شناسایی بد افزارهای اندرویدی با ترکیبی از الگوریتم های یادگیری ماشین

سید محمد رضا پورهاشمی
دانشجو کارشناسی ارشد دانشگاه آزاد اسلامی شیراز
کاظم براتی مهر
عضو هیئت علمی دانشگاه آزاد اسلامی شیراز

چکیده:
در عصر دیجیتال کنونی، با افزایش استفاده از دستگاه‌های هوشمند و محبوبیت سیستم‌عامل اندروید، تهدیدات سایبری به طور قابل توجهی افزایش یافته است. بدافزارها تلاش می‌کنند با سوءاستفاده از این پلتفرم، به اطلاعات شخصی کاربران دسترسی پیدا کنند. شناسایی و تشخیص این تهدیدات به یک چالش جدی تبدیل شده و نیاز به روش‌های نوین برای حفاظت از امنیت کاربران را ضروری می‌سازد.تحقیق حاضر به بررسی شناسایی و تشخیص بدافزارهای اندرویدی پرداخته و از تکنیک‌های پیشرفته‌ای همچون ترکیب الگوریتم‌های یادگیری ماشین با استفاده از روش استکینگ بهره می‌گیرد. این الگوریتم‌ها با تحلیل داده‌های بزرگ و الگوهای پیچیده می‌توانند کمک شایانی به شناسایی این تهدیدات نمایند. هدف اصلی این تحقیق توسعه الگوریتم‌های هوش مصنوعی و بهبود عملکرد روش های موجود با تمرکز بر استفاده همزمان از ویژگی‌های پویا و ایستای برنامه‌ها برای بهبود دقت شناسایی و کاهش خطای تشخیص است .از طرف دیگر، این تحقیق به بررسی تأثیر بدافزارها بر عملکرد کل سیستم و تجربه کاربری می‌پردازد. شناسایی و تحلیل رفتار بدافزارها و الگوهای مربوط به آن‌ها از جمله اقداماتی است که این مطالعه انجام می‌دهد. نتایج حاصل از این تحقیق می‌تواند به شناسایی مؤثرتر تهدیدات سایبری و بهبود روش‌های موجود کمک کرده و ابزارهای لازم برای حفاظت از اطلاعات شخصی کاربران را ارائه دهد. در نهایت، امید است که این پژوهش به توسعه تکنیک‌های پیشرفته و کارآمد برای مقابله با تهدیدات بدافزاری در پلتفرم‌های اندرویدی منجر شود.

واژگان کلیدی: بد افزار، یادگیری ماشین، اندروید، استکینگ

مقدمه (فونت B Nazanin - اندازه ۱۲ - پررنگ)

با پیشرفت فناوری اطلاعات و ارتباطات در دهه‌های اخیر، استفاده از دستگاه‌های هوشمند به بخشی جدایی‌ناپذیر از زندگی روزمره انسان‌ها تبدیل شده است. سیستم‌عامل اندروید به عنوان یکی از پلتفرم‌های محبوب، به علت قابلیت‌های گسترده و دسترسی آسان به برنامه‌های متنوع، به محبوبیت بیشتری دست یافته است. اما در کنار این مزایا، محبوبیت اندروید باعث افزایش تهدیدات سایبری و تولید بدافزارها شده است. بدافزارها با اتخاذ روش‌های مختلف، سعی در نفوذ به دستگاه‌های کاربران دارند و این امر، امنیت اطلاعات شخصی و مالی آنان را تحت تأثیر قرار می‌دهد. شناسایی و تشخیص بدافزارها به یک چالش جدی تبدیل شده است، زیرا این نرم‌افزارهای مخرب نه تنها می‌توانند به داده‌های حساس دسترسی پیدا کنند، بلکه به عملکرد کلی دستگاه و تجربه کاربری نیز آسیب می‌زنند. در این راستا، نیاز به روش‌های نوین و کارآمد برای شناسایی این تهدیدات احساس می‌شود. از جمله این روش‌ها، الگوریتم‌های یادگیری ماشین هستند که به دلیل توانایی‌های پیشرفته خود در پردازش و تحلیل داده‌های بزرگ، به عنوان ابزاری مؤثر برای شناسایی بدافزارها مطرح شده‌اند.
این مقاله با هدف بررسی روش‌های نوین شناسایی و تشخیص بدافزارهای اندرویدی، به تحلیل تکنیک‌های پیشرفته یادگیری ماشین و به‌خصوص روش استکینگ می‌پردازد. با تمرکز بر ویژگی‌های پویا و ایستا در برنامه‌ها و بررسی تأثیرات بدافزارها بر عملکرد سیستم، این تحقیق به دنبال دستیابی به نتایج قابل توجهی در بهبود دقت شناسایی و کاهش خطای تشخیص است. در نهایت، امید است که این پژوهش به توسعه ابزارها و تکنیک‌های کارآمدی برای مقابله با تهدیدات بدافزاری حتی در برابر روش‌های نوین حمله منجر شود و امنیت اطلاعات کاربران را در پلتفرم‌های اندرویدی تقویت کند.
محققان در تشخیص بد افزار از شیوه ها و ویژگی های مختلفی استفاده نموده اند. برخی از این شیوه ها تحلیل ایستا، تحلیل پویا و به صورت ترکیبی می باشند. در میان تحقیقات مختلف با روش های گوناگون استفاده از ویژگی های مختلف نیز بسیار حائز اهمیت بوده و مورد نظر محقیقن می باشد. همچون استفاده از میزان هر کدام از نرم افزارهای خوش خیم و بد افزار ها از باتری، مموری، بهره گیری از ویژگی هایی همچون api call، opcode، permitions و ... . نمونه های این مقالات موارد زیر می باشد:
(Sahs and Khan ۲۰۱۲) یک رویکرد یادگیری ماشین برای شناسایی بدافزار در اندروید در یک محیط ایستا پیشنهاد کردند. این مدل یک SVM تک‌کلاسی بود که بر روی نمونه‌های بی‌خطر آموزش دیده بود. مدل از مجوزهای داخلی و کد بایت خام که به صورت گراف جریان کنترل ارائه شده بود، به عنوان ویژگی‌ها برای تحلیل استفاده کرد. این کار از یک مجموعه داده محدود شامل ۲۰۸۱ فایل بی‌خطر و ۹۱ فایل مخرب استفاده می‌کند. گره‌های موجود در گراف‌های جریان کنترل بر اساس آخرین دستورالعمل برچسب‌گذاری شده و کوچک هستند. چنین برچسب‌هایی قادر به ضبط اطلاعات مهم موجود در کد اصلی نیستند. تشخیص بدافزارهای اندروید با استفاده از تحلیل ایستا ، که در آن برنامه برای بررسی وجود هرگونه کد مخرب تجزیه و تحلیل می‌شود، یک رویکرد محبوب است. چندین راه‌حل با استفاده از رویکرد ایستا توسعه یافته‌اند که از ویژگی‌هایی مانند مجوزها، فراخوانی‌های API، دستورات و Intents استفاده می‌کنند. (Cen et al. ۲۰۱۵) و (Fan et al. ۲۰۱۷) و (Yerima and Sezer ۲۰۱۹) نمونه‌هایی از راه‌حل‌های تشخیص مبتنی بر تحلیل ایستا هستند.

در روش استفاده شده در مقاله ۱ (Elish et al. ۲۰۱۵) ، طبقه‌بندی دقیق برای شناسایی برنامه‌های مخرب اندروید ارائه شده است که از ویژگی‌های جریان داده‌ای استفاده می‌کند تا نشان دهد چگونه ورودی‌های کاربر منجر به فراخوانی‌های حساس API می‌شوند. ارزیابی این روش با ۱۴۳۳ برنامه مخرب و ۲۶۸۴ برنامه رایگان پرطرفدار، نرخ منفی کاذب ۲.۱٪ و نرخ مثبت کاذب ۲.۰٪ را نشان می‌دهد. این سیستم همچنین قادر به شناسایی برنامه‌های مخرب جدید در بازار گوگل پلی است که ابزارهای اسکن ویروس نمی‌توانند آن‌ها را تشخیص دهند. در این روش الگوریتم پیشنهاد شده شامل استخراج نقاط ورودی کاربر، آنالیز گراف وابستگی داده، و بررسی مسیرهای بین ورودی‌های کاربر و نقاط فراخوانی برای محاسبه متریک‌های ویژه برای طبقه‌بندی برنامه‌ها است. در روش های پویا میتوان به مقاله زیر اشاره نمود:

در روش (Song et al. ۲۰۲۳) با ادغام چندین مدل یادگیری عمیق که از ویژگی‌های دینامیک ناهمگن نمونه‌های بدافزار استفاده می‌کند به‌طور خاص، سه مدل یادگیری عمیق ناهمگن پیاده‌سازی شده که ویژگی‌های مختلفی را از سه نمایش متفاوت یاد می‌گیرند: ۱) توالی نام‌های API، ۲) گراف منابع API و ۳) گراف فراخوانی API. هر یک از این نمایش‌ها از ردیابی زمان اجرای بدافزار ساخته شده است. همچنین، روش‌هایی مانند مکانیزم توجه و شبکه‌های عصبی گراف در مدل‌های پایه به‌کار گرفته شده‌اند تا با ویژگی‌های فراخوانی‌های API سازگار شوند. در نهایت، یک الگوریتم تجمیع برای ادغام خروجی‌های سه مدل پایه استفاده شده است. مدل هترونت[۱] شامل ادغام چندین مدل یادگیری عمیق و استفاده از ویژگی‌های پویا ناهمگن، می باشد.

در مقاله (Gupta et al. ۲۰۲۵)، یک مدل شناسایی بدافزار اندروید را معرفی می‌کنند که از ترکیب شبکه‌های حافظه کوتاه‌مدت بلند (LSTM) و واحدهای تکراری دروازه‌دار (GRU) الهام گرفته و با استفاده از الگوریتم بهینه‌سازی کرم خاکی (EOA) بهینه‌سازی شده است.

در این تحقیق، از مدل جنگل تصادفی برای انتخاب ویژگی‌ها استفاده شده است. مدل پیشنهادی با دقت ۹۹٪ و کمترین مقادیر ضرر، عملکرد بهتری نسبت به مدل‌های مرسوم مانند GRU، LSTM، RNN، رگرسیون لجستیک و SVM از خود نشان می‌دهد.

باتوجه به بررسی های مقالات مختلف روش های سنتی و روش های ایستا به تنهایی در شناسایی بد افزارهای اندرویدی موثر نبوده و استفاده از روش های پویا در این مهم تاثیر بالاتری دارد، اما روش های موجود گاها در استفاده از روش های ترکیبی نیز با مشکلاتی همراه می باشد همچون دقت پایین در شناسایی بد افزارها، روش پیش رو تلاش می کند که با ترکیب چند الگوریتم مختلف یاگیری ماشین با استفاده از روش استکینگ[۲] روشی جدید را ارائه دهد. امید است که نتیجه بهتری را به نسبت روش های موجود در حوزه شناسایی ترکیبی ارائه نماید. اهداف این مقاله با فرض استفاده از الگوریتم‌های یادگیری ترکیبی به نسبت روش‌های یادگیری غیر ترکیبی، دقت بالاتری در شناسایی و طبقه‌بندی بدافزارهای اندروید دارد، شامل طراحی و اجرای الگوریتم‌های هوش مصنوعی و یادگیری ماشین (شامل یادگیری نظارت‌شده و بدون نظارت) برای شناسایی و طبقه‌بندی برنامه‌ها بر اساس ویژگی‌های پویا و ایستا آن‌ها و همچنین استفاده از تکنیک‌های متنوع (همچون روش های بهینه‌سازی) و ترکیبی روش‌های مدرن یادگیری ماشین به امید بهینه‌سازی روش‌های شناسایی، افزایش دقت و کاهش سطح خطا می باشد.

روش تحقیق

در این مقاله، قصد داریم به بررسی و ترکیب الگوریتم‌های XGBoost، SVM، RF، DT و GB با استفاده از روش Stacking بپردازیم. Stacking یک روش ensemble learning است که با ترکیب چندین مدل یادگیری ماشین، سعی در بهبود عملکرد و دقت پیش‌بینی دارد. در این روش، ابتدا مدل‌های پایه (Base Models) روی داده‌های آموزشی آموزش داده می‌شوند و سپس یک مدل Meta-Learner روی خروجی مدل‌های پایه آموزش می‌بینند تا پیش‌بینی نهایی را انجام دهد.

یادگیری ماشین به عنوان یکی از شاخه‌های مهم هوش مصنوعی، در سال‌های اخیر پیشرفت‌های چشمگیری داشته است. الگوریتم‌های مختلفی برای حل مسائل یادگیری ماشین ارائه شده‌اند که هر کدام دارای مزایا و معایب خاص خود هستند. در این میان، الگوریتم‌های ensemble learning به دلیل توانایی بالا در ترکیب مدل‌های مختلف و بهبود عملکرد، توجه بسیاری را به خود جلب کرده‌اند. Stacking یکی از روش‌های قدرتمند ensemble learning است که در این مقاله به بررسی آن می‌پردازیم.

در این مقاله، از سه الگوریتم یادگیری ماشین به عنوان مدل پایه استفاده می‌کنیم:

- **SVM (Support Vector Machine)**: یک الگوریتم دسته‌بندی که با یافتن بهترین ابرصفحه (hyperplane) در فضای ویژگی، داده‌ها را به بهترین شکل ممکن جدا می‌کند.
- **DT (Decision Tree)**: یک الگوریتم ساده و قابل تفسیر که با تقسیم داده‌ها به صورت سلسله مراتبی، تصمیم‌گیری می‌کند.

۱ HeteroNet
۲ Stacking

- **۳. روش Stacking**

روش Stacking به صورت کلی شامل مراحل زیر است:

۱. **آموزش مدل‌های پایه:** ابتدا هر یک از مدل‌های پایه روی داده‌های آموزشی آموزش داده می‌شوند.

۲. **ایجاد داده‌های جدید:** خروجی مدل‌های پایه روی داده‌های آموزشی و داده‌های آزمون به عنوان ویژگی‌های جدید برای مدل Meta-Learner در نظر گرفته می‌شوند.

۳. **آموزش مدل Meta-Learner:** مدل Meta-Learner روی ویژگی‌های جدید (خروجی مدل‌های پایه) آموزش داده می‌شود تا پیش‌بینی نهایی را انجام دهد.

مجموعه داده:

در این مقاله از یک مجموعه داده از kaggel استفاده شده است که شامل ۳۴ ویژگی مرتبط با دسته‌بندی برنامه‌های اندروید است. در این مجموعه داده تعداد ۵۰۰۰۰ بد افزار و ۵۰۰۰۰ نرم افزار خوش خیم قرار داده شده است. در نمودار شماره ۱ میتوان تعادل این مجموعه داده را دید. تعادل اهمیت زیادی دارد زیرا اطمینان می‌دهد که طبقه‌بند قادر است بین دو دسته بدون تمایل به سمت هیچ یک از کلاس‌ها تمایز قائل شود. علاوه بر این، برای ارزیابی کارایی مدل پیشنهادی، داده‌ها را به نسبت ۸۰ به ۲۰ به مجموعه‌های آموزشی و آزمایشی تقسیم شده است.

پیاده‌سازی

برای پیاده‌سازی روش Stacking، می‌توان از کتابخانه‌های مختلفی مانند scikit-learn در پایتون استفاده کرد. در این پیاده‌سازی، ابتدا مدل‌های پایه (SVM، Decision tree) را روی داده‌های آموزشی آموزش می‌دهیم. سپس خروجی این مدل‌ها را به عنوان ویژگی‌های جدید در مدل stacking بررسی می گردد.

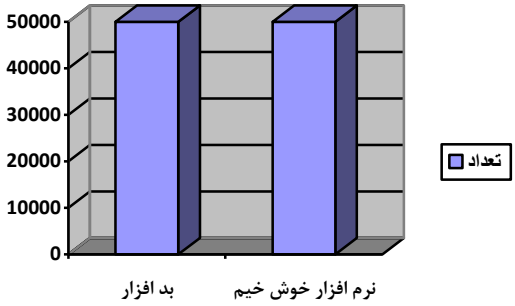
یافته ها

در این مطالعه، ما به شناسایی بدافزارهای اندرویدی پرداخته‌ایم. با استفاده از روش ترکیب(Stacking) ، دو مدل قدرتمند یادگیری ماشین، یعنی **(SVM)** و درخت تصمیم **(Decision tree)** را به کار گرفتیم.

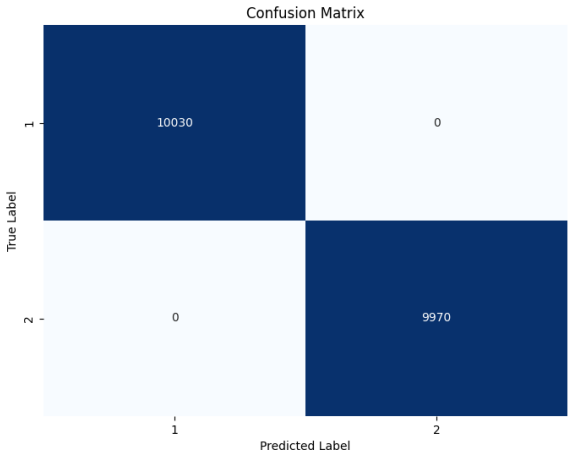
پس از آموزش و ارزیابی مدل‌های ترکیبی، نتایج به‌دست‌آمده نشان داد که مدل ترکیبی قادر به شناسایی دقیق بدافزارها با **دقت ۱۰۰ درصد** بوده است (نمودار ۲و۳). این دقت بالای مدل به ما امکان می‌دهد تا اطمینان حاصل کنیم که روش‌های شناختی ما توانایی تشخیص درست بدافزارهای مختلف را دارند.

این نتایج نشان‌دهنده کارایی و قابلیت‌های بالای روش‌های ترکیبی در شناسایی تهدیدات سایبری و بدافزارها می‌باشد و اهمیت استفاده از تکنیک‌های پیشرفته یادگیری ماشین در حوزه امنیت سایبری را روشن می‌سازد.

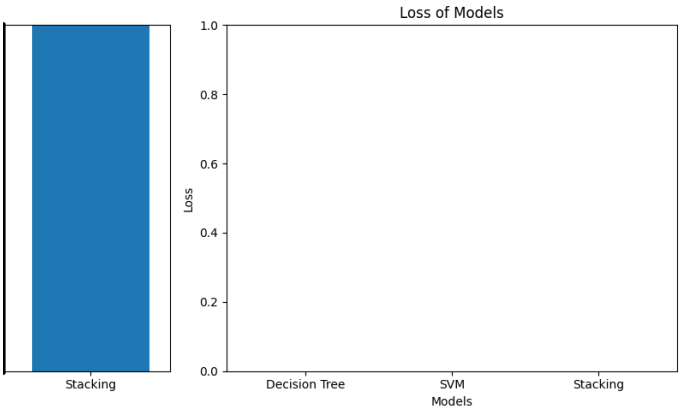
جداول، شکل ها و نمودارها



نمودار ۱. میزان تعادل در مجموعه داده



نمودار۲. ماتریس سردرگمی



نمودار۳. درصد دقت روش استکینگ

	support	f۱-score	recall	precision
۰	۹۹۷۰	۱,۰۰	۱,۰۰	۱,۰۰
۱	۱۰۰۳۰	۱,۰۰	۱,۰۰	۱,۰۰
accuracy	۲۰۰۰۰	۱,۰۰		
macro avg	۲۰۰۰۰	۱,۰۰	۱,۰۰	۱,۰۰
weighted avg	۲۰۰۰۰	۱,۰۰	۱,۰۰	۱,۰۰

جدول ۱. ارزیابی مدل ترکیبی

فرمول‌ها و روابط ریاضی

برای محاسبه هر کدام از قسمت های ارزیابی مدل فرمول های زیر مورد استفاده قرار است:

۱- Precision = True Positives / (True Positives + False Positives)
۲- Recall = True Positives / (True Positives + False Negatives)
۳- F۱ Score = ۲ * Precision * Recall / (Precision + Recall)
٤- Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)
٥- Macro Avg = (Precision_Class۰ + Precision_Class۱) / ۲
٦- Weighted Avg = (Precision_Class۰ * Support_Class۰ + Precision_Class۱ * Support_Class۱) / (Support_Class۰ + Support_Class۱)

بحث و نتیجه‌گیری

ارزیابی عملکرد:برای ارزیابی عملکرد مدل Stacking، می‌توان از معیارهای مختلفی مانند دقت،Recall، F۱-score و AUC استفاده کرد. این معیارها با مقایسه پیش‌بینی‌های مدل با داده‌های واقعی، میزان دقت و صحت مدل را ارزیابی می‌کنند.

:در این مقاله، به بررسی و ترکیب الگوریتم‌های SVM و RF با استفاده از روش Stacking پرداختیم. Stacking به عنوان یک روش قدرتمند ensemble learning، می‌تواند با ترکیب مدل‌های مختلف و بهبود عملکرد، در مسائل یادگیری ماشین مختلف مورد استفاده قرار گیرد. در این مجموعه داده با استفاده از ترکیب چندالگوریتم، با دقت ۱۰۰ درصد در کلاس بندی بد افزارها و نرم افزارها موثر باشد. این روش به نسبت روش ارائه شده در مقاله (Gupta et al. ۲۰۲٥) یک درصد بهبود یافته و دقت بالاتری را کسب نموده است.

منابع

١. Cen, Lei, Christoher S. Gates, Luo Si, and Ninghui Li. ٢٠١٥. “A Probabilistic Discriminative Model for Android Malware Detection with Decompiled Source Code.” *IEEE Transactions on Dependable and Secure Computing* ١٢ (٤): ٤٠٠–٤١٢. https://doi.org/١٠,١١٠٩/TDSC.٢٠١٤,٢٣٥٥٨٣٩.

٢. Elish, Karim O., Xiaokui Shu, Danfeng Yao, Barbara G. Ryder, and Xuxian Jiang. ٢٠١٥. “Profiling User-Trigger Dependence for Android Malware Detection.” *Computers and Security* ٤٩ (٥٤٠): ٢٥٥–٧٣. https://doi.org/١٠,١٠١٦/j.cose.٢٠١٤,١١,٠٠١.

٣. Fan, Ming, Jun Liu, Wei Wang, Haifei Li, Zhenzhou Tian, and Ting Liu. ٢٠١٧. “DAPASA: Detecting Android Piggybacked Apps Through Sensitive Subgraph Analysis.” *IEEE Transactions on Information Forensics and Security* ١٢ (٨): ١٧٧٢–٨٥. https://doi.org/١٠,١١٠٩/TIFS.٢٠١٧,٢٦٨٧٨٨٠.

٤. Gupta, Brij B., Akshat Gaurav, Varsha Arya, Shavi Bansal, Razaz Waheeb Attar, Ahmed Alhomoud, and Konstantinos Psannis. ٢٠٢٥. “Earthworm Optimization Algorithm Based Cascade LSTM-GRU Model for Android Malware Detection.” *Cyber Security and Applications* ٣ (December):١٠٠٠٨٣. https://doi.org/١٠,١٠١٦/j.csa.٢٠٢٤,١٠٠٠٨٣.

٥. Sahs, Justin, and Latifur Khan. ٢٠١٢. “A Machine Learning Approach to Android Malware Detection.” *Proceedings - 2012 European Intelligence and Security Informatics Conference, EISIC 2012*, ١٤١–٤٧. https://doi.org/١٠,١١٠٩/EISIC.٢٠١٢,٣٤.

٦. Song, Runhan, Lun Li, Lei Cui, Qiqi Liu, and Jin Gao. ٢٠٢٣. “Binary Malware Detection via Heterogeneous Information Deep Ensemble Learning.” *Proceedings of the International Conference on Parallel and Distributed Systems - ICPADS*, ١١٤٧–٥٦. https://doi.org/١٠,١١٠٩/ICPADS٦٠٤٥٣,٢٠٢٣,٠٠١٦٨.

٧. Yerima, Suleiman Y., and Sakir Sezer. ٢٠١٩. “DroidFusion: A Novel Multilevel Classifier Fusion Approach for Android Malware Detection.” *IEEE Transactions on Cybernetics*. https://doi.org/١٠,١١٠٩/tcyb.٢٠١٧,٢٧٧٧٩٦٠.

٨. **N. Saravana, Malware detection, ٢٠١٨, (https://www.kaggle.com/datasets/nsaravana/malware-detection). Accessed: ٢٠٢٤-٠١-٣٠.**