ارزیابی ویژگی‌های جدید برای شناسایی روبات‌های مخرب وب بر پایه‌ی روش‌های یادگیری ماشین

ناصر یوسفی[۱]، مجید وفایی جهان[۲] و جواد حاجیان نژاد
[۱] دانشجوی کارشناسی ارشد فناوری اطلاعات گرایش امنیت اطلاعات، دانشگاه بین المللی امام رضا (ع)، مشهد، yousofi@ymail.com
[۲] استاد یار گروه مهندسی کامپیوتر، دانشگاه آزاد اسلامی واحد مشهد، VafaeiJahan@mshdiau.ac.ir

چکیده - امروزه انواع مختلفی از روبات‌های وب هم‌زمان با کاربران انسانی به اطلاعات سایت‌ها دسترسی پیدا می‌کنند. دسته بندی کاربران و جداسازی انسان‌ها از روبات‌ها مسئله‌ای مهم برای سرورهای ارائه دهنده‌ی خدمات محسوب می‌شود چرا که گاهی خدمت‌دهی به روبات‌ها مانع از ارائه‌ی خدمات کافی برای کاربر انسانی می‌شود. از طرفی روبات‌ها خودشان به دسته‌های خوش رفتار و مخرب تقسیم‌بندی می‌شوند و نباید از دسترسی همه‌ی آن‌ها جلوگیری به عمل آورد. از جمله روبات‌های مخربی که برای حمله به سایت‌ها استفاده می‌شوند روبات‌های حمله‌ی منع سرویس هستند که همه ساله خسارات فراوانی از خود برجای می‌گذارند. ما در این تحقیق با استفاده از تکنیک‌های یادگیری ماشین بر روی فایل ثبت وقایع سایت، به مدلی مناسب برای شناسایی روبات‌های مخرب وب با صحت بالای ٪۹۶ دست پیدا کردیم. همچنین ما با نگاهی ویژه به روبات‌های مخربی که رفتار انسان‌ها را تقلید می‌کنند، روش جدیدی برای برچسب گذاری مجموعه‌ی داده پیشنهاد دادیم و با ارائه‌ی ۴ ویژگی جدید نشان دادیم که روش‌های یادگیری ماشین با استفاده از ویژگی‌های ارائه شده در پژوهش‌های قبلی و افزودن این ویژگی‌ها عملکرد بهتری از خود نشان می‌دهند.

کلید واژه- روبات وب، مخرب، دسته بندی، حمله‌ی DDoS، Bot.

۱- مقدمه

امروزه اینترنت نقش گسترده‌ای در زندگی مردم دنیا بازی می‌کند؛ خدمت دهنده‌های اینترنتی به صورت گسترده و افزاینده‌ای جایگزین بسیاری از خدمات مورد نیاز مردم از جمله در زمینه‌های بانکداری، آموزش، کسب و کار و... شده‌اند. تا آنجایی که شرکت‌های اینترنتی جایگاه ثروتمندترین و بزرگ‌ترین شرکت‌های دنیا را از آن خود کردند. با این وجود، آسیب‌پذیری‌های فطری معماری اینترنت، فرصت‌های مفیدی برای انواع مختلف حمله‌های امنیتی به کاربردهای مبتنی بر اینترنت فراهم می‌کنند. به‌طوری‌که تعداد و قدرت ابزارهای مخرب وب هرچه بیشتر در حال افزایش است و در عین حال استفاده از این ابزارها برای حمله به وب سایت‌ها همگانی‌تر و ساده‌تر می‌شود.

در میان حملاتی که به خدمت‌گزارها و خدمت گیرنده‌های وب صورت می‌گیرد، حجم بسیار زیادی به صورت خودکار و با استفاده از ابزاری با نام روبات‌های اینترنتی یا روبات‌های وب انجام می‌گیرد. روبات‌های وب برنامه‌های خودکاری هستند که برای کاربردهای مختلفی طراحی می‌شوند و وظایف متناسبی را در اینترنت دنبال می‌کنند. روبات‌های اینترنتی استفاده‌های بسیار زیاد و متنوعی دارند که از مهم‌ترین آن‌ها روبات‌های موتورهای جست‌وجوی وب است. وظیفه‌ی اصلی آن‌ها ایندکس کردن صفحه‌های وب و جمع آوری اطلاعات با خزیدن در تارنمای وب برای بسیاری از سایت‌ها بخصوص موتورهای جستجوگری وب است. در این بین روبات‌های مخربی هم وجود دارند که یا به قصد خرابکارانه و یا به خاطر طراحی ناشیانه توسط طراح روبات، باعث ایجاد خسارات بسیار سنگینی به شبکه و سرویس دهنده‌ی وب می‌شوند. آن‌ها به سرعت در اینترنت می‌خزند، پخش می‌شوند، بدون اجازه تبلیغ می‌کنند، اطلاعات را می‌دزدند و برای صاحبانشان ارسال می‌کنند، منابع را هدر می‌دهند و مانع دسترسی کاربران خوش‌رفتار وب می‌شوند. از این‌رو نیاز به راهکارهایی برای تفکیک روبات‌های وب مخرب از روبات‌های خوش‌رفتار و کاربران انسانی دیده می‌شود.

۲- کارهای مرتبط

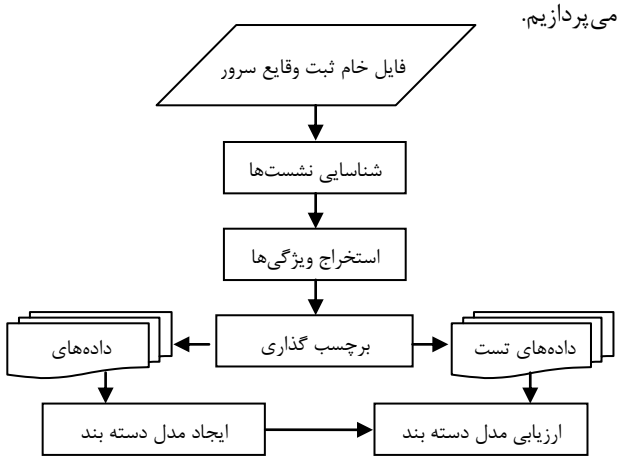
تا قبل از سال ۲۰۰۲ از چند تکنیک ساده برای تشخیص روبات‌های وب و جلوگیری از آن‌ها استفاده می‌شد. یکی از این روش‌ها استفاده از فایلی با نام robot.txt بود که توسط Koster [۱, ۲] در سال ۱۹۹۴ ارائه شد و در سال ۱۹۹۶ توسط Kollar [۳] مورد استناد و بازبینی، قرار گرفت. این تکنیک، که به صورت استانداردی با عنوان «جلوگیری از روبات» مطرح شد، به مدیر سایت توانایی مشخص کردن قسمت‌هایی از وب سایت، که نباید روبات‌ها در آنجا حضور پیدا کنند یا حضور گسترده داشته باشند را ارائه می‌کرد. در رهنمودهایی که در

سال‌های ۱۹۹۳ تا ۱۹۹۵ توسط Koster [۲] و Eichmann [۴] ارائه شده بود، طراح روبات وب اخلاقا می‌بایست نوع روبات را به نوعی در فیلد User Agent بیاورد تا از کاربران انسانی تمایز یابد. بررسی آدرس IP نیز یکی از تکنیک‌های اولیه برای جلوگیری و تشخیص روبات‌های وب بود. مشکلات این روش این است که لیست مورد نظر بسیار عظیم خواهد بود و همچنان رو به افزوده شدن است. مطالعات انجام گرفته در زمینه‌ی دسته بندی روبات‌های وب عموماً در ۴ دسته‌ی عمده قرار می‌گیرند [۵]: تحلیل نحوی فایل ثبت وقایع، بررسی الگوی ترافیک کاربران، تکنیک‌های یادگیری تحلیلی و سیستم تست تورینگ.

یکی از اولین مطالعاتی که بر روی کشف روبات‌های وب با استفاده از روش‌های داده کاوی انجام گرفت، توسط TAN و همکارانش [۶] در سال ۲۰۰۲ بود که نشان داد، روبات‌های وب الگوی ناوبری مشخص و مشابهی دارند، بنابراین می‌توان یک مدل دقیق برای شناسایی حضور روبات‌های وب بر اساس الگوی ناوبریشان، ایجاد کرد. آن‌ها ۲۵ ویژگی مختلف از هر نشست را مورد بررسی قرار دادند و از بین این ۲۵ ویژگی، ۳ ویژگی را به عنوان بیشترین متمایز کننده‌ی روبات‌ها از انسان‌ها نام می‌برد. در [۷, ۸] با آنالیز فایل خام دسترسی سایت‌های ۵ دانشگاه از ۳ کشور نشان دادند، که مجموع درخواست‌های روبات‌های مشهور حدود ۱۰ درصد از کل درخواست‌های HTTP را شامل می‌شوند. همچنین نشان دادند که الگوی بازدید روبات‌ها تفاوت قابل ملاحظه‌ای با الگوی بازدید انسان‌ها دارد. در [۹] با بررسی توزیع ساعتی درخواست‌های دریافتی از ۵ روبات سایت‌های معروف جستوجو در ۱۸۰ روز نشان دادند که تنها ترافیک مربوط به این روبات‌ها ۱,۶ درصد از کل ترافیک HTTP را شامل می‌شد. به منظور دسته‌بندی نشست‌های فایل ثبت وقایع، از الگوریتم‌های داده کاوی دیگری نیز استفاده شده است. به عنوان مثال در [۱۰] و [۱۱] نویسنده‌ها به ترتیب از روش شبکه‌های عصبی و بیزین بهره برده‌اند. در زمینه‌ی تشخیص روبات‌های وب مخرب نیز مطالعاتی در [۱۲, ۱۳] صورت گرفته است. یکی از پژوهش‌هایی که به طور خاص به تشخیص روبات‌های مخرب پرداخته است، مقاله آقای Stevanovic [۵] است که از ۷ روش دسته بندی با استفاده از ۹ ویژگی که دوتای آن‌ها ویژگی‌های جدید بودند، استفاده کرده است. همچنین افراد دیگری از روش‌های غیر مرتبط با داده کاوی بهره برده‌اند از جمله [۱۴] که از روش مدل زنجیره‌ی مخفی مارکوف استفاده کرده است. در [۱۵] نیز از تست تورینگ و در [۱۶] از روش تحلیل ترافیک بهره برده شده است.

هدف نهایی ما ارائه‌ی مدلی برای نگاشت هر نشست به کلاس‌های از پیش تعیین شده‌ی "مخرب" و "خوش‌رفتار" است. از این‌رو نیاز به شناسایی نشست‌ها، استخراج ویژگی مربوط به هر نشست و برچسب گذاری نشست‌ها داریم. مراحل کلی این پژوهش در شکل (۱) مشاهده می‌گردد.

ما در بخش ۲ مقاله روش شناسایی نشست‌ها را شرح می‌دهیم، در بخش ۳ استخراج ویژگی‌ها را با توضیح چهار ویژگی جدید بیان می‌کنیم، برچسب گذاری نشست‌ها را در بخش ۴ و مشخصات مجموعه داده‌های استفاده شده را در بخش ۵ بررسی می‌کنیم و در نهایت در بخش ۶ به نحوه‌ی پیاده سازی و ارزیابی مدل دسته بند ایجاد شده می‌پردازیم.



شکل (۱) : مراحل تحقیق

۳- شناسایی نشست‌ها

ما درخواست‌های کاربران را بر اساس آدرسIP و فیلد User Agent و با رویکرد توجه به زمان در قالب نشست‌ها دسته بندی می‌کنیم. بطوری‌که اگر فاصله‌ی زمانی دو درخواست با هم کمتر از نیم ساعت باشد این دو درخواست HTTP در یک نشست قرار می‌گیرند.

۴- استخراج ویژگی

برای کمک به الگوریتم یادگیری، به ویژگی‌هایی نیاز داریم که به شکل بهتری، جداکننده‌ی کاربران مخرب از کاربران خوش‌رفتار باشند. برای این منظور از اطلاعات مربوط به نشست استفاده کرده و ویژگی‌های مورد نظر را برای هر نشست می‌یابیم.

ما در اینجا از ۱۱ ویژگی عنوان شده در مراجع [۵, ۶, ۱۱, ۱۷-۱۹] استفاده کرده‌ایم که در زیر لیست آن‌ها را مشاهده می‌کنید.

(۱) تعداد درخواست‌های نشست (۲) مدت زمان نشست (۳) نسبت درخواست صفحه‌های HTML به تصویر (۴) درصد درخواست فایل‌های PDF و PS (۵) درصد درخواست‌هایی که با خطاهای 4XX مواجه می‌شوند (۶) درصد درخواست‌های HTTP request با نوع Head (۷) درصد درخواست‌های از نوع Unassigned Referer (۸) درخواست فایل Robots.txt (۹) انحراف از معیار عمق صفحه های درخواستی (۱۰) درصد بالای درخواست‌های پی‌درپیHTTP (۱۱) تعداد بایت نشست.

در این پژوهش ما از ۴ ویژگی جدید برای تفکیک روبات‌های مخرب از کاربران خوش رفتار استفاده کردیم که این ویژگی‌ها به شرح زیر می‌باشند.

۱.تعداد درخواست به زمان نشست: همان‌طور که بیان کردیم تاکنون از تعداد درخواست‌های نشست و زمان نشست به عنوان ویژگی‌های مفید برای تشخیص روبات‌های وب بکار برده می‌شدند. اما نسبت تعداد درخواست‌های هر نشست به زمان آن نیز می‌تواند اطلاعات خوبی در مورد رفتار کاربر وب به ما بدهد، چرا که این نسبت برای کاربران انسانی گاهی بسیار کمتر از روبات‌های وب مخرب است.

۲.میانگین اندازه‌ی هر درخواست: گاهی اندازه‌ی درخواست‌های روبات‌ها از الگوی خاصی تبعیت می‌کند. به عنوان مثال تعدادی از روبات‌ها تصاویر صفحه را نادیده گرفته و تعدادی دیگر فقط اشیاء صفحه مانند تصاویر و فایل‌های فلش را درخواست می‌دهند.

۳.درصد درخواست فایل‌های js: فایل جاوا اسکریپت حاوی کدهای سمت کاربر هستند که معمولا روبات‌های وب علاقه‌ای به درخواست کردن این فایل‌ها ندارند.

۴.تعداد روی هم افتادگی نشست‌های با IP مشابه: بسیاری از روبات‌های مخرب وب از تعداد زیادی User Agent برای درخواست‌های HTTP خود استفاده می‌کنند. آن‌ها برای جلوگیری از مسدود شدن خود پس از چند درخواست فیلد User Agent درخواست‌های بعدی خود را تغییر می‌دهند. بنابراین با توجه به روش شناسایی نشست که عنوان شد، نشست‌های این نوع روبات‌ها به چند نشست کوچک تقسیم می‌شوند و با تغییر مشخصات نشست، شناسایی آن‌ها به عنوان یک نشست مخرب دشوار می‌شود. از این رو ما برای هر نشست تعداد نشست‌های با IP مشابه که در یک لحظه با این نشست اشتراک زمانی دارد و در اصطلاح روی هم افتادگی دارد را محاسبه کرده و به عنوان یک ویژگی مفید استفاده می‌کنیم. برای محاسبه‌ی این ویژگی، از زمان آغاز و پایان نشست بهره می‌گیریم، بطوری‌که اگر زمان آغاز نشستی با IP مشابه، در بین زمان آغاز و پایان نشست جاری باشد یک روی هم افتادگی برای نشست جاری محسوب می‌شود.

۵- برچسب گذاری نشست‌ها

در مطالعات گذشته برای برچسب گذاری کاربران سایت از روش‌های مختلفی استفاده می‌شد([۶, ۱۱, ۱۷]) که اغلب آن‌ها تنها به تفکیک روبات‌ها از کاربران انسانی اقدام می‌کردند. اما در تحقیقات اخیر تفکیک روبات‌های مخرب از خوش‌رفتار نیز مدنظر محققان قرار گرفته است. از جمله‌ی این تحقیقات مراجع [۵, ۱۸, ۱۹] هستند که روش برچسب گذاری آن‌ها به شرح زیر است.

اگر فیلد User Agent مربوط به یک مرورگر شناخته شده یا یک روبات وب شناخته شده باشد به آن برچسب گروه ۱ می‌دهیم، اما اگر این فیلد مربوط به یک روبات وب مخرب شناخته شده یا ناشناس باشد آن را با برچسب گروه ۲ علامت گذاری می‌کنیم.

اما با توجه اینکه بسیاری از روبات‌های مخرب وب از فیلد User Agent مرورگرهای شناخته شده استفاده می‌کنند در نتیجه به عنوان گروه ۱ یا همان کاربران مجاز شناخته شده، برچسب گذاری می‌شوند. به همین منظور ما روش جدیدی برای برچسب گذاری نشست‌های وب پیشنهاد می‌دهیم که با توجه به ویژگی‌های فیلد User Agent طراحی شده است.

۵-۱- رویکردی جدید در برچسب گذاری داده‌ها

ما در اینجا نشست‌های مجموعه داده را بر اساس فیلد User Agent و دسترسی به robot.txt و با مراحل زیر برچسب گذاری می‌کنیم. هدف نهایی برچسب زنی تمام نشست‌ها به دو گروه "کاربران خوش‌رفتار" و "کاربران مخرب یا ناشناخته" است.

۱.نشست‌هایی که فیلد User Agent آن‌ها مربوط به روبات‌های وب

۲.مخرب شناخته شده یا کاربران ناشناس باشند را در گروه کاربران مخرب یا ناشناخته قرار می‌دهیم.

۳.نشست‌هایی که IP آن‌ها در لیست سیاه روبات‌های وب مخرب باشند را در گروه کاربران مخرب یا ناشناخته قرار می‌دهیم.

۴.نشست‌هایی که User Agent آن‌ها مربوط به مرورگرهای شناخته شده باشند ولی فایل robot.txt را فراخوانی کرده باشند در گروه کاربران مخرب یا ناشناخته قرار می‌گیرند.

از آنجایی که احتمال فراخوانی فایل robot.txt توسط کاربر انسانی و با مرورگر بسیار پایین و در حد صفر است. تقریباً تمامی نشست‌هایی که این فایل را فراخوانی کرده‌اند به احتمال فراوان روبات‌های مخربی بوده‌اند که از User Agent یک مرورگر شناخته شده استفاده کرده‌اند. یا هیچ کاربری این فایل را با مرورگر خود باز نکرده است یا آنقدر تعداد آن‌ها کم است که می‌توان به عنوان نویز داده‌ها از آن‌ها چشم پوشی کرد.

۵.نشست‌های مربوط به مرورگرهای شناخته شده و روبات‌های وب خوش رفتار شناخته شده در گروه کاربران خوش رفتار قرار می‌گیرند.

همان‌طور که می‌دانیم بسیاری از روبات‌های وب مخرب از فیلد User Agent مرورگرهای شناخته شده استفاده می‌کنند. به همین دلیل ابتدا باید روبات‌های وب مخرب را برچسب گذاری کرد و در انتها باقی مانده را بر اساس این‌که ناشناخته یا شناخته شده خوش رفتار هستند، برچسب گذاری شوند. با استفاده از رویکرد بالا می‌توان بسیار دقیق‌تر برچسب گذاری نشست‌ها را انجام داد.

۶- مجموعه داده

یک نمونه از رکوردهای این مجموعه داده دارای فیلدهای آدرس IP، زمان درخواست، User agent، مسیر فایل درخواستی، وضعیت پاسخ HTTP، اندازه درخواست، نوع درخواست HTTP و فیلد refer است.

مشخصات این مجموعه داده در جدول (۱) آمده است.

جدول (۱): مشخصات مجموعه داده‌های سایت Articlebaz.com.

نشست های تست	نشست های آموزش	تعداد کل نشست‌ها	تعداد کل رکوردها	مجموعه داده
۹۷۹	۲۲۸۴	۳۲۶۳	۱۶۲۴۸۷	Articlebaz.com

۷- پیش پردازش‌های فایل مجموعه داده‌ها

همان‌طور که گفتیم برای برچسب‌گذاری رکوردها از فیلد User Agent استفاده می‌شود. به همین منظور ما با استفاده از منابع و ابزارهای [۲۰-۲۳] نوع User Agent هر نشست را مشخص کردیم تا مرحله‌ی برچسب گذاری را انجام دهیم. پس از برچسب گذاری نشست‌ها با روشی که پیش‌تر عنوان کردیم، به اطلاعات جدول (۲) دست پیدا کردیم.

جدول (۲): تنوع کلاس‌های مجموعه داده.

تعداد نشست‌های کاربران مخرب یا ناشناس	تعداد نشست‌های کاربران خوش‌رفتار	تعداد کل نشست‌ها
۴۵۶	۲۸۰۷	۳۲۶۳

۸- نتایج

الگوریتم‌های دسته‌بندی که در این مطالعه ارزیابی گردیده‌اند عبارتند از: درخت تصمیم [۲۴]، ID3 [۲۵]، C4.5 [۲۶] و شبکه‌ی عصبی، ماشین بردار پشتیبانی و روش Naive Bayesian که در [۲۷] می‌توان یافت. تمامی این الگوریتم‌ها با استفاده از نرم افزار RapidMiner و با پارامترهای پیش فرض آن پیاده سازی و ارزیابی شده‌اند.

۸-۱- ماتریس تداخل و معیارهای ارزیابی

در روش‌های یادگیری با ناظر از ماتریس تداخل برای بررسی میزان صحت الگوریتم دسته بندی استفاده می‌شود. این ماتریس میزان برچسب گذاری اشتباه دسته بند برای هر کلاس را به صورت بصری نشان می‌دهد که در مورد مسئله‌ی دسته‌بندی کاربران وب همانند جدول (۳) می‌باشد.

جدول (۳): جدول تداخل.

	کلاس واقعی	
	مخرب	خوش رفتار
پیش بینی مخرب	True Positives(TP)	False Positives(FP)
پیش بینی خوش رفتار	False Negatives(FN)	True Negatives(TN)

معیارهای متفاوتی برای ارزیابی الگوریتم‌های دسته‌بندی وجود دارد که در بین آن‌ها صحت یک معیار مناسب برای این منظور است. معیار ارزیابی صحت نشان دهنده‌ی میزان درستی جواب‌های یک روش دسته بندی است که نحوه‌ی محاسبه‌ی آن را در فرمول (۱) مشاهده می‌کنید. [۲۸, ۲۹]

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \quad (1)$$

اما همان‌طور که مشخص است، در صورتی که مجموعه‌ی داده‌ها نامتوازن باشد و تعداد نمونه‌های یک کلاس خیلی بیشتر از کلاس دیگر باشد، این معیار در نشان دادن صحیح توزیع خطا بر روی کلاس‌ها دچار مشکل می‌شود. به عنوان مثال در پیاده سازی ما این مشکل در بدترین حالت در مورد روش SVM بدون ویژگی‌های پیشنهادی اتفاق افتاد که همان‌طور که جدول تداخل آن نشان می‌دهد (جدول (۴))، با وجود دستیابی به صحت بالای ٪۸۵ در تشخیص کاربران مخرب بسیار ضعیف عمل کرده است که این معیار ناتوان از بیان آن است. همان‌طور که مشخص است در مورد مثال قبل معیار فراخوانی مقدار بسیار پایین ٪۱,۴۲ را به خود می‌گیرد و ضعف این روش دسته‌بندی در شناسایی صحیح روبات‌های مخرب (TP) را آشکار می‌کند.

جدول (۴): جدول تداخل روش SVM بدون استفاده از ویژگی‌های پیشنهادی.

	کلاس واقعی	
	مخرب	خوش رفتار
پیش بینی مخرب	۲	۲
پیش بینی خوش رفتار	۱۹۳	۸۳۶

بنابراین با توجه به مشکل ذکر شده ما برای ارزیابی روش‌های دسته بندی از معیارهای دیگری نیز استفاده می‌کنیم که برای یادگیری نامتوازن استفاده می‌شوند و تکمیل کننده‌ی معیار صحت هستند. این معیارها شامل فراخوانی، دقت و معیار F1-measure هستند که در فرمول (۲) مشاهده می‌شوند[۲۸].

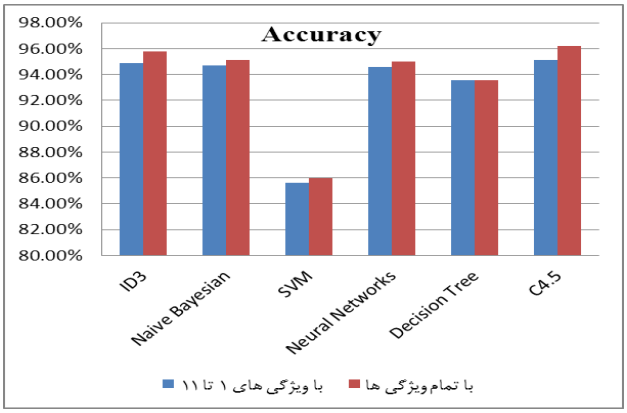
$$\text{Re } call\ (R) = \frac{TP}{TP + FN}\ ,\ \text{Pr } ecision\ (P) = \frac{TP}{TP + FP}$$

$$F1 = \frac{2RP}{R + P} \quad (۲)$$

معیار دقت نیز میزان نزدیکی جواب‌های آزمون گرفته شده از روش دسته بندی، فارغ از صحت آن‌ها را نشان می‌دهد. از آنجایی که در ارزیابی یک روش یادگیری ماشین هر دو معیار فراخوانی و دقت اهمیت دارند و نزدیکی جواب‌های این دو معیار به هم نیز مهم است از معیار F1-measure که ترکیبی از این دو معیار است، استفاده می‌شود.

در آزمایشی که برای بررسی دقت دسته‌بندی الگوریتم‌های یادگیری در شناسایی روبات‌های مخرب وب طراحی کردیم، یک‌بار تنها ویژگی‌های ۱ تا ۱۱ را بکار بردیم و بار دیگر با تمام ویژگی‌ها این آزمایش را انجام دادیم تا تأثیر استفاده از ۴ ویژگی پیشنهادی در صحت دسته‌بندی این الگوریتم‌ها را بررسی کنیم. نمودار نمایش دهنده‌ی میزان صحت

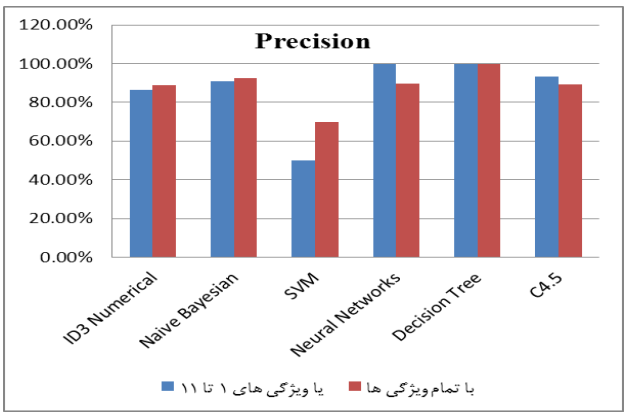
تشخیص ۶ روش یادگیری ماشین استفاده شده را در شکل (۲) مشاهده می‌کنید.



شکل (۲) : نمودار ارزیابی میزان صحت روش‌های یادگیری.

همان‌طور که در شکل (۲) پیداست، استفاده از ۴ ویژگی پیشنهادی میزان صحت دسته‌بندی کاربران وب را افزایش داده است به طوری که روش C4.5 با استفاده از تمام ویژگی‌ها به صحت ٪۹۶,۲۲ که بالاترین میزان صحت دسته‌بندی در بین روش‌های یادگیری می‌باشد، رسیده است.

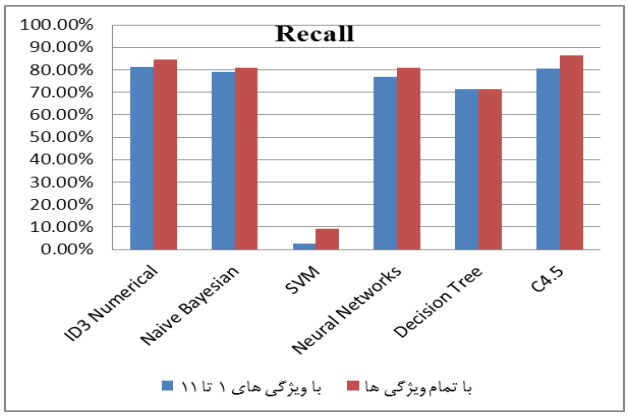
از بین ۶ روش یادگیری استفاده شده، دقت ۳ روش بین ٪۱,۸۳ تا ٪۲۰ افزایش یافته است و بیشترین دقت مربوط به درخت تصمیم است که در هر دور آزمایش به دقت ٪۱۰۰ رسیده است. شکل (۳) گویای این مطلب است. همچنین شکل (۴) و شکل (۵) به ترتیب نشان دهنده‌ی میزان افزایش فراخوانی و F1-measure با استفاده از ویژگی‌های جدید می‌باشد.



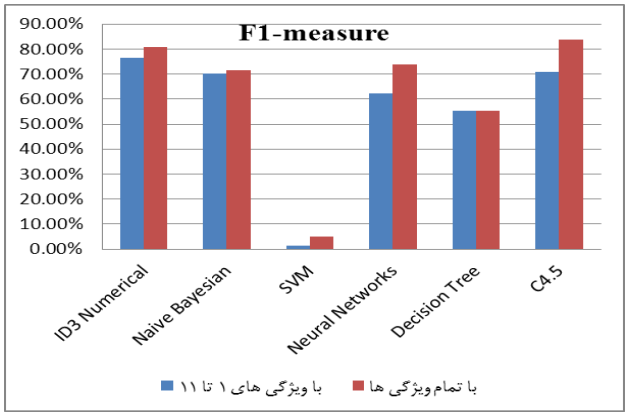
شکل (۳) : نمودار ارزیابی میزان دقت روش‌های یادگیری.

با دقت در دو شکل (۳) و (۴) متوجه تاثیر بسیار جالب استفاده از ویژگی‌های جدید در کارایی روش‌های استفاده شده می‌شویم. نکته‌ی جالب استفاده از این ویژگی‌ها کاهش مقدار دقت در روش‌های شبکه‌ی عصبی و C4.5 با وجود افزایش بقیه روش‌ها (بجز درخت تصمیم که ثابت مانده) است. همان‌طور که گفتیم بیشتر از اینکه تنها یک کدام از این دو معیار برای ما مهم باشد ترکیب این دو معیار در قالب F1-measure برای ما مهم است.

از آن جایی که دقت و فراخوانی به هم وابستگی دارند به طوری که گاهی افزایش یکی باعث کاهش و ضربه به دیگری می‌شود، بنابراین می‌توانیم کاهش این مقدار در روش‌های شبکه عصبی و C4.5 را در جهت مصالحه‌ای مفید بین دقت و فراخوانی در نظر بگیریم، چرا که موجب شده است تا این دو روش در معیار فراخوانی بیشترین افزایش یعنی حدوداً بین ۱۱ تا ۱۲ درصد را داشته باشند. همان‌طور که از شکل (۵) نیز پیداست، در تمامی روش‌ها بجز درخت تصمیم که ثابت مانده است، میزان F1-measure با استفاده از ویژگی‌های پیشنهادی افزایش یافته است.



شکل (۴) : نمودار ارزیابی میزان فراخوانی روش‌های یادگیری.



شکل (۵) : نمودار ارزیابی میزان F1-measure روش‌های یادگیری.

۸-۲- رتبه بندی ویژگی‌ها بر اساس مربع همبستگی

ضریب همبستگی عددی بین ۱- و ۱ است که پیوند بین دو ویژگی را نشان می‌دهد. اگر همبستگی بین دو ویژگی مثبت باشد، پیوند مستقیم برقرار است و اگر منفی باشد پیوند دو ویژگی بصورت معکوس است. در صورتی که ضریب همبستگی مقدار صفر به خود بگیرد به معنای استقلال دو ویژگی است. ما از مربع همبستگی برای دسته بندی ویژگی‌ها استفاده کرده‌ایم. بطوری که هرچه میزان توان دو همبستگی ما بین یک ویژگی و ویژگی برچسب، به ۱ نزدیک‌تر باشد، ویژگی بهتری محسوب می‌شود. جدول (۵) نشان می‌دهد که ویژگی

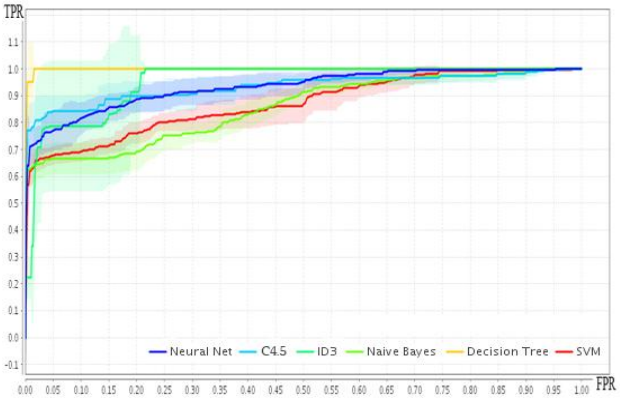
درصد فایل‌های js در جایگاه بسیار خوب ۴ام و سه ویژگی دیگر در رده‌های ۶ام، ۸ام و ۱۱ام قرار می‌گیرند.

جدول (۵): رتبه بندی ویژگی‌ها بر اساس همبستگی.

رتبه	ویژگی
۱	درصد درخواست‌های HTTP request با نوع Head
۲	درصد درخواست‌های از نوع Unassigned Refer
۳	انحراف از معیار عمق صفحه های درخواستی
۴	درصد درخواست فایل‌های js
۵	نسبت درخواست صفحه‌های HTML به تصویر
۶	میانگین اندازه‌ی هر درخواست
۷	مدت زمان نشست
۸	تعداد درخواست به زمان نشست
۹	تعداد درخواست‌های نشست
۱۰	درصد بالای درخواست‌های پی‌درپیHTTP
۱۱	تعداد روی هم افتادگی نشست‌های با IP مشابه
۱۲	درصد درخواست‌هایی که با خطاهای XX۴ مواجه می‌شوند
۱۳	تعداد بایت نشست
۱۴	درخواست فایل Robots.txt

۹- نمودار ROC

نمودار ROC به صورت بصری مصالحه‌ی بین نرخ true positive (TPR) و نرخ false positive (FPR) را نمایش می‌دهد. روش ارزیابی ROC از دو معیار ارزیابی تک ستونه TPR (محور عمودی) و FPR (محور افقی) استفاده می‌کند و یک گراف با ترسیم نرخ TP بر روی FP به دست می‌دهد. در واقع هر نقطه در این فضا نشان‌دهنده کارایی یک دسته بند برای یک توزیع داده شده است. نحوه‌ی محاسبه‌ی TPR و FPR در فرمول (۳) آمده است [۲۷, ۲۸].



شکل (۶) : نمودار ROC روش‌های یادگیری ماشین با استفاده از تمام ویژگی‌ها.

$$TPR = \frac{TP}{TP + FN} \quad \text{و} \quad FPR = \frac{FP}{TN + FP} \qquad (۳)$$

بهترین مدل دسته‌بندی باید تا جای ممکن به گوشه‌ی سمت چپ بالای نمودار نزدیک شود. همان‌طور که از شکل (۶) پیداست، با توجه به این معیار ارزیابی، روش درخت تصمیم و پس از آن C4.5 نسبت به بقیه عملکرد بهتری دارند.

۱۰- نتیجه گیری

در این مقاله مشاهده کردیم که استفاده از فیلد User Agent در برچسب گذاری مجموعه داده نیاز به رعایت نکات ظریفی دارد که در صورتی که به خصوصیات این فیلد و به سازوکار شبکه توجه کافی نشود، از کیفیت برچسب گذاری کاسته می‌شود. از بین ۴ ویژگی جدید که پیشنهاد کردیم ۲ ویژگی جدید به‌ویژه برای تشخیص روبات‌های مخربی که رفتار کاربران انسانی را تقلید می‌کنند می‌توانند مناسب باشند. "تعداد روی هم افتادگی نشست‌های با IP مشابه" و "نسبت درخواست به زمان نشست" ویژگی‌هایی هستند که به‌خصوص در روبات‌های حمله‌ی DDoS بیشتر دیده می‌شوند. در برچسب گذاری نیز با نگاهی جدید به فراخوانی فایل robot.txt توسط نشست، سعی بر آن داشتیم تا میزان خطا را کاهش دهیم و روبات‌های مخرب بیشتری را برچسب گذاری کنیم. در نهایت با ارزیابی روش‌های یادگیری به نتایج زیر دست پیدا کردیم: (۱) روش‌های یادگیری C4.5، Naive Bayesian و ID3 به صحت بالای ٪۹۵ دست پیدا کردند. درخت تصمیم با دقت ٪۱۰۰ و SVM، C4.5 و ID3 با دقتی بالای ٪۸۹ دسته‌بندی کاربران وب را انجام دادند. فراخوانی برای C4.5 به ٪۸۳,۶۹ و F1-measure به ٪۸۶,۴۵ رسید که بالاترین میزان در بین بقیه‌ی روش‌ها بود. نمودار ROC نیز درخت تصمیم و C4.5 را به ترتیب به عنوان کاراترین روش‌ها برگزید. (۲) استفاده از ۴ ویژگی پیشنهادی در کنار ویژگی‌های تحقیقات گذشته و مقایسه‌ی آن با حالتی که تنها از ویژگی‌های گذشته استفاده شده بود نشان داد که این ۴ ویژگی میزان صحت و F1-measure تمام روش‌ها بجز درخت تصمیم (که تغییری در آن ایجاد نشده است) را افزایش داده است.

۱۱- مراجع

[1] Koster,M., *A standard for robot exclusion*.1994:NEXOR.
[2] Koster,M., *Guidelines for robot writers.* Nexor Corp., http://web. nexor. co. uk/mak/doc/robots/guidelines. html, 1993.
[3] Kollar, C.P., J.R.R. Leavitt, and M. Mauldin, *Robot exclusion standard revisited.*
[4] URL: http://www.kollar.com/robots.html, 1996.
[5] Eichmann, D., *Ethical web agents.* Computer Networks and ISDN Systems, 1995. **28**(1): p. 127-136.
[6] Stevanovic, D., A. An, and N. Vlajic, *Feature evaluation for web crawler detection with data mining techniques.* Expert Systems with Applications, 2012. **39**(10): p.8707-8717.

[7] Tan, P.-N. and V. Kumar, *Discovery of Web Robot Sessions Based on their Navigational Patterns.* Data Mining and Knowledge Discovery, 2002. **6**(1): p. 9-35.

[8] Dikaiakos, M., A .Stassopoulou, and L. Papageorgiou, *Characterizing crawler behavior from web server access logs.*E-Commerce and Web Technologies,2003:p369-378.

[9] Dikaiakos,M.D., A. Stassopoulou, and L. Papageorgiou, *An investigation of web crawler behavior : characterization and metrics.* Computer Communications , 2005. **28**(8): p. 880-897.

[10] Ye, S., G. Lu, and X. Li. Workload-aware web crawling and server workload detection. in Proceedings of the second Asia-Pacific advanced network research workshop. 2004.

[11] Bomhardt, C., W. Gaul, and L. Schmidt-Thieme, *Web Robot Detection - Preprocessing Web Logfiles for Robot Detection*, in *New Developments in Classification and Data Analysis*, H.H. Bock, et al., Editors. 2005, Springer Berlin Heidelberg. p. 113-124.

[12] Stassopoulou, A. and M.D. Dikaiakos, *Web robot detection: A probabilistic reasoning approach.* Computer Networks, 2009. **53**(3): p. 265-278.

[13] Lin, J.L., *Detection of cloaked web spam by using tag-based methods.* Expert Systems with Applications, 2009. **36**(4): p. 7493.۷۴۹۹–

[14] Hou, Y.T., et al., *Malicious web content detection by machine learning.* Expert Systems with Applications, 2010. **37**(1): p. 55-60.

[15] Lu, W.Z. and S.Z. Yu. Web robot detection based on hidden Markov model. in Communications, Circuits and Systems Proceedings, 2006 International Conference on. 2006. IEEE.

[16] Von Ahn, L., et al., *CAPTCHA: Using hard AI problems for security.* Advances in Cryptology—EUROCRYPT 2003, 2003: p. 646-646.

[17] Lin, X., L. Quan, and H. Wu. An automatic scheme to categorize user sessions in modern HTTP traffic. in Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE. 2008. IEEE.

[18] Lourenço, A. and O. Belo, Applying Clickstream Data Mining to Real-Time Web Crawler Detection and Containment using ClickTips Platform, in Advances in Data Analysis. 2007, Springer. p. 351-358.

[19] Stevanovic, D., N. Vlajic, and A. An, Unsupervised Clustering of Web Sessions to Detect Malicious and Non-malicious Website Users. Procedia Computer Science, 2011. **5**: p. 123-131.

[20] Stevanovic, D., N. Vlajic, and A. An, Detection of malicious and non-malicious website visitors using unsupervised neural network learning. Applied Soft Computing, 2012.

[21] *The Web Robots Pages.* url: http://www.robotstxt.org last accessed December 2012, 20۰۰

[22] *List of User Agent Strings.* url: http://user-agent-string.info/ last accessed May 2013, 2011.

[23] Staeding, A., *List of User-Agents (Spiders, Robots, Crawler, Browser).* URL http://www. user-agents. org, 2008.

[24] *List of User Agent Strings.* url: http://www.useragentstring.com/ last accessed May 2013, 2011.

[25] Mitchell, T.M., *Machine learning. 1997.* Burr Ridge, IL: McGraw Hill, 1997. **45**.

[26] Quinlan, J.R., *Induction of decision trees.* Machine learning, 1986. **1**(1): p. 81-106.

[27] Quinlan, J.R., *C4 :۵ .programs for machine learning*. Vol. 1. 1993: Morgan kaufmann.

[28] Han, J., M. Kamber, and J. Pei, *Data mining: concepts and techniques*. 2006: Morgan kaufmann.

[29] Powers, D.M., *Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation.* School of Informatics and Engineering, Flinders University, Adelaide, Australia, Tech. Rep. SIE-07-001, 2007.

[30] Tan, P.-N., *Introduction to data mining*. 2007: Pearson Education India.