

علمی- پژوهشی

مقایسه الگوریتم‌های یادگیری ماشین نظارتی در تشخیص الگوریتم‌های تولید دامنه شبکه‌های بات

مهدی اسدی^۱، محمدعلی جبرئیل جمالی^{۲*}، سعید پارسا^۳، وحید مجید نژاد^۲

۱- دانشجوی دکتری، گروه مهندسی کامپیوتر، واحد شبستر، دانشگاه آزاد اسلامی، شبستر، ایران، ۲- استادیار، گروه مهندسی کامپیوتر، واحد شبستر،

دانشگاه آزاد اسلامی، شبستر، ایران، ۳- دانشیار، گروه مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران

(دریافت: ۱۳۹۸/۰۶/۱۲، پذیرش: ۱۳۹۸/۱۱/۱۲)

چکیده

الگوریتم‌های تولید دامنه در شبکه‌های بات به‌عنوان نقاط ملاقات مدیر بات با خدمت‌دهنده فرمان و کنترل آن‌ها مورد استفاده قرار می‌گیرند و می‌توانند به‌طور مداوم تعداد زیادی از دامنه‌ها را برای گریز از تشخیص توسط روش‌های سنتی از جمله لیست سیاه، تولید کنند. شرکت‌های تأمین‌کننده امنیت اینترنتی، معمولاً لیست سیاه را برای شناسایی شبکه‌های بات و بدافزارها استفاده می‌کنند، اما الگوریتم تولید دامنه می‌تواند به‌طور مداوم دامنه را به‌روز کند تا از شناسایی لیست سیاه جلوگیری کند. شناسایی شبکه‌های بات مبتنی بر الگوریتم تولید دامنه یک مسئله چالش‌برانگیز در امنیت سامانه‌های کامپیوتری است. در این مقاله، ابتدا با استفاده از مهندسی ویژگی‌ها، سه نوع ویژگی (ساختاری، آماری و زبانی) برای تشخیص الگوریتم‌های تولید دامنه استخراج شده و سپس مجموعه داده جدیدی از ترکیب یک مجموعه داده با دامنه‌های سالم و دو مجموعه داده با الگوریتم‌های تولید دامنه بدخواه و ناسالم تولید می‌شود. با استفاده از الگوریتم‌های یادگیری ماشین، رده‌بندی دامنه‌ها انجام شده و نتایج به‌صورت مقایسه‌ای جهت تعیین نمونه با نرخ صحت بالاتر و نرخ مثبت نادرست کمتر جهت تشخیص الگوریتم‌های تولید دامنه مورد بررسی قرار می‌گیرد. نتایج به‌دست آمده در این مقاله، نشان می‌دهد الگوریتم جنگل تصادفی، نرخ صحت، نرخ تشخیص و مشخصه عملکرد پذیرنده بالاتری را به ترتیب برابر با $0.89/32$ ، $0.91/67$ و $0.88/9$ ارائه می‌دهد. همچنین در مقایسه با نتایج سایر الگوریتم‌های بررسی شده، الگوریتم جنگل تصادفی نرخ مثبت نادرست پایین‌تری برابر با $0.37/3$ نشان می‌دهد.

کلیدواژه‌ها: شبکه‌بات، الگوریتم‌های تولید دامنه، الگوریتم‌های یادگیری ماشین، فهرست سیاه، خدمت‌دهنده فرمان و کنترل

۱- مقدمه

معمولاً، ارائه‌دهندگان امنیت با استفاده از مهندسی معکوس، الگوریتم‌ها را رمزگشایی کرده و یک لیست از دامنه‌ها را با ترافیک بالقوه فرمان و کنترل ایجاد می‌کنند. استفاده از خصوصیات آماری دامنه‌های مشابه، یکی دیگر از راهبردهای یافتن یک دامنه تولیدشده توسط الگوریتم‌های تولید دامنه است. ضعف اصلی راهبردهای سنتی، عدم توانایی تشخیص به‌صورت بلادرنگ است. کشف الگوریتم تولید دامنه در جوامع امنیت یک مسئله حیاتی بوده و راه‌حل‌های موجود عمدتاً بر اساس مهندسی معکوس و فهرست سیاه دامنه‌های فرمان و کنترل برای شناسایی ۵ بات‌ها و ترافیک آن‌ها است. مهندسی معکوس نیازمند یک نمونه بدافزار است که ممکن است در عمل همیشه ممکن نباشد [۱]. از سوی دیگر، ایجاد لیست سیاه بسیار دشوار و هزینه‌بر است زیرا میزان دامنه‌های تولیدشده به‌سرعت افزایش می‌یابد [۳]. شرکت‌های تأمین‌کننده امنیت اینترنتی، راه‌کارهای متعددی برای ردیابی^۲ ترافیک الگوریتم تولید دامنه ارائه کرده‌اند. قبل از استفاده از الگوریتم تولید دامنه‌ها، مدیران بات‌ها یک فهرست ایستا از

شبکه‌بات مجموعه‌ای از دستگاه‌های آلوده به بدافزار است و مجرمان اینترنتی برای ارسال هرزنامه‌ها، سرقت اطلاعات شخصی و انتشار حملات انکار سرویس توزیع شده از این شبکه استفاده می‌کنند [۱]. امروزه اغلب بات‌ها بر الگوریتم‌های تولید دامنه تمرکز دارند تا فهرستی از نام‌های دامنه را برای اتصال با خدمت‌دهنده فرمان و کنترل خود ایجاد کنند. الگوریتم تولید دامنه یک نوع ابزار مخرب است که توسط مدیر بات‌ها برای ایجاد تعداد زیادی از نام‌های دامنه استفاده می‌شود و با این کار مدیر بات می‌تواند خدمت‌دهنده فرمان و کنترل خود را مخفی نگه دارد تا با استفاده از روش‌های امنیتی استاندارد سایبری شناسایی نشود. این روش که شار دامنه^۱ نامیده می‌شود شبیه مخفی کردن خدمت‌دهنده فرمان و کنترل مدیر بات در یک لیست طولانی آدرس‌های آی‌پی است [۲].

* رایانامه نویسنده مسئول: m_jamali@itrc.ac.ir

¹ Domain Flux

² Trace

به صورت بلادرنگ است. عملکرد هریک از الگوریتم‌ها با سنجه‌هایی از جمله نرخ تشخیص^{۱۱}، نرخ صحت^{۱۲}، نرخ مثبت نادرست^{۱۳} و مشخصه سیستم پذیرنده^{۱۴} مورد بررسی قرار گرفته است. مشخصه عملیاتی پذیرنده در این مقاله برای هریک از الگوریتم‌ها محاسبه شده و منحنی آن برای نمایش میزان توانایی هریک از روش‌ها در تشخیص الگوریتم‌های تولید دامنه ترسیم شده است. ساختار این مقاله بدین شرح است: در بخش ۲، مرور کلی در مورد الگوریتم‌های تولید دامنه، مهندسی ویژگی‌ها^{۱۵} و ویژگی‌های استخراج شده^{۱۶} از مجموعه داده موجود ارائه شده است. الگوریتم‌های یادگیری ماشین به صورت مختصر در بخش ۳ مورد بحث قرار گرفته است. بخش ۴، آزمایش‌های مختلف و نتایج آن‌ها را ارائه می‌دهد. بخش ۵ به نتیجه‌گیری و کارهای آتی اختصاص یافته است.

۲- الگوریتم‌های تولید دامنه و مهندسی ویژگی‌ها

۲-۱- الگوریتم‌های تولید دامنه

الگوریتم‌های تولید دامنه بر ایجاد یک رشته شبه تصادفی متشکل از حروف و مقادیر عددی به صورت ایستا تمرکز دارند. همچنین برای ایجاد یک نام دامنه مناسب، دامنه سطح بالا^{۱۷} (TLD) مانند .com، .ir را به رشته ایجاد شده اضافه می‌کنند. آدرس‌های سخت‌افزاری و آدرس آی‌پی می‌تواند در فهرست سیاه قرار گرفته و مسدود شود. شبکه‌های بات جدید از الگوریتم‌های تولید دامنه برای ایجاد یک زیرساخت فرمان و کنترل پایدار استفاده می‌کنند [۲]. استفاده از دامنه‌های پویا، شناسایی و حذف شبکه‌های بات را برای مؤسس‌های امنیتی اینترنتی دشوار کرده است [۱۱].

با توجه به احتمال شناسایی دامنه‌های شبکه‌های بات در مدت‌زمان کم، دامنه‌های استفاده شده در شبکه‌های بات دارای مدت‌زمان حیات کمی بوده و به سرعت تغییر می‌یابند؛ بنابراین، استفاده از لیست سیاه برای شناسایی این نوع شبکه‌های بات کارآمد نیست. یکی از روش‌های تولید دامنه روش شار دامنه^{۱۸} است. شار دامنه فنی است که توسط شبکه‌های بات و خدمت دهنده‌های فرمان و کنترل استفاده می‌شود تا دامنه‌های زیادی را با استفاده از الگوریتم تولید دامنه ایجاد کند. تمام شبکه‌های بات و خدمت دهنده‌های فرمان و کنترل که دارای زیرساخت مشابه هستند، از یک الگوریتم مشابه استفاده می‌کنند به طوری که همه

نام‌های دامنه را استفاده می‌کردند و مدافعان سایبری با استفاده از فهرست‌های نام دامنه خاص، شبکه‌های بات ایجاد شده را شناسایی می‌کردند. با این حال، با معرفی الگوریتم‌های تولید دامنه، دامنه‌ها به طور مداوم در حال تغییر بوده و مدافعان سایبری از طریق یک لیست سیاه، نمی‌توانند تمامی دامنه‌های مهاجم را شناسایی و مسدود کنند. علاوه بر این، حتی اگر مدافع بتواند با مهندسی معکوس یک الگوریتم تولید دامنه، به آن دست یابد عملیات مهندسی معکوس زمانبر است بنابراین، استفاده از فن‌های یادگیری ماشین برای شناسایی دامنه‌های بدخواه و ناسالم می‌تواند به عنوان یک رویکرد مؤثر و سریع مطرح شود.

برای تشخیص الگوریتم‌های تولید دامنه و رده‌بندی آن‌ها با استفاده از صفات مختلف تلاش‌هایی انجام شده است. مک‌گراس و گوپتا [۴] از ویژگی‌هایی مانند سوابق ویز^۱، ویژگی‌های لغوی^۲ و آدرس‌های آی‌پی بدخواه شناخته شده استفاده کردند. محققان دیگر با به کارگیری درخت تصمیم J48 [۵] و استفاده از ویژگی‌های مبتنی بر زمان، مبتنی بر پاسخ به خدمت دهنده نام دامنه (DNS^۳) و مبتنی بر دامنه سعی در شناسایی الگوریتم‌های تولید دامنه کردند. ویژگی‌هایی مانند طول دامنه و نام میزبان برای شناسایی هرزنامه‌های تبلیغاتی مورد استفاده قرار گرفته است [۶]. محققان دیگری از ویژگی‌هایی مانند روش دو گرام^۴، توزیع خصوصیات^۵ و ویژگی‌های ساختاری دامنه‌ها مانند طول نام دامنه و کلمات موجود در دامنه و فن‌هایی مانند رگرسیون^۶ [۷]، درخت تصمیم متناوب^۷ [۸]، ماشین بردار پشتیبان^۸ [۹] و شبکه‌های عصبی اسپایک تکامل یافته^۹ استفاده کرده‌اند.

علاوه بر این، روش‌های یادگیری عمیق و روش‌های استفاده کننده از توالی منابع محلی جهانی^{۱۰} با استفاده از ویژگی‌های مبتنی بر دامنه و ویژگی‌های مبتنی بر آدرس اینترنتی برای تشخیص الگوریتم‌های تولید دامنه مورد استفاده قرار گرفته است [۱۰]. مسئله اصلی این روش‌ها این است که روش‌های ارائه شده معمولاً نیاز به اعمال پیش‌پردازش داشته و بنابراین، نمی‌توانند به صورت بلادرنگ اجرا شوند. شناسایی شبکه‌های بات مبتنی بر الگوریتم‌های تولید دامنه یک مسئله چالش برانگیز در امنیت سامانه‌های کامپیوتری است. تمرکز این مقاله، بررسی مقایسه‌ای یازده الگوریتم یادگیری ماشین نظارتی و الگوریتم شبکه عصبی جهت تشخیص الگوریتم تولید دامنه

¹ Whois records

² Lexical characteristics

³ Domain Name Server (DNS)

⁴ Bigram

⁵ Distribution of characters

⁶ Regression

⁷ Alternating Decision Tree

⁸ Support Vector Machines (SVM)

⁹ evolving Spiking Neural Networks (eSNNs)

¹⁰ Universal Resource Locator (URL)

¹¹ Detection Rate or True Positive Rate (TPR)

¹² Accuracy

¹³ False Positive Rate (FPR)

¹⁴ Receiver operating characteristic (ROC)

¹⁵ Feature Engineering

¹⁶ Extracted Features

¹⁷ Top Level Domain

¹⁸ Domain Fluxing

مختلف آن‌ها، داده‌های دامنه به یک جدول ساخت یافته تبدیل شده و به‌عنوان ورودی این الگوریتم‌ها مورداستفاده قرار می‌گیرند. مهندسی ویژگی‌ها بر روی نام دامنه‌های مجموعه داده انجام یافته و در نهایت بر اساس مطالعه الگوریتم‌ها و روش‌های پیشین، سه نوع ویژگی ساختاری^۴، زبانی^۵ و آماری^۶ برای تولید مجموعه داده جدید تولید می‌شود.

در شبکه‌های باتی که از الگوریتم‌های تولید دامنه برای تولید نام دامنه شبه تصادفی برای ارتباطات با سرویس‌دهنده فرمان و کنترل خود استفاده می‌کنند با محاسبه آنتروپی شانون^۷ [۱۴]، امتیاز آنتروپی یک دامنه محاسبه شده و دامنه‌های ناسالم از دامنه‌های پاک و سالم تشخیص داده می‌شوند. میزان آنتروپی برای یک زیر دامنه، پس از محاسبه احتمال رخداد یک کاراکتر $P(x_i)$ در یک زیر دامنه و با استفاده از رابطه (۱) برای تمامی دامنه‌ها به‌دست می‌آید:

$$Entropy = -\sum_{i=1}^n (P(x_i) \log_2 P(x_i)) \quad (1)$$

در این مقاله از مقدار آنتروپی محاسبه‌شده برای هر دامنه به‌عنوان یک ویژگی آماری از مجموعه ویژگی‌ها جهت شناسایی دامنه‌های بدخواه استفاده شده است که این ویژگی از نوع ویژگی‌های آماری در نظر گرفته شده است.

جدول (۱) ویژگی‌های ساختاری، جدول (۲) ویژگی‌های زبانی و جدول (۳) ویژگی‌های آماری استخراج شده از مجموعه داده تولیدشده را نشان می‌دهد.

آن‌ها یک دامنه مشابه و یکسانی را به‌صورت مستقل برای خود ایجاد می‌کنند. زیرمجموعه‌ای از این دامنه‌ها توسط خدمت‌دهنده‌های فرمان و کنترل ثبت می‌شوند و مدیر بات از این دامنه‌ها جهت ارتباط با شبکه‌بات خود استفاده می‌کند. برای این‌که فرآیند تشخیص الگوریتم تولید دامنه پیچیده‌تر شود، خدمت‌دهنده‌های فرمان و کنترل به‌طور مرتب دامنه‌های جدیدی با استفاده از الگوریتم تولید دامنه ایجاد می‌کنند که سبب ایجاد لیست سیاه بزرگ‌تری شده و تلاش برای شناسایی آن‌ها را سخت‌تر می‌کند. فن‌های تولید دامنه دارای پیچیدگی متنوعی در تولید نام دامنه هستند. برای مثال، روش رامیت^۱ دامنه‌هایی را با مجموعه‌ای از ضرب و تقسیم‌ها و اعمال مازول‌هایی بر روی هسته^۲ اولیه محاسبه می‌کند [۱۲]، درحالی‌که روش ساپوباکس^۳ دامنه‌ها را با تلفیق دو رشته تصادفی گرفته‌شده از زبان انگلیسی ایجاد می‌کند [۱۳].

۲-۲- مهندسی ویژگی‌ها

در این مقاله با توجه به استفاده از الگوریتم‌های رده‌بندی و یادگیری ماشین نظارتی، از روش مدیریت داده‌ها باید استفاده شود. ابتدا، داده‌های موجود که نام دامنه‌ها هستند باید به داده‌های ساخت یافته برای پذیرش توسط الگوریتم‌های رده‌بندی و یادگیری ماشین تبدیل شوند. درعین حال، در رده‌بندی، تنها ویژگی نام دامنه کافی نبوده و نیاز به برخی از ویژگی‌های دیگر است. اغلب الگوریتم‌های رده‌بندی و یادگیری ماشین نظارتی عملکرد خوبی در پیش‌بینی دارند و با توجه به ویژگی‌های

جدول (۱): ویژگی‌های ساختاری

ویژگی	توضیح	نوع ویژگی	مثال ۱ bxjofordlinnetavox.com	مثال ۲ prata.pt
HwP	دارای پیشوند www	دودویی	۰	۰
DNL	طول نام دامنه	عدد صحیح	۲۲	۸
SLM	میانگین طول زیر دامنه	عدد اعشاری	۱۸/۰	۵/۰
NoS	تعداد زیر دامنه‌ها	عدد صحیح	۱	۱
CTS	دارای TLD به‌عنوان زیر دامنه	دودویی	۰	۰
CSCS	دارای زیر دامنه تک کاراکتری	دودویی	۰	۰
UR	نسبت زیر دامنه به کل دامنه	عدد اعشاری	۰/۰	۰/۰
CIPA	دارای آدرس آی پی	دودویی	۰	۰
HVTLTD	دارای TLD معتبر	دودویی	۱	۱

⁴ Structural

⁵ Linguistic

⁶ Statistical

⁷ Shannon's Entropy

Ramnit

² Seed

³ Suppobox

جدول (۲): ویژگی‌های زبانی

مثال ۲	مثال ۱	نوع ویژگی	توضیح	ویژگی
prata.pt	bxjofordlinnetavox.com	دودویی	دارای رقم در زیر دامنه	Contains_digit
۰/۴	۰/۳۳	عدد اعشاری	نرخ حروف صدادار (A, E, I, O, U) در زیر دامنه	Vowel_ratio
۰/۰	۰/۰	عدد اعشاری	نرخ ارقام در زیر دامنه	Digit_ratio

جدول (۳): ویژگی‌های آماری

مثال ۲	مثال ۱	نوع ویژگی	توضیح	ویژگی
prata.pt	bxjofordlinnetavox.com	عدد اعشاری	نرخ تعداد تکرار کاراکترها در زیر دامنه	RRC
۰/۲۵	۰/۲۱	عدد اعشاری	نرخ حروف بی‌صدا در زیر دامنه	RCC
۰/۴	۰/۴۴	عدد اعشاری	نرخ ارقام متوالی در زیر دامنه	RCD
۰/۰	۰/۰	عدد اعشاری	آنترپی زیر دامنه	Entropy
۱/۹۲۱	۳/۶۸۳	عدد اعشاری		

۳- الگوریتم‌های یادگیری ماشین

رویکرد داده‌کاوی متناسب با تشخیص و تفکیک دامنه‌های موجود به دو رده دامنه‌های سالم و دامنه‌های ناسالم، رویکرد رده‌بندی خواهد بود. با استفاده از رویکرد رده‌بندی مشخصه‌های تأثیرگذار بر ناسالم بودن این دامنه‌ها نیز پیش‌بینی می‌شود. با توجه به استفاده از مجموعه داده‌های برچسب‌دار^۱ در این مقاله، یازده الگوریتم یادگیری ماشین نظارتی و یک الگوریتم شبکه عصبی مورد استفاده قرار گرفته است. به منظور آموزش الگوریتم‌های یادگیری نظارتی، داده‌ها یا ویژگی‌هایی به عنوان ورودی اولیه به سیستم وارد می‌شود که قبلاً توسط عامل انسانی با خروجی مورد انتظار برچسب‌زده شده است و ورودی‌ها و خروجی‌های مطلوب متناظر با آن‌ها به صورت جفت‌های ورودی/خروجی در اختیار سیستم قرار می‌گیرند. با استفاده از این مجموعه داده‌ها، الگوریتم، رابطه میان ورودی‌ها و پاسخ درست از پیش تعیین شده (برچسب) را آموزش می‌بیند. فرآیند آموزش تا زمانی که نمونه پیش‌بینی الگوریتم به دقت کافی برسد و نتایج پیش‌بینی برابر و یا نزدیک به نتایج از قبل تعیین شده باشند ادامه خواهد یافت. در یادگیری نظارتی، سیستم پس از آموزش و با استفاده از الگوهای که در طی آموزش ایجاد نموده است، قادر خواهد بود تا در مواجهه با داده‌های بدون برچسب، خروجی مناسب را پیش‌بینی نماید. نمونه‌های استفاده شده از این الگوریتم‌ها در این مقاله عبارت‌اند از رگرسیون منطقی^۲، درخت تصمیم^۳، جنگل تصادفی^۴،

K نزدیک‌ترین همسایه^۵، سه نوع شبکه بیزی^۶ - گاوسی^۷، برنولی^۸ و چندجمله‌ای^۹، الگوریتم درخت‌های اضافی^{۱۰}، دو نوع الگوریتم تقویتی^{۱۱} و ماشین بردار پشتیبان^{۱۲}. همچنین شبکه عصبی چندلایه^{۱۳} به عنوان یک الگوریتم رده‌بندی دیگر مطالعه شده است. در ادامه توضیح مختصری از این الگوریتم‌ها ارائه داده می‌شود.

۳-۱- رگرسیون منطقی

رگرسیون منطقی یکی از فن‌های کاربردی برای تحلیل داده‌های رده‌بندی شده است. زمانی که متغیر هدف، متغیری کیفی با دو سطح باشد، دیگر نمونه‌های رگرسیون معمولی قابل استفاده نیستند. در این گونه موارد، از رگرسیون منطقی استفاده می‌شود. هدف رگرسیون منطقی، تعیین احتمال شرطی مربوط به مشاهده‌های مشخص یک رده با توجه به متغیرهای مستقل است. به عبارت ساده‌تر، با گرفتن متغیر ورودی، مقدار متغیر وابسته به آن را پیش‌بینی می‌کند [۱۸].

۳-۲- شبکه بیزی

شبکه بیزی، نمونه مقدماتی از نمونه احتمال بیزی است. این

^۵ K-Nearest Neighbors

^۶ Bayesian Network

^۷ Gaussian

^۸ Bernoulli

^۹ Multinomial

^{۱۰} Extra Trees

^{۱۱} Boosting Algorithms

^{۱۲} Support Vector Machine

^{۱۳} Multilayer Perceptron

^۱ Labeled

^۲ Logistic Regression

^۳ Decision Tree

^۴ Random Forest

مرتب کردن آن‌ها در درخت از گره ریشه، گره‌های که در بالای درخت قرار دارد، به سمت گره‌های برگ، گره‌های انتهایی درخت که فقط از یک سو به سایر گره‌ها متصل هستند، رده‌بندی می‌کند. هر گره داخلی (غیر برگ) از درخت، متناظر با یک مشخصه از مشاهدات بوده و هر یالی که از آن خارج می‌شود متناظر با یک مقدار برای آن مشخصه است. در نهایت هر گره برگ، در یک رده متغیر هدف، رده‌بندی می‌شود [۳۳]. درخت‌های تصمیم بالا به پایین یکی از الگوریتم‌های رایج رده‌بندی می‌باشند [۱۷]. از مهم‌ترین دلایل رایج بودن این الگوریتم‌ها شفافیت و قابلیت تفسیر بالای آن‌ها است. مزیت دیگر آن‌ها موجود بودن پیاده‌سازی‌های قوی همچون درخت‌های ID3، C4.5 و CHAID است. تنها تفاوت این الگوریتم با درخت تصمیم ساده این است که این روش برای ساخت درخت، سطح اهمیتی از آزمون کای مربع^۵ را مشخص می‌کند تا رشد درخت را متوقف کند [۳۴]. برای مثال الگوریتم ID3 از الگوریتم‌های ساده درخت‌های تصمیم بوده و ایده اساسی این روش، فرایند جستجوی حریصانه بالا به پایین در مجموعه داده، به منظور ارزیابی هر مشخصه در هر گره بوده و در آن انتخاب‌های قبلی هرگز مورد بازبینی قرار نمی‌گیرند [۳۵].

۳-۵- جنگل تصادفی

درخت‌های تصمیم به دلیل کاربرد آسان و تفسیر آن‌ها در حوزه رده‌بندی‌های دوسطحی بسیار اهمیت یافته‌اند. کاربرد این درخت‌ها، امکان استفاده از متغیرهای پیش‌بینی کننده با مقیاس‌های متفاوت را فراهم می‌کند. عدم ثبات و پایداری درخت‌های تصمیم و ایجاد راه‌حل‌های بهینه محلی یکی از مشکلات کاربرد آن‌ها است. فن جنگل تصادفی از فن‌های دارای رویکرد رده‌بندی است که برای رفع مشکلات موجود در فن درخت تصمیم ارائه شده است. در این فن، مجموعه‌ای از درخت‌های تصمیم ایجاد شده و هر درخت به مهم‌ترین رده رأی می‌دهد. با ادغام رأی درخت‌های مختلف، برای هر نمونه یک رده پیش‌بینی می‌شود. در این روش که برای افزایش دقت درخت تصمیم طراحی شده است، تعداد درخت بیشتری تولید می‌شود تا برای پیش‌بینی رده باهم اقدام به رأی‌گیری کنند. این روش یک رده‌بندی کننده جمعی است که از تعدادی درخت تصمیم تشکیل شده است و نتیجه نهایی، میانگین نتیجه تک‌تک درخت‌ها است [۳۱]. به عبارت دیگر، جنگل تصادفی چندین درخت تصمیم ساخته و با ادغام آن‌ها پیش‌بینی‌های درست‌تری ارائه می‌دهد. از مزایای جنگل تصادفی قابلیت استفاده آن، هم برای مسائل رده‌بندی و هم مسائل رگرسیون است که غالب سامانه‌های یادگیری ماشین کنونی را تشکیل می‌دهند [۱۹].

نمونه بر پایه احتمال وقوع یا عدم وقوع پدیده‌ها است. عملکرد آن، بر فرضیات استقلال قوی استوار بوده و احتمال رخداد یک صفت روی احتمال سایر صفات بی‌تأثیر است. به این معنی که در این نمونه، احتمال رخداد نتیجه نهایی، بر اساس احتمالات رخداد متغیرهای مستقل به شرط رخداد همان نتیجه به دست آمده و احتمال رخداد هر یک از متغیرهای مستقل به شرط رخداد یک نتیجه نهایی خاص، مستقل از احتمال رخداد سایر متغیرهای مستقل به شرط رخداد همان نتیجه است. به این ترتیب، یک مسئله چند متغیره p بعدی به تخمین p مسئله یک متغیره کاهش پیدا می‌کند. این امر، سبب کاهش پیچیدگی‌های محاسبات می‌شود [۹]. سه نوع نمونه بیزی در این مقاله مورد بررسی قرار گرفته‌اند که عبارت‌اند از:

- بیزی گاوس^۱ که در رده‌بندی استفاده می‌شود و فرض می‌شود که ویژگی‌ها توزیع نرمال را دنبال می‌کنند.
- بیزی چندجمله‌ای^۲ برای اعداد و مقادیر گسسته^۳ استفاده می‌شود.
- بیزی برنولی^۴، اگر بردارهای ویژگی، دوتایی باشد نمونه دو جمله‌ای مناسب است (به عنوان مثال ۰ و ۱).

۳-۳- K نزدیک ترین همسایه

هدف از فن K نزدیک‌ترین همسایه، رده‌بندی یک عضو جدید بر اساس ویژگی نمونه‌های آموزش‌دهنده است. در این فن نمونه جدید بر اساس اکثریت K رده که نزدیک‌ترین همسایه‌ها را با آن نمونه داشته باشند، تقسیم‌بندی می‌شود. به طور کلی می‌توان بیان کرد که روش K نزدیک‌ترین همسایه، یک روش تشخیص الگوهای غیر پارامتری است که تعداد K تا از نزدیک‌ترین الگوهای مشابه را پیدا کرده و بر اساس آن‌ها، ارزش نمونه مورد مطالعه را پیش‌بینی می‌کند. این الگوریتم بر اساس حداقل فاصله نمونه مورد بررسی تا نمونه‌های موجود دیگر برای تعیین K نزدیک‌ترین همسایه‌ها کار می‌کند و نمونه را متعلق به رده‌ای می‌داند که بیشترین آرا را در بین K نزدیک‌ترین همسایه داشته باشد [۲۱ و ۳۲].

۳-۴- درخت تصمیم

درخت تصمیم از معروف‌ترین فن‌های رده‌بندی است. هر درخت تصمیم از تعدادی گره و یال تشکیل شده است. در ساخته شدن درخت و تشکیل هر گره، الگوریتم درخت تصمیم به دنبال انتخاب بهترین مشخصه برای شکستن درخت به دو یا چند زیر درخت است. درخت تصمیم، مشاهدات وارد شده به نمونه را با

⁵ Chi Square

¹ Gaussian Naïve Bayesian

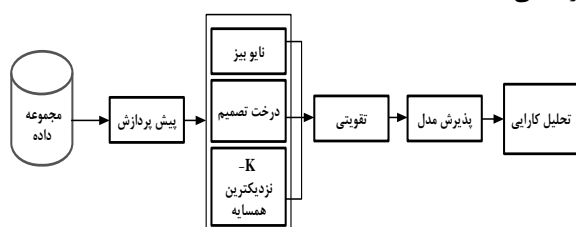
² Multinomial Naïve Bayesian

³ Discrete counts

⁴ Bernolli Naïve Bayesian

۳-۶- ماشین بردار پشتیبان

وزنشان کاهش خواهد یافت. سپس وزن دیگری به صورت مجزا به هر رده‌بندی کننده با توجه به دقت کلی آن اختصاص داده می‌شود که بعداً در فاز آزمایش (آزمون) مورد استفاده قرار می‌گیرد. رده‌بندی کننده‌های دقیق از ضرب اطمینان بالاتری برخوردار خواهند بود. در نهایت هنگام ارائه یک نمونه جدید هر رده‌بندی کننده، یک وزن پیشنهاد می‌دهد و کلاس بارأی اکثریت انتخاب خواهد شد. پارامتری که رده‌بندی کننده جمعی دارد تعداد تکرار است. با توجه به این که این روش یک زیر فرآیند دارد که این زیر فرآیند باید یک الگوریتم یادگیرنده داشته باشد، از الگوریتم‌های یادگیری همچون درخت تصمیم، K نزدیک‌ترین همسایه، بیسی ساده و غیره استفاده می‌شود. شکل (۱) نمایی از این نمونه را ارائه می‌دهد.



شکل (۱): نمایی از نمونه جمعی

ماشین بردار پشتیبان یک الگوریتم رده‌بندی مبتنی بر نظریه یادگیری آماری است و از الگوریتم‌های بر پایه هسته در یادگیری ماشین محسوب می‌شود. الگوریتم ماشین بردار پشتیبان یکی از الگوریتم‌های معروف در زمینه یادگیری نظارتی است که برای رده‌بندی و رگرسیون استفاده می‌شود. این الگوریتم به‌طور هم‌زمان حاشیه‌های هندسی را بیشینه کرده و خطای رده‌بندی را کمینه می‌کند لذا به‌عنوان رده‌بندی حداکثر حاشیه نیز نامیده می‌شود. یک مسئله با دو کلاس نتیجه خطوط بی‌شماری ممکن است وجود داشته باشد که توسط آن‌ها رده‌بندی انجام شود ولی فقط یکی از این خطوط ماکزیمم تفکیک و جداسازی را فراهم می‌آورد. از بین جداسازهای خطی، آن جداسازی که حاشیه داده‌های آموزشی را حداکثر می‌کند خطای تعمیم را حداقل خواهد کرد. نقاط داده‌ای ممکن است ضرورتاً نقاط داده‌ای در فضای دوبعدی R^2 نباشند و ممکن است در فضای چندبعدی R^n مربوطه باشند. رده‌بندی کننده‌های خطی متعددی ممکن است این خصوصیت را فراهم کنند اما الگوریتم ماشین بردار پشتیبان به دنبال جداکننده‌ای است که حداکثر جداسازی را برای دسته‌ها انجام دهد [۲۰].

۳-۷- الگوریتم درخت‌های اضافی^۱

الگوریتم درخت‌های اضافی [۳۷] یک گروه از درخت‌های تصمیم و رگرسیون را مطابق با رویه بالا به پایین^۲ کلاسیک ایجاد می‌کند. دو تفاوت اصلی این الگوریتم با سایر الگوریتم‌های گروهی مبتنی بر درخت از جمله جنگل تصادفی در این است که اولاً گره‌ها را با انتخاب نقاط برش به‌طور تصادفی تقسیم می‌کند و ثانیاً از کل نمونه‌های یادگیری برای گسترش درخت‌ها استفاده می‌کند.

۳-۸- رده‌بندی کننده جمعی^۳

الگوریتم رده‌بندی کننده جمعی [۲۲] از کل مجموعه داده به‌منظور آموزش هر رده‌بندی کننده استفاده می‌کند، اما بعد از هر بار آموزش، بیشتر بر روی داده‌های سخت تمرکز می‌کند تا به‌درستی رده‌بندی شوند. این الگوریتم تکراری، تغییر انطباقی به توزیع داده‌های آموزش با تمرکز بیشتر بر روی نمونه‌هایی است که قبلاً به‌طور درست رده‌بندی نشده‌اند. در ابتدا تمام رکوردها وزن یکسانی می‌گیرند و وزن‌ها در هر تکرار افزایش پیدا خواهند کرد. وزن نمونه‌هایی که به‌اشتباه رده‌بندی شده‌اند افزایش خواهد یافت در حالی که آن دسته از نمونه‌هایی که به‌درستی رده‌بندی شده‌اند

۳-۸-۱- الگوریتم تقویتی تطبیقی^۴

الگوریتم تقویتی تطبیقی [۲۳] یک الگوریتم بزرگ است که جهت افزایش دقت رده‌بندی به‌همراه دیگر الگوریتم‌های یادگیری استفاده می‌شود. در این الگوریتم، رده‌بندی کننده در هر مرحله جدید، به نفع نمونه‌های نادرست رده‌بندی شده در مراحل قبل تنظیم می‌شود. هدف الگوریتم تقویتی تطبیقی افزایش میزان یادگیری رده‌بندی شده است. این الگوریتم با ترکیب چند رده‌بندی کننده ضعیف مرز مناسبی جهت تفکیک داده‌های بین دو کلاس به‌دست می‌آورد. رده‌بندی کننده‌ها همیشه به نفع داده‌هایی که در مرحله قبل به‌اشتباه رده‌بندی شده‌اند، عمل می‌کنند. الگوریتم تقویتی تطبیقی نسبت به داده‌های اختلال‌دار^۵ و پرت^۶ حساس بوده ولی نسبت به مسئله بیش‌برازش^۷ از اغلب الگوریتم‌های یادگیری بهتر عمل می‌کند.

۳-۸-۲- الگوریتم تقویتی گرادیان^۸

الگوریتم تقویتی گرادیان [۲۴]، نیز مانند جنگل تصادفی با استفاده از درخت‌های تصمیم ضعیف عمل می‌کند. تفاوت این دو

^۴ Adaptive Boosting (AdaBoost)

^۵ Noisily

^۶ Outlier

^۷ Over-fitting

^۸ Gradient Boosting

^۱ Extra-Trees

^۲ Top-Down Procedure

^۳ Boosting

است، استفاده شده است. پلتفرم مورد استفاده برای یادگیری، ویندوز ۷ بوده و بسیاری از آزمایش‌ها برای تعیین پارامترهای تنظیم مناسب این الگوریتم‌ها انجام یافته است.

۴-۲- مجموعه داده

در این مقاله از سه مجموعه داده اولیه برای تولید مجموعه داده جدید استفاده شده است که عبارت‌اند از:

۱. داده‌های سالم (دامنه‌های پاک) مجموعه Alexa

۲. داده‌های بدخواه (دامنه‌های ناسالم):

الف. مجموعه داده Bambenek

ب. مجموعه داده 360 Lab

با ترکیب و درهم‌سازی این سه مجموعه داده، مجموعه داده جدیدی حاوی دامنه‌های سالم و ناسالم برای استخراج ویژگی‌های ساختاری، زبانی و آماری اشاره شده در بخش ۲-۲، ایجاد شد. این مجموعه داده حاوی ۲,۴۵۸,۸۳۶ رکورد (دامنه) بوده و پس از استخراج ویژگی‌ها، مجموعه داده‌ای با ۱۶ ویژگی تولید شد. برای انجام آزمایش‌ها و ارزیابی عملکرد الگوریتم‌های یادگیری ماشین و الگوریتم شبکه عصبی از ۱۰۰ هزار رکورد تصادفی انتخاب شده از تمام رکوردهای مجموعه داده استفاده شد. برای انجام آزمایش‌ها، ۷۰ درصد مجموعه داده برای آموزش الگوریتم و ۳۰ درصد باقیمانده جهت آزمون الگوریتم تخصیص داده شد.

۴-۳- ارزیابی عملکرد و نتایج

برای ارزیابی الگوریتم‌های اشاره شده از روش اعتبارسنجی متقابل^۴ با $K\text{-fold}=10$ استفاده شد. با این روش میزان عملکرد الگوریتم‌ها به صورت دقیق‌تر مورد ارزیابی قرار گرفت. میانگین نتایج، با ۳۰ بار آزمایش بر روی مجموعه داده محاسبه شده و برای هر یک از الگوریتم‌ها ارائه شده است.

۴-۳-۱- سنج‌های ارزیابی

یکی از مهم‌ترین مراحل پس از طراحی یا ساخت نمونه یا الگوریتم، ارزیابی کارایی^۵ آن است. در الگوریتم‌های یادگیری ماشین نظارتی و مسائل رده‌بندی برای ارزیابی عملکرد هر یک از الگوریتم‌ها، سنج‌هایی از جمله نرخ صحت^۶، نرخ تشخیص^۷ یا نرخ بازخوانی^۸، نرخ مثبت نادرست^۹ و مشخصه عملیاتی پذیرنده^{۱۰} استفاده می‌شود که در زیر نحوه محاسبه هر یک از آن‌ها اشاره می‌شود:

الگوریتم آن است که در الگوریتم تقویتی گرادیان درخت‌ها یکی پس از دیگری آموزش داده شده و هر درخت زیرمجموعه، ابتدا با داده‌هایی که به‌اشتباه توسط درخت قبلی پیش‌بینی شده‌اند آموزش داده می‌شود. این امر سبب می‌شود نمونه بیشتر بر روی موارد پیچیده متمرکز شده و کمتر به مسائلی که پیش‌بینی در آن‌ها آسان است اهمیت دهد.

۳-۹- الگوریتم شبکه عصبی چندلایه

مفهوم شبکه‌های عصبی از مغز انسان الهام گرفته شده است و برای مسائلی مانند رده‌بندی و پیش‌بینی استفاده شده است. شبکه‌های عصبی استفاده وسیعی در شناسایی الگوها دارند؛ چون قابلیت پاسخگویی به ورودی‌های غیرمنتظره را دارند. در ساخت شبکه عصبی، نورون‌ها یاد می‌گیرند که چگونه الگوهای ویژه مختلفی را تشخیص دهند. اگر الگویی پذیرفته شود درحالی که در طول یادگیری، ورودی با خروجی مرتبط نباشد، نورون‌ها از مجموعه الگوهایی که قبلاً یاد گرفته، آن خروجی را که بیشترین شباهت را به الگو داشته و کم‌ترین تفاوت را با ورودی دارد، انتخاب می‌کند. این فرآیند از سه لایه شامل لایه‌های ورودی، لایه‌های میانی، لایه‌های خروجی تشکیل شده است. لایه‌های ورودی، اطلاعات را از یک منبع خارجی دریافت می‌کند و یک یا چندلایه خروجی نیز اطلاعات را به سیگنال‌هایی برای استفاده منبع خروجی تبدیل می‌کند. لایه‌های میانی نیز واسطی بین لایه‌های ورودی و لایه‌های خروجی است و در حقیقت، فرآیند پردازش داده را انجام می‌دهند [۲۵ و ۳۶].

۴- نتایج و اعتبارسنجی

۴-۱- جزئیات پیاده‌سازی

برای انجام آزمایش‌های مختلف دستگاهی با پردازنده اینتل Core i7-2670QM با سرعت 2.20 گیگاهرتز، حافظه اصلی ۸ گیگابایت و کارت گرافیکی NVIDIA GeForce GT540 با ۲ گیگابایت حافظه مورد استفاده قرار گرفته است. الگوریتم‌های یادگیری ماشین نظارتی و الگوریتم شبکه عصبی با استفاده از زبان برنامه‌نویسی پایتون نسخه ۳,۶ در محیط ژوپیتِر نوت‌بوک^۱ و تنسورفلو^۲ [۲۶] پیاده‌سازی و آموزش داده شده و از ابزار کراس نسخه ۲,۲,۲ [۲۷] که یک واسط برنامه‌نویسی کاربردی^۳ (API) برای نمونه سریع از تنسورفلو نسخه ۱,۱۰,۰ در پردازنده (CPU)

⁴ Cross Validation (CV)

⁵ Performance Evaluation

⁶ Accuracy

⁷ True Positive Rate (TPR)

⁸ Recall

⁹ False Positive Rate (FPR)

¹⁰ Receiver Operating Characteristic (ROC)

¹ Jupyter Notebook

² Tensorflow

³ Application programming Interface

یک از این الگوریتم‌ها حاصل شد. الگوریتم جنگل تصادفی با توجه به بررسی انواع درخت‌ها با حالت‌های مختلف سعی در یافتن بهترین درخت داشته و نسبت به سایر الگوریتم‌ها نتایج بهتری در تمامی سنج‌های ارزیابی در این مقاله ارائه داده است. برای مثال با تنظیم تعداد تخمین‌کننده^۲ در جنگل تصادفی و با افزایش یا کاهش این پارامتر نتایج متفاوتی حاصل شد ولی با تنظیم این پارامتر با مقدار ۱۰، بالاترین نرخ صحت و نرخ تشخیص و پایین‌ترین نرخ مثبت نادرست به دست آمد.

زمان آموزش و آزمایش این الگوریتم نیز با توجه به نتایج حاصل مناسب بود. در الگوریتم‌های درخت تصمیم و درخت اضافی نیز سعی شد تا پارامترهای مختلف بررسی و نتایج آن‌ها مقایسه شود که بهترین نتایج در درختانی با ماکزیمم عمق^۳ ۱۰ حاصل شد. زمان آموزش و آزمایش این الگوریتم‌ها نیز مقداری قابل پذیرش بود.

در الگوریتم‌های ماشین بردار پشتیبان و K نزدیک‌ترین همسایه علی‌رغم دستیابی به نتایج مناسب در نرخ صحت، نرخ تشخیص و نرخ مثبت نادرست، دارای مدت زمان آموزش و آزمایش بسیار بالا بوده و عملکرد زمانی بسیار ضعیف‌تری نسبت به سایر الگوریتم‌ها ارائه دادند. بنابراین، با در نظر گرفتن این نقطه ضعف این الگوریتم‌ها برای به‌کارگیری بر روی مجموعه داده‌های بزرگ‌تر از جمله تمام داده‌های مجموعه داده استفاده شده در این مقاله مناسب نخواهند بود. الگوریتم‌های تقویتی نیز نتایج خوبی در تشخیص دامنه‌های سالم و دامنه‌های ناسالم ارائه دادند ولی مدت زمان آزمایش این الگوریتم‌ها نیز نسبتاً زیاد بود.

در الگوریتم شبکه عصبی چندلایه با تغییر تعداد لایه‌های مخفی و میانی و تغییر تعداد گره‌های هر لایه نتایج مختلفی حاصل شد. با انجام آزمایش‌های مختلف، بهترین نتایج در شبکه با ۳ لایه و با ۳۰ گره در هر لایه حاصل شد. لازم به یادآوری است که به‌کارگیری تعداد لایه‌ها و گره‌های بیشتر تأثیری در نرخ صحت و نرخ تشخیص نداشته است. الگوریتم‌های بی‌زی در تشخیص دامنه‌های سالم و دامنه‌های ناسالم عملکرد ضعیف‌تری نسبت به سایر الگوریتم‌های اشاره‌شده ارائه داده ولی زمان آموزش و زمان آزمایش بسیار پایین‌تر و بهتری نسبت به سایر الگوریتم‌ها داشتند.

در مجموع نتایج حاصل نشان از عملکرد مناسب الگوریتم جنگل تصادفی در سنج‌های ارزیابی دارد. بالاترین نرخ صحت برابر با ۸۹/۳۲٪، بالاترین نرخ تشخیص برابر ۹۱/۶۷٪، با مقدار مشخصه عملیاتی پذیرنده ۰/۹۰۲ و کمترین نرخ مثبت نادرست ۰/۳۷۳ با به‌کارگیری الگوریتم جنگل تصادفی به دست آمد.

- مثبت درست (TP): تعداد دامنه‌های ناسالمی که به‌درستی به‌عنوان دامنه ناسالم شناسایی شده‌اند.
- مثبت نادرست (FP): تعداد دامنه‌های سالمی که به‌اشتباه به‌عنوان دامنه‌های ناسالم شناسایی شده‌اند.
- منفی درست (TN): تعداد دامنه‌های سالمی که به‌درستی به‌عنوان دامنه عادی شناسایی شده‌اند.
- منفی نادرست (FN): تعداد دامنه‌های ناسالمی که به‌اشتباه به‌عنوان دامنه‌های عادی شناسایی شده‌اند.

• **صحت:** درصد پیش‌بینی‌های درست تمام دامنه‌ها را نشان می‌دهد.

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

• **نرخ تشخیص یا نرخ بازخوانی:** نشان‌دهنده درصد دامنه‌های ناسالمی است که به‌درستی به‌عنوان یک دامنه ناسالم پیش‌بینی شده است.

$$True Positive Rate (TPR) = \frac{TP}{TP + FN} \quad (2)$$

• **نرخ مثبت نادرست:** درصد دامنه‌های سالمی را که به‌اشتباه به‌عنوان دامنه‌های ناسالم رده‌بندی شده‌اند، نشان می‌دهد.

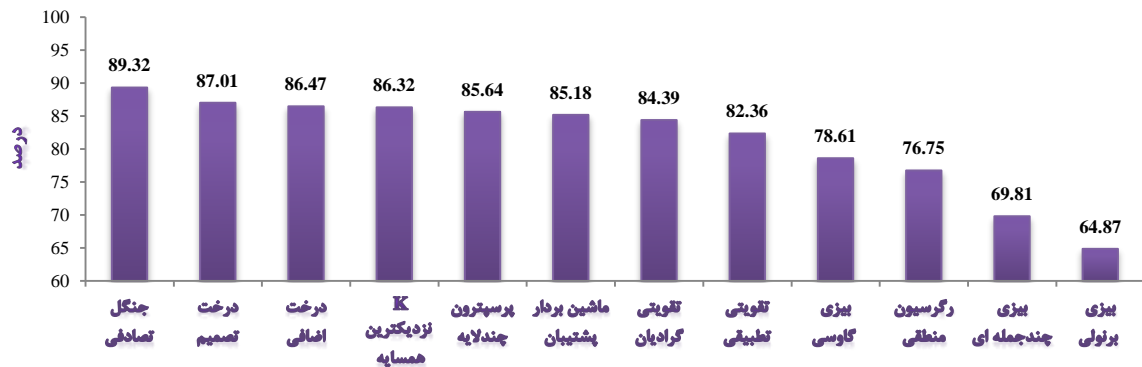
$$FPR = \frac{FP}{FP + TN} \quad (3)$$

• **منحنی مشخصه عملیاتی پذیرنده:** نشان‌دهنده میزان عملکرد الگوریتم برای حل مسئله‌ای است که بر روی آن اجرا می‌شود. هر چه نرخ مثبت درست^۱ بیشتر از نرخ مثبت نادرست باشد این مشخصه مقدار بالاتر و بهتری را نشان می‌دهد در نتیجه الگوریتم موردنظر عملکرد بهتری برای حل مسئله خواهد داشت.

۴-۳-۲- نتایج

جدول (۴) و شکل‌های (۲) تا (۴)، نرخ صحت، نرخ تشخیص یا بازخوانی، نرخ مثبت نادرست و مشخصه عملیاتی پذیرنده الگوریتم‌های یادگیری ماشین نظارتی اشاره‌شده و شبکه عصبی چندلایه را نشان می‌دهد. مشخصه عملیاتی پذیرنده به صورت مجزا برای هر یک از الگوریتم‌ها محاسبه و منحنی آن در شکل‌های (۹ - ۶) ارائه شده است.

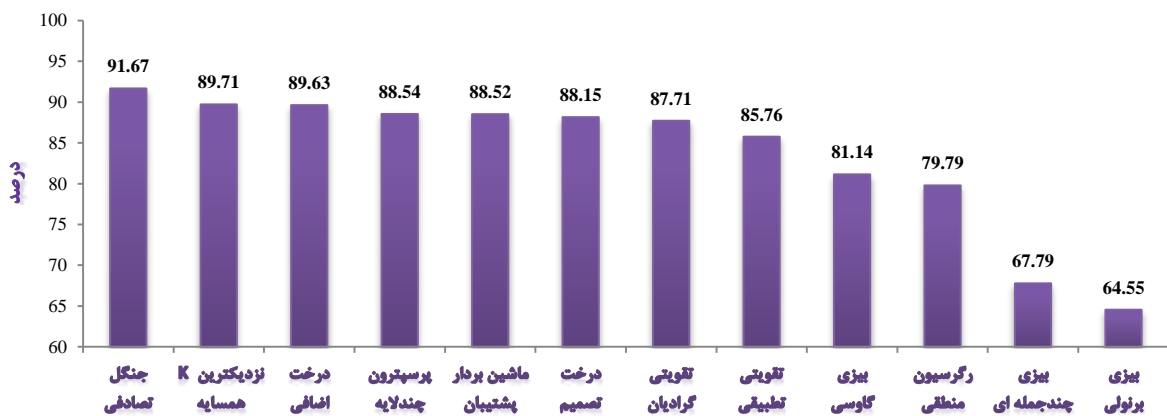
با انجام آزمایش‌های متعدد نتایج ارائه‌شده در جدول (۴) حاصل شد. این نتایج نشان می‌دهند الگوریتم‌های مبتنی بر درخت از جمله جنگل تصادفی، درخت تصمیم و درخت اضافی نسبت به سایر الگوریتم‌ها عملکرد بهتری ارائه می‌دهند. با بررسی و تغییر پارامترهای تنظیم الگوریتم‌ها، نتایج متفاوتی توسط هر



شکل (۲): نمودار نرخ صحت الگوریتم‌ها در تشخیص DGA

الگوریتم جنگل تصادفی ارائه دادند. ضعیف‌ترین نتایج در این سنجه را الگوریتم بیزی برنولی ارائه داده است.

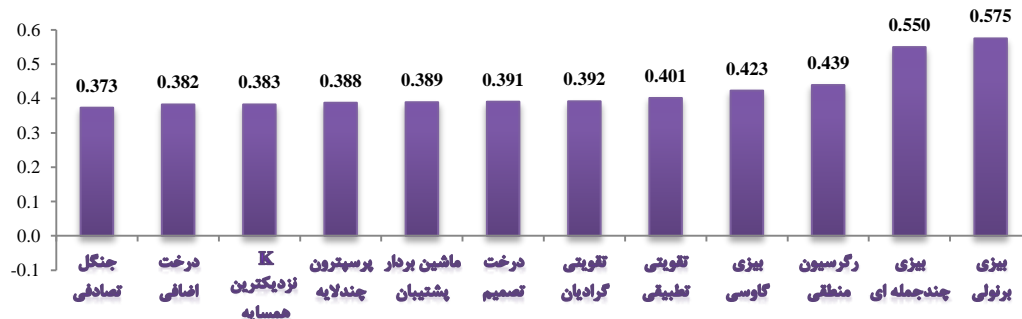
در نتایج حاصل از پیاده‌سازی سنجه نرخ تشخیص، الگوریتم جنگل تصادفی بهترین مقدار را برابر با ۸۹/۳۲ ارائه داده است و الگوریتم‌های درخت تصمیم و درخت اضافی نیز نتایج نزدیک به



شکل (۳): نمودار نرخ تشخیص الگوریتم‌ها در تشخیص DGA

نزدیک به الگوریتم جنگل تصادفی ارائه دادند. ضعیف‌ترین نتایج در این سنجه را الگوریتم بیزی برنولی ارائه داده است.

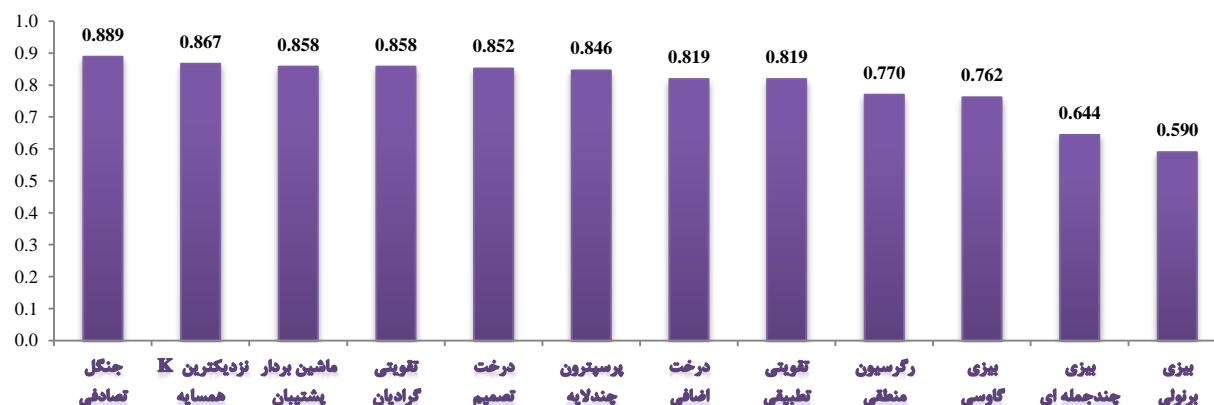
در نتایج حاصل از پیاده‌سازی سنجه نرخ تشخیص، الگوریتم جنگل تصادفی بهترین مقدار را برابر با ۹۱/۶۷ ارائه داده است و الگوریتم‌های K نزدیک‌ترین همسایه و درخت اضافی نیز نتایج



شکل (۴): نمودار نرخ مثبت نادرست الگوریتم‌ها در تشخیص DGA

نتیجه و بالاترین نرخ مثبت نادرست توسط الگوریتم بیزی برنولی برابر با ۰/۵۷۵ حاصل شده است.

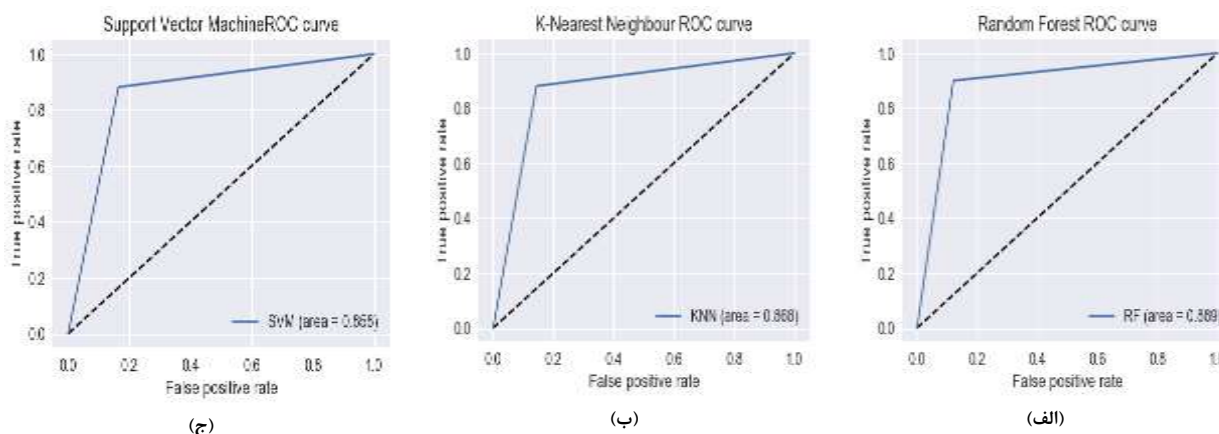
بهترین و کمترین نرخ مثبت نادرست توسط الگوریتم‌های جنگل تصادفی با مقدار ۰/۳۷۳ ارائه شده و درخت اضافی و K نزدیک‌ترین همسایه نتایج نزدیک به آن ارائه داده‌اند. ضعیف‌ترین



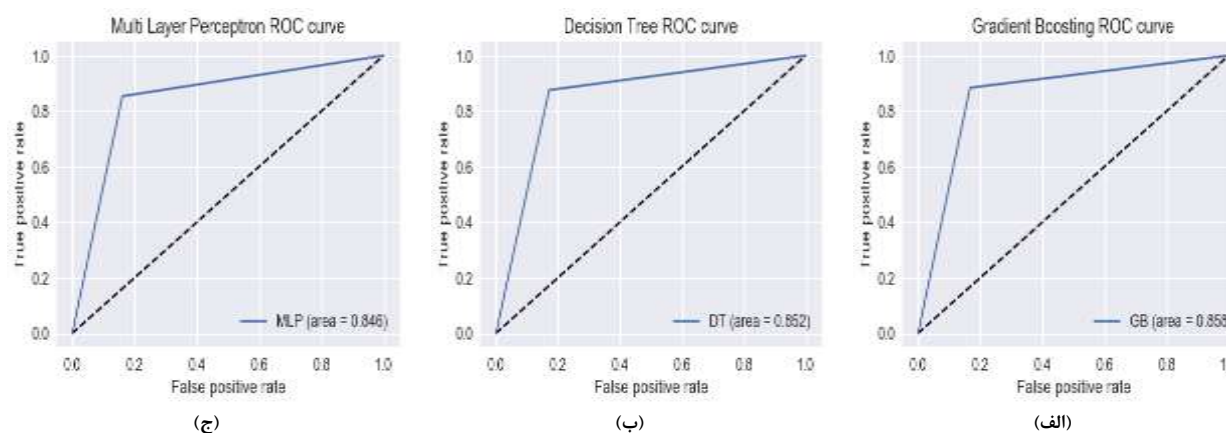
شکل (۵): نمودار مشخصه عملیاتی پذیرنده الگوریتم‌ها در تشخیص DGA

توانایی شناسایی تعداد دامنه‌های ناسالم بیشتری را نسبت به سایر الگوریتم‌ها داشته است. به عبارت دیگر عملکرد الگوریتم جنگل تصادفی در تشخیص دامنه‌های ناسالم بالاتر است.

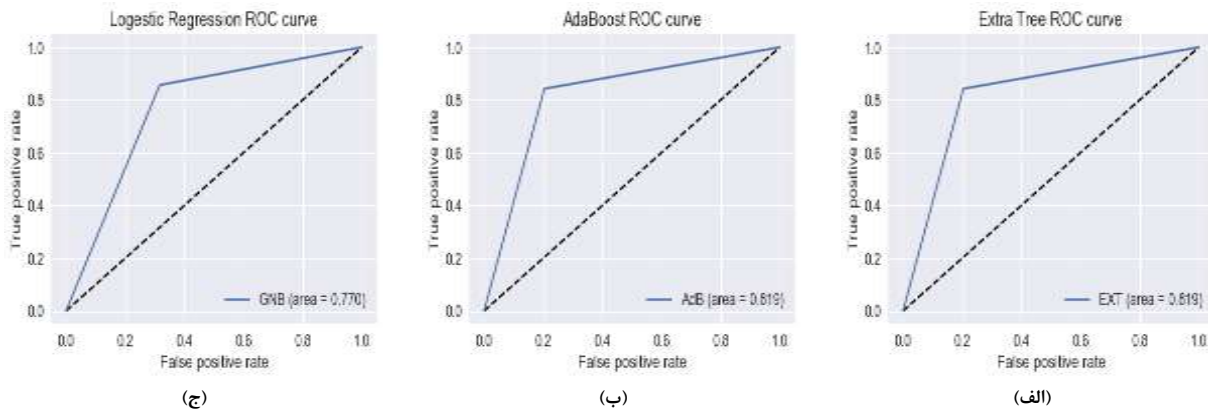
با توجه به شکل ۵ و مقادیر به دست آمده از سنجه مشخصه عملکرد پذیرنده می‌توان نتیجه گرفت با توجه به مقدار ۰/۸۸۹ که توسط الگوریتم جنگل تصادفی ارائه شده است این الگوریتم



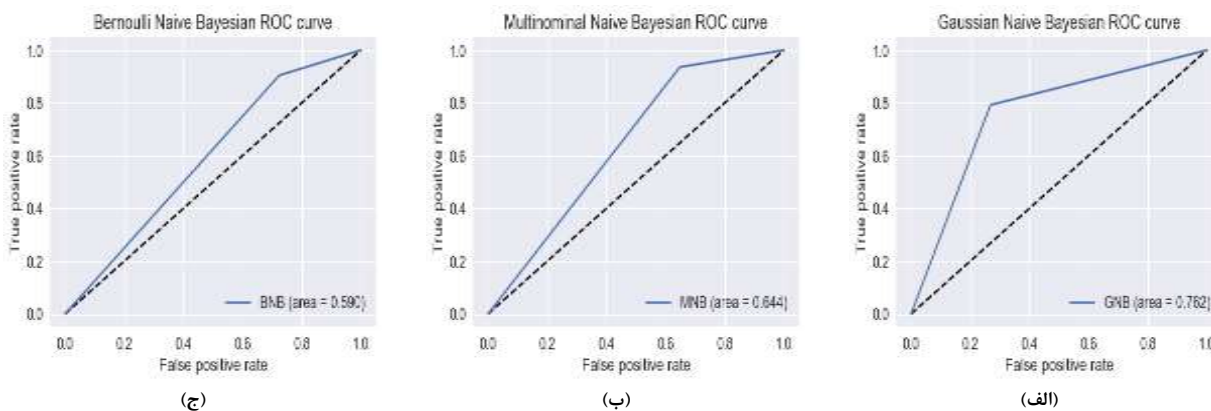
شکل (۶): منحنی مشخصه عملکرد پذیرنده (الف) جنگل تصادفی، (ب) K نزدیک‌ترین همسایه و (ج) ماشین بردار پشتیبان در تشخیص DGA



شکل (۷): منحنی مشخصه عملکرد پذیرنده (الف) تقویتی گرادیان، (ب) درخت تصمیم و (ج) چندلایه در تشخیص DGA



شکل (۸): منحنی مشخصه عملکرد پذیرنده (الف) درخت اضافی، (ب) تقویتی تطبیقی و (ج) رگرسیون منطقی در تشخیص DGA



شکل (۹): منحنی مشخصه عملکرد پذیرنده (الف) بی‌زی گاوسی، (ب) بی‌زی چندجمله‌ای و (ج) بی‌زی برنولی در تشخیص DGA

الگوریتم‌ها ارائه دادند. بنابراین، با توجه به این نقطه ضعف، الگوریتم K نزدیک‌ترین همسایه و الگوریتم ماشین بردار پشتیبان بر روی مجموعه داده استفاده‌شده در این مقاله مناسب نخواهند بود. مدت زمان آزمایش الگوریتم‌های تقویتی نیز نسبتاً زیاد بود ولی نتایج ارائه‌شده برای تشخیص دامنه‌های سالم و دامنه‌های ناسالم توسط این الگوریتم‌ها به نسبت سایر الگوریتم‌ها مناسب بود. الگوریتم شبکه عصبی چندلایه نیز با سرعت نسبتاً مناسبی در مراحل آموزش و آزمایش به نتایج خوبی دست یافت. هر چند الگوریتم‌های بی‌زی در تشخیص دامنه‌های سالم و دامنه‌های ناسالم عملکرد ضعیف‌تری نسبت به سایر الگوریتم‌های اشاره‌شده ارائه دادند ولی از نظر زمان آموزش و زمان آزمایش عملکرد بسیار بهتری نسبت به سایر الگوریتم‌ها داشتند.

۵- نتیجه‌گیری و کارهای آتی

کارهای متعددی در تشخیص شبکه‌های بات انجام‌یافته است، با این حال، اخیراً شبکه‌های باتی که از الگوریتم‌های تولید دامنه برای ارتباط مدیر بات با خدمت‌دهنده فرمان و کنترل خود استفاده می‌کنند افزایش چشمگیری داشته‌اند. استخراج ویژگی از مجموعه داده‌ها در ارتباط با شناسایی شبکه‌های بات استفاده

مشخصه عملیاتی پذیرنده سه الگوریتم در هر یک از شکل‌های (۶-۹) ارائه شده است که به ترتیب بهترین و بالاترین میزان دستیابی الگوریتم‌های مطرح‌شده در این سنجه ترسیم‌شده است. این منحنی‌ها و مقادیر سنجه مشخصه عملیاتی پذیرنده بیانگر قدرت تشخیص دامنه‌های ناسالم توسط هر یک از الگوریتم‌های یادگیری ماشین است.

۴-۳-۳- تحلیل نتایج

با بررسی کلی نتایج می‌توان توانایی الگوریتم‌های استفاده‌کننده از درخت‌های تصمیم در تشخیص دامنه‌های تولیدشده توسط الگوریتم‌های تولید دامنه و دامنه‌های ناسالم را مشاهده کرد. این الگوریتم‌ها عملکرد مناسبی را نسبت به سایر الگوریتم‌های یادگیری ماشین بررسی‌شده در تفکیک دامنه‌های سالم از دامنه‌های ناسالم ارائه دادند. زمان آموزش و آزمایش این الگوریتم‌ها نیز با توجه به نتایج حاصل مناسب بود.

در الگوریتم‌های ماشین بردار پشتیبان و K نزدیک‌ترین همسایه علی‌رغم دستیابی به نتایج مناسب در نرخ صحت، نرخ تشخیص و نرخ مثبت نادرست، مدت زمان آموزش و آزمایش بسیار بالا بوده و عملکرد زمانی بسیار ضعیف‌تری نسبت به سایر

- [8] M. Antonakakis, R. Perdisci, Y. Nadji, N. Vasiloglou, S. Abu-Nimeh, W. Lee, and D. Dagon, "From Throw-Away Traffic to Bots: Detecting the Rise of DGA-Based Malware," In Proceedings of the 21st USENIX Security Symposium, Bellevue, WA, USA, 8-10 August 2012.
- [9] D. Nhaou and K. Sung-Ryul, "Classification of malicious domain names using support vector machine and bi-gram method," J. Secur. Appl., vol. 7, pp. 51-58, 2013.
- [10] K. Demertzis and L. Iliadis, "Evolving smart URL filter in a zone-based policy firewall for detecting algorithmically generated malicious domains," In International Symposium on Statistical Learning and Data Sciences; Springer:Cham, Switzerland, pp. 223-233, 2015.
- [11] J. Hagen and S. Luo, "Why domain generation algorithms (DGA)?," Trend Micro, 18 August 2016.
- [12] Symantec, W32.Ramnit analysis, Version 1.0, 2015-02-24.
- [13] J. Geffner, "End-to-end analysis of a domain generating algorithm malware family," Black Hat USA, 2013.
- [14] C. E. Shannon, "A Mathematical Theory of Communication," Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, 1948.
- [15] M. Mohri, A. Rostamizadeh, and A. Talwalkar, "Foundations of machine learning," MIT press, 2018.
- [16] I. Rish, "An empirical study of the naive Bayes classifier," International Joint Conferences on Artificial Intelligence 2001 Workshop on Empirical Methods in Artificial Intelligence, pp. 41-46, 2001.
- [17] L. Rokach and O. Z. Maimon, Data mining with decision trees: theory and applications, 2008.
- [18] J. Harrell and E. Frank, "Regression modeling strategies: with applications to linear models, logistic and ordinal regression, and survival analysis," Springer, 2015.
- [19] D. Denisko and M. M. Hoffman, "Classification and interaction in random forests," in Proceedings of the National Academy of Sciences, vol. 115, no. 8, pp. 1690-1692, 2018.
- [20] C. Cortes and V. Vapnik, "Support vector networks," Machine Learning, vol. 20, no. 3, pp. 273-297, 1995.
- [21] G. Shakhnarovich, T. Darrel, and P. Indyk, "Nearest-neighbor methods in learning and vision: theory and practice (neural information processing)," The MIT press, 2006.
- [22] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Boosting Algorithms as Gradient Descent," Advances in Neural Information Processing Systems 12, MIT Press, pp. 512-518, 2000.
- [23] D. P. Solomatine and D. L. Shrestha, "AdaBoost. RT: a boosting algorithm for regression problems," 2004 IEEE International Joint Conference on Neural Networks, pp. 1163-1168, 2004.

کننده از الگوریتم‌های تولید دامنه بسیار حائز اهمیت است. در این مقاله سعی گردید با استفاده از استخراج سه نوع ویژگی (ساختاری، زبانی و آماری) از مجموعه داده‌های دارای انواع دامنه‌های سالم و ناسالم، یازده الگوریتم یادگیری ماشین نظارت شده و نیز شبکه عصبی بر اساس میزان رده‌بندی درست دامنه‌ها مقایسه و برتری هر کدام در سنج‌های ارزیابی تعیین شود. نتایج آزمایش‌ها نشان داد الگوریتم جنگل تصادفی دارای نرخ صحت و نرخ تشخیص بالاتری با نرخ مثبت نادرست پایین‌تری است.

مطالعات آتی در راستای گسترش و به کارگیری روش‌های انتخاب ویژگی‌های تأثیرگذار جهت افزایش کارایی و بالا بردن نرخ دقت و صحت تشخیص دامنه‌های ناسالم و همچنین کاهش خطای تشخیص در این الگوریتم‌ها خواهد بود. در کارهای آتی می‌توان از مهندسی ویژگی‌ها در جهت تولید خودکار ویژگی‌ها و نیز با استفاده روش‌های یادگیری عمیق، الگوریتم‌های تولید دامنه ناسالم و بدخواه را با نرخ دقت و صحت بالایی تشخیص داد.

۶- مراجع

- [1] S. Parsa, H. Mortazi, "Botnet Detection with Flow Behavior Analysis Approach," Journal of Electronical & Cyber Defence, vol. 5, no. 4, 2017. (In Persian)
- [2] S. Schiavoni, F. Maggi, L. Cavallaro, and S. Zanero, Phoenix, "DGA-based botnet tracking and intelligence," in: Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment (DIMVA), in: Lecture Notes in Computer Science, 8550, pp. 192-211, 2014.
- [3] J. Woodbridge, H. S. Anderson, A. Ahuja, and D. Grant, "Predicting domain generation algorithms with long short-term memory networks," CoRR abs/1611.00791. arXiv:1611.00791, 2016.
- [4] D. K. McGrath and M. Gupta, "Behind Phishing: An Examination of Phisher Modi Operandi," In Proceedings of the First USENIX Workshop on Large-Scale Exploits and Emergent Threats, LEET 08, San Francisco, CA, USA, 15 April 2008.
- [5] L. Bilge, E. Kirda, C. Kruegel, and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," In Proceedings of the Network and Distributed System Security Symposium, NDSS 2011, San Diego, CA, USA, 6-9 February 2011.
- [6] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: Learning to detect malicious web sites from suspicious URLs," In Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, pp. 1245-1254, 2009.
- [7] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated domain-flux attacks with DNS traffic analysis," IEEE/ACM Trans. Netw., vol. 20, pp. 1663-1677, 2012.

- [31] A. Cutler, D. R. Cutler, and J. R. Stevens, *Ensemble Machine Learning: Random Forests*, Springer US., 2012.
- [32] V. Thanuja, B. Venkateswarlu, and G. S. G. N. Anjaneyulu, "Applications of Data Mining in Customer Relationship Management", *J. Comp. & Math. Sci.*, vol. 2, pp. 423-433, 2011.
- [33] C. F. Tsai and M. Y. Chen, "Variable selection by association rules for customer churn prediction of multimedia on demand," *Expert Systems with Applications*, vol. 37, pp. 2006-2015, 2010.
- [34] J. R. Quinlan, *C4.5: Programs for machine learning*, San Francisco, CA: Morgan Kaufman, 1993.
- [35] J. A. Michael and S. L. Gordon, *Data Mining Technique: For Marketing, Sales and Customer Support*, Wiley, New York, 1997.
- [36] W. A. Au, K. C. C Chan, and X. Yao, "A Novel Evolutionary Data Mining Algorithm With Application to Churn Prediction," *IEEE Transactions on Evolutionary Computation*, vol. 7, pp. 532-545, 2003.
- [37] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Mach. Learn.*, vol. 63, no. 1, pp. 3-42, 2006.
- [24] J. H. Friedman, "Stochastic gradient boosting," *Computational Statistics & Data Analysis*, vol. 38, no. 4, pp. 367-378, 2002.
- [25] D. Kriesel, *A Brief Introduction to Neural Networks*, 2007. Retrieved from www.dkriesel.com/_media/science/neuronalenetze-en-zeta2-2col-krieselcom.pdf
- [26] M. Abadi et al., "Tensorflow: a system for large-scale machine learning," in *OSDI*, vol. 16, pp. 265-283, 2016.
- [27] F. Chollet, Keras, Accessed: 2017-05-28. [Online]. Available: <https://github.com/fchollet/keras>.
- [28] Alexa Top 1 Million Sites: The Alexa Top Sites web service provides access to lists of websites ordered by Alexa Traffic Rank.
(<https://www.kaggle.com/cheedcheed/top1m>)
- [29] Bambenek Consulting provided malicious algorithmically-generated domains.
(<http://osint.bambenekconsulting.com/feeds/dga-feed.txt>)
- [30] 360 Lab DGA Domains: A collection of domains generated by DGA and it is maintained by 360-a Chinese security vendor. This dataset keeps updated every day.
(<https://data.netlab.360.com/feeds/dga/dga.txt>)

Comparison of Supervised Machine Learning Algorithms in Detection of Botnets Domain Generation Algorithms

M. Asadi, M. A. Jabraeil Jamali*, S. Parsa, V. Majidnezhad

*Islamic Azad University, Shabestar Branch, Shabestar, Iran

(Received: 03/09/2019, Accepted: 01/02/2020)

ABSTRACT

Domain generation algorithms (DGAs) are used in Botnets as rendezvous points to their command and control (C&C) servers, and can continuously provide a large number of domains which can evade detection by traditional methods such as Blacklist. Internet security vendors often use blacklists to detect Botnets and malwares, but the DGA can continuously update the domain to evade blacklist detection. In this paper, first, using features engineering; the three types of structural, statistical and linguistic features are extracted for the detection of DGAs, and then a new dataset is produced by using a dataset with normal DGAs and two datasets with malicious DGAs. Using supervised machine learning algorithms, the classification of DGAs has been performed and the results have been compared to determine a DGA detection model with a higher accuracy and a lower error rate. The results obtained in this paper show that the random forest algorithm offers accuracy rate, detection rate and receiver operating characteristic (ROC) equal to 89.32%, 91.67% and 0.889, respectively. Also, compared to the results of the other investigated algorithms, the random forest algorithm presents a lower false positive rate (FPR) equal to 0.373.

Keywords: Botnet, Domain Generation Algorithms (DGAs), Machine Learning Algorithms, Blacklist, C-&-C Server