دوره شانزدهم، شماره بهار و تابستان ۱۴۰۲

مجله فناوری اطلاعات در طراحی مهندسی

Information Technology in Engineering Design

http://ited.sinaweb.net

تشخیص بدافزار با یک رویکرد حساس به هزینه مبتنی بر ترکیب طبقه‌بندها با روش پلکانی

اعظم سادات مقدم قدیری جلالی [۱] حسن شاکری*[۲] یاسر علمی [۳]

(۱) گروه مهندسی کامپیوتر، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

(۲) گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران*

(۳) گروه مهندسی کامپیوتر، واحد سبزوار، دانشگاه آزاد اسلامی، سبزوار، ایران

(تاریخ دریافت :۱۴۰۲/۰۵/۱۰ تاریخ پذیرش: ۱۴۰۲/۰۶/۲۹)

چکیده

رویکردهای مختلفی برای افزایش دقت تشخیص بدافزار پیشنهاد شده است که ازجمله می‌توان به ترکیب طبقه‌بندها اشاره کرد. همچنین تحقیقات گوناگونی با هدف کاهش هزینه‌های مختلف *IDS* انجام شده است. با وجود این نیاز به ارائه رویکردی جهت کاهش هزینه سیستم‌های مبتنی بر ترکیب طبقه‌بندها وجود دارد. این مقاله راهکاری برای تشخیص بدافزارهای اندروید پیشنهاد می‌دهد. این رویکرد شامل دو مرحله است. اولین گام، انتخاب مناسب‌ترین ویژگی‌ها با استفاده از الگوریتم کای‌مربع است. در گام دوم به‌عنوان نوآوری عمده این پژوهش، یک مدل پلکانی برای تشخیص بدافزار با استفاده از ترکیب دو طبقه‌بند مورد استفاده قرارمی‌گیرد که براساس سطح ریسک موجود و حساسیت مورد نیاز، یک مصالحه مطلوب بین نرخ هشدار اشتباه و نرخ منفی کاذب برقرار می‌کند. در مدل پیشنهادی طبقه‌بند دوم فقط بر روی رکوردهایی که با طبقه‌بند اول تعیین تکلیف نشده‌اند عمل می‌کند تا هزینه زمانی تشخیص در مقایسه با روش‌های پرهزینه‌ای مانند رأی‌گیری اکثریت کاهش یابد. نتایج ارزیابی راهکار پیشنهادی برروی یک مجموعه‌داده معتبر نشان داد که مدل ما صحت تشخیص را به بیش از ۹۵٪ و نرخ تشخیص اشتباه را به کمتر از ۰٫۰۳٪ می‌رساند که بهبود قابل ملاحظه‌ای نسبت به کارهای قبلی محسوب می‌شود. همچنین کارایی مدل در چهار سطح امنیتی بررسی شد.

کلمات کلیدی: تشخیص بدافزار، یادگیری ماشین، ترافیک شبکه، ترکیب طبقه‌بندها

*عهده‌دار مکاتبات:

حسن شاکری

نشانی: گروه مهندسی کامپیوتر، واحد مشهد، دانشگاه آزاد اسلامی، مشهد، ایران

پست الکترونیکی:**shakeri@mshdiau.ac.ir**

۱- مقدمه

یک گزارش تحقیقاتی توسط شرکت بین المللی داده (IDC)[۱] نشان می‌دهد که سیستم عامل اندروید گوگل با سهم ۸۷٫۶٪ پیشرو مطلق بازارسیستم‌عامل‌های تلفن‌های هوشمند بوده است، در حالیکه IOS اپل جایگاه دوم را با سهم ۱۱٫۷٪ به خود اختصاص داده است. این موفقیت بی‌نظیر پلت‌فرم اندروید با این ویژگی‌ها یک سری خطرات بالقوه ایجاد می‌کند که از جمله می‌توان به نفوذ بدافزارها در این محیط اشاره کرد.

این نفوذ نرم‌افزارهای مخرب بر روی پلت‌فرم اندروید به صورت گسترده‌ای در تحقیقات و صنعت مورد توجه قرار گرفته است. محققان زیادی پیشنهادهای متفاوتی برای مقابله با این مشکل ارائه کرده‌اند. استفاده از روشهای یادگیری ماشین یکی از روشهای معمول مورد استفاده در این زمینه است [۲, ۳] که البته برای دستیابی به دقت تشخیص بهتر می‌توان از ترکیب طبقه‌بندها استفاده کرد. یک مطالعه‌ی جدید [۴] طیف وسیعی از رفتارهای مخرب بدافزارها را بررسی می‌کند و روش های تشخیص بدافزار موبایل موجود را به دو گروه دسته بندی می‌کند: روشهای تجزیه و تحلیل ایستا وتجزیه و تحلیل پویا. در مطالعات دیگری نیز این تقسیم‌بندی انجام شده است[۵-۷].

تعداد زیادی از کارهای گذشته از تحلیل ایستا برای تشخیص رخنه در ناحیه خصوصی، بدافزارها و آسیب‌پذیری برنامه‌های کاربردی اندروید استفاده کرده‌اند[۸-۱۰]. با این حال تجزیه و تحلیل ایستا توسط کدهای چندریختی و کدهای مخرب که بدافزارهای مختلف برای فرار از تشخیص استفاده می‌کنند به چالش کشیده شده است. روش‌های پویا، سیستم عامل دستگاه را برای ردیابی و دسترسی به اطلاعات حساس زمان اجرا تغییر می‌دهند[۱۱]. تجزیه و تحلیل پویا امیدوار کننده است اما نیاز به مجموعه‌ی بزرگی از اجراها برای پوشش رفتارهای برنامه‌های کاربردی دارد. بنابراین انجام تجزیه و تحلیل پویا در منابع محدود دستگاه‌های هوشمند چالش برانگیز است.

برای مقابله بهتر با بدافزارهای موبایل محققان شروع به کاوش برای راه‌حل‌های جدید برای تشخیص بدافزار براساس ترافیک شبکه کردند. با استفاده از ترافیک شبکه کشف بدافزارهای پنهانی امیدوار کننده است زیرا بدافزارها معمولا رفتارهای مخربی از طریق ارتباط شبکه انجام می‌دهند. بنابراین در این پژوهش از ویژگیهای استخراج شده از فایلهای ترافیکی شبکه والگوریتمهای یادگیری ماشین و ترکیب پلکانی طبقه‌بندها برای تشخیص بدافزارهای اندروید استفاده شده است.

راهکار پیشنهادی ما شامل دو بخش است:

۱. یافتن ویژگیهای بهینه و کاهش ویژگیها برای تشخیص مناسب بدافزار با استفاده از مجموعه‌داده.

۲. ارائه یک رویکرد پلکانی برای تشخیص بدافزار با استفاده از الگوریتم J48 و الگوریتم جنگل تصادفی.

این راهکار از دو جهت، حساس به هزینه است: اول این که در مقایسه با سیستم‌های تشخیص نفوذی که مبتنی بر فقط یک طبقه‌بند هستند دقت تشخیص را افزایش و هزینه ناشی از تشخیص نادرست را کاهش می‌دهد. دوم این که در مقایسه با سیستم‌های تشخیص نفوذ مبتنی بر رأی‌گیری بین چند طبقه‌بند، هزینه عملیاتی کمتری دارد زیرا در مرحله اول فقط یک طبقه‌بند عمل می‌کند و در مرحله دوم برحسب نتیجه مرحله اول و سطح ریسک قابل پذیرش، ممکن است یک طبقه‌بند دیگر هم اجرا شود و یا نیازی به اجرای آن نباشد.

بقیه مقاله به صورت زیر سازماندهی شده است: بخش دوم ارائه‌ی کارهای مرتبط و در بخش سوم توضیح روش پیشنهادی برای تشخیص بدافزار، بخش چهارم بحث و نتایج ارزیابی‌ها و بخش پنجم شامل نتیجه‌گیری و کارهای آینده می‌باشد.

۲- کارهای مرتبط

در سالهای اخیر مکانیزم‌های تشخیص بدافزار با استفاده از تکنیک‌های داده‌کاوی برای تشخیص فایلهای مخرب به کمک یادگیری ماشین افزایش یافته است. روشهای یادگیری ماشین می‌توانند نمونه‌های پنهان از مجموعه داده‌هایی شامل هردو گروه بدافزار و بی‌خطر را تشخیص داده و بدافزارها را از کدهای بی‌خطر تمیز دهند[۴].

با توجه به محدودیت‌های تجزیه و تحلیل ایستا و پویا پژوهشگران شروع به استفاده از ترافیک شبکه برای شناسایی و تجزیه و تحلیل برنامه‌های مخرب کردند[۱۲, ۱۳]. این روش‌های مرتبط به سه گروه تقسیم می‌شوند:

- روشهای مبتنی بر امضای شبکه[۱]: در این روش تشخیص بدافزار براساس مقایسه با امضاهای از پیش تعریف شده‌ی بدافزار می‌باشد. تولید خودکار امضاهای شبکه در کارهای مختلف قبلی مورد بررسی قرار گرفته است[۱۴-۱۶] .
- روشهای مبتنی بر ویژگی‌های ایستای ترافیکی شبکه[۲]: در این روش از ویژگیهای ایستای ترافیک شبکه مانند میانگین اندازه‌ی بسته‌ها، نسبت بایتهای ورودی به خروجی، مدت زمان ارسال بسته‌ها و ویژگیهای ایستای دیگر برای تشخیص ترافیک مخرب از بی‌خطر استفاده می‌شود. فریم‌ورکی به نام App Scanner [۱۷] با استفاده از ویژگیهای ترافیکی ایستای رمزنگاری شده به تشخیص برنامه‌های کاربردی آسیب‌رسان می‌پردازد و در کاری که توسط کنتی و همکارانش [۱۸] انجام شد اعمال کاربر توسط تجزیه و تحلیل ویژگیهای آماری رمزنگاری شده تجزیه و تحلیل شد.
- روشهای مبتنی بر ویژگی‌های متنی ترافیکی شبکه[۳]: در این روشها از آنالیز متن جریان‌های ترافیکی برای تشخیص بدافزار استفاده می‌شود. کارهای انجام شده [۱۲, ۱۹, ۲۰] از این روش برای تشخیص بدافزار استفاده کرده‌اند.

از طرف دیگر یک رویکرد کارآمد در سیستم‌های تشخیص نفوذ، رویکرد طبقه‌بندی ترکیبی[۴] است. رویکرد ترکیبی یک مجموعه از طبقه‌بندهای ضعیف را انتخاب می‌کند و خروجی آنها را با یکدیگر ترکیب می‌نماید تا طبقه‌بند نهایی را به گونه‌ای بسازد که کارایی آن از کارایی تک‌تک طبقه‌بندهای استفاده شده در الگوریتم بیشتر باشد. نهایتاً دسته رکوردهای دیده‌نشده در مرحله‌ی ارزیابی با ترکیب‌کردن خروجی تک‌تک طبقه‌بندهای جزئ استفاده‌شده تعیین می‌شود.

برای تعیین دسته یک رکورد جدید آن رکورد به تمام طبقه‌بندهای کوچک نشان داده می‌شود. هر کدام از این طبقه‌بندها یک خروجی برای آن رکورد تولید می‌کنند در نهایت خروجی‌ها با یکدیگر ترکیب می‌شوند تا طبقه نهایی رکورد ورودی مشخص شود. از جمله این روش‌ها می‌توان روش Bagging، Boosting و رای‌گیری مبتنی بر اکثریت[۵] را نام برد[۲۱].

کارهای زیادی در زمینه‌ی تشخیص بدافزار اندروید با استفاده از ترافیک شبکه ارائه شده است که در ادامه به گروهی از کارهای گذشته اشاره می‌شود. در کاری که توسط چاکر ابورتی و تنموئی پیراززی [۲۲] انجام شد رویکردی پیشنهاد شده که اولین الگوریتمی است که به صورت موثر نمونه‌های بدافزار را به دو گروه خانواده‌های بزرگ و خانواده‌های کوچک (حتی اگر قبلاً مشاهده نشده‌اند) تقسیم می‌کند این روش با استفاده از ویژگیهای ایستا و پویا و ترکیب الگوریتم های طبقه‌بندی و خوشه‌بندی انجام شد با این رویکرد خانواده‌های جدید بدافزار نیز طبقه‌بندی شدند. پژوهشی توسط چن وهمکارانش [۲۳] در سال ۲۰۱۷ انجام شد که از ویژگیهای

---

[۱] Network Signature

[۲] Packet/Flow statistical Features

[۳] Packet/Flow Textual Features

[۴] Ensemble Classification Method

[۵] Majority Vote

آماری ترافیک شبکه استفاده کردند با این حال آنها همچنین نقص روش‌های یادگیری ماشین را در مواجهه با داده‌های نامتقارن نشان دادند. علاوه بر این چندین روش طبقه‌بند داده‌های نامتقارن از قبیل (SMOTE)+(SVM)، (SVMCS) و C4.5، C4.5CS را روی همان مجموعه‌داده ترافیکی اجرا کردند نتایج تجربی نشان داد که با وجود اینکه روشهای یادگیری ماشین نامتعادل می‌تواند دقت تشخیص را بهبود دهند کارایی این الگوریتم‌ها زمانی که درجه‌ی عدم‌تعادل از آستانه‌ی خاصی می‌گذرد کاهش می‌یابد.

بنابراین الگوریتم اصلاح شده‌ای برای حل مشکل طبقه‌بندی داده‌های نامتقارن ارائه شد. در یکی دیگر از پژوهش‌های اخیر[۲۴] یک رویکرد ترکیبی طبقه‌بندی جدید براساس معماری چندسطحی که ترکیب موثری از الگوریتم‌های یادگیری ماشین، برای بهبود دقت فراهم می‌کند ارائه شده است. چارچوبی به نام Droid Fusion یک مدل را با آموزش طبقه‌بندهای پایه در سطح پایین‌تر تولید می‌کند و سپس مجموعه‌ای از الگوریتم‌های مبتنی بر رتبه‌بندی را بر روی دقت‌های پیش‌بینی آنها در سطح بالاتر برای نتیجه‌گیری نهایی اعمال کردند. در مرجع [۲۵] رویکردی شامل دو مرحله ارائه شد که در مرحله‌ی اول از الگوریتم‌های ترکیبی IG[۱] و PC[۲] برای انتخاب بهترین ویژگی‌ها استفاده شد و سپس رویکردی ترکیبی از سیستم استنتاج فازی عصبی و الگوریتم ازدحام ذرات برای تشخیص بدافزارها از برنامه‌های بی‌خطر ارائه شد. پژوهشی توسط ونگ و همکارانش [۱۲] انجام شد که در آن برای تشخیص بدافزار موبایل از ویژگیهای استخراج شده از اطلاعات متنی هدر http و https برای تشخیص استفاده شد.

در مرجع [۲۶] یک روش شناسایی بدافزارهای اندرویدی براساس تشخیص انحراف از ترافیک نرمال پیشنهاد شده است که براساس ۹ ویژگی ترافیکی، تشخیص نرم‌افزارهای مخرب و نقاب‌زنی[۳] را انجام می‌دهد. روش مذکور همچنین قادر است نرم‌افزارهای مخرب خاص مانند بدافزارهای تبلیغاتی را شناسایی کند.

در مرجع [۲۷] رویکردی برای تشخیص بدافزارهای اندرویدی با استفاده از شبکه پیچشی گرافی[۴] (GCN) ارائه شده است. ایده اصلی در این رویکرد نگاشت برنامه‌ها و APIهای اندرویدی به یک گراف ناهمگن بزرگ و تبدیل مسأله اولیه به یک مسأله طبقه‌بندی گره‌ها است. سپس گراف ناهمگن به عنوان ورودی به مدل GCN داده می‌شود. در تحقیق مذکور همچنین یک نمونه اولیه از سیستم تشخیص بدافزار به نام Groid معرفی شده است که مؤلفان نرخ مثبت کاذب پایین را یکی از مهم‌ترین امتیازات آن عنوان کرده‌اند.

از طرف دیگر برخی از کارهای تحقیقاتی ازجمله [۲۸-۳۲] مسأله هزینه در سیستم‌های تشخیص نفوذ را مورد توجه قرار داده و مدل‌های حساس به هزینه برای این منظور ارائه کرده‌اند. یکی از مهم‌ترین جنبه‌های هزینه در سیستم‌های طبقه‌بندی، هزینه مربوط به تشخیص اشتباه شامل مثبت کاذب و منفی غلط است. برای کاهش این نوع هزینه، الگوریتم‌های boosting حساس به هزینه جزو کارآمدترین تکنیک‌ها محسوب می‌شوند [۲۸]. اما در چندین فاکتور دیگر هزینه نیز به ویژه در مورد سیستم‌های تشخیص نفوذ اهمیت دارند که از جمله می‌توان به هزینه توسعه سیستم، هزینه عملکرد، و هزینه پاسخ دستی یا خودکار به حملات اشاره کرد.

با وجود این نیاز به ارائه رویکردی در جهت کاهش هزینه سیستم‌های تشخیص نفوذ مبتنی بر ترکیب طبقه‌بندها وجود دارد به طوری که ضمن کاهش خطای تشخیص و هزینه ناشی از آن، تا حد امکان مانع اجرای طبقه‌بندهای ثانویه شود تا به این ترتیب، هزینه عملیاتی سیستم تشخیص نفوذ ترکیبی نیز کاهش یابد.

---

[۱] Information Gain

[۲] Pearson Corr Coaf

[۳] Masquerading

[۴] Graph Convolutional Network

۳- روش پیشنهادی

با توجه به روش‌های موجود برای تشخیص بدافزارها در این بخش روش پیشنهادی ما که یک روش ترکیبی پلکانی از الگوریتم‌های یادگیری ماشین است ارائه می‌شود. یک روش ترکیبی سعی می‌کند که با استفاده گروهی از طبقه‌بندها نتایج بهتری را به نسبت حالت استفاده از هر کدام از طبقه‌بندها به تنهایی ایجاد کند.

فلوچارت مدل پیشنهادی در شکل ۱ ارائه شده است.

الف. انتخاب ویژگی‌ها

به دلایل زیر ما نیاز به کاهش ویژگی‌ها داریم:

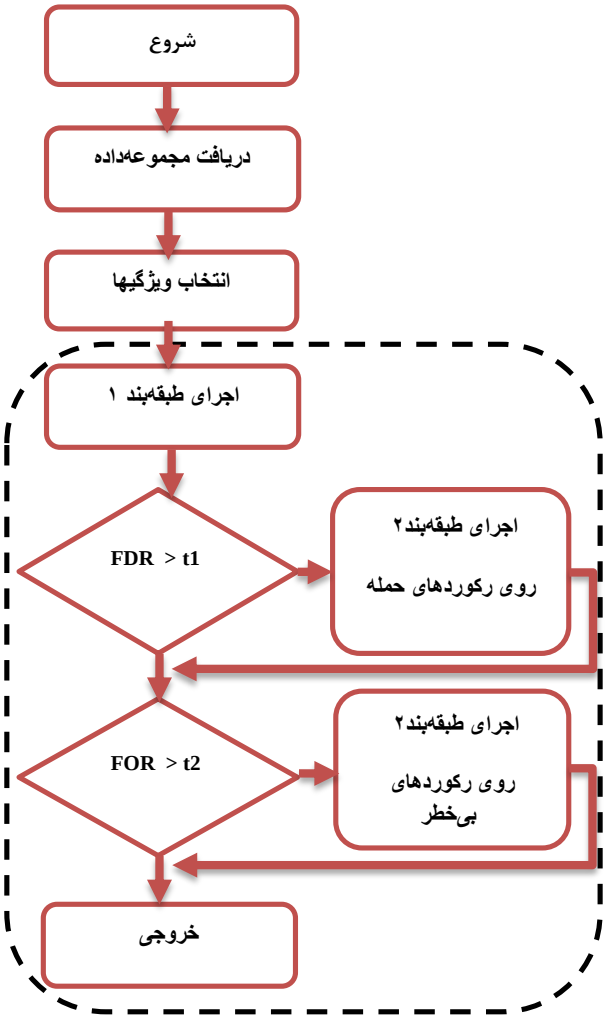
۱)کاهش پیچیدگی. الگوریتم های داده کاوی زمانی که ویژگی های آنها زیاد می شوند نیاز به زمان و منابع زیادی دارند بنابراین کاهش تعداد ویژگی برای صرفه جویی در زمان و منابع اهمیت دارد.

۲)کاهش نویز. ویژگی‌های اضافی همیشه منجر به بهبود عملکرد الگوریتم نمی‌شود برعکس آنها ممکن است مسئله بیش‌برازش[۱] را در برداشته باشند در نتیجه انتخاب یک مجموعه ویژگی مناسب می‌تواند شانس بیش‌برازش را کاهش دهد.

همه‌ی ویژگیها در مجموعه ویژگی‌های ما برای مدل تشخیصی بدافزار مفید نیستند. این ویژگیهای غیرضروری می‌تواند کارکرد عادی الگوریتم را منفی کند. بنابراین از الگوریتم کای‌مربع استفاده می‌کنیم.

آزمون کای‌مربع آزمون آماری است که به صورت گسترده ای برای تعیین اینکه آیا توزیع‌های مورد نظر از متغیرهای قطعی به صورت قابل ملاحظه‌ای از مشاهدات متفاوت است مورد استفاده قرار می‌گیرد. این آزمون یک روش اندازه‌گیری برای اختصاص ویژگی وزن خاص است که مقدار آزمون کای‌مربع نامیده می‌شود و برای توصیف همبستگی بین گروه‌ها مورد استفاده قرار می‌گیرد.

[۱] Overfitting



شکل ۱- فلوچارت روش پیشنهادی

برای انتخاب ویژگیها با استفاده از آزمون کای‌مربع می توان به دو صورت کار کرد یک روش به این صورت است که یک حد آستانه‌ی ثابت را تنظیم کنیم و یا K مقدار که مقدار آزمون کای‌مربع بالایی را دارند برای انتخاب ویژگی گزینش کنیم. نتایج آزمون اهمیت هر ویژگی برای تعیین کلاس‌بندی را نشان می‌دهد. در این پژوهش با استفاده از این روش آماری وزن هریک از ویژگیها تعیین می‌شود و سپس K ویژگی که وزن بالاتری داشته باشند به عنوان ویژگیهای انتخابی گزینش می‌شوند.

ب. طبقه‌بندی پلکانی آگاه از حساسیت

در تمام پژوهش‌هایی که دارای مرحله‌ی یادگیری ماشین هستند دستیابی به قدرت تشخیص بالاتر مدنظر است ما نیز جهت افزایش قدرت تشخیص مدل خود از رویکرد ترکیب طبقه‌بندها به صورت پلکانی استفاده کردیم. این مراحل پژوهش در شکل ۱ به صورت خط‌چین مشخص شده است. بعد از جداسازی داده‌ها به دو بخش آموزش و آزمایشی ابتدا دو طبقه‌بند با استفاده از داده‌های آموزشی مدل خود را ایجاد می‌کنند با ساخت مدل‌های یادگیری مرحله‌ی اول اجرای مدل برروی داده‌های آزمایشی آغاز می‌شود در گام اول

طبقه‌بند ۱ داده‌های مجموعه‌داده را به دو گروه حمله و نرمال دسته‌بندی می‌کند سپس مقادیر TP[۱]، FP[۲]، FN[۳] و TN[۴] در مدل اولیه محاسبه شده و دو مقدار نرخ مثبت کاذب (FDR[۵]) و نرخ منفی کاذب(FOR[۶]) برطبق رابطه‌ی ۱و ۲ که در زیر آمده محاسبه می‌شود.

$$FDR = \frac{FP}{TP + FP} \quad (۱)$$

$$FOR = \frac{FN}{TN + FN} \quad (۲)$$

در این رابطه‌ها TP، تعداد رکوردهای حمله وTN تعداد رفتارهای نرمال هستند که به درستی تشخیص داده شده است. همچنین FP تعداد رفتار نرمال است که به اشتباه، حمله تشخیص داده شده و FN تعداد رفتارحمله است که به اشتباه، نرمال اعلام شده است. در گام دوم دو حدآستانه‌ی t_1 و t_2 با توجه به شرایط در نظر گرفته می‌شود این دو مقدار می‌تواند مقدار سخت‌گیری و یا سهل‌گیری مدل را با توجه به شرایط تغییر دهد. با کم شدن مقادیر t_1 و t_2 حساسیت مدل افزایش یافته و با زیاد شدن آنها حساسیت مدل کم می‌شود و ضمناً در این مدل می‌توان با تغییر حدآستانه‌ی خاصی حساسیت مدل را برروی داده‌های مورد تمرکز آن آستانه که می‌تواند رکوردهای حمله و یا نرمال باشد تغییر داد. به عنوان مثال در صورتی که مقدار t_1 کمتر شود مدل با کنترل بیشتر رکوردهای حمله را مجدد با طبقه بند ۲ گروه‌بندی می‌کند و سعی در کاهش میزان نرخ مثبت کاذب دارد، این حالت می‌تواند در مورد بدافزارهای تبلیغاتی کاربرد داشته باشد و یا کاهش مقدار t_2 می‌تواند مدل را برای بدافزارهای پرخطر آماده‌تر کند. درمدل پیشنهادی علاوه بر افزایش دقت تشخیص می‌توان کنترل بیشتری بر روی مدل داشت و آن را برای شرایط مختلف تنظیم کرد و در صورت نیاز تعداد سطوح آن را افزایش داد، علاوه براین از عملیات پردازشی سنگین بر روی کل رکوردها و افزایش هزینه جلوگیری کرد.

۴- ارزیابی ونتایج

الف. آزمایش‌ها

مجموعه‌داده استفاده شده در این پژوهش که جزئیات آن در زیربخش بعدی ارائه شده است، در سه کلاس آگهی‌های تبلیغاتی، بدافزارهای عمومی و بی‌خطر کلاس‌بندی شده است. قبل از اجرای مدل، ما مجموعه‌داده موجود را به یک مجموعه‌داده دو کلاسی تبدیل می‌کنیم. در این مسئله تشخیص دو کلاس بدافزار و بی‌خطر مورد توجه می‌باشد. ابتدا با رویکرد توضیح داده شده در بخش قبل به انتخاب مناسب‌ترین ویژگیها برای مدل پرداختیم. از آنجا که F-Measure به صورت جامعی میزان تشخیص و میزان خطای مدل را در شناسایی نمونه‌های مخرب نشان می‌دهد بنابراین برای انتخاب تعداد ویژگهای مدل از این معیار استفاده شده تا تعداد ویژگی مطلوب و بهینه مدل شناسایی شوند. نتایج این بررسی در قسمت نتایج ارائه شده است.

سپس برای ارزیابی معیارهای مشخص شده‌ی مدل، چهار سطح امنیتی به صورت زیر تعریف شد:

- سطح امنیتی پایین: در این سطح امنیتی حساسیت مدل را با گذاشتن آستانه‌های t_1 و t_2 بزرگ کاهش داده، به طوریکه فقط مرحله‌ی اول اجرای مدل، اجرا شود. این سطح امنیتی می‌تواند زمانی که هدف کاهش حجم پردازش و دسترسی به یک سطح امنیتی پایین مطلوب می‌باشد کاربرد داشته باشد.

---

[۱] True positive
[۲] False Positive
[۳] False Negative
[۴] True Negative
[۵] False Discovery Rate
[۶] False Omission Rate

- سطح امنیتی میانی حساس به تشخیص بی‌خطر: در این سطح مقدار t_1 و t_2 طوری تنظیم شده است که t_1 یک مقدار کوچک و t_2 مقدار بزرگتری خواهد داشت. در این شرایط مدل، کار تشخیص را فقط بر روی رکوردهایی که در مرحله‌ی اول بی‌خطر شناسایی شده است با طبقه‌بند دوم انجام می‌دهد.
- سطح امنیتی میانی حساس به بدافزار: در سطح امنیتی حساس به بدافزار مقدار t_1 و t_2 طوری تنظیم شده است که t_1 یک مقدار بزرگ و t_2 مقدار کوچکتری خواهد داشت دراین شرایط مدل کار تشخیص را فقط بر روی رکوردهایی که در مرحله‌ی اول حمله شناسایی شده است با طبقه‌بند دوم انجام می‌دهد.
- سطح امنیتی بالا: در این سطح امنیتی مقادیر t_1 و t_2 بسیار کم داده شد تا مدل در مرحله‌ی دوم به صورت کامل اجرا شود یعنی هم رکوردهای بی‌خطر و هم رکوردهای حمله به ترتیب با استفاده از طبقه‌بند دوم به صورت مجدد دسته‌بندی شوند.

نتایج اجرای این سطوح مختلف در قسمت نتایج ارائه و تحلیل شده است. تمام آزمایش‌ها با نرم‌افزار داده‌کاوی RapidMiner انجام شده است.

ب. مجموعه داده

برای ایجاد یک سیستم تشخیص بدافزار به مجموعه‌داده‌ای نیاز داریم تا مدل‌ها بتوانند از آن یاد بگیرند. برای انجام این تحقیق از مجموعه‌داده موسسه سایبری کانادا (CIC)که زیر نظر دانشگاه نیوبرانزویک است استفاده شده است. از آنجا که نرم‌افزارهای مخرب پیشرفته قادر به شناسایی حضور شبیه‌ساز هستند و رفتار خود را تغییر می‌دهند تا از شناسایی آنها جلوگیری شوند. برای غلبه بر این مسئله، جهت جمع‌آوری این مجموعه‌داده برنامه‌های اندروید بر روی دستگاه واقعی نصب شده‌اند و ترافیک شبکه بدست آمده است.

مجموعه داده CICAAGM [۲۶] با نصب برنامه‌های اندروید در تلفن‌های هوشمند نیمه‌اتوماتیک گرفته شده است. این مجموعه داده که از طریق [۳۳] قابل دسترسی است، از ۱۹۰۰ برنامه‌ی کاربردی که در سه گروه قراردارند تشکیل شده است.۲۵۰ برنامه‌ی ابزار تبلیغاتی و۱۵۰ بدافزار عمومی و۱۵۰۰ برنامه‌ی بی‌خطر است.

ویژگی‌های این مجموعه‌داده حدود ۸۰ ویژگی است که در چند گروه دسته‌بندی می‌شوند که خلاصه‌ای از آنها در جدول ۱ موجود است.

جدول ۱- برخی از ویژگی‌های رکوردهای مجموعه‌داده

ویژگیهای رفتاری	
Duration	مدت‌زمان اجرا
ویژگیهای مبتنی بر تعدادبایتها	
Total Forward Bytes	مجموع بایت‌ها در جهت رو به جلو
Total Backward Bytes	مجموع بایت‌ها در جهت رو به عقب
Forward Header Length	مجموع بایت‌ها استفاده شده برای هدرها در جهت رو به جلو
Backward Header Length	مجموع بایت‌ها استفاده شده برای هدرها در جهت رو به عقب
ویژگیهای مبتنی بر مشخصات بسته‌ها	
Total Forward Packets	مجموع بسته‌ها در جهت رو به جلو
Total Backward Packets	مجموع بسته‌ها در جهت رو به عقب

Forward packet length (Min, Mean, Max, Std)	مینیمم و ماکزییم ومیانگین و انحراف معیاراندازه‌ی بسته‌ها در جهت رو به جلو
Backward packet length (Min, Mean, Max, Std)	مینیمم و ماکزییم ومیانگین و انحراف معیاراندازه‌ی بسته‌ها در جهت رو به عقب
ویژگیهای مبتنی بر جریان	
Flow Packet Length (Min, Mean, Max, Std)	مینییمم،میانگین،ماکزیمم و انحراف معیار طول یک جریان
Flow Forward Bytes	میانگین بایتهای یک زیر جریان در جهت رو به جلو
Flow Backward Bytes	میانگین بایتهای یک زیر جریان در جهت رو به عقب
Backward Variance Data Byte	انحراف معیار مجموع بایت‌های استفاده‌شده در یک جریان رو به عقب
Forward Variance Data Byte	انحراف معیار مجموع بایت‌های استفاده‌شده در یک جریان رو به جلو
Flow FIN	تعداد بسته‌ها با FIN
Idle (Max, Min)	ماکزیمم و مینیمم زمانی که یک جریان قبل از فعال شدن بیکار است.
Initial Window Forward	مجموع بایتهای ارسال شده به سمت جلو درآغاز به کار ویندوز
Initial Window Backward	مجموع بایت های ارسال شده به سمت عقب درآغاز به کار ویندوز
Segment Size Forward (Max, Min)	مینیمم و ماکزییم اندازه‌ی سگمنت‌های مشاهده شده در جهت رو به جلو
Segment Size Backward (Max, Min)	مینیمم و ماکزییم اندازه‌ی سگمنت‌های مشاهده شده در جهت رو به عقب
ویژگیهای مبتنی بر زمان	
Forward Arrival (Min, Mean, Max, Std)	مینییمم، میانگین، ماکزیمم و انحراف معیار زمان بین ارسال دو بسته روبه جلو
Backward Arrival Time (Min, Mean, Max, Std)	مینییمم، میانگین، ماکزیمم و انحراف معیار زمان بین ارسال دو بسته رو به عقب
Idle Time (Min, Mean, Max, Std)	مینییمم، میانگین، ماکزیمم و انحراف معیار زمانی یک جریان قبل از فعال شدن بیکاراست
Active Time (Min, Mean, Max, Std)	مینییمم، میانگین، ماکزیمم و انحراف معیار زمانی یک جریان قبل از بیکاری فعال است

ج. تنظیمات آزمایش‌ها

با توجه به محدودیت توان پردازشی از یک طرف و لزوم رسیدن به دقت کافی در ارزیابی شاخص‌های عملکرد از طرف دیگر، الگوریتم پیشنهادی در ۲۰ دور اجرا شد و میانگین مقادیر شاخص‌ها در این ۲۰ دور تعیین و جهت مقایسه با نتایج کارهای تحقیقاتی موجود مورد استفاده قرار گرفت.

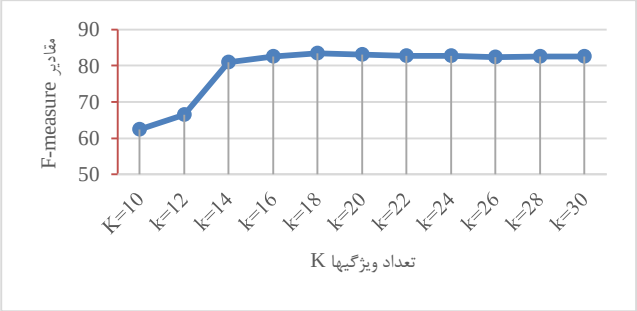
همان‌گونه که در زیربخش قبلی اشاره شد، مجموعه‌داده مورد استفاده برای ارزیابی راهکار پیشنهادی شامل رکوردهایی متعلق به سه کلاس آگهی‌های تبلیغاتی، بدافزارهای عمومی و نرم‌افزارهای بی‌خطر بود ولی با توجه به هدف این پژوهش، ما قبل از انجام آزمایش‌ها رکوردهای مربوط به کلاس آگهی‌های تبلیغاتی را حذف کردیم و یک مجموعه‌داده دوکلاسی شامل نرم‌افزارهای مخرب (بدافزار) و بی‌خطر حاصل شد. از طرف دیگر مجموعه‌داده از نظر تعداد رکوردهای این دو کلاس، نامتوازن است به طوری که تعداد بدافزارها ۱۵۰ و تعداد نرم‌افزارهای بی‌خطر ۱۵۰۰ مورد است. در نتیجه برای برقراری توازن و افزایش دقت تشخیص، در هر دور از آزمایش‌ها نمونه‌برداری با انتخاب ۱۰۰درصد رکوردهای کلاس بدافزار و ۱۰درصد رکوردهای کلاس بی‌خطر انجام شد.

د. ارائه و تحلیل نتایج آزمایش‌ها

در این بخش ارزیابی‌های انجام شده در بخش‌های مختلف تشریح شده است.

انتخاب ویژگیها: اولین آزمایش با هدف محاسبه‌ی تعداد ویژگیهای بهینه از مجموعه‌داده می‌باشد. در مجموعه‌داده این پژوهش نزدیک به ۸۰ ویژگی از ویژگیهای ترافیکی نرم‌افزارهای مخرب و بی‌خطر ارائه شده بود. ما در این آزمایش تعداد K ویژگی را که

وزن بالاتری داشته و در تشخیص مدل مفیدتر هستند را گزینش کردیم. نمودار شکل ۲ میزان تغییر معیار F-Measure مدل را با تعداد Kهای متفاوت نشان می‌دهد.



شکل ۲- نمودار تعیین میزان تعداد ویژگی براساس مقادیر F-Measure

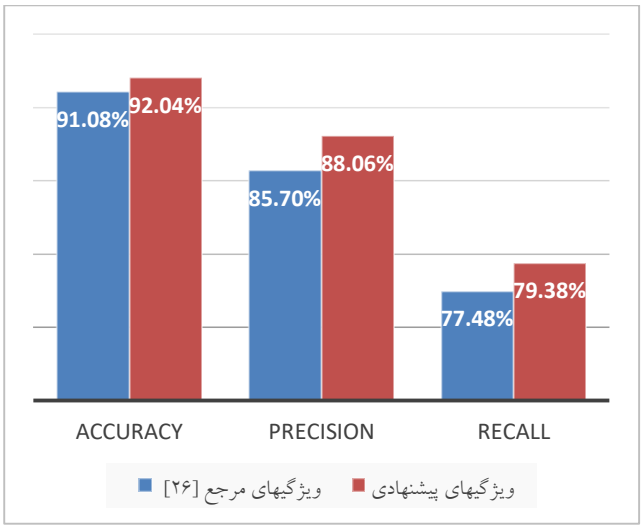
در شکل ۲ محور افقی تعداد ویژگیها را که با گام‌های دوتایی تغییر کرده‌اند، نمایش می‌دهد که از مقدار ۱۰ تا ۳۰ تغییر می‌کند و محور عمودی مقدار معیار F-Measure را با تغییرات K نشان می‌دهد. با توجه به ارزیابی انجام شده بالاترین مقدار F-Measure را برای K=۱۸ داریم البته تغییرات نمودار برای مقادیر بین ۱۶ تا ۳۰ ناچیز می‌باشد اما برای بهینه بودن و کاهش حجم پردازشی در پژوهش ما از K=۱۸ استفاده شده است.

در آزمایش دوم مقایسه‌ای بین ویژگیهای انتخابی مدل ما و ویژگیهای انتخابی مدل ارائه شده در مرجع [۲۶] ارائه شده است که نتایج این مقایسه در شکل‌های ۳ و ۴ ارائه‌شده است. نتایج این مقایسه‌ها نشان می‌دهد که ویژگیهای انتخابی ما عملکرد بهتری به نسبت مدل مرجع [۲۶] دارد. این مقایسه با ۱۰٪ داده‌های مجموعه‌داده و با الگوریتم جنگل تصادفی[۱] و الگوریتم J48 به دست آمده است.

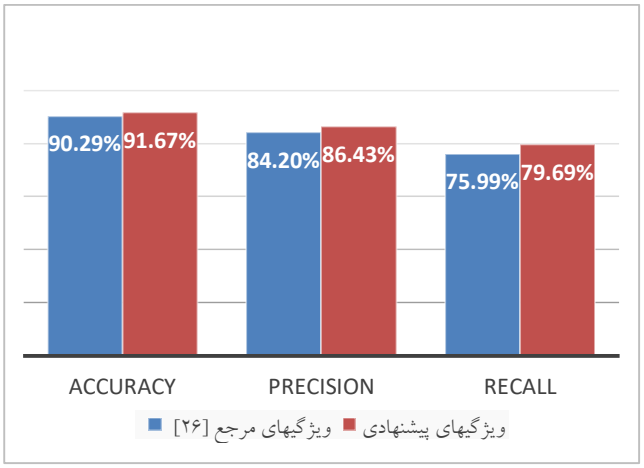
ارزیابی مدل در چهار سطح امنیتی متفاوت: با استفاده از مدل پیشنهادی می‌توان چهار سطح امنیتی متفاوت را برای تشخیص بدافزار در مدل طراحی کرد که کاربر با توجه به شرایط پردازشی و همچنین شرایط امنیتی خود این سطوح را با تغییر مقادیر t_1 و t_2 در مدل تنظیم کند در شکل ۵ معیار FPR مدل در چهار سطح امنیتی ارائه شده است. در سطح امنیتی حساس به تشخیص بدافزار و سطح امنیتی بالا کمترین مقدار نرخ تشخیص اشتباه بدافزار را داریم و نتایج نشان می‌دهد که اجرای این سطوح امنیتی می‌تواند نرخ خطای مدل را به خوبی کاهش داده و مقدار FPR را به کمتر از ۰,۰۳٪ برساند.

همچنین عملکرد راهکار پیشنهادی با مدل Groid [۲۷] مورد مقایسه قرار گرفته است. نتایج این مقایسه در جدول ۲ ارائه شده است که نشان می‌دهد عملکرد روش پیشنهادی از نظر معیارهای مختلف صحت، دقت، فراخوانی و F-Measure بهتر از مدل Groid است.

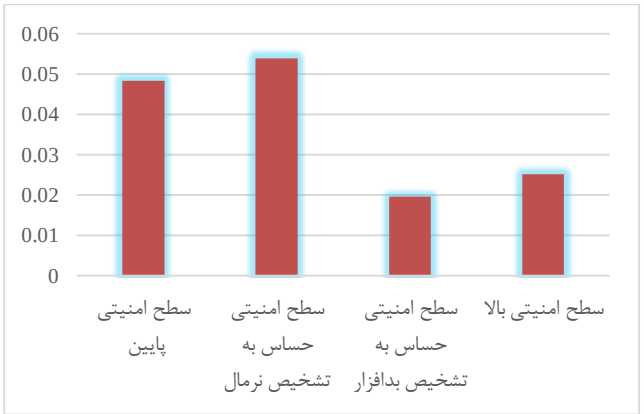
[۱] Random Forest



شکل ۳- نمودار مقایسه معیارهای ارزیابی با الگوریتم جنگل تصادفی



شکل ۴- نمودار معیارهای ارزیابی با الگوریتم J48



شکل ۵- نمودار مقایسه ی FPR در سطوح امنیتی مدل

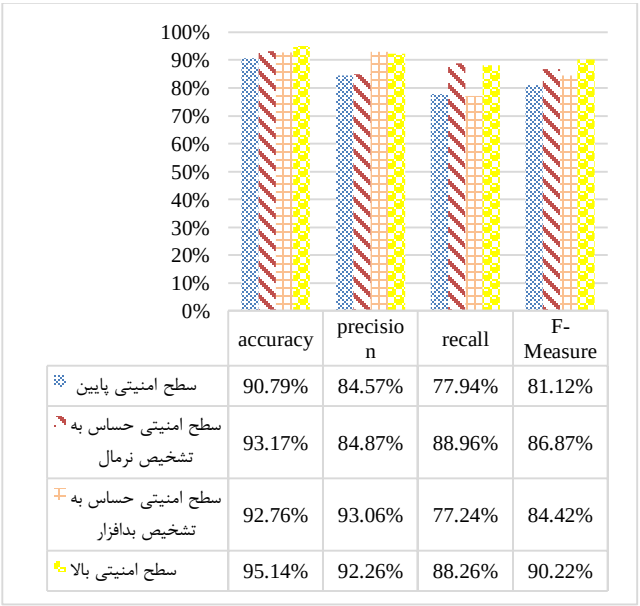
جدول ۲- مقایسه عملکرد راهکار پیشنهادی با مدل Groid [۲۷]

	صحت	دقت	فراخوانی	F-Measure
Groid [۲۷]	۹۶,۹۸	۰,۹۶۹	۰,۹۵۹	۰,۹۶۰
روش پیشنهادی	۹۷,۴۲	۰,۹۷۳	۰,۹۷۰	۰,۹۷۱

در شکل ۶ نمودار مقایسه‌ای معیارهای دیگر مدل در چهار سطح امنیتی نشان داده شده است. نتایج نشان می‌دهد که اجرای مدل در هر یک از سطوح متوسط یا بالا میزان Accuracy و F-Measure به میزان قابل ملاحظه‌ای افزایش یافته و عملکرد مدل بهبود یافته است. در سطح امنیتی حساس به نرمال و سطح امنیتی بالا میزان Recall افزایش یافته به این معنی که نسبت مقادیری از ترافیک که حمله تشخیص داده شده به نسبت کل ترافیکی که واقعا حمله هستند بالاتر از بقیه سطوح می‌باشد. در سطح امنیتی حساس به تشخیص بدافزار و سطح امنیتی بالا میزان معیار Precision به نسبت سطوح دیگر بیشترین افزایش را داشته است. به این ترتیب درصدی از مقادیری که به درستی به عنوان ترافیک حمله تشخیص داده شده به نسبت کل ترافیک‌هایی که حمله تشخیص داده شده است بیشترین مقدارها را به نسبت بقیه سطوح نشان می‌دهد.

نتایج بدست آمده در این قسمت نیز با استفاده از الگوریتم J48 به عنوان طبقه‌بند اول و الگوریتم جنگل تصادفی به عنوان طبقه‌بند دوم و با ۱۰٪ داده‌های مجموعه‌داده بدست آمده است.

به طور کلی مدل ایجاد شده با افزایش مقدار صحت به بیش از ۹۵٪ و کاهش نرخ تشخیص اشتباه به کمتر از ۰,۰۳٪ دقت کافی برای تشخیص بدافزار را خواهد داشت. با توجه محدودیت منابع پردازشی مقادیر داده شده فقط با استفاده از ۱۰٪ داده‌های مجموعه‌داده انجام شده و انتظار داریم در صورت افزایش توان پردازشی می توان صحت مدل را تا ۹۸٪ افزایش داد.



	accuracy	precision	recall	F-Measure
سطح امنیتی پایین	90.79%	84.57%	77.94%	81.12%
سطح امنیتی حساس به تشخیص نرمال	93.17%	84.87%	88.96%	86.87%
سطح امنیتی حساس به تشخیص بدافزار	92.76%	93.06%	77.24%	84.42%
سطح امنیتی بالا	95.14%	92.26%	88.26%	90.22%

شکل ۶- نمودار مقایسه‌ی معیارهای ارزیابی در سطوح مختلف امنیتی تعریف شده

۵- نتیجه‌گیری وکارهای آینده

بدافزارهای اندروید تهدیدی جدی با رشدی فزاینده هستند. به عنوان یک راه‌حل، ما برای تشخیص بدافزارهای اندروید از ویژگیهای ترافیکی شبکه استفاده کردیم. در مرحله‌ی اول از آنجا که کیفیت مدل تشخیصی به کیفیت ویژگیهای انتخابی وابستگی زیادی دارد سعی شد که با انتخاب روشی کم‌هزینه و مناسب، ویژگیهای مجموعه‌داده گزینش شوند. برای این منظور از آزمون کای‌مربع که یک آزمون آماری است استفاده شد و در قسمت ارزیابی نیز آزمایش‌هایی اجرا شد تا مناسبترین ویژگیها برای ساخت مدل از مجموعه داده انتخاب شوند. نکته‌ای که در این نتیجه‌ی آزمایش‌ها وجود داشت این بود که ویژگیهای انتخابی مدل ما که در جدول ۱ دسته‌بندی شده‌اند همه متعلق به دسته‌بندی مبتنی بر جریان هستند. این نشان می‌دهد که استفاده از این ویژگیهای ترافیکی می‌تواند مدل را برای تشخیص بدافزار آماده‌تر کند شاید علت این موضوع این باشد که ممکن است هکرها بتوانند ویژگیهای ترافیکی مانند مدت زمان جریان را تقلید کنند اما الگوگیری از ضرایب مبتنی بر جریان کار پیچیده‌ای می‌باشد.

پس از انتخاب ویژگی‌های مناسب، مدل در چهار سطح امنیتی طراحی شد این چهار سطح امنیتی با استفاده از دو طبقه‌بند که به صورت پلکانی اجرا می‌شوند ساخته شد در پایین‌ترین سطح امنیتی فقط طبقه‌بند اول اجرا و ارزیابی شد و سپس در دو سطح میانی با تغییر دو آستانه‌ی t_1 و t_2 و تنظیم آن به صورتی اجرا شد که در سطح امنیتی حساس به بی‌خطر رکوردهایی که بی‌خطر اعلام شده بودند توسط طبقه‌بند دوم کلاس‌بندی شدند و در سطح امنیتی حساس به بدافزار نیز با تنظیم آستانه‌های تعریف شده طبقه‌بند دوم فقط بر روی رکوردهایی که در مرحله‌ی اول، حمله تشخیص داده شده‌اند اعمال شد. ارزیابی‌ها نشان داد که مدل در تمام سطوح امنیتی به صحت بالای ۹۰٪ رسیده است و در سطوح میانی نیز میزان تشخیص به میانگین حدود ۹۳٪ رسیده و در سطح امنیتی بالا مدل توانسته به صحتی بیش از ۹۵٪ برسد و نرخ تشخیص اشتباه بدافزار را به کمتر از ۰٫۰۳٪ برساند. نکته‌ای که در این بررسی‌ها وجود دارد این است که مدل پس از اجرای مرحله‌ی دوم می‌تواند صحت خود را به میزان حدود ۵٪ افزایش دهد. این در حالی است که مدل در مرحله‌ی دوم بر روی همه‌ی رکوردهای دریافتی از مرحله‌ی اول اجرا نشده و این می تواند نقطه قوت این مدل در مقایسه با روشهای پرهزینه‌ی دیگری مانند رای‌گیری اکثریت که از ترکیب برنامه طبقه‌بندها برای ساخت مدل خود استفاده می‌کنند باشد. به عبارت دیگر راهکار پیشنهادی هم از نظر کاهش هزینه ناشی از تشخیص اشتباه و هم کاهش هزینه عملیاتی یک روش حساس به هزینه محسوب می‌شود.

در آینده می توان مدل را به صورت چند سطحی اجرا کرد و همچنین معیارهای دیگری برای اجرای سطوح امنیتی انتخاب کرد که بسته به کاربرد خاص محیط مناسبتر باشند همچنین می‌توان بر روی آستانه‌های اعمال شده کار کرد و نشان داد که چطور اگر آستانه‌های تعیین شده از حد خاصی بالاتر و یا پایین‌تر باشند نتیجه‌ی منفی در کارایی مدل خواهند داشت.

مراجع

[1] O S. Smartphone, Market Share Q2. 2016 [cited 2016; Available from: https://www.idc.com/promo/smartphone-market-share/os.

[2] Souri, A., N.J. Navimipour, and A.M. Rahmani, Formal verification approaches and standards in the cloud computing: A comprehensive and systematic review. Computer Standards & Interfaces, 2018. 58: p. 1-22.

[3] Souri, A., M. Norouzi, and P. Asghari, An analytical automated refinement approach for structural modeling large-scale codes using reverse engineering. International Journal of Information Technology, 2017. 9(4): p. 329-333.

[4] Souri, A. and R. Hosseini, A state-of-the-art survey of malware detection approaches using data mining techniques. Human-centric Computing and Information Sciences, 2018. 8(1): p. 3.

[5] Damodaran, A., et al., A comparison of static, dynamic, and hybrid analysis for malware detection. Journal of Computer Virology and Hacking Techniques, 2017. 13(1): p. 1-12.

[6] Shijo, P. and A. Salim, Integrated static and dynamic analysis for malware detection. Procedia Computer Science, 2015. 46: p. 804-811.

[7] Tong, F. and Z. Yan, A hybrid approach of mobile malware detection in Android. Journal of Parallel and Distributed computing, 2017. 103: p. 22-31.
[8] Arzt, S., et al. Flowdroid: Precise context, flow, field, object-sensitive and lifecycle-aware taint analysis for android apps. in Acm Sigplan Notices. 2014. ACM.
[9] Feng, Y., et al. Apposcopy: Semantics-based detection of android malware through static analysis. in Proceedings of the 22nd ACM SIGSOFT International Symposium on Foundations of Software Engineering. 2014. ACM.
[10] Lu, L., et al. Chex: statically vetting android apps for component hijacking vulnerabilities. in Proceedings of the 2012 ACM conference on Computer and communications security. 2012. ACM.
[11] Enck, W., et al., TaintDroid: an information-flow tracking system for realtime privacy monitoring on smartphones. ACM Transactions on Computer Systems (TOCS), 2014. 32(2): p. 5.
[12] Wang, S., et al., Detecting android malware leveraging text semantics of network flows. IEEE Transactions on Information Forensics and Security, 2017. 13(5): p. 1096-1109.
[13] Zaman, M., et al. Malware detection in Android by network traffic analysis. in 2015 international conference on networking systems and security (NSysS). 2015. IEEE.
[14] Newsome, J., B. Karp, and D. Song. Polygraph: Automatically generating signatures for polymorphic worms. in 2005 IEEE Symposium on Security and Privacy (S&P'05). 2005. IEEE.
[15] Singh, S., et al. Automated Worm Fingerprinting. in OSDI. 2004.
[16] Yegneswaran, V., et al. An Architecture for Generating Semantic Aware Signatures. in USENIX Security Symposium. 2005.
[17] Taylor, V.F., et al. Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic. in 2016 IEEE European Symposium on Security and Privacy (EuroS&P). 2016. IEEE.
[18] Conti, M., et al., Analyzing android encrypted network traffic to identify user actions. IEEE Transactions on Information Forensics and Security, 2015. 11(1): p. 114-125.
[19] Pandita, R., et al. {WHYPER}: Towards Automating Risk Assessment of Mobile Applications. in Presented as part of the 22nd {USENIX} Security Symposium ({USENIX} Security 13). 2013.
[20] Ren, J., et al. Recon: Revealing and controlling pii leaks in mobile network traffic. in Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services. 2016. ACM.
[21] Goyal, A. and R. Kaur, A Survey on Ensemble Model for Loan Prediction. International Journal of Advanced research and Innovative Ideas in Education (IJARIIE), 2016. 2(1): p. 623-628.
[22] Chakraborty, T., F. Pierazzi, and V. Subrahmanian, EC2: ensemble clustering and classification for predicting android malware families. IEEE Transactions on Dependable and Secure Computing, 2017.
[23] Chen, Z., et al., Machine learning based mobile malware detection using highly imbalanced network traffic. Information Sciences, 2018. 433: p. 346-364.
[24] Yerima, S.Y. and S. Sezer, Droidfusion: A novel multilevel classifier fusion approach for android malware detection. IEEE transactions on cybernetics, 2018. 49(2): p. 453-466.
[25] Altaher, A. and O.M. Barukab, Intelligent Hybrid Approach for Android Malware Detection based on Permissions and API Calls. International Journal of Advanced Computer Science and Applications, 2017. 8(6): p. 60-67.
[26] Arash Habibi Lashkari, A.F.A.K., Hugo Gonzalez, Kenneth Fon Mbah, Ali A. Ghorbani Towards a Network-Based Framework for Android Malware Detectionand Characterization. 15th International Conference on Privacy, Security and Trust, 2017: p. 10.
[27] Gao H, Cheng S, Zhang W. GDroid: Android malware detection and classification with graph convolutional network. Computers & Security. 2021 Jul 1;106:102264.
[28] Nikolaou N, Edakunni N, Kull M, Flach P, Brown G. Cost-sensitive boosting algorithms: Do we really need them?. Machine Learning. 2016 Sep;104:359-84.
[29] Gupta N, Jindal V, Bedi P. CSE-IDS: Using cost-sensitive deep learning and ensemble algorithms to handle class imbalance in network-based intrusion detection systems. Computers & Security. 2022 Jan 1;112:102499.
[30] Telikani A, Gandomi AH. Cost-sensitive stacked auto-encoders for intrusion detection in the Internet of Things. Internet of Things. 2021 Jun 1;14:100122.
[31] Telikani A, Shen J, Yang J, Wang P. Industrial IoT intrusion detection via evolutionary cost-sensitive learning and fog computing. IEEE Internet of Things Journal. 2022 Jul 4;9(22):23260-71.
[32] Zhang G, Wang X, Li R, Song Y, He J, Lai J. Network intrusion detection based on conditional Wasserstein generative adversarial network and cost-sensitive stacked autoencoder. IEEE access. 2020 Oct 19;8:190431-47.
[33] https://www.unb.ca/cic/datasets/android-adware.html