینترنت اشیا توضیح داده می‌شود. بخش ۵ نتایج به دست آمده را مورد بررسی قرار می‌دهد. در نهایت، بخش ۶ به نتیجه‌گیری اختصاص دارد.

[1] Ensemble learning [2] Tree-Structured Parzen Estimator [3] Synthetic Minority Over-sampling Technique

۲ پیشینه تحقیق

در مقاله [۷]، الگوریتم‌های SVM[۱]، RNN[۲] و LSTM[۳] بر روی مجموعه داده Bot-IoT آموزش داده شده‌اند. انتخاب ویژگی‌ها با استفاده از دو روش ضریب همبستگی پیرسون (PCC[۴]) و آنتروپی انجام شده و ۱۰ ویژگی برتر از هر روش برای آموزش مدل‌ها انتخاب گردیده است. الگوریتم‌ها هم بر روی کل مجموعه داده بدون انتخاب ویژگی و هم بر روی داده‌های ۱۰ ویژگی برتر آموزش داده شده‌اند. نتایج نشان می‌دهد که مدل SVM با استفاده از ۱۰ ویژگی برتر دقت ٪۸۸ و با کل مجموعه داده دقت ٪۹۹ را به دست آورده است. مدل RNN دقت ٪۹۹ برای ۱۰ ویژگی برتر و دقت ٪۹۷ برای کل مجموعه داده را نشان می‌دهد. همچنین، مدل LSTM دقت ٪۹۹ برای ۱۰ ویژگی برتر و دقت ٪۹۸ برای کل مجموعه داده را به دست آورده است. یکی از چالش‌های این مجموعه داده، عدم توازن در تعداد نمونه‌های نرمال و حمله است. این عدم توازن در داده‌ها منجر به بایاس[۵] شدن مدل می‌شود. بایاس شدن مدل زمانی رخ می‌دهد که تعداد نمونه‌های متعلق به یک کلاس بسیار بیشتر یا کمتر از کلاس‌های دیگر باشد، که در نتیجه آن مدل تمایل پیدا می‌کند تا کلاس غالب را بهتر تشخیص دهد و در تشخیص کلاس‌های کمتر ضعیف عمل کند.

در مقاله [۴]، برای تشخیص حملات باتنت اینترنت اشیا از روش یادگیری عمیق استفاده کرده است. نویسندگان از معماری ResNet-18 به منظور تشخیص حملات باتنت در مراحل اسکن و حملات DDoS استفاده کرده‌اند. داده‌های مورد استفاده در این مقاله، از مجموعه داده‌های ScanLab و DDoSLab است که توسط خود نویسندگان جمع‌آوری شده است. برای انتخاب ویژگی‌های هر دو مجموعه داده، از الگوریتم Lo-gistic Regression استفاده شده و ۱۵ ویژگی برتر هر مجموعه داده برای آموزش مدل‌ها انتخاب شده است. مدل‌های آموزش‌داده شده به نام‌های ResNetScan و ResNetDDoS معرفی شده‌اند. این مدل‌ها علاوه بر آموزش و ارزیابی بر روی مجموعه داده‌های جمع‌آوری شده، بر روی سه مجموعه داده‌ی BoT-IoT، CICIDS-19 و CICIDS-17 نیز آموزش و ارزیابی شده‌اند. نتایج به دست آمده نشان می‌دهد که مدل‌های ResNetScan-1 و ResNetDDoS-1 که روی مجموعه داده‌های ScanLab و DDoSLab آموزش داده شده‌اند، عملکرد بهتری هم بر روی مجموعه داده‌ی آموزشی و هم بر روی سه مجموعه داده‌ی دیگر داشته‌اند. مدل‌های پیشنهادی به میانگین دقت ٪۹۸/۸۹ دست یافته است. هایپرپارامترهای مورد استفاده برای آموزش مدل در مقاله ذکر شده‌اند، اما روشی که برای یافتن آن‌ها استفاده شده، در مقاله اشاره نشده است. همچنین تعداد نمونه‌های مورد استفاده در هر کلاس نیز در مقاله ذکر نشده است.

در مقاله [۸]، از مدل‌های یادگیری ماشین KNN[۶]، MLP-ANN[۷] و Naïve Bayes برای آموزش مدل‌ها بر روی مجموعه داده Bot-IoT استفاده شده است. به منظور رفع مشکل عدم توازن داده‌ها، از تکنیک SMOTE بهره گرفته شده است. نویسندگان مقاله، تاثیر استفاده و عدم استفاده از روش SMOTE را در نتیجه‌ی طبقه‌بندی مقایسه کرده‌اند. به این ترتیب، مدل‌های پیشنهادی را یک بار با مجموعه داده‌ای که روش SMOTE بر روی آن اعمال شده و یک بار دیگر با داده‌های نامتوازن آموزش داده‌اند. نتایج نشان می‌دهد که الگوریتم KNN در هر دو حالت عملکرد بهتری نسبت به دو الگوریتم دیگر دارد. مدل KNN برای داده‌های نامتوازن دقت ٪۹۹/۶ و برای داده‌های متوازن دقت ٪۹۲/۱ را به دست آورده است. همچنین، امتیاز ROC_AUC برای داده‌های نامتوازن ٪۹۹/۲ و برای داده‌های متوازن ٪۹۲/۲ بوده است. نتایج سایر مدل‌ها نیز در مقاله ارائه شده است. برای انتخاب ویژگی، از روش Chi-Square که یکی از روش‌های انتخاب ویژگی فیلتر است، استفاده شده و ۸ ویژگی برتر مجموعه داده برای آموزش مدل‌ها انتخاب شده‌اند. روش‌های فیلتر یکی از رایج‌ترین تکنیک‌های انتخاب ویژگی هستند که به صورت مستقل از الگوریتم یادگیری عمل کرده و ویژگی‌ها را بر اساس خصوصیات ذاتی داده‌ها رتبه‌بندی می‌کنند. در این مقاله، هیچگونه بهینه‌سازی برای هیچ‌یک از الگوریتم‌ها انجام نشده است.

در مقاله [۹]، مقایسه‌ای بین سه مدل یادگیری ترکیبی و مدل پیشنهادی ELBA-IoT صورت گرفته است. برای پیاده‌سازی ELBA-IoT، ابتدا سه مدل یادگیری ترکیبی Adaboost، RUSboost و Bagged به طور مستقل آموزش داده شدند. سپس با استفاده از روش رای‌گیری، نتایج پیش‌بینی این سه مدل ترکیب شدند. مدل ELBA-IoT با این سه مدل مقایسه شده است و به دقت ٪۹۹/۶ دست یافته که در مقایسه با سایر مدل‌ها نتیجه بهتری است. مجموعه داده مورد استفاده برای آموزش مدل‌ها، N-BaIoT بوده است. برای انتخاب ویژگی‌ها، از روش PCC استفاده شده است. یکی از محدودیت‌های مجموعه داده N-BaIoT، محدودیت پروتکل‌های استفاده شده در آن است؛ در این مجموعه داده تنها از پروتکل‌های TCP و UDP استفاده شده است. همچنین، این مجموعه داده دربرگیرنده ویژگی‌های مهمی مانند پورت مبدأ و مقصد و بسیاری از ویژگی‌های حیاتی دیگر نیست که می‌تواند بر دقت و کارایی مدل تأثیرگذار باشد. هرچند مقادیر هایپرپارامترهای استفاده شده در مقاله ذکر شده است، اما روش و دلایل انتخاب این مقادیر توضیح داده نشده است.

در مقاله [۱۰] برای تشخیص حملات باتنت در اینترنت اشیا از الگوریتم یادگیری ترکیبی Random Forest استفاده شده است. مجموعه داده‌های مورد استفاده برای آموزش و ارزیابی مدل شامل Bot-IoT و NF-UNSW-NB15_V2 می‌باشد. فرآیند انتخاب ویژگی‌ها با بهره‌گیری از الگوریتم Isolated Forest برای تشخیص ناهنجاری در داده‌ها و روش PCC برای کاهش ابعاد ویژگی‌ها صورت گرفته است. روش‌های انتخاب ویژگی در سه حالت مختلف انجام شده است. در حالت اول، تنها از Iso-lated Forest استفاده شده و سپس مدل آموزش داده شده است. در حالت دوم، ابتدا Isolated Forest اعمال شده و سپس از PCC برای کاهش ابعاد ویژگی‌ها استفاده شده است. در حالت سوم، ابتدا از PCC برای

[1]Support Vector Machine [2]Recurrent Neural Network [3]Long Short--Term Memory [4]Pearson Correlation Coefficient [5]Bias [6]K-Nearest Neighbor [7]Multi-layer Perceptron-Artificial Neural Network

کاهش ابعاد استفاده شده و سپس Isolated Forest اعمال شده است. مدل پیشنهادی به دقت ٪۹۹٫۹۸ برای مجموعه داده Bot-IoT و ٪۹۹٫۳۰ برای مجموعه داده NF-UNSW-NB15_V2 رسیده است. یکی از چالش‌های مهم این مقاله، نامتوازن بودن داده‌های هر کلاس در هر دو مجموعه داده است. در مجموعه داده Bot-IoT، تعداد نمونه‌های نرمال ۹۵۳۴ و تعداد نمونه‌های مربوط به حملات ۷۳۳۶۰۹۰۰ می‌باشد. همچنین در مجموعه داده NF-UNSW-NB15_V2، تعداد نمونه‌های نرمال ۱۵۵۰۷۱۲ و تعداد نمونه‌های مربوط به حملات ۷۲۴۰۶ است.

در مقاله [۱۱]، مقایسه‌ای بین چهار الگوریتم یادگیری ماشین Decision Tree، Naïve Bayes، Logistic Regression، KNN و دو الگوریتم یادگیری ترکیبی XGboost و Random Forest انجام شده است. مجموعه داده استفاده شده در این مقاله Ton-IoT است که شامل ٪۶۵٫۰۷ داده‌های نرمال و ٪۳۴٫۹۳ داده‌های حمله است. برای انتخاب ویژگی‌ها از روش Information Gain، که یکی از روش‌های فیلتر انتخاب ویژگی است، استفاده شده است. در این مقاله، داده‌ها به صورت دو کلاسه و چند کلاسه طبقه‌بندی شده‌اند. نتایج نشان می‌دهد که الگوریتم‌های Logistic Regression و Naïve Bayes عملکرد مناسبی در طبقه‌بندی دو کلاسه نداشته و به ترتیب دقت ٪۷۴ و ٪۴۳ را به دست آورده‌اند. همچنین در طبقه‌بندی چند کلاسه، این دو الگوریتم به ترتیب به دقت ٪۶۵ و ٪۱۲ رسیده‌اند که نشان‌دهنده عملکرد ضعیف آن‌ها در این نوع طبقه‌بندی است. از سوی دیگر، الگوریتم‌های Decision Tree، Random Forest، KNN و XGboost به دقت بالایی در هر دو نوع طبقه‌بندی دست یافته‌اند. در طبقه‌بندی چند کلاسه با استفاده از ۱۸ ویژگی، این الگوریتم‌ها به دقت ٪۹۸ دست یافته‌اند. همچنین، در مقاله به تاثیر تعداد ویژگی‌ها بر دقت مدل‌ها پرداخته شده است. در این مقاله، هیچگونه بهینه‌سازی بر روی مدل‌های پیاده‌سازی شده انجام نشده و هیچ اقدامی برای توازن داده‌ها بین کلاس‌های مختلف صورت نگرفته است.

در مقاله [۱۲]، مقایسه‌ای بین مدل پیشنهادی به نام Extra Tree Voting Classifier و سایر مدل‌های یادگیری ماشین انجام شده است. مدل پیشنهادی برای آموزش از ترکیب الگوریتم‌های Extra Tree و Random Forest استفاده می‌کند. سه روش برای انتخاب ویژگی‌ها مورد بررسی قرار گرفته است: در روش اول، انتخاب ویژگی‌ها از طریق الگوریتم Extra Tree انجام شده و سپس مدل آموزش داده شده است. روش دوم از الگوریتم Random Forest برای انتخاب ویژگی‌ها استفاده کرده است. در روش سوم، ویژگی‌هایی که به طور مشترک بین دو الگوریتم مذکور انتخاب شده‌اند، مورد استفاده قرار گرفته‌اند. نتایج مقایسه بین این سه روش نشان می‌دهد که روش سوم عملکرد بهتری را در مقایسه با دو روش دیگر دارد. علاوه بر این، مدل پیشنهادی با چندین الگوریتم دیگر از جمله Decision Tree، Extra Tree، Naïve Bayes، KNN و Random Forest مقایسه شده است. نتایج این مقایسه نشان دهنده عملکرد بهتر مدل پیشنهادی نسبت به سایر مدل‌ها می‌باشد. برای آموزش مدل، از مجموعه داده MedBIoT استفاده شده است که شامل داده‌های مربوط به سه باتنت اینترنت اشیا با نام‌های Mirai، Bashlite و Torii می‌باشد. این مدل توانسته دقت ٪۹۹٫۹۱ را برای این مجموعه داده به دست آورد. از مشکلات موجود در مجموعه داده MedBIoT، محدودیت ویژگی‌ها آن از جمله عدم وجود پورت مبدا و مقصد و نبود آدرس IP مبدا و دیگر ویژگی‌های مهمی است که برای تشخیص حمله‌ها ضروری هستند.

در مقاله [۱۳] نیز از رویکرد یادگیری ترکیبی برای توسعه مدل استفاده شده است. این مقاله به کارگیری الگوریتم‌های Naïve Bayes و LightGBM به عنوان طبقه‌بندهای پایه پرداخته است و سپس از نتایج پیش‌بینی این الگوریتم‌ها به عنوان ورودی برای آموزش یک طبقه‌بند متا به نام TPE-LightGBM استفاده کرده است. مدل پیشنهادی بر روی دو مجموعه داده N-BaIoT و UNSW-NB15 آموزش و ارزیابی شده است. برای انتخاب ویژگی‌ها، از روش امتیاز فیشر که یک تکنیک فیلتر برای انتخاب ویژگی است، استفاده شده و ۳۰ ویژگی برتر مجموعه داده ها برای آموزش مدل انتخاب شده است. این مدل به دقت ٪۹۹٫۶۸ دست یافته است. یکی از چالش‌های مطرح شده در این مقاله، مشکلات موجود در مجموعه داده N-BaIoT است که پیش‌تر به برخی از آن‌ها اشاره شده بود. علاوه بر این، عدم توازن در تعداد داده‌های مربوط به حملات مختلف نیز از دیگر مشکلات این مجموعه داده است.

در مقاله [۱۴]، دو مدل 1D-CNN [1] و LSTM-CNN برای تشخیص حملات باتنت اینترنت اشیا با استفاده از مجموعه داده N-BaIoT به کار گرفته شده است. برای انتخاب ویژگی‌ها، از روش ANOVA استفاده شده و ۱۰ ویژگی برتر برای آموزش مدل‌ها انتخاب شده‌اند. مدل LSTM-CNN به امتیاز $F1$ معادل ٪۸۹ و مدل 1D-CNN به امتیاز $F1$ معادل ٪۸۳ ودست یافته‌اند. این مدل‌ها به طور کلی عملکرد خوبی در تشخیص نوع حملات و باتنت‌های حمله‌کننده نشان داده‌اند، به جز در موارد مربوط به حملات Bashlite TCP و Bashlite UDP که عملکرد مطلوبی نداشته‌اند.

مقاله [۱۵] از مجموعه داده Bot-IoT برای آموزش دو مدل یادگیری ماشین، هشت مدل یادگیری ترکیبی، و پنج مدل یادگیری عمیق بهره گرفته است. مدل‌های یادگیری ماشین شامل SVM و KNN هستند. مدل‌های یادگیری عمیق نیز شامل LSTM، GRU [2]، RNN، MLP و ANN می‌باشند و مدل‌های ترکیبی مورد استفاده شامل Random Forest، Extra Tree، LightGBM، XGboost، Catboost، Adaboost، Histboost و Bagged می شود. برای انتخاب ویژگی‌ها از ضریب همبستگی و الگوریتم Random Forest استفاده شده است. همچنین، برای رفع مشکل عدم توازن داده‌ها، روش SMOTE به کار گرفته شده و تاثیر استفاده از این روش در مقایسه با عدم استفاده از آن بررسی شده است. در این مقاله، بدترین دقت مربوط به الگوریتم Adaboost است که در صورت عدم استفاده از SMOTE دقت ٪۱۲ و در صورت استفاده از آن دقت ٪۴۳ را نشان می‌دهد. بهترین دقت در حالت عدم استفاده از SMOTE مربوط

[1] One Dimensional-Convolutional Neural Network [2] Gated Recurrent Unit

به مدل‌های XGboost و Catboost است که به ترتیب به دقت ٪۹۸٫۵ و ٪۹۸٫۱۹ دست یافته‌اند. در حالت استفاده از SMOTE، بهترین دقت مربوط به مدل‌های XGboost و LightGBM است که به ترتیب دقت ٪۹۷٫۹ و ٪۹۸ را نشان می‌دهند. در این مقاله، هایپرپارامترهای استفاده شده و روش بهینه‌سازی آنها برای هیچ یک از مدل‌ها ذکر نشده است. جدول ۱، خلاصه از مطالعات بررسی شده در این بخش را نشان می دهد.

۳ مجموعه داده و الگوریتم‌های مورد استفاده

۱.۳ یادگیری ترکیبی

یادگیری ترکیبی یک روش در یادگیری ماشین است که به ترکیب چندین مدل برای بهبود دقت و عملکرد کلی می‌پردازد. ایده اصلی در یادگیری ترکیبی این است که با ادغام چندین مدل مختلف، می‌توان نقاط ضعف هر مدل تکی را کاهش داد و به یک مدل قدرتمندتر و قابل اعتمادتر دست یافت. در این روش، مدل‌های مختلف به صورت مستقل آموزش داده می‌شوند و سپس خروجی‌های آنها با استفاده از تکنیک‌های مختلف ترکیب می‌گردد. این ترکیب می‌تواند به صورت میانگین‌گیری، رأی‌گیری اکثریت یا ترکیب وزنی خروجی‌ها باشد. هدف اصلی یادگیری ترکیبی افزایش دقت مدل و کاهش خطر بیش‌برازش و کم‌برازش است [۶].

۲.۳ مجموعه داده BoT-IoT

مجموعه داده BoT-IoT در سال ۲۰۱۸ توسط دانشگاه نیو ساوت ولز با هدف فراهم آوردن یک نمایش واقعی از حملات باتنت بر روی دستگاه‌های اینترنت اشیا توسعه یافت. این مجموعه شامل داده‌های مربوط به حملات DDoS، DoS، سرقت داده و اسکن است که بر روی پنج دستگاه مختلف اجرا شده است. در این مجموعه داده، حملات اسکن به دو دسته اصلی اسکن پورت و اثرانگشت‌گیری سیستم‌عامل تقسیم می‌شوند. حملات DDoS در BoT-IoT شامل بهره‌برداری از مکانیزم‌های پروتکل‌های اینترنتی برای تخلیه منابع پردازشی و حافظه هستند که منجر به عدم توانایی ماشین هدف در پاسخگویی به درخواست‌ها می‌شود. پروتکل‌های مورد استفاده در این حملات شامل TCP، UDP، و HTTP می‌باشد [۷].

از ویژگی‌های برجسته BoT-IoT تنوع بالای حملات مختلف است که بر روی دستگاه‌های اینترنت اشیا توسط باتنت‌ها انجام می‌شود. این مجموعه داده شامل حملاتی است که بر روی دستگاه‌های خانگی و صنعتی اینترنت اشیا اجرا شده‌اند، که این تنوع دستگاه‌ها یکی از نقاط قوت اصلی آن محسوب می‌شود. علاوه بر این، BoT-IoT تلاش می‌کند تا سناریوهای حملات واقعی را شبیه‌سازی کند که به پژوهشگران امکان می‌دهد روش‌های دفاعی خود را در شرایط نزدیک به واقعیت آزمایش کنند. حجم بزرگ داده‌های موجود در این مجموعه به محققان اجازه می‌دهد تحلیل‌های عمیق‌تری انجام دهند و به نتایج قابل اعتمادتری برسند. در مقایسه با مجموعه داده‌های UNSW-NB15 و CICIDS-17 که بر حملات عمومی تمرکز دارند، BoT-IoT به طور خاص برای بررسی حملات باتنت بر روی دستگاه‌های اینترنت اشیا طراحی شده است. همچنین،در مقایسه با مجموعه داده‌های N-BaIoT و MedBIoT، این مجموعه داده دارای ویژگی‌های کاملی است که استفاده از آن را در مدل‌های یادگیری ماشین تسهیل می‌کند. مستندات جامع و منابع آموزشی همراه این مجموعه داده به محققان کمک می‌کند تا به سرعت با آن آشنا شوند و از آن بهره‌برداری کنند.

۳.۳ الگوریتم LightGBM

الگوریتم LightGBM متشکل از چندین Decision Tree است که به صورت ترتیبی در فرآیند تقویت گرادیانی [1] آموزش داده می‌شوند. در این روش، هر درخت تلاش می‌کند تا خطای درخت قبلی را کاهش دهد و به این ترتیب مدل نهایی بهینه‌تر می‌شود. LightGBM با معرفی چندین روش نوآورانه از جمله الگوریتم ترکیب ویژگی‌های مستقل، استراتژی برگ‌محور با محدودیت عمق و الگوریتم هیستوگرام، از درخت‌های تقویت گرادیانی سنتی بهره می‌گیرد. ایده اصلی این الگوریتم، ترکیب خطی M درخت ضعیف برای تبدیل آن‌ها به درخت‌های قوی است. این فرآیند با استفاده از معادله زیر به دست می‌آید:

$$F(x) = \sum_{m=1}^{M} f_m(x) \qquad (1)$$

در اینجا، $F(x)$ خروجی نهایی و $f_m(x)$ خروجی مربوط به mامین درخت ضعیف است. بهبودهایی که توسط LightGBM ارائه می‌شود، به طور قابل توجهی مصرف حافظه و زمان آموزش را کاهش می‌دهد و در عین حال، دقت مدل را در مقایسه با الگوریتم‌های تقویت گرادیانی پیشین مانند XGboost افزایش می‌دهد. این ویژگی‌ها LightGBM را برای مسائلی با داده‌های بزرگ، مانند طبقه‌بندی نوع حمله، بسیار مناسب می‌سازد [۱۳].

۴.۳ الگوریتم Random Forest

Forest یکی از الگوریتم‌های یادگیری ترکیبی است که می‌توان از آن در مسائل رگرسیون و طبقه‌بندی استفاده کرد. در این روش، مدلی متشکل از چندین Decision Tree ایجاد می‌شود که هر کدام به صورت مستقل و با استفاده از نمونه‌های مختلف از داده‌های ورودی آموزش می‌بینند. نتایج نهایی با ترکیب خروجی‌های حاصل از هر یک از این درخت‌ها به دست می‌آیند. در مسائل طبقه‌بندی، نتیجه نهایی از طریق رأی‌گیری اکثریت تعیین می‌شود که به صورت Prf در معادله زیر تعریف می‌گردد:

$$P_r f = mode\{DT_1(s), DT_2(s), \ldots, DT_c(s)\} \qquad (2)$$

که در آن $(DT_1(s)DT_2(s)\ldots DT_c(s))$ تعداد c درخت تصمیم‌گیری (DT) در جنگل هستند [۱۲].

[1] Gradient Boosting

جدول ۱. خلاصه کارهای مرتبط

مقاله	سال انتشار	مجموعه داده	الگوریتم	انتخاب ویژگی	بهینه‌سازی هایپرپارامترها	توازن تعداد داده ها هر کلاس
[۷]	۲۰۱۹	Bot-IoT	SVM, RNN, LSTM	PCC, Entropy	✓	✗
[۴]	۲۰۲۱	ScanLab, DDosLab	CNN (ResNet-18)	Logistic Regression	✓	✗
[۸]	۲۰۲۱	Bot-IoT	KNN, Naïve Bayes, MLP-ANN	Chi-Square	✗	✓
[۹]	۲۰۲۲	N-BaIoT	Voting Classifier (Adaboost, RUSboost, Bagged)	PCC	✓	✗
[۱۰]	۲۰۲۲	NF-UNSW-NB15_V2, Bot-IoT	Random Forest	Isolated Forest, PCC	✓	✗
[۱۱]	۲۰۲۲	Ton-IoT	Decision Tree, Naïve Bayes, Random Forest, KNN, Logistic Regression, XGboost	Information Gain	✗	✗
[۱۲]	۲۰۲۳	MedBIoT Voting	Classifier (Extra Tree, Random Frest)	Extra Tree, Random Forest	✓	✓
[۱۳]	۲۰۲۳	N-BaIoT, UNSW-NB15	Stacked Classifier (Naïve Bayes, LightGBM)	Fisher Score	✓	✗
[۱۴]	۲۰۲۳	N-BaIoT	1D-CNN, LSTM-CNN	ANOVA	✓	✗
[۱۵]	۲۰۲۳	Bot-IoT	KNN, SVM, Random Forest, XGboost, Catboost, Adaboost, LightGBM, Histboost, Bagged, Extra Tree, LSTM, GRU, RNN, MLP, ANN	Correlation Coefficient, Random Forest	✗	✓

۵.۳ روش بهینه‌سازی TPE

روش TPE یک تکنیک بهینه‌سازی است که برای انتخاب هایپرپارامترها از قضیه بیز استفاده می‌کند. در طی آموزش مدل، از این بهینه‌ساز برای تنظیم هایپرپارامترها با هدف بهبود عملکرد طبقه‌بندی و جلوگیری از بیش‌برازش بهره می‌بریم. این روش قادر است از طیف گسترده‌ای از متغیرها در فضای جستجو، شامل مقادیر یکنواخت، لگاریتم–یکنواخت و توزیع نرمال، پشتیبانی کند. فرآیند بهینه‌سازی TPE به صورت زیر است:

$$x^* = \text{argmin} f(x) \qquad (۳)$$

در معادله (۳)، $f(x)$ تابع هدف است و $x*$ مجموعه‌ای از هایپرپارامترها در فضای جستجو x است که مقدار تابع هدف $f(x)$ را به حداقل می‌رساند. هدف الگوریتم بهینه‌سازی بیزی TPE، بهینه‌سازی هایپرپارامترهای مدل با مدل‌سازی همزمان $p(x|y)$ و $p(y)$ است. در اینجا، $p(y)$ توزیع راه‌حل‌ها و $p(x|y)$ توزیع پارامتر x در صورت معلوم بودن راه‌حل است. علاوه بر این، برای هدایت جستجو در فضای پیکربندی از معیار بهبود مورد انتظار استفاده می‌شود. محاسبه $p(x|y)$ مطابق معادله (۴) انجام می‌شود:

$$p(x|y) = \begin{cases} l(x) & y < y' \\ g(x) & y \geq y' \end{cases} \qquad (۴)$$

که در آن y' آستانه‌ای است که از پیش تعریف شده است، $l(x)$ نمایانگر برآورد چگالی است که تابع اتلاف مقدار مشاهده‌شده $x(i)$ کوچک‌تر از y' است، و $g(x)$ نمایانگر ترکیب چگالی است که تابع اتلاف مقدار مشاهده‌شده $x(i)$ بزرگ‌تر از y' است [۱]. الگوریتم ۱، مراحل پیاده سازی TPE برای دو مدل LightGBM و Random Forest نشان می دهد.

الگوریتم ۱ مراحل پیاده سازی TPE برای بهینه سازی هایپرپارامترها

Input:hyperparameters $X = \{x_1, x_2, \ldots, x_n\}$; Model M; C to record the configurations and loss

1: Initialize M_0; $C = \{\}$
2: **for** $(n = 1, 2, 3, \ldots)$ **do**
3: Find the set of optimal hyperparameters x^* by minimizing the M_{n-1}:
4: $x^* = \text{argmin} M_{n-1}(x)$
5: Calculate the loss c under the set x^* of loss function $L : c = L(x^*)$
6: Store x^* and loss c in C
7: Fit a new model $M_{new} = M \cup (x^*, C)$
8: **end for**

Output:set of optimal hyperparameters of LightGBM and Random Forest x^* and minimum loss c in C

۴ روش پیشنهادی

پیاده‌سازی یک مدل یادگیری ترکیبی شامل مراحل مهمی است که هر یک به‌طور مستقیم بر موفقیت نهایی مدل تأثیرگذارند. این مراحل شامل تعیین مسئله و جمع‌آوری داده‌ها، پیش پردازش داده‌ها جمع‌آوری شده، انتخاب الگوریتم یادگیری مناسب برای آموزش مدل، ارزیابی مدل و در نهایت تنظیم و بهبود مدل در صورت عدم رضایت از عملکرد مدل می‌شود.

در فرآیند پیش پردازش مجموعه داده Bot-IoT، ابتدا تمام داده‌های حاوی مقادیر گم‌شده و تکراری پاکسازی می‌شوند. این مجموعه داده شامل ۴۶ ویژگی است که ۳۷ ویژگی عددی و ۹ ویژگی متنی دارند. نمونه‌های حمله با مقدار یک و نمونه‌های نرمال با مقدار صفر برچسب‌گذاری شده‌اند تا برای آموزش و ارزیابی مدل‌های یادگیری ماشین به روش طبقه‌بندی دودویی استفاده شوند. علاوه بر این، ویژگی‌های دسته‌بندی حملات و زیردسته‌ها نیز معرفی شده‌اند که برای آموزش و ارزیابی مدل‌های طبقه‌بندی چندکلاسه مورد استفاده قرار می‌گیرند. از آنجا که فقط از داده‌های نرمال، اسکن و DDoS استفاده شده است، ویژگی‌های Attack و Category از مجموعه داده حذف شده‌اند. ویژگی‌های stime و ltime نشان‌دهنده زمان شروع و زمان آخرین بسته برای هر نمونه هستند. با این حال، از آنجا که این اطلاعات در ویژگی‌های dur، rate، srate و drate گنجانده شده است، ویژگی‌های stime و ltime به منظور جلوگیری از بیش‌برازش حذف شدند. ویژگی‌های saddr و daddr که مربوط به آدرس‌های IP مبدأ و مقصد هستند نیز حذف شده‌اند، زیرا آدرس‌های IP خصوصی می‌توانند از شبکه‌ای به شبکه دیگر متفاوت باشند. این امر می‌تواند باعث ناکارآمدی مدل‌ها در صورت تغییر آدرس‌های IP در مجموعه داده‌های جدید شود. ستون‌های متنی flgs، proto و state نیز حذف و معادل عددی آن‌ها flgs_number، proto_number، state_number استفاده شده است. نمونه‌هایی که از پروتکل ICMP استفاده می‌کنند، برای ویژگی‌های sport و dport ممکن است مقادیر هگزادسیمال داشته باشند. بنابراین، مقادیر هگزادسیمال این ویژگی‌ها به ۱- تغییر داده شده‌اند تا نشان‌دهنده مقدار نامعتبر باشند. برچسب داده‌های مربوط به اثرانگشت‌گیری سیستم‌عامل و اسکن سرویس به Scan و برچسب داده‌های مربوط به UDP، TCP و HTTP به DDoS تغییر داده شده است. سپس برچسب‌ها با استفاده از Label Encoder کدگذاری شده و نرمال‌سازی Min-Max برای قرار دادن داده‌ها در محدوده بین صفر و یک اعمال شده است. یکی از چالش‌های این مجموعه داده، عدم توازن در تعداد نمونه‌های هر کلاس است. برای رفع این مشکل، از روش SMOTE استفاده شده است. این روش با تولید مصنوعی نمونه‌های جدید برای کلاس اقلیت، تنوع داده‌های این کلاس را افزایش می‌دهد و به بهبود تعادل مجموعه داده کمک می‌کند. جدول ۲، تعداد داده ها مجموعه داده قبل و بعد از پیاده سازی روش SMOTE نشان می دهد.

پس از ایجاد نمونه‌های جدید توسط SMOTE، با استفاده از تابع `np.random.permutation` از کتابخانه‌ی NumPy، ترتیب ردیف‌های مجموعه داده به‌صورت تصادفی تغییر می‌کند تا نمونه‌ها شافل شوند. این کار می‌تواند در کاهش سوگیری مدل‌های یادگیری ماشین، به ویژه هنگام توزیع تصادفی نمونه‌ها برای آموزش و تست، بسیار مؤثر باشد.

جدول ۲. تعداد داده های هر دسته بندی قبل و بعد SMOTE

دسته بندی	قبل SMOTE	بعد SMOTE
نرمال	۴۷۷	۳۵۷۶۸۸۴
اسکن	۹۱۰۸۲	۳۵۷۶۸۸۴
DDoS	۳۵۷۶۸۸۴	۳۵۷۶۸۸۴

جدول ۳. مقادیر هایپرپارامترهای انتخاب شده در مدل‌ها

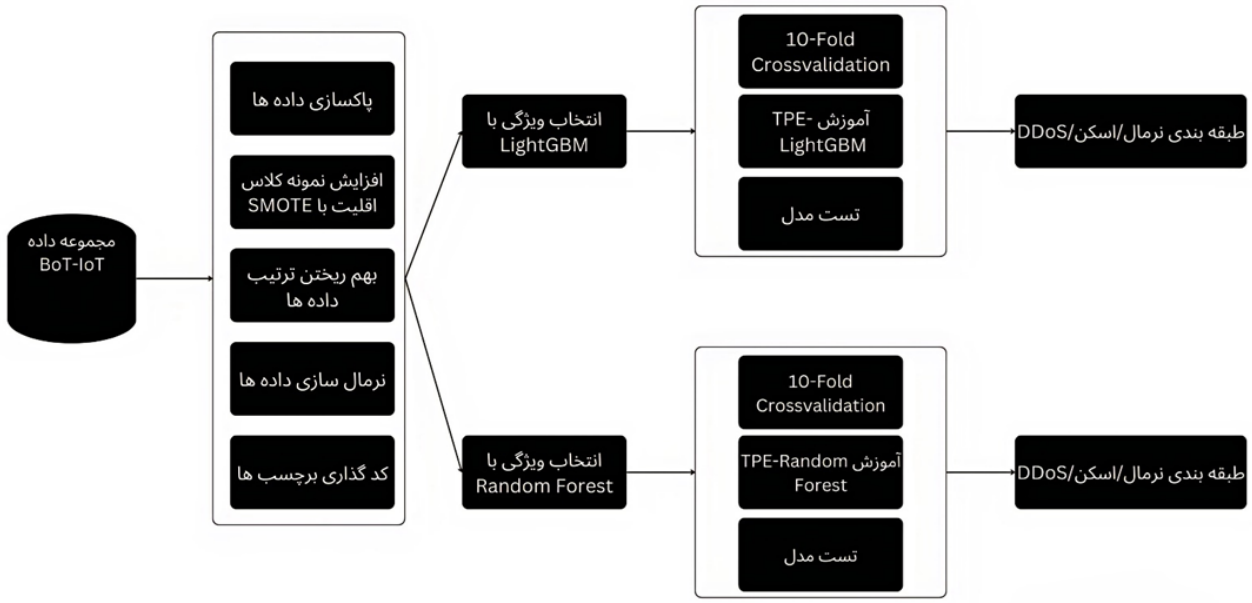
Random Forest		LightGBM	
max_leaf_nodes	۴۸۰	learning_rate	۰٫۰۰۷۴۰۲۹ ۸۸۵۷۷۰۸۷۸۲۲
n_estimators	۲۷۱	n_estimators	۲۶۶
max_depth	۴۵۶	num_leaves	۳
		min_child_samples	۱۱۸

برای انتخاب ویژگی، از دو الگوریتم مبتنی بر یادگیری ترکیبی استفاده شد. به این منظور، از لیست‌های اهمیت ویژگی[1] تولید شده توسط الگوریتم‌های Random Forest و LightGBM به عنوان مبنای رتبه‌بندی ویژگی‌ها استفاده شد. سپس، ۲۰ ویژگی با بالاترین رتبه از هر لیست انتخاب شدند.

برای یافتن بهترین مقادیر هایپرپارامترها در الگوریتم‌های یادگیری ترکیبی انتخابی، از روش TPE استفاده شد. Hyperopt، که یک کتابخانه متن‌باز پایتون برای بهینه‌سازی بیزی است، از TPE بهره می‌گیرد. اگرچه کتابخانه‌های متعددی برای بهینه‌سازی بیزی وجود دارند، اما Hyperopt به دلیل اینکه تنها کتابخانه‌ای است که به طور خاص TPE را ارائه می‌دهد، متمایز است. جدول ۳، لیست هایپرپارامترها هر مدل به همراه مقدارشان را نمایش می‌دهد.

برای آموزش مدل‌ها نیز از دو الگوریتم LightGBM و Random Forest، استفاده شده است. داده‌ها به نسبت ۷۰ درصد برای آموزش و ۳۰ درصد برای تست تقسیم‌بندی شدند. همچنین برای ارزیابی مدل‌های پیشنهادی از روش K-Fold Cross-Validation با K برابر ۱۰ استفاده شده است. K-Fold Cross-Validation روشی برای ارزیابی مدل‌های یادگیری ماشین است که داده‌ها را به K بخش تقسیم می‌کند. در هر مرحله، یک بخش برای تست و بقیه برای آموزش استفاده می‌شوند. این فرآیند K بار تکرار شده و در نهایت میانگین نتایج ارزیابی به عنوان عملکرد کلی مدل در نظر گرفته می‌شود. شکل ۲، مراحل پیاده سازی مدل پیشنهادی را نشان می دهد.

[1] Feature importance



شکل ۲. مراحل پیاده سازی مدل‌های پیشنهادی

۵ ارزیابی نتایج

برای ارزیابی عملکرد مدل‌ها، از معیارهای دقت [۱]، صحت [۲]، فراخوانی [۳] و امتیاز $F1$ استفاده شده است. این معیارها به منظور سنجش کارایی مدل در مسائل طبقه‌بندی به کار می‌روند. TP [۴] حالتی که مدل به درستی یک جریان حمله را به عنوان حمله شناسایی می‌کند. TN [۵] حالتی که مدل به درستی یک جریان عادی را به عنوان عادی شناسایی می‌کند. FP [۶] حالتی که مدل به اشتباه یک جریان عادی را به عنوان حمله شناسایی می‌کند. FN [۷] حالتی که مدل به اشتباه یک جریان حمله را به عنوان عادی شناسایی می‌کند.

$$\text{Accuracy} = \frac{TP+TN}{Total} \times ۱۰۰ \quad (۵)$$

$$\text{Precision} = \frac{TP}{TP+FP} \times ۱۰۰ \quad (۶)$$

$$\text{Recall} = \frac{TP}{TN+FN} \times ۱۰۰ \quad (۷)$$

$$\text{F1-score} = ۲ \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \times ۱۰۰ \quad (۸)$$

جدول ۴ نتایج ارزیابی مدل‌ها با استفاده از معیارهای تعیین‌شده را نشان می‌دهد. همچنین میانگین دقت به دست آمده در مرحله K-Fold Cross-Validation برای مدل LightGBM برابر با ۹۹٫۰۱٪ و برای مدل Random Forest برابر با ۹۹٪ است. نرخ خطای هر دو مدل به طرز چشمگیری پایین است؛ LightGBM با نرخ خطای ۰٫۹۸٪ و Random

[1]Accuracy [2]Precision [3]Recall [4]True Positive [5]True Negative
[6]False Positive [7]False Negative

جدول ۴. نتایج مدل پیشنهادی

معیار ارزیابی	LightGBM	Random Forest
Accuracy	۹۹٫۰۲٪	۹۹٫۹۹٪
Precision	۹۹٫۰۲٪	۹۹٫۹۹٪
Recall	۹۹٫۰۲٪	۹۹٫۹۹٪
F1-score	۹۹٫۰۲٪	۹۹٫۹۹٪

جدول ۵. مقایسه زمان آموزش و پیشبینی مدل‌ها

الگوریتم	زمان آموزش	زمان پیشبینی
LightGBM	۱۶۸۱ ثانیه	۷۴ ثانیه
Random Forest	۱۳۶۶۴ ثانیه	۲۱۹ ثانیه

Forest با نرخ خطای ۰٫۰۱٪، نشان‌دهنده دقت بالای این مدل‌ها هستند. جدول ۵، مدت زمان آموزش و پیشبینی دو مدل پیشنهادی ما را نشان می‌دهد.

در ادامه، مقایسه‌ای بین مدل‌های مختلف در [۷]، [۸]، [۱۰] و [۱۵] با مدل پیشنهادی انجام شده است. جدول ۶ نتایج مدل های مورد بررسی را نمایش می دهد. همه مدل‌های مقایسه‌شده بر روی مجموعه داده Bot-IoT آموزش دیده و ارزیابی شده‌اند. مقاله [۷] از الگوریتم‌های SVM، LSTM و RNN استفاده کرده و به نتایج قابل‌توجهی دست یافته است. با این حال، به منظور افزایش سرعت آموزش، تعداد اپوک‌های مدل یادگیری عمیق به ۴ کاهش یافته است. این اقدام، اگر چه سرعت را افزایش داده، اما منجر به کم‌برازش مدل در پیش‌بینی داده‌های جدید می‌شود که ضعف قابل توجهی است. مقاله [۸] با استفاده از مدل‌های KNN، Naïve

جدول ۶. مقایسه دقت مدل‌های پیشنهادی با سایر مدل‌های موجود

مقاله	الگوریتم	دقت
[۷]	SVM	۸۸٫۳۷٪
	LSTM	۹۹٫۷۴٪
	RNN	۹۹٫۷۴٪
[۸]	KNN	۹۲٫۱٪
	Naïve Bayes	۵۱٫۵٪
	MLP-ANN	۸۵٫۸٪
[۱۰]	Random Forest	۹۹٫۹۸٪
[۱۵]	LightGBM	۹۸٫۲۴٪
	Random Forest	۹۴٫۲۵٪
مدل پیشنهادی	LightGBM	۹۹٫۰۲٪
	Random Forest	۹۹٫۹۹٪

Bayes و MLP-ANN به نتایج نسبتا ضعیفی دست یافته است. یکی از علل این موضوع عدم بهینه‌سازی مدل‌هاست که باعث کاهش دقت نهایی شده است. در مقاله [۱۰]، الگوریتم Random Forest به کار رفته و دقت بالایی حاصل شده است. با این حال، به منظور افزایش سرعت آموزش، بخشی از داده‌های مجموعه داده حذف شده است. این کاهش حجم داده می‌تواند منجر به کاهش عملکرد مدل در پیش‌بینی داده‌های جدید شود. همچنین، زمان آموزش مدل در این مطالعه ۲۱۸ ثانیه و زمان پیش‌بینی ۶ ثانیه گزارش شده است که اگر چه سریع‌تر از مدل پیشنهادی ما است، اما با کاهش دقت همراه بوده است.

مقاله [۱۵] نیز به بررسی الگوریتم‌های یادگیری ماشین و یادگیری عمیق پرداخته، اما در اینجا فقط LightGBM و Random Forest مورد بررسی قرار گرفته‌اند. هرچند نتایج خوبی ارائه شده، اما این مدل‌ها بهینه‌سازی نشده‌اند و در مقایسه با مدل ما، دقت کمتری دارند. بهینه‌سازی هایپرپارامترها، همان‌طورکه در مدل ما انجام شده، می‌تواند از بیش‌برازش جلوگیری کرده و مدل را برای پیش‌بینی داده‌های ناشناخته قوی‌تر سازد، هر چند این فرآیند زمان‌بر و محاسباتی پیچیده است. مدل‌های پیشنهادی ما، با صرف زمان بیشتر برای آموزش و پیش‌بینی، دقت بالاتری را در مقایسه با مدل‌های پیشین ارائه می‌دهند. این تبادل بین دقت و زمان، یک استراتژی بهینه‌سازی است که به منظور افزایش قابلیت تعمیم مدل بر روی داده‌های جدید و جلوگیری از بیش‌برازش انتخاب شده است. در کاربردهای واقعی که دقت پیش‌بینی اهمیت بیشتری نسبت به زمان اجرا دارد، مدل‌های ما یک گزینه برتر هستند، حتی اگر مدت زمان بیشتری نیاز داشته باشند.

۶ نتیجه‌گیری

در این مقاله، یک سیستم تشخیص ناهنجاری مبتنی بر یادگیری ترکیبی با هدف پیشگیری و تشخیص حمله باتنت اینترنت اشیا معرفی شد. این سیستم با استفاده از الگوریتم‌های قدرتمند LightGBM و Random Forest و با بهره‌گیری از مجموعه داده Bot-IoT، که شامل نمونه‌هایی از حملات پیشرفته مانند DDoS و اسکن است، توانست عملکردی دقیق و موثر در تشخیص ناهنجاری‌ها ارائه دهد. یکی از عوامل کلیدی در بهبود دقت سیستم، فرآیند انتخاب ویژگی بود که به وسیله دو الگوریتم مذکور انجام شد. این فرآیند منجر به شناسایی و استفاده از ویژگی‌های مهم و مؤثر از بین داده‌های موجود شد که به کاهش ابعاد داده و افزایش کارایی مدل‌ها کمک کرده است. استفاده از روش SMOTE برای رفع مشکل عدم توازن داده‌ها و همچنین بهینه‌سازی هایپرپارامترها با استفاده از الگوریتم TPE، باعث افزایش دقت و کاهش نرخ خطا شد. نتایج نشان داد که LightGBM با نرخ خطای ۰٫۹۸٪ و دقت ۹۹٫۰۲٪ و Random Forest با نرخ خطای ۰٫۰۱٪ و دقت ۹۹٫۹۹٪ عملکرد بسیار مطلوبی در تشخیص حملات به دست آورده‌اند. این نتایج نشان‌دهنده کارایی بالای سیستم پیشنهادی در تشخیص حملات باتنت در مراحل مختلف چرخه حیات آن در اینترنت اشیا است.

مراجع

[1] Cao, Yongzhong, Wang, Zhihui, Ding, Hongwei, Zhang, Jiale, and Li, Bin. Iot botnet attacks detection and classification based on ensemble learning. in *International Symposium on Artificial Intelligence and Robotics*, pp. 45–55. Springer, 2023.

[2] Kalakoti, Rajesh, Nõmm, Sven, and Bahsi, Hayretdin. In-depth feature selection for the statistical machine learning-based botnet detection in iot networks. *IEEE Access*, 10:94518–94535, 2022.

[3] Kumari, Pooja and Jain, Ankit Kumar. A comprehensive study of ddos attacks over iot network and their countermeasures. *Computers & Security*, 127:103096, 2023.

[4] Hussain, Faisal, Abbas, Syed Ghazanfar, Pires, Ivan Miguel, Tanveer, Sabeeha, Fayyaz, Ubaid U, Garcia, Nuno M, Shah, Ghalib A, and Shahzad, Farrukh. A two-fold machine learning approach to prevent and detect iot botnet attacks. *Ieee Access*, 9:163412–163430, 2021.

[5] Leevy, Joffrey L, Hancock, John, Khoshgoftaar, Taghi M, and Peterson, Jared M. Iot information theft prediction using ensemble feature selection. *Journal of Big Data*, 9(1):6, 2022.

[6] Dong, Xibin, Yu, Zhiwen, Cao, Wenming, Shi, Yifan, and Ma, Qianli. A survey on ensemble learning. *Frontiers of Computer Science*, 14:241–258, 2020.

[7] Koroniotis, Nickolaos, Moustafa, Nour, Sitnikova, Elena,

and Turnbull, Benjamin. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796, 2019.

[8] Pokhrel, Satish, Abbas, Robert, and Aryal, Bhulok. Iot security: botnet detection in iot using machine learning. *arXiv preprint arXiv:2104.02231*, 2021.

[9] Abu Al-Haija, Qasem and Al-Dala'ien, Mu'awya. Elba-iot: An ensemble learning model for botnet attack detection in iot networks. *Journal of Sensor and Actuator Networks*, 11(1):18, 2022.

[10] Mohy-Eddine, Mouaad, Guezzaz, Azidine, Benkirane, Said, Azrour, Mourade, and Farhaoui, Yousef. An ensemble learning based intrusion detection model for industrial iot security. *Big Data Mining and Analytics*, 6(3):273–287, 2023.

[11] Ahmed, Aqeel and Tjortjis, Christos. Machine learning based iot-botnet attack detection using real-time heterogeneous data. in *2022 International Conference on Electrical, Computer and Energy Technologies (ICECET)*, pp. 1–6. IEEE, 2022.

[12] Sarwar, Ayesha, Mushtaq, Muhammad Faheem, Akram, Urooj, Rustam, Furqan, Hamza, Ameer, Rupapara, Vaibhav, and Ullah, Saleem. Iot networks attacks detection using multi-novel features and extra tree random-voting ensemble classifier (er-vec). *Journal of Ambient Intelligence and Humanized Computing*, 14(12):16637–16651, 2023.

[13] Cao, Yongzhong, Wang, Zhihui, Ding, Hongwei, Zhang, Jiale, and Li, Bin. An intrusion detection system based on stacked ensemble learning for iot network. *Computers and Electrical Engineering*, 110:108836, 2023.

[14] Majd, Nahid Ebrahimi and Gudipelly, Dhatri Sai Kumar Reddy. Iot botnet classification using cnn-based deep learning. in *2023 IEEE International Performance, Computing, and Communications Conference (IPCCC)*, pp. 46–51. IEEE, 2023.

[15] Alkhudaydi, Omar Azib, Krichen, Moez, and Alghamdi, Ans D. A deep learning methodology for predicting cybersecurity attacks on the internet of things. *Information*, 14(10):550, 2023.

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2024, Volume 22, Number 2 (pp. 45–55)
https://monadi.isc.org.ir

Presented at the ISCISC 2024 in Tarbiat Modares University & Research Institute for Information and Communication Technology, Tehran, Iran

Prevention and detection of botnet attacks in IoT using ensemble learning methods★

Fateme Pishdad * and Reza Ebrahimi Atani

Department of Computer Engineering, Faculty of Engineering, University of Guilan, Rasht, Iran

ARTICLE INFO.

Keywords:
Internet of things
Botnet
Anomaly detection
Ensemble learning
Feature selection

Type: Research paper

ABSTRACT

With the advancement and development of Internet of Things (IoT) applications, the need for securing infrastructure in this domain has gained particular importance due to the limitations of computational and storage resources. Botnets are among IoT security challenges in which, by infecting computational nodes of this technology, they are capble of turning the network into a collection of compromised machines under the control of attackers. This paper proposes an anomaly detection system based on ensemble learning to prevent and identify IoT botnet attacks at the initial scanning stage and DDoS attacks. This system uses feature selection and optimal hyperparameter tuning for each classifier to increase model accuracy and prevent overfitting. The data used in this paper is taken from the BoT-IoT dataset, which covers activities related to different stages of the botnet lifecycle. For feature selection and classification, two ensemble learning algorithms, LightGBM and Random Forest, are used, and hyperparameter optimization is performed using the TPE method. Results demonstrated that the LightGBM algorithm achieved an error rate of 0.98% and an accuracy of 99.02%, while the Random Forest algorithm exhibited an error rate of 0.01% and an accuracy of 99.99%, indicating highly satisfactory performance in attack detection. The proposed models, with increased training and prediction time, have offered significantly higher accuracy compared to previous models.

© 2024 ISC

★ The ISCISC 2024 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: pishdadfateme@webmail.guilan.ac.ir (Fateme Pishdad), rebrahimi@guilan.ac.ir (Reza Ebrahimi Atani)
© 2024 ISC. All rights reserved.