

بهبود افزایش امنیت و تشخیص بدافزار با استفاده از الگوریتم‌های تجمیعی

زهره بوشهری¹، پروانه اصغری^{1*}

تاریخ دریافت: 1401/05/02

تاریخ پذیرش: 1401/07/23

چکیده

یکی از راه‌های امنیت، تشخیص بدافزار در سامانه‌های کامپیوتری توسط روش‌های شناسایی بدافزار می‌باشد. از آنجایی که این امر هزینه مالی، زمانی و انسانی زیادی به همراه دارد، تحقیق پیش‌رو درصدد بوده تا با تکیه بر استخراج اطلاعات مفید داده‌ها از روی دیناست بدافزار میکروسافت با نام BIG 2015 یک کلاسه‌کننده که هم در زمینه‌ی استخراج ویژگی و هم در زمینه‌ی ساز و کار طبقه‌بند که بسیار ساده و پیچیدگی محاسباتی کمی دارد هزینه‌های ذکر شده را کاهش دهد. در این راستا جهت دستیابی به پیش‌بینی‌های بهتر و دقت بالاتر از الگوریتم‌های Xgboost و Lgb، مجموعه ویژگی‌های مبتنی بر محتوا و تکنیک‌های Feature selection، Feature Importance و permutation Importance استفاده شده است. از میان 1804 ویژگی استخراج شده ویژگی section_name_headre با وزن 0/2160 نقش مهم‌تر و پررنگ‌تری در کلاسه‌بندی ایفا کرده است. که میزان دقت کلاسه‌کننده 99/81 و خطای پیش‌بینی‌کننده به میزان 0/00774 بدست آمده است. لذا با بهره‌گیری از یافته‌های این تحقیق در سامانه‌های IDS و IPS می‌توان دقت تشخیص بدافزار را افزایش و میزان خطای تشخیص را کاهش داد.

کلید واژه : یادگیری ماشین، بدافزار، انتخاب ویژگی، Xgboost، permutation Importance

¹ گروه مهندسی کامپیوتر، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران. z.boushehri9699@gmail.com

¹ گروه مهندسی کامپیوتر، واحد تهران مرکزی، دانشگاه آزاد اسلامی، تهران، ایران. p_asghari@iauctb.ac.ir

۱. مقدمه

با فراگیر شدن استفاده از شبکه‌های کامپیوتری و افزایش ارتباطات در حوزه علوم کامپیوتری، یکی از مسائلی که توجه کارشناسان امر را به خود مشغول داشته است، امنیت سیستم‌های کامپیوتری است. از حوزه‌های مهم این علم می‌توان به شناسایی و کشف نرم‌افزارهای مخرب اشاره کرد، نرم‌افزارهای مخرب را بدافزار می‌نامند. بدافزار یک تکه کد است که رفتار سیستم عامل یا برخی از برنامه‌های حساس امنیتی را بدون رضایت کاربر تغییر می‌دهد. شناسایی این تغییرات با استفاده از ویژگی‌های ثابت اعمال سیستم عامل یا هر برنامه‌ی دیگری ممکن نیست. با پیشرفت سریع و توسعه وب، امروزه یکی از مهمترین خطرات دیجیتالی بشمار می‌آیند که در سیستم‌های کامپیوتری پخش شده و رفتار عادی سیستم را دچار اختلال می‌کنند [1]. از زمان پیدایش اولین کد مخرب کامپیوتری در سال 1986 هر ساله تعداد قابل توجهی از کدهای مخرب جدیدی ظهور پیدا می‌کنند. این رشد سریع کدهای مخرب، تلاش‌های متخصصین امنیتی برای ارائه راه‌حلی به منظور تشخیص و حذف آن‌ها از سیستم کاربران را بالا برده است [2] [3]. این نرم‌افزارها طیف وسیعی از خطرات و تهدیدها را در بر می‌گیرند و براساس نحوه آسیب رسانی به سیستم قربانی به دسته‌های مختلفی از جمله ویروس‌ها، کرم‌ها، اسب‌های تروا، جاسوس‌افزارها، آگهی‌افزارها، روت‌کیت‌ها و هرزنامه‌ها تقسیم می‌شوند [4]. روش‌های شناسایی بدافزار از دیدگاه‌های مختلفی

طبقه‌بندی شده‌اند که در این مقاله روش‌های شناسایی بدافزار به سه گروه مبتنی بر ناهنجاری، ویژگی و امضا تقسیم می‌شود که در ادامه توضیح داده می‌شوند [3].

روش مبتنی بر ناهنجاری: از دانش گذشته‌ی خود در مورد آنچه آن را رفتار عادی می‌داند جهت تشخیص بدافزار مورد بررسی استفاده می‌نماید. نوع خاصی از روش مبتنی بر ناهنجاری با نام روش مبتنی بر ویژگی شناخته می‌شود. این روش از یک سری ویژگی‌ها یا قوانینی از رفتار درست به عنوان ابزاری جهت تشخیص بدافزار بودن فایل استفاده می‌کنند. برنامه‌ای که دارای آن ویژگی‌ها نباشد مشکوک و یا بدافزار تلقی می‌شوند. روش مبتنی بر ناهنجاری معمولاً در دو مرحله انجام می‌شود. یک مرحله آموزش و یک مرحله تشخیص در طول مرحله آموزش تشخیص - گرسعی در رفتار عادی دارد. یکی از مزیت‌های اصلی روش مبتنی بر ناهنجاری توانایی آن در تشخیص حملات روز نخست است.

تشخیص مبتنی بر ویژگی: نوعی از تشخیص مبتنی بر ناهنجاری است که سعی دارد هشدار اشتباه بالای روش‌های مبتنی بر ناهنجاری را بهبود ببخشد. در روش مبتنی بر ویژگی در فاز یادگیری مجموعه‌ای از قواعد ایجاد می‌گردد که این قواعد مشخص کننده تمامی رفتارهای درستی است که هر برنامه‌ای در سیستم در حال بررسی می‌تواند از خود بروز دهد. بزرگترین

مزیت روش مبتنی بر ویژگی آن است که مشخص نمودن درست و دقیق مجموعه رفتارهایی که برنامه می‌تواند از خود بروز دهد [5,6].

روش مبتنی بر امضا: از امضای خاصی که به هر فایل تعلق دارد جهت تشخیص بدافزار بودن فایل‌های مشابه استفاده می‌نماید. بصورت ایده‌آل یک امضا باید بتواند هر بدافزاری را که رفتاری شبیه آنچه در امضا مشخص شده از خود بروز دهد را شناسایی نماید. این امضاها نیازمند مخزنی جهت ذخیره سازی هستند. این مخزن تمامی دانشی است که روش مبتنی بر امضا دارا است. هنگامی که روش مبتنی بر امضا سعی در کشف یک بدافزار درون فایل دارند این مخزن جستجو می‌شود. در حال حاضر امضاهایی که نشان دهنده بدافزار هستند عمدتاً توسط عوامل انسانی خیره توسعه داده می‌شوند. هنگامی که امضای جدیدی تشخیص داده می‌شود به مخزن دانش اضافه می‌گردد. یکی از اشکالات عمده روش مبتنی بر امضا عدم تشخیص حملات روز نخست است زیرا هیچ امضایی از این نوع در مخزن وجود ندارد (اسکندری و هاشمی 2012، سانتوز و دیگران 2013، ایگل و دیگران 2007).

هر روش تشخیصی می‌تواند بر یکی از سه روش ایستا، پویا یا ترکیبی باشد (روپیا و دیگران، 2009). روش ایستا از ویژگی‌هایی مانند کد برنامه جهت تشخیص بدافزار بودن آن استفاده می‌کند در حالی که روش‌های پویا از اطلاعات اجرایی مانند پشته برنامه

در حال اجرا بهره می‌برد. به صورت کلی روش ایستا قبل از اجرای برنامه بدافزار بودن آن را تشخیص می‌دهد ولی روش پویا سعی در تشخیص بدافزار در حین اجرا یا پس از اجرای آن دارد. روش‌های ترکیبی نیز وجود دارند (رابک و دیگران، 2003) که در آن‌ها اطلاعات ایستا و پویا جهت تشخیص بدافزار بودن استفاده می‌گردد. برای شناسایی بدافزارها، روش‌های متعددی پیشنهاد شده است. یکی از روش‌هایی که اخیراً در این حوزه مطرح شده است، استفاده از الگوریتم‌های تجمیعی برای شناسایی خانواده بدافزارها می‌باشد. استفاده از این روش‌ها به دلیل ماهیت عملکرد بالا، می‌تواند موثر واقع شود.

در این مقاله، از مجموعه داده‌های **Big Data** تشخیص بدافزار سایت **kaggle** استفاده می‌شود و از دقت و خطا به عنوان دو معیار جهت ارزیابی و آزمون کارایی روش پیشنهادی استفاده می‌شود. برای دستیابی به این دو معیار در این مقاله برای انتخاب ویژگی در بخش کلاس‌بندی از الگوریتم **LGB** و الگوریتم **Xgboost** که هر دو الگوریتم زیر مجموعه‌ی جنگل تصادفی و از الگوریتم‌های تجمیعی هستند، استفاده شده است. در انتخاب ویژگی‌های مناسب از تکنیک‌های **Feature Importance** و **permutation Importance** استفاده می‌شود که اساس و پایه‌ی این پژوهش جهت نوآوری تکنیک‌های جدید انتخاب ویژگی می‌باشد. در این تحقیق با استفاده از روش‌های انتخاب ویژگی

برای دستیابی به بهترین زیرمجموعه ویژگی‌ها و همچنین دستیابی به پیش بینی‌های بهتر و رسیدن به بالاترین دقت بر اساس این زیرمجموعه کمینه از ویژگی‌ها از مدلسازی پیش‌بینانه استفاده شده است. در روش پیشنهادی انواع مختلف تکنیک‌های داده‌کاوی که می‌توانند در زمینه ترکیب طبقه‌بندها مورد استفاده قرار گیرد، مورد بحث و بررسی قرار می‌گیرد. همچنین مفاهیم پایه امنیت و راه‌های تشخیص بدافزار مورد بحث و بررسی قرار می‌گیرد تا ساختارهای مورد استفاده برای ترکیب تشریح گردد.

ساختار این مقاله به این شرح است: در بخش 2، به بیان پیشینه این تحقیق می‌پردازیم. در بخش 3، روش پیشنهادی به تفصیل توضیح داده می‌شود. در بخش 4، نتایج ارزیابی روش پیشنهادی در مقایسه با سایر روش‌ها ارائه می‌شود و در بخش 5 به نتیجه‌گیری و کارهای آتی اشاره خواهد شد.

2. پیشینه تحقیق

یکی از مقوله‌های امنیت فناوری اطلاعات امن بودن فضای تبادل اطلاعات است. بدون شک بدافزارها یکی از مهمترین تهدیدهای امنیتی برای فناوری اطلاعات بوده و هستند. پیشرفت در توسعه‌ی بدافزارها از یک سو همچنین اهمیت امنیت سیستم‌های کامپیوتری از سوی دیگر، مقابله با این تهدید بزرگ به یکی از مباحث به روز در حوزه‌ی امنیت سیستم‌های کامپیوتری تبدیل شده است.

ریچا شارما و همکارانش در سال 2020 یک تکنیک جدید در شناسایی بد افزار Trojan با استفاده از کلاس طبقه بندی شده XGBoost ارائه نمودند. تکنیک‌های موجود در تشخیص بد-افزار تروجان قادر به تشخیص Trojan ها به طور دقیق نیستند، تعادل کلاس را به طور کارآمد کنترل نمی‌کنند، از اهمیت ویژگی درختکاری استفاده نمی‌شود و دارای نرخ مثبت و منفی زیادی هستند، بنابراین برای غلبه بر این مشکلات، این مقاله یک تکنیک جدید تشخیص وزن مبتنی بر Extreme Gradient Boosting (CWXGB) را ارائه می‌دهد، که Trojan ها را با استفاده از بهترین مجموعه مقادیر ترکیبی و پی در پی SCOAP و HT را از netlist سطح گیت تشخیص می‌دهد. یک طرح توزین در مدل CW-XGB برای حل مشکل عدم تعادل کلاس با اختصاص وزنه‌های بالاتر به اقلیت وارد شده در تروجان ارائه شده است. علاوه بر این، یک روش انتخاب ویژگی جدید است که بهترین مجموعه ویژگی‌ها را با استفاده از importance Permutation انتخاب می‌کند و همچنین از تکرار مدل جلوگیری می‌کند. نتایج ارزیابی نشان می‌دهد که روش پیشنهادی به طور متوسط، دقت و یادآوری 04:99٪، 86:98٪، و سرانجام، دقت کل مدل را 73:98٪ ارائه می‌دهد که اندازه گیری بالایی از قابلیت تفکیک را نشان می‌دهد.

در مقاله [9]، سونیتا چوداری و همکارش در سال 2020 الگوی رفتاری مطالعه خود را با استفاده از تکنیک‌ها و الگوریتم‌های یادگیری ماشین که از طریق تجزیه و تحلیل استاتیک یا دینامیکی بوده است را برای تشخیص و طبقه‌بندی بدافزارها بدست آوردند، در این تحقیق، روش‌های تشخیص مبتنی بر رفتار [10] را برای استفاده از الگوریتم‌های یادگیری ماشین مورد بحث قرار دادند تا مدل شناسایی و طبقه‌بندی بدافزار مبتنی بر رفتار خود را چارچوب‌بندی کنند. نتایج ارزیابی آزمایش‌های این تحقیق جهت تشخیص بدافزار که با استفاده از برنامه ی J48 بدست آمده است دقت 96.8 درصد را نشان می‌دهد.

روهیت سریواستا و همکارانش در 5 دسامبر 2020 در مقاله [11] به مطالعه بر روی تشخیص حملات بدافزارها در دستگاه‌های تلفن همراه که از سیستم عامل اندروید استفاده می‌شود پرداختند. روش ارائه شده توسط این گروه محقق ایجاد یک تشخیص‌گر و تحلیل‌گر بدافزار می‌باشد در این پژوهش همچنین بینش‌هایی را درباره حملات بدافزار COVID-19 در دستگاه‌های تلفن همراه پیوند می‌دهد. برای دستیابی به این اهداف، مدلی پیاده‌سازی شده است که ویژگی‌های ذاتی فایل برنامه اندروید را استخراج کرده و آن‌ها را برای تجزیه و تحلیل سریع و دقیق تجزیه و تحلیل می‌کند که 400 برنامه را به طور کامل از بازار رسمی اندروید گوگل پلی -

دیمین ماریون و همکارانش در سال 2021 در مقاله [7] رویکردی با استفاده از تابش الکترومغناطیسی برای آشکارساختن مبهمات بدافزارها جهت شناسایی انواع تهدیدات و طبقه‌بندی آن‌ها که دستگاه‌های اینترنت اشیا را مورد هدف قرار می‌دهند، ارائه نمودند. آنها با گرفتن 100000 نمونه اندازه‌گیری از یک سیستم اینترنت اشیا آلوده به نمونه‌های مختلف بدافزار دست پیدا نمودند که دقت پیش‌بینی آزمایش آنها عدد 99.89 درصد را نشان می‌دهد. علاوه بر این، نتایج تحقیق آنها نشان می‌دهد نمونه‌های بدافزار تغییر یافته را با تکنیک‌های مبهم‌سازی در مرحله آموزش طبقه‌بندی و تعیین کردند که چه نوع مبهم‌هایی وجود دارد که برای تجزیه و تحلیل بدافزارها برای تحلیل گران مفید واقع شود. در مقاله [8]، بوراک طه‌تاجی و همکارش در سال 2020 در کنفرانس نوآوری در سیستم‌های هوشمند و برنامه‌های کاربردی روشی برای تشخیص بدافزار در دستگاه‌های تلفن هوشمند اندرویدی با استفاده از الگوریتم‌های یادگیری ماشین ارائه نمودند. در این مطالعه فایل‌های smali که بسته‌های اندروید دیکامپایل شده هستند را با استفاده از ویژگی‌های n-gram که از مدل‌های یادگیری ماشین می‌باشد را استخراج کردند و همچنین ترکیب استخراج ویژگی و انتخاب ویژگی را از مدل‌های آموزش دیده انجام داده‌اند.

نهایی اعمال می‌کند. مدل DroidFusion چند سطحی می‌تواند بعنوان یک پیش‌بینی‌کننده دقت بهبود یافته برای تشخیص بدافزار اندرویدی مورد استفاده قرار گیرد.

نظروال هوک و همکارانش در سال ۲۰۱۸ در مقاله [14]، یک روش انتخاب ویژگی مجموعه برای طبقه بندی مجموعه داده‌ها از روش‌های انتخاب ویژگی با نام Ensemble Feature Selection با استفاده از اطلاعات متقابل (EFS-MI) معرفی نمودند که ترکیبی از زیر مجموعه‌های ویژگی‌های انتخاب شده توسط فیلترهای مختلف مانند InfoGain، GainRatio، ReliefF، Chi-square و عدم قطعیت متقارن و زیر مجموعه‌ای بهینه از ویژگی‌ها را به همراه دارد است. برای ارزیابی عملکرد این روش، آنها از داده‌های NSL-KDD و TUIDS برای تایید صحت طبقه بندی روش خود با استفاده از طبقه‌بندی کننده‌های مختلف، مانند درختان تصمیم‌گیری، جنگل‌های تصادفی، KNN و SVM در مجموعه داده‌های استفاده کردند و به دقت ۹۸/۹۲ درصد رسیدند. آنها در این مقاله از تکنیک جستجوی حریص در زیر مجموعه داده‌های بهینه استفاده کردند و معتقد بودند که برنامه‌های مخرب به سرعت در حال توسعه و گسترش هستند. بنابراین ضروری است که دقت تشخیص بدافزارها را در برنامه‌های سیستم اندرویدی همانند برنامه‌های سیستم ویندوزی باید بهبود بخشید.

استور جمع آوری کرده اند و با استفاده از الگوریتم K-means که مبتنی بر ویژگی‌های ترکیبی هستند به دقت 96.12٪ رسیدند.

الخاندر و مارتینا و همکارانش در سال ۲۰۱۹ در مقاله [12] یک روش شناسایی بدافزار را بر اساس تلفیق ویژگی‌های استاتیک و پویا از طریق ترکیب طبقه بندی کننده‌های گروه برای سیستم‌های اندرویدی ارائه نمودند. در این تحقیق مکانیسم‌های تشخیص، از یادگیری ماشین برای ساخت طبقه بندی چارچوب AndroPyTool و مجموعه داده OmniDroid در تشخیص نرم افزارهای مخرب یا خوش خیم که در تشخیص موثر است استفاده کردند. برای این منظور از ۲۲۰۰۰ نمونه بدافزارهای واقعی نرم افزاری را با هدف کمک به سازندگان و محققان ابزارهای ضد بدافزار هنگام بهبود یا توسعه مکانیسم‌ها و ابزارهای جدید برای شناسایی بدافزارهای اندرویدی ارائه نمودند.

سلیمان یریما و همکارش در سال ۲۰۱۸ در مقاله [13]، یک رویکرد ترکیبی طبقه‌بندی جدید مبتنی بر یک معماری چندسطحی را ارائه نمودند. که این رویکرد، ترکیبی موثر از الگوریتم‌های یادگیری ماشین برای بالا بردن بهبود دقت می‌باشد. این چارچوب که معرف به DroidFusion است مدلی را ایجاد می‌کند که با آموزش طبقه بندی کننده‌های پایه در یک سطح پایین، مجموعه‌ای از الگوریتم‌های مبتنی بر رتبه‌بندی را بر روی دقت‌های پیش‌بینی کننده آنها در سطح بالاتر به منظور به دست آوردن یک طبقه‌بندی

در مقاله [15]، جی فانگ و همکارانش در سال ۲۰۱۹، یک روش تشخیص ترکیبی برای بدافزارهای سیستم‌های اندروید ارائه نمودند که تشخیص پویا را بر روی برنامه‌های مشکوک به دست آمده از تشخیص استاتیک انجام می‌دهد. نتایج آزمایشات این مقاله نشان می‌دهد که این روش از دقت بالایی برخوردار است و پیچیدگی زمان کم دارد و الگوریتم *xgboost* با عملکرد بهینه دقت تشخیص این روش را بالاتر از تشخیص استاتیک و تشخیص پویا با نرخ ۹۴.۶٪ انجام می‌دهد.

در گذشته بسیاری از پژوهشگران روی نوع خاصی از مسائل تمرکز می‌کردند و تعداد بسیار کمی از آن‌ها روی چندین بهبود برخی هدف‌ها به صورت همزمان کار می‌کنند. در [۱۶]، سیستم‌های یادگیری مبتنی بر دقت *DTGA* ارائه شده‌است که هدف آن بهبود دقت پیش‌گویی با توجه به مسائل مربوط به دامنه، اندازه، ابعاد مجموعه داده و توزیع شدگی کلاس‌ها می‌باشد. دقت و قابلیت تفسیر به صورت همزمان مورد توجه قرار گرفته است و در این مقاله تلاش شده تا علاوه بر دقت، قابلیت درک مسئله نیز افزایش یابد و هرچه تعداد قانون کمتر باشد، آن قانون قابل تفسیرتر می‌باشد.

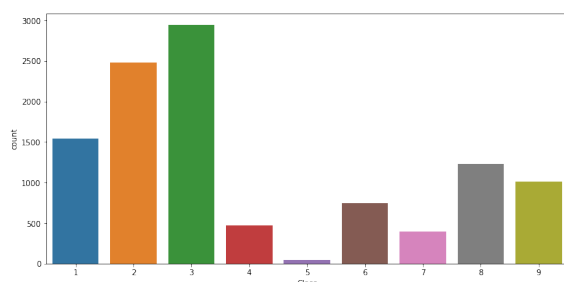
دیوید دونالد و همکارش در سال ۲۰۱۷ در مقاله [۱۷]، برای بهبود شناسایی بدافزار در فضای اینترنت از روش گروه‌انتخابی برای طبقه‌بندی ترافیک اینترنت استفاده نمودند، طبقه‌بندی ترافیک

اینترنت برای فعالیت‌های مختلف شبکه مانند شناسایی بدافزار یک رویکرد کارآمد و بسیار حیاتی محسوب می‌شود. در این پژوهش از تکنیک هرس چند مرحله‌ای برای هرس طبقه‌بندی کننده‌های گروه بزرگ و همچنین از اسکریپت نویسی *Groovy* استفاده شده‌است. که این تکنیک ابزاری قدرتمند است که باعث کاهش چشمگیر زمان کدگذاری می‌شود. استفاده از اسکریپت نویسی می‌تواند یکی از روش‌های مناسب در طبقه‌بندی داده‌های بزرگ باشد. برای تجزیه و تحلیل این تحقیق از نرم افزار داده‌کاوی و بدافزار *n-gram* به عنوان مجموعه داده از مخازن داده و همچنین در آزمایشات آنها از *SimpleCLI* در *WEKA* برای تولید و اجرای همه طبقه‌بندی‌ها در طول آزمایش استفاده شده‌است. معیارهای عملکرد در این مقاله تضمین می‌کند که مدل هرس ساخته شده قوی است تا بدافزارها را به طور موثر تشخیص دهد. از متریک سطح زیر منحنی، برای ارزیابی اثربخشی طبقه بندی استفاده شده است.

3. روش پیشنهادی

انتخاب ویژگی یکی از روش‌های موثر در کاهش ابعاد داده و در نتیجه، محدود کردن فضای مسأله برای الگوریتم‌های یادگیری ماشین است. هدف انتخاب ویژگی که از آن به عنوان انتخاب زیرمجموعه ویژگی‌ها هم یاد می‌شود، حذف ویژگی‌های بی ربط و زائد است. اخیراً روش‌های مبتنی بر الگوریتم‌های *XGBM*

10873 رکورد متعلق به مجموعه داده آزمایش هستند. در تعداد رکوردهای متعلق به هر گروه از بدافزار نشان داده شده است که طبق آن، کلاس 3 دارای بیشترین و کلاس 5 دارای کمترین تعداد رکوردها در مجموعه داده آموزش است.

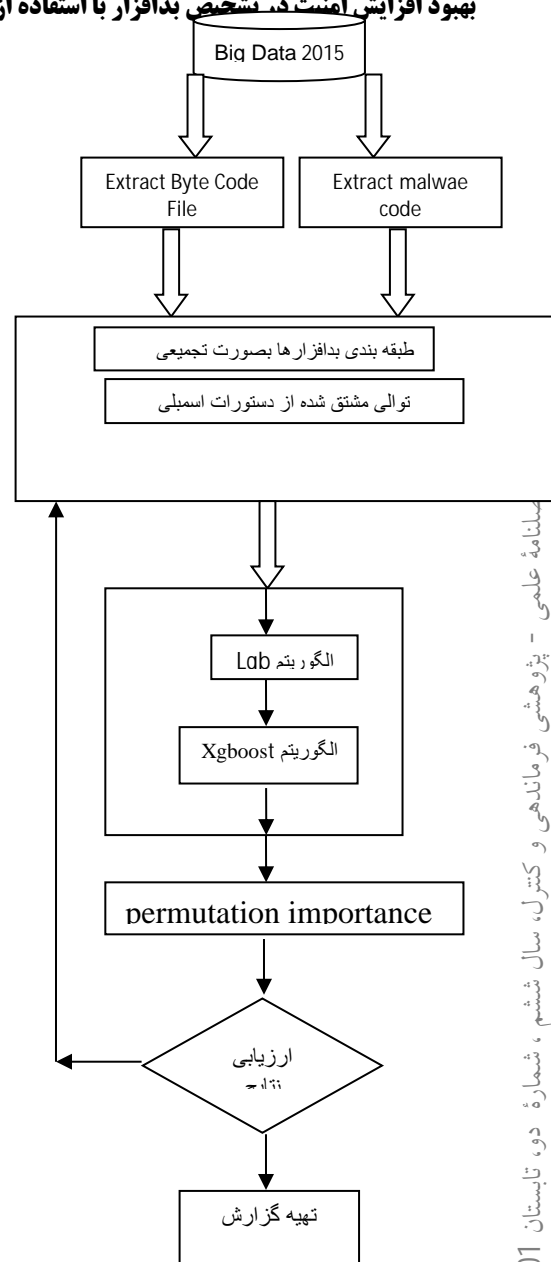


شکل 1. گروه های مختلف بدافزار

و LIGHTGBM (که به اختصار `xgb` و `lgb` نامیده می شوند) به عنوان روش هایی کارآمد برای انتخاب ویژگی، پذیرفته شده اند. از این رو در این مقاله از `feature importance` این الگوریتم ها برای انتخاب بهترین زیرمجموعه ویژگی ها بر اساس سطوح مختلف اهمیت استفاده می شود. مایکروسافت تقریباً نیمی از اطلاعات مربوط به 21741 نمونه بدافزار را منتشر کرده است که 10868 نمونه از آن برای آموزش مدل کار پیش رو استفاده شده است. شناسه هر نمونه بدافزار یک کد هش 20 کارکتری است. این فایل ها از 9 خانواده مخرب به نام های `Vundo(V)`, `Simda(S)`, `Tracur(T)`, `kelihos_ver1`, `Ramnit(R)`, `Lollipop(L)`, `kelihos_ver3(K3)`, `Gatak(G)`, `Obfuscator.ACY(O)`, `K1` تشکیل شده اند [18]. برچسب کلاس ها یک عدد صحیح از 1 تا 9 می باشد، اگر "1" باشد اولین خانواده ی مخرب در لیست فوق را نشان می دهد و اگر 9 باشد آخرین خانواده را نشان می دهد. دو فایل برای هر نمونه مخرب وجود دارد، یکی حاوی کد هگز و دیگری حاوی کد دیس اسمبل شده است. مایکروسافت هدر PE را برای اطمینان از عدم فاش شدن هویت بدافزارها از بین برده است. در شکل 1 توزیع داده ها در 9 خانواده ابر `CoLab` نشان داده شده است. در فاز اول این پروژه، از مجموعه داده دسته بندی بدافزار مایکروسافت، 1804 ویژگی برای 9 دسته از بدافزارها استخراج شد. در مجموعه داده حاصل، 10868 رکورد متعلق به آموزش و

بدافزارها می باشند. در نهایت این ویژگی ها بصورت تجمیعی برای طبقه بندی بدافزارها مورد استفاده قرار می گیرد. بدست آوردن ویژگی های متعدد و موثر از دیتاست تا زمانی که کلاسه کننده بتواند بهترین شکل کار خود را انجام دهد، ادامه خواهد داشت. لذا در روش فوق از چندین الگوریتم برای استخراج ویژگی ها استفاده شده است. در واقع ابتدا مدل سازی 90 درصد از داده های آموزش (10درصد از داده به عنوان داده اعتبارسنجی در ارزیابی stratified kfold cross validation)، با الگوریتم های xgb و lgb انجام شده و بر اساس feature importance هر یک از این دو الگوریتم، چند زیر مجموعه از لیست ویژگی های مجموعه داده آموزش انتخاب می شود. سپس مدل سازی و پیش بینی جدید بر اساس مجموعه ویژگی های کاهش یافته صورت گرفته و سرانجام، میزان دقت پیش بینی آنها بر اساس معیار logloss با هم مقایسه می شود.

علاوه بر این، با استفاده از permutation importance هم به انتخاب زیرمجموعه ویژگی ها پرداخته شده است. در این روش، برای هر مجموعه داده به تعداد ویژگی های آن مدل سازی و ارزیابی انجام می گیرد. بدین صورت که در هر مرحله از این روش، مقادیر یک ویژگی، در هم سازی شده و سپس مدل سازی می گردد و بر اساس میزان تفاوت خروجی مدل بین دو حالت اولیه و در هم سازی شده و مهم ترین ویژگی ها استخراج شد. روش های فوق، روش های کاهنده ابعاد داده ها هستند. زیرا از بخشی از ویژگی های موجود که بیشترین تأثیر را در پیش بینی دارند، انتخاب می کنند. هدف این مقاله، استفاده از روش های انتخاب ویژگی برای بهترین زیرمجموعه ویژگی ها و دستیابی به پیش بینی های بهتر بر اساس این زیرمجموعه کمینه از ویژگی - هاست. در مقابل، روش های افزایش ابعاد داده ها، نظیر استخراج



شکل 2. فرآیند روش پیشنهادی

در واقع، در روش پیشنهادی داده کاوی بر روی بدافزارها و استخراج ویژگی هاست که بیشترین تأثیر را در تشخیص دسته بندی بدافزارها خواهد داشت. لذا در این راهکار استخراج ویژگی های عمومی بدافزارها به منظور طبقه بندی آن ها با توجه به فضای مساله صورت گرفته است که این ویژگی ها در تمامی بدافزارها به صورت مشترک مورد استفاده قرار گرفته است. که شامل توالی مشتق شده از دستورات اسمبلی و فراخوانی توابع سیستمی توسط

ویژگی می‌توان استفاده کرد که ویژگی‌های جدیدی را به مجموعه ویژگی‌های موجود می‌افزایند. در این مقاله، با استفاده از استخراج ویژگی‌ها، چهار ویژگی جدید به ویژگی‌های داده‌ها افزوده شد.

3-1. بررسی و پیش پردازش داده‌ها

3-1-1. بررسی همبستگی ویژگی‌ها

میزان همبستگی ویژگی‌ها با ویژگی هدف محاسبه و نمایش داده می‌شود. همانگونه که در جدول 1 مشاهده می‌شود، ویژگی‌های `_setusermaterr`، `_p_fmde` و `HeapCreate` دارای بیشترین همبستگی با ویژگی هدف هستند. ماتریس همبستگی نشان می‌دهد که بسیاری از ویژگی‌ها مانند "TB-f0" و "img161" و "img321" دارای همبستگی بالایی با یکدیگر هستند. این موضوع می‌تواند منجر به multicollinearity شود.

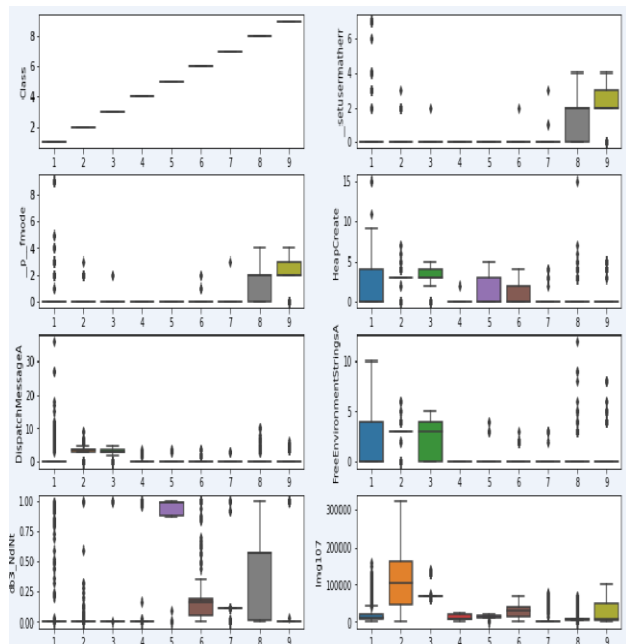
جدول 1. ویژگی‌های دارای بیشترین همبستگی با ویژگی class

Class	1.000000
__setusermaterr	0.545989
__p_fmde	0.533037
HeapCreate	0.509319
DispatchMessageA	0.471076
FreeEnvironmentStringsA	0.457317
db3_NdNt	0.433308
Img107	0.407566
TB_89	0.404929
TB_f0	0.397401
FreeEnvironmentStringsW	0.396155
Img32.1	0.395749
TlsAlloc	0.395407
_acmdln	0.390966
img16.1	0.389151

با به دست آوردن میزان همبستگی ویژگی‌ها نسبت به ویژگی -

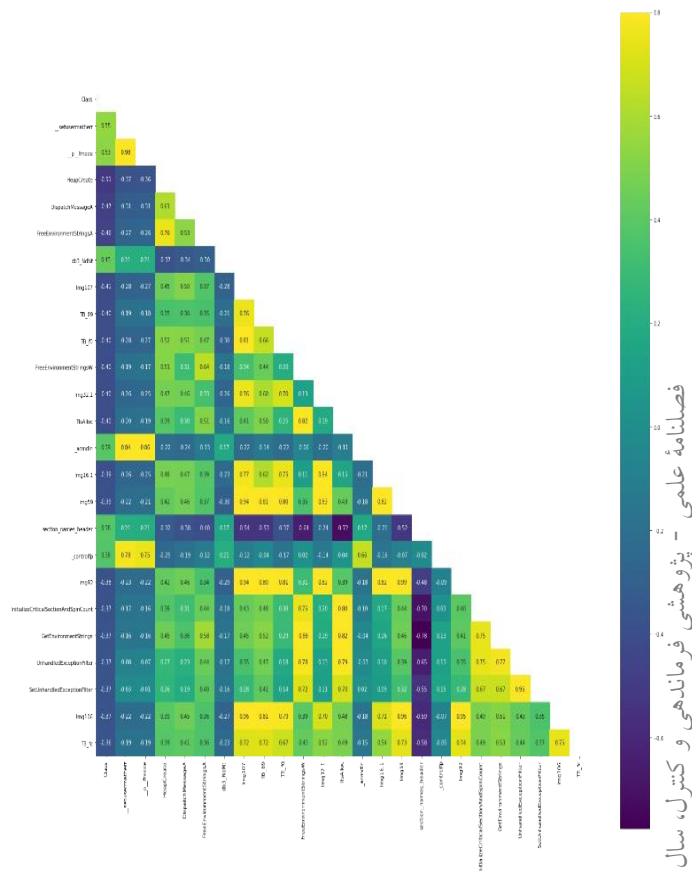
ها، ویژگی کلاس آن دسته از ویژگی‌ها که `null` بوده‌اند با

میانگین جایگزین شده‌اند. لازم به ذکر است که در ماتریس همبستگی آن دسته از ویژگی‌ها مقداری نزدیک به صفر داشته‌اند نقش کم‌رنگی در دسته‌بندی دارند و هرچه عدد ویژگی در این ماتریس به 1 و 1- نزدیک تر باشد نقش موثرتری در دسته بندی دارد. بعنوان مثال : BYTES با CLASS نسبت معکوس قوی دارد و در دسته بندی بسیار موثر است. نظر به خروجی می‌توان دریافت که رسم یک جعبه برای هر ترکیب سطرها و ستون‌ها، در اصل کاری است که نقشه حرارتی انجام می‌دهد. رنگ جعبه، به مقدار آن خانه بستگی دارد. مثلاً در شکل 3، چنانچه همبستگی بالایی میان دو ویژگی وجود داشته باشد، خانه یا جعبه متناظر، زرد است، از طرف دیگر، چنانچه همبستگی وجود نداشته باشد، خانه متناظر، بنفش باقی می‌ماند. این طیف رنگی از کمترین مقدار تا بیشترین مقدار تمام خانه‌های ماتریس تغییر می‌کند که در سمت راست جدول قابل مشاهده است. مقادیر همبستگی را نیز، از طریق ارسال `True` برای پارامتر `annot` در نقشه حرارتی می‌توان مشاهده نمود.



شکل 4-الف. نمودار جعبه ای ویژگی های دارای بیشترین همبستگی با

ویژگی class

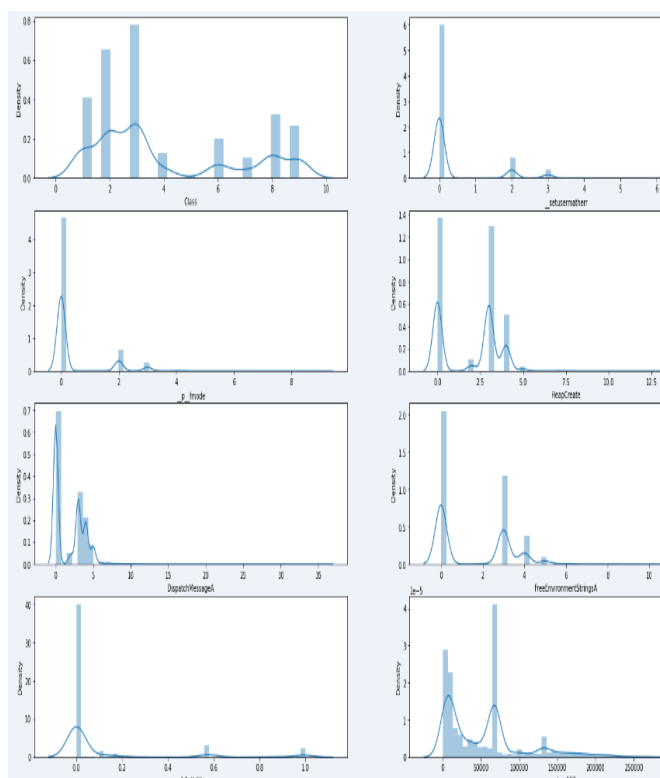


شکل 3. ویژگی های دارای بیشترین همبستگی با ویژگی class

3-1-2. بررسی پرت بودن داده بر اساس نمودار جعبه ای

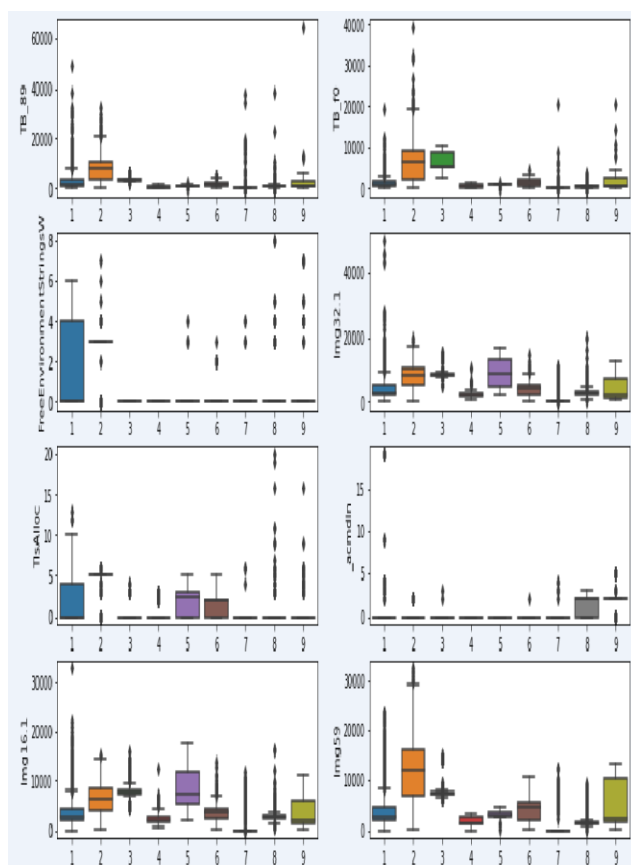
در این بخش با استفاده از نمودارهای جعبه‌ای شکل‌های 4-الف و 4-ب، میزان پرت بودن مقادیر بخشی از ویژگی‌هایی که دارای بیشترین میزان همبستگی با متغیر هدف هستند مورد بررسی قرار گرفته است. کاملاً مشخص است که تعدادی از مقادیر اکثر ویژگی‌ها، پرت هستند.

بخشی از متغیرهای دارای بیشترین میزان همبستگی با متغیر هدف بررسی شد. این موضوع در شکل های 5-الف و 5-ب نشان داده شده است که بر اساس آن، اکثر متغیرها دارای توزیع غیر نرمال هستند. محاسبه میزان چولگی مقادیر ویژگی ها نیز این موضوع را تأیید می کند.



شکل 5-الف. نمودار توزیع ویژگی های دارای بیشترین همبستگی با

ویژگی class



شکل 4-ب. نمودار جعبه ای ویژگی های دارای بیشترین همبستگی با

ویژگی class

3-3-1. بررسی وضعیت نرمال بودن داده ها

بسیاری از الگوریتم های یادگیری ماشین براساس مجموعه ای از فرضیات و قواعد آماری عمل می کنند. و این فرضیات و قواعد برای داده هایی طراحی شده اند که توزیع آنها نرمال است. هر چند در صورتی که مجموعه داده بزرگ باشد، توزیع داده ها، نرمال فرض می شود، ولی در این صورت هم بهتر است نرمال بودن داده ها بررسی شده و تا حد ممکن با تبدیلات داده ای لازم، توزیع داده ها به توزیع نرمال نزدیک شود. در این بخش میزان نرمال بودن

۳- مقادیر ویژگی‌هایی که چولگی آنها کمتر از 0.5- است،

به مجذور مقادیر همان ویژگی تبدیل می‌شوند.

```
skews= pd.Series(index= df_x.columns)
for col in df_x.columns:
    skews[col] = df_x[col].skew()

index_list= list(skews.index)
for x, i in enumerate(index_list):
    if x_train[i].skew() < -0.5:
        for j in range(len(x_train)):
            x_train.iloc[j, x] = np.square(x_train.iloc[j, x])
    if x_train[i].skew() > 0.5:
        for j in range(len(x_train)):
            if x_train.iloc[j, x] != 0:
                x_train.iloc[j, x] = np.log(np.abs(x_train.iloc[j, x]))

return x_train
```

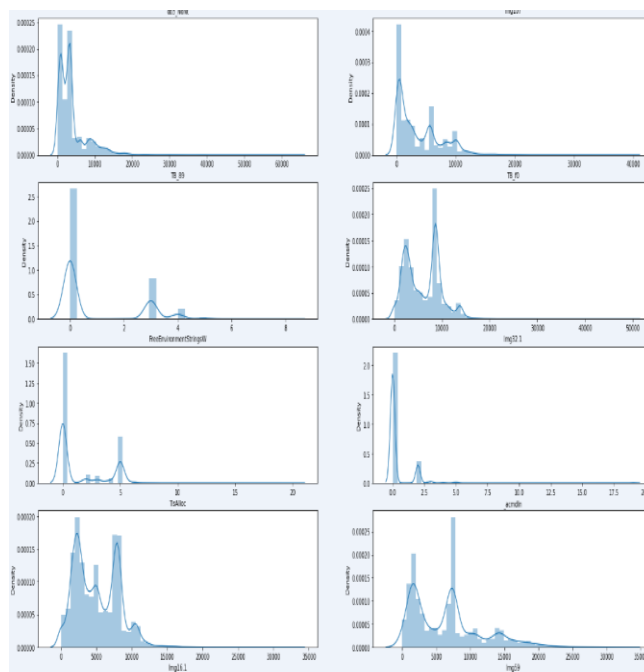
شکل 6. روش کاهش چولگی مقادیر ویژگی‌ها

بررسی چولگی اولیه داده‌ها در شکل 7 نمایش داده

شده‌است.

MethCallEngine	104.249700
_vbaErase	104.249700
_vbaVarZero	104.249700
_vbaInStrVar	104.249700
_vbaAryVar	104.249700
...	
ent_p_19	-4.767019
ent_q_diff_diffs_median	-7.770609
ent_q_diff_diffs_min	-11.153566
ent_q_diff_diffs_0_max	-11.537138
ent_q_diffs_max	-13.403123

شکل 7. بررسی چولگی اولیه داده‌ها



شکل 5-ب. نمودار توزیع ویژگی‌های دارای بیشترین همبستگی با

ویژگی class

3-1-4. تبدیل داده به منظور رفع چولگی داده‌ها:

معمولا برای رفع چولگی غیر نرمال داده‌ها، مجموعه‌ای از

تبدیلات داده‌ای روی داده‌ها انجام می‌شود. در این مقاله، از روش

شکل 6 برای کاهش چولگی مقادیر ویژگی‌ها استفاده شده است:

۱- توزیع ویژگی‌هایی که چولگی آن‌ها بین 0.5- و 0.5

است، نرمال فرض می‌شود.

۲- مقادیر غیر صفر ویژگی‌هایی که چولگی آنها بیش از

0.5 است، به لگاریتم مقادیر همان ویژگی تبدیل

می‌شوند.

logloss_xgb_raw_data	0.01194
logloss_xgb_sk_handled_data	0.01576
logloss_lgb_raw_data	0.0082
logloss_lgb_sk_handled_data	0.01225

2-3 استخراج ویژگی

استخراج ویژگی از ویژگی های موجود، از تکنیک های موثر در

بهبود یادگیری الگوریتم‌هاست. بدین منظور در این مقاله، چهار

ویژگی به شرح زیر به مجموعه ویژگی ها، اضافه شد:

```
x_train_feature_added = x_train.copy()
x_train_feature_added['sumImg'] = x_train_feature_added['Img107']
for i in range(107):
    feat_name = 'Img{0}'.format(i)
    x_train_feature_added['sumImg'] = x_train_feature_added['sumImg'] + x_train_feature_added[feat_name]
    x_train_feature_added['sumImport'] = x_train_feature_added['Import'] + x_train_feature_added['Imports']
    x_train_feature_added['loc_by_mean'] = x_train_feature_added['loc']/x_train_feature_added['loc'].mean()
    x_train_feature_added['var_by_mean'] = x_train_feature_added['var']/x_train_feature_added['var'].mean()
    x_train_feature_added['Forwarder_by_mean'] = x_train_feature_added['Forwarder']/x_train_feature_added['Forwarder'].mean()
x_train_feature_added.head()
```

در جدول 3 تأثیر اضافه شدن چهار ویژگی جدید به مجموعه

ویژگی‌ها نشان داده شده است. نتایج نشان می‌دهد که افزودن این

ویژگی‌ها باعث بهبود عملکرد در هر یک از مدل های lgb و

xgb شده است.

بررسی چولگی داده‌های نرمال شده در شکل 8 نمایش داده شده است.

```
x_train_sk_handled.skew().sort_values(
    __vbaStrI2          104.249700
    __vbaVarZero       104.249700
    __vbaErase         104.249700
    __vbaAryConstruct2 104.249700
    __vbaInStrVar      104.249700
    ...
    ent_q_diff_diffs_min -5.124846
    tls_por              -5.602426
    ent_q_diff_block_2_20 -5.699199
    dd5_NdNt            -8.298382
    db3_idata           -48.985028
    Length: 1804, dtype: float64
```

شکل 8. بررسی چولگی داده های نرمال شده

3-4-1 آموزش و ارزیابی با مقادیر تبدیل یافته (تبدیلات

کاهش چولگی داده‌ها)

ضرورت و نحوه تبدیل داده‌ها به منظور کاهش چولگی داده‌ها

در بخش سوم این مقاله مورد بحث قرار گرفت. در ادامه ی این

بخش و در این مرحله، مدلسازی و ارزیابی هر یک از الگوریتم

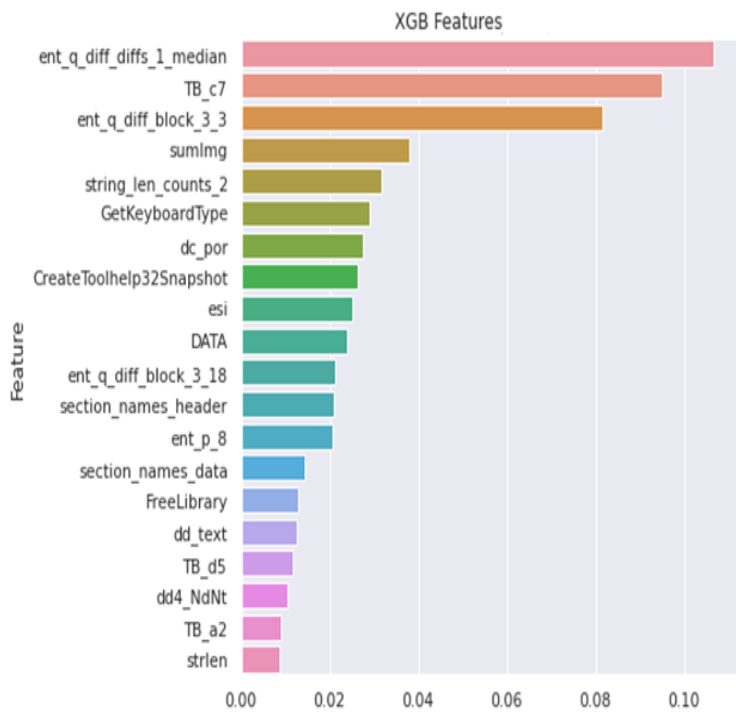
های lgb و xgb با استفاده مجموعه مقادیر اولیه و مجموعه

مقادیر تبدیل یافته داده‌ها انجام گرفته است و نتایج ارزیابی‌ها

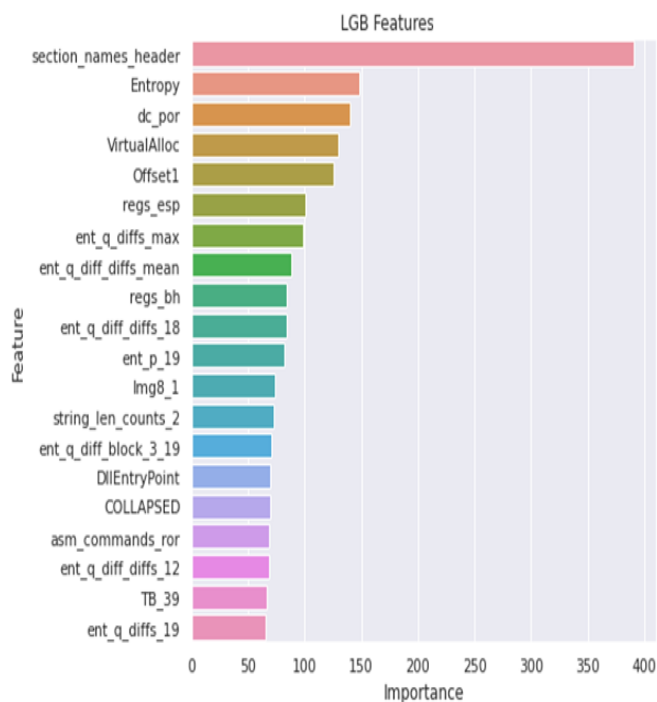
مطابق جدول 2 نشان می‌دهد که این تبدیلات موجب کاهش

عملکرد مثبت مدل می‌شوند.

جدول 2. مقایسه نتایج مقادیر اولیه و مجموعه مقادیر تبدیل یافته داده‌ها



شکل 9.20 ویژگی مهم حاصل از xgb



شکل 10.20 ویژگی مهم حاصل از lgb

جدول 3. مقایسه نتایج آموزش مدل‌ها با ویژگی‌های اولیه و همه ویژگی‌ها

logloss_xgb_raw_data	0.01194
logloss_xgb_allfeat	0.01148
logloss_lgb_raw_data	0.0082
logloss_lgb_allfeat	0.00744

3-2-1 استخراج اهمیت ویژگی‌ها

ویژگی‌های مختلف از اهمیت برابری برای شرکت در تعیین نتایج الگوریتم‌های یادگیری ماشین برخوردار نیستند. یکی از روش‌های استخراج میزان اهمیت ویژگی‌ها، استفاده از **feature importance** الگوریتم‌های **lgb** و **xgb** است. در این مرحله ابتدا هر یک از این الگوریتم‌ها با همه ویژگی‌ها، آموزش دیده و در نتیجه آن، میزان اهمیت ویژگی‌ها استخراج شده است. در شکل 9 و 10، 20 مورد از مهم‌ترین ویژگی‌های مجموعه داده به همراه میزان اهمیت آنها، نشان داده می‌شود.

3-3 آموزش و ارزیابی با زیرمجموعه ویژگی ها

logloss_xgb_raw_data	0.01194
logloss_xgb_best_features_50	0.01194
logloss_xgb_best_features_100	0.01194
logloss_xgb_best_features_150	0.01194
logloss_xgb_best_features_250	0.01194
logloss_lgb_raw_data	0.0082
logloss_lgb_best_features_50	5.21675
logloss_lgb_best_features_100	4.86224
logloss_lgb_best_features_150	5.21675
logloss_lgb_best_features_250	4.43379

3-3-1 آموزش و ارزیابی با زیرمجموعه ویژگی ها (با

سطوح مختلف (feature importance)

در این مرحله، ابتدا الگوریتم های **xgb** و **lgb** با استفاده از همه ویژگی‌ها (ویژگی های اولیه و ویژگی‌های افزوده شده) آموزش دیدند. سپس با استفاده از **feature importance** هر یک از آنها، مجموعه‌های 50، 100، 150 و 250 تایی از بهترین ویژگی‌ها استخراج شد و در مرحله پایانی، مدل سازی و ارزیابی با استفاده از هر یک از این مجموعه ویژگی ها انجام گرفت. نتایج و ارزیابی های بدست آمده هر یک از این مدل ها مطابق جدول 4 نشان می‌دهد که دقت الگوریتم **xgb** با 50 ویژگی مهم از مجموعه ویژگی‌ها تفاوت فاحشی با دقت آن با استفاده کل مجموعه ویژگی های این مسأله ندارد.

این موضوع در مورد استفاده از 100، 150 و 250 ویژگی مهم هم صدق می‌کند ولی دقت الگوریتم **lgb** با زیرمجموعه‌ای از ویژگی‌ها (زیرمجموعه ای از بهترین ویژگی‌های حاصل از **feature importance** الگوریتم **lgb**) به مراتب بدتر از دقت آن با استفاده کل ویژگی‌هاست. این تفاوت به تفاوت سیاست ساخت درخت این دو الگوریتم برمی‌گردد.

جدول 4. مقایسه نتایج آموزش با کل ویژگی‌ها و زیرمجموعه ویژگی‌ها

حاصل از **feature importance**

3-3-2 آموزش و ارزیابی با زیرمجموعه ویژگی ها (با

سطوح مختلف (permutation Importance)

در روش **permutation Importance** برای هر مجموعه داده به تعداد ویژگی‌های آن مدل سازی و ارزیابی انجام می‌گیرد. بدین صورت که در هر مرحله از این روش، مقادیر یک ویژگی، در هم‌سازی شده و سپس مدل سازی می‌گردد و بر اساس میزان تفاوت خروجی مدل بین دو حالت اولیه و در هم‌سازی شده (حالت اول: بدون در هم سازی هیچ یک از ویژگی‌ها و حالت دوم: با در هم سازی یک ویژگی)، مهم ترین ویژگی‌ها استخراج می‌شود. الگوریتم **permutation Importance**، الگوریتمی بسیار سنگین است و پردازش آن نیاز به منابع پردازشی زیادی دارد. به همین دلیل در این پروژه، به منظور اجرای الگوریتم **permutation Importance** با استفاده از هر یک از الگوریتم‌های **xgb** و **lgb**، تنها 300 مورد از بهترین ویژگی‌های آنها (300 ویژگی برتر حاصل از **feature importance** هر یک از این الگوریتم‌ها) انتخاب شده و بر اساس آنها، **permutation Importance** انجام گرفت. در مرحله آخر

200 ویژگی از بهترین ویژگی‌های حاصل از permutation

Importance انتخاب شده و با استفاده از آنها، آموزش هر یک

از الگوریتم‌های xgb و lgb صورت گرفت. در شکل 11 و 12

مهم‌ترین ویژگی‌های استخراج شده از این روش، به ترتیب

اهمیت نشان داده شده‌اند.

ویژگی section_names_header در هر دو مدل، بیشترین

امتیاز را در مقایسه با سایر ویژگی‌ها کسب نموده است.

Weight	Feature
0.1336 ± 0.0026	section_names_header
0.0004 ± 0.0001	Entropy
0.0004 ± 0.0001	Unknown_Sections_lines_por
0.0003 ± 0.0001	asm_commands_std
0.0002 ± 0.0002	ent_q_diff_diffs_1_median
0.0001 ± 0.0001	sumlmg
0.0001 ± 0.0000	regs_esp
0.0001 ± 0.0001	Offset1
0.0001 ± 0.0000	ent_q_diffs_max
0.0001 ± 0.0001	section_names_rsrc
0.0001 ± 0.0001	VirtualAlloc
0.0001 ± 0.0002	dc_por
0 ± 0.0000	SetStdHandle
0 ± 0.0000	TB_40
0 ± 0.0000	TB_b3
0 ± 0.0000	edx
0 ± 0.0000	lmg2_1
0 ± 0.0000	ImageList_Destroy
0 ± 0.0000	WriteProcessMemory
0 ± 0.0000	ent_q_diff_diffs_median
... 280 more ...	

شکل 12. مهم‌ترین ویژگی‌های حاصل از

(xgb) permutation importance

نتایج ارزیابی‌های صورت گرفته مطابق جدول 5 از مقایسه نتیجه

آموزش با همه ویژگی‌ها و زیرمجموعه ویژگی‌ها حاصل از

permutation importance نشان می‌دهد که استفاده از چنین

زیرمجموعه‌ای از ویژگی‌ها تأثیر چشمگیری بر عملکرد مدل ندارد

و نتایج این دو حالت بسیار نزدیک به هم خواهد بود.

Weight	Feature
0.2160 ± 0.0000	section_names_header
0.0012 ± 0.0000	regs_esp
0.0007 ± 0.0000	Entropy
0.0006 ± 0.0000	Offset1
0.0005 ± 0.0000	VirtualAlloc
0.0004 ± 0.0000	ent_q_diffs_max
0.0004 ± 0.0000	asm_commands_ror
0.0001 ± 0.0000	Unknown_Sections_lines_por
0 ± 0.0000	ent_q_diff_diffs_0_min
0 ± 0.0000	section_names_text
0 ± 0.0000	GetStartupInfoA
0 ± 0.0000	ent_q_diffs_11
0 ± 0.0000	regs_dl
0 ± 0.0000	ent_q_diff_diffs_median
0 ± 0.0000	ent_q_diff_diffs_1_var
0 ± 0.0000	int1
0 ± 0.0000	Plus
0 ± 0.0000	TB_a4
0 ± 0.0000	regs_ebx
0 ± 0.0000	ent_q_diff_diffs_0_var
... 280 more ...	

شکل 11. مهم‌ترین ویژگی‌های حاصل از

(lgb) permutation importance

جدول 5. مقایسه نتیجه آموزش با همه ویژگی‌ها و زیرمجموعه ویژگی‌ها

حاصل از permutation importance

logloss_xgb_allfeat	0.01148
logloss_xgb_best_features_perm	0.01209
logloss_lgb_allfeat	0.00744
logloss_lgb_best_features_perm	0.009968

نشان می‌دهد و در پایان با استفاده از خروجی‌های دو الگوریتم xgb و lgb نتایج جدول 9 حاصل گردیده است. همانطور که در جدول 9 نمایش داده شده است، میزان خطای پیش‌بینی و دقت در روش ارائه شده به مراتب بهتر از روش احمدی و همکاران می‌باشد.

جدول 6. مقایسه نتیجه مدل‌سازی با مدل‌های منفرد و stacking

logloss_xgb_allfeat	0.01148
logloss_xgb_best_features_perm	0.01209
logloss_sclf_xgb_best_features_perm	0.190646
logloss_lgb_allfeat	0.00744
logloss_lgb_best_features_perm	0.009968
logloss_sclf_lgb_best_features_perm	0.190646

جدول 7. نتایج دقت همه مدل‌ها

acc_xgb_raw_data	0.9972401103955841
acc_xgb_allfeat	0.9981600735970562
acc_xgb_best_features_50	0.9972401103955841
acc_xgb_best_features_100	0.9954001839926403
acc_xgb_best_features_150	0.9972401103955841
acc_xgb_best_features_250	0.9981600735970562
acc_xgb_sk_handled_data	0.9972401103955841
acc_lgb_raw_data	0.9990800367985281
acc_lgb_allfeat	0.9981600735970562
acc_lgb_best_features_50	0.0229990800367985
acc_lgb_best_features_100	0.1665133394664213
acc_lgb_best_features_150	0.0229990800367985
acc_lgb_best_features_250	0.2483900643974241
acc_lgb_sk_handled_data	0.9981600735970562
acc_xgb_best_features_perm	0.9972401103955841
acc_lgb_best_features_perm	0.9963201471941122
acc_sclf_xgb_best_features_perm	0.9963201471941122
acc_sclf_lgb_best_features_perm	0.9944802207911684

4. ارزیابی روش پیشنهادی در مقایسه با مدل‌های منفرد و

stacking

در مرحله آخر، روش پیشنهادی با استفاده از مدل‌های نهایی مرحله قبل و GaussianNB، پشته‌ای از مدل‌ها پیاده‌سازی شد. بدین صورت که در لایه اول، مدل‌های lgb و xgb آموزش دیده با 200 ویژگی مهم حاصل از permutation importance و در لایه دو GaussianNB به عنوان کلاس‌کننده نهایی مورد استفاده قرار گرفت. نتایج و ارزیابی‌های صورت گرفته مطابق جدول 6 نشان می‌دهد که عملکرد هر یک از مدل‌های lgb و xgb به تنهایی، بهتر از عملکرد چنین پشته‌ای است. در جدول 7 نتایج دقت همه مدل‌ها ذکر شده است و مطابق این جدول ویژگی‌های اولیه با استفاده از الگوریتم lgb بیشترین دقت را به خود اختصاص داده‌اند. نتایج و ارزیابی‌های صورت گرفته مطابق با جدول 8 میزان خطای پیش‌بینی الگوریتم lgb از همه‌ی مدل‌ها، کمتر بوده در نتیجه بهترین عملکرد و دقت را داشته است. جدول 9. نتایج confusion matrix و accuracyscore را

جدول 8. نتایج همه مدل‌ها

logloss_lgb_best_features_perm: 0.009968037082013147 acc_lgb_best_features_perm: 0.9963201471941122 cm_lgb_best_features_perm: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 1 0 0 72 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 99 2] [0 0 0 0 0 0 0 0 1 86]]	logloss_lgb_best_features_perm: 0.0 acc_lgb_best_features_perm: 0.996320 cm_lgb_best_features_perm: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 1 0 0 72 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 99 2] [0 0 0 0 0 0 0 0 1 86]]
logloss_scif_xgb_best_features_perm: 0.12709761322875074 acc_scif_xgb_best_features_perm: 0.9963201471941122 cm_scif_xgb_best_features_perm: [[161 0 0 0 2 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 1 72 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 101 0] [0 0 0 0 0 0 0 0 1 86]]	logloss_scif_lgb_best_features_perm: 0.9 acc_scif_lgb_best_features_perm: 0.9963201471941122 cm_scif_lgb_best_features_perm: [[161 0 0 0 2 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 1 72 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 99 1] [0 0 0 0 0 0 0 0 1 86]]

جدول 9. نتایج مقایسه کارهای انجام شده

Accuracy	Log loss	محققین
99/77	0/009656	احمدی و همکارانش
99/81	0/007604	مهدوی
99/81	0/00774	کار انجام شده

5. نتیجه گیری

برای مقابله با بدافزارها ابتدا باید راهکارهایی برای شناسایی و تجزیه تحلیل آن‌ها داشته باشیم. یکی از مشکلات در تشخیص بدافزارها، پیچیده تر شدن بدافزارها توسط تکنیک‌های مبهم سازی است که در این صورت مقابله با بدافزارها نیازمند تشخیص و تمایز میان بدافزارها و نرم افزارهای قانونی است. در این مقاله، روشی جهت بهبود افزایش امنیت و تشخیص بدافزار با استفاده از الگوریتم تجمیعی ارائه شد. همچنین با استفاده از روش‌های

logloss_xgb_raw_data	0.01194
logloss_xgb_sk_handled_data	0.01576
logloss_xgb_best_features_50	0.01194
logloss_xgb_best_features_100	0.01194
logloss_xgb_best_features_150	0.01194
logloss_xgb_best_features_250	0.01194
logloss_xgb_allfeat	0.01148
logloss_lgb_allfeat	0.00744
logloss_lgb_raw_data	0.0082
logloss_lgb_sk_handled_data	0.01225
logloss_lgb_best_features_50	5.21675
logloss_lgb_best_features_100	4.86224
logloss_lgb_best_features_150	5.21675
logloss_lgb_best_features_250	4.43379
logloss_lgb_best_features_perm	0.00997
logloss_xgb_best_features_perm	0.01209
logloss_scif_xgb_best_features_perm	0.19065
logloss_scif_lgb_best_features_perm	0.19065

جدول 9. نتایج confusion matrix و accuracyscore

logloss_xgb_allfeat: 0.010852646612050954 acc_xgb_allfeat: 0.9981600735970562 cm_xgb_allfeat: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 0 73 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 100 1] [0 0 0 0 0 0 0 0 1 86]]	logloss_xgb_sk_handled_data: 0.01537082264501708 acc_xgb_sk_handled_data: 0.9972401103955841 cm_xgb_sk_handled_data: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 0 73 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [1 0 0 0 0 0 0 0 99 1] [0 0 0 0 0 0 0 0 1 86]]
logloss_lgb_allfeat: 0.0074439245786956385 acc_lgb_allfeat: 0.9981600735970562 cm_lgb_allfeat: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 0 73 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [0 0 0 0 0 0 0 100 1] [0 0 0 0 0 0 0 0 1 86]]	logloss_lgb_sk_handled_data: 0.010453178169007513 acc_lgb_sk_handled_data: 0.9981600735970562 cm_lgb_sk_handled_data: [[163 0 0 0 0 0 0 0 0 0] [0 266 0 0 0 0 0 0 0 0] [0 0 296 0 0 0 0 0 0 0] [0 0 0 49 0 0 0 0 0 0] [0 0 0 0 2 0 0 0 0 0] [0 0 0 0 0 73 0 0 0 0] [0 0 0 0 0 0 50 0 0 0] [1 0 0 0 0 0 0 0 100 1] [0 0 0 0 0 0 0 0 1 86]]

انتخاب ویژگی جهت دستیابی به پیش‌بینی‌های بهتر و دقت بالاتر از دو الگوریتم **xgboost** و **lgb** استفاده شد. در روش پیشنهادی، یک کلاسه‌کننده طراحی شد که هم در زمینه استخراج ویژگی و هم در زمینه ساز و کار کلاسه‌کننده بسیار ساده و پیچیدگی محاسباتی کمی دارد. چهار ویژگی جدید پیشنهاد شد و نتایج نشان می‌دهد که افزودن این ویژگی‌ها باعث بهبود عملکرد در هر یک از مدل‌های **lgb** و **xgb** و همچنین دقیق‌تر شدن کلاسه‌کننده و بهبود دقت آن نسبت به تحقیقات پیشین شده‌است. استخراج ویژگی‌های ساختاری جدید در مقایسه با ویژگی‌های مبتنی بر محتوا، آسان‌تر بوده و طبقه‌بندی پوشه‌های مخرب و بسته‌بندی شده به آسانی صورت می‌پذیرد و این امر سبب پایین آمدن هزینه محاسباتی و زمانی شده‌است. پیاده‌سازی‌ها و روش‌های گوناگونی برای هوش مصنوعی و یادگیری ماشین به منظور حل مسائل جهان واقعی وجود دارد و یادگیری نظارت شده (**Supervised Learning**) یکی از پرکاربردترین رویکردها است. یادگیری با نظارت یکی از زیرمجموعه‌های یادگیری ماشینی است. در این روش مدل با دریافت اطلاعات برچسب زده شده آموزش می‌بیند و سعی می‌کند الگوی بین داده‌ها و برچسب‌هایشان را به صورت یک تابع یاد گرفته و برچسب داده‌های جدید و دیده نشده را پیش‌بینی کند. از این روش هم در مسائل طبقه‌بندی و هم در مسائل رگرسیون استفاده می‌شود.

در این تحقیق تنها به 9 مدل بدافزار و نحوه ی تفکیک آن‌ها از یکدیگر اکتفا شده است اما می‌توان با توسعه ی این روش، به روشی کارآمد برای شناسایی طیف وسیعی از بدافزارها، بدون توجه به دسته بندی آن‌ها اقدام کرد و همچنین در کارهای آینده ی هم‌راستا با این تحقیق، می‌توان علاوه بر ابزارهای استفاده شده، سایر روش‌های ترکیبی را مورد استفاده قرار داده و برای اطمینان از عملکرد صحیح آنها، علاوه بر مجموعه داده‌های استفاده شده، می‌توان از مجموعه داده‌های مشابه نیز استفاده کرده و روش پیشنهادی را در زمینه‌های جدیدی بکار گرفت. همچنین، استفاده از یک مجموعه‌ی کاهش‌یافته به گونه‌ای که تنها شامل مهمترین ویژگی‌ها باشد می‌تواند باعث کاهش زمان و دقت بالاتر کلاسه‌کننده شود و از بیش برآزش جلوگیری نماید و کار را برای تحلیلگران سهل کند. تمرکز روی مبحث استخراج ویژگی‌های ساختاری جدید از ویژگی‌های مبتنی بر محتوا در عین کم هزینه بودن از نظر محاسباتی نیز سریع می‌باشد و همین امر می‌تواند دلیلی بر گرایش محققان به سمت اینگونه روش‌ها جهت تحقیقات آتی باشد.

مراجع

- [۱۱] S. srivastavai, R.P Mishra, V. Kumar, H. Kumar Shukla, N. Goyal, C. Singh. (2020) , “Android Malware Detection Amid COVID-19,” International Conference on System Modeling & Advancement in Research Trends, Faculty of Engineering & Computing Sciences, Teerthanker Mahaveer University, Moradabad, India
- [۱۲] Alejandro Martin, R.Lara-Cabrera, D.Camacho. (2019) , “Android malware detection through hybrid features fusion and ensemble classifiers: the AndroPyTool framework and the OmniDroid dataset” IEEE.
- [۱۳] Suleiman Yerima , Y. Member, and Sakir Sezer. (2018) , “ DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection,” IEEE .
- [۱۴] Nazrul Hoque , Mihir Singh, K.Dhruba Bhattacharyya. (2018) , “ EFS-MI: an ensemble feature selection method for classification An ensemble feature selection method,” IEEE.
- [۱۵] Qi Fang , X.Yang , C.Ji . (2019) , “A Hybrid Detection Method For Android Malware,” IEEE.
- [۱۶] B.K Sarkar, S. Sana. (2011) , “A Genetic Algorithm-Based Rule Extraction System,” Journal of Applied Soft Computing, vol. ۱۲, pp. ۲۳۸-۲۵۴.
- [۱۷] A. David Donald , G. Murali. (2017) , “ Selective Ensemble of Internet Traffic Classifiers for Improving Malware Detection,” IEEE.
- [۱۸] G. Serazzi, & S.Zanero. (2004) , “Computer Virus Propagation Models”, Computer Science , 2965 , 26-50. Computer Virus Propagation Models. Computer Science , 2965 , 26-50.
- [19] M. Ahmadi, D. Ulyanov, S .Semenov, M .Trofimov, G .Giacinto. (2016) “Novel featur extraction, selection and fusion for effective malware family classification”, Proceeding of the sixth ACM conference on data and application security and privacy;; ACM.
- [19] مهدوی، راحله (۱۳۹۷)، تشخیص بدافزار با استفاده از شبکه های عصبی عمیق و الگوریتم xgboost ، تهران
- [۱] Alazab M, Layton R, Venkataraman S, Watters P. (2010) , “Malware detection based on structural and behavioural features of api calls”, IEEE International, Conference.
- [۲] Aycock.J (2006) , “Computer Viruses and Malware”. Springer, Heidelberg.
- [3] F. Mira. (2021) , “ A Systematic Literature Review on Malware Analysis”. IEEE International IOT, Electronics and Mechatronics Conference (IEMTRONICS).
- [۴] RK .Shahzad, SI .Haidar, N .Lavesson. (2010) , “ Detection of spyware by mining executable files”, 2010 International Conference on Availability, Reliability and Security; IEEE.
- [۵] J. Z. Kolter and M. A. Maloof . (2006) , “Learning to Detect and Classify Malicious Executables in the Wild”, J. Mach. Learn. Res., vol. 7, pp.2721–2744, Dec.
- [۶] M. Gheorghescu. (2005) , “An automated virus classification system ”, in Virus Bulletin Conference, pp. 294–300.
- [۷] D. Marion, D. Phuc Pham, A. Heuser . (2021) , “ Obfuscation Revealed - Using Electromagnetic Emanation to Identify and Classify Malware,” IEEE European Symposium on Security and Privacy (EuroS&P) Vienna, Austria.
- [۸] B. Tahtaci, B. canbay. (2020) , “Android Malware Detection Using Machine Learning,” Innovations in Intelligent Systems and Applications Conference (ASYU), Istanbul, Turkey.
- [۹] S. Choudhary, A. Sharma. (2020) , “Malware Detection & Classification using Machine Learning,” International Conference on Emerging Trends in Communication, Control and Computing (ICONC3) Mody University of Science and Technology, Lakshmanagarh.
- [۱۰] H. S. Galal, Y. B. Mahdy, M. A. Atiea. (2016) , “ Behavior-based features model for Malware detection,” JCVHT, vol. 12, pp. 59-67.