دوفصل‌نامه علمی منادی امنیت فضای تولید و تبادل اطلاعات (افتا)
سال ۱۴۰۱، جلد ۲۰، شماره ۱ (۸۳-۹۳)
https://monadi.isc.org.ir

ارائه‌شده در هجدهمین کنفرانس بین‌المللی انجمن رمز ایران در دانشگاه اصفهان، ۱۰ و ۱۱ شهریور ۱۴۰۰

ابداع روشی جهت نجات گره مه از نفوذ★

سیدامید آذرکسب*، سیدحسین خواسته و سعید صدیقیان کاشی

دانشکده مهندسی کامپیوتر، دانشگاه صنعتی خواجه نصیرالدین طوسی، تهران، ایران

اطلاعات مقاله

کلمات کلیدی:
رایانش مه
اینترنت اشیاء
ماشین بردار پشتیبان
حملات تزریق
رایانش ابری

dor: 20.1001.1.24763047.1401.11.1.10.7

نوع مقاله: پژوهشی

چکیده

مه، ابر نزدیک به زمین است. اجزای مه و ابر مکمل یکدیگر می‌باشند. این اجزا سرویس‌های وابسته به یکدیگر و با مزایای دو جانبه را، برای ایجاد ارتباطات، پردازش، کنترل و ذخیره‌سازی در سراسر شبکه فراهم می‌کنند. حمله به گره مه همانند حمله به ابر، از درجه اهمیت بالایی برخوردار است. از آنجایی که گره مه منابع محدودتری را در اختیار دارد بیشتر مورد توجه و هدف نفوذی‌ها قرار می‌گیرد. علاوه بر این، گره‌های مه برای مهاجمان جذاب‌تر هستند، زیرا آنها توان محاسباتی کمتری داشته و نسبت به ابر در مکان نزدیک‌تری به مهاجم قرار دارند. اما نکته کلیدی این است که دسترسی به منابع محدود، نجات گره مه را آسان‌تر می‌کند؛ زیرا مه پیچیدگی‌های ابر را نداشته و به راحتی می‌توان سیستم تشخیص نفوذ را بر روی آن اجرا کرد. ما در این مقاله با تمرکز بر محدودیت منابع در گره مه، به ابداع روشی برای نجات گره مه می‌پردازیم. در روش پیشنهادی از تکنیک ماشین بردار پشتیبان استفاده می‌شود. از مزایای استفاده از ماشین بردار پشتیبان می‌توان به گرفتار نشدن در دام بهینه‌های محلی، حل مسئله بیش‌برازش و سهولت در کار با داده‌های با ابعاد بالا اشاره داشت. بر اساس تحقیقات انجام‌شده، ماشین بردار پشتیبان بیشترین و پرکاربردترین روش یادگیری ماشین استفاده شده برای مقالات امنیتی اینترنت اشیاء، در ادبیات موجود است. در این مقاله جهت انجام آزمایش‌ها، طبق آمارهای جهانی منتشر شده، مهم‌ترین دسته حملات وب، یعنی حملات تزریق رخنه مورد توجه قرار می‌گیرد. میانگین دقت تشخیص به دست آمده و نتایج ارزیابی‌ها بیانگر کارایی قابل قبول روش پیشنهادی می‌باشد.

۱ مقدمه

توسعه‌های اینترنت اشیاء در مقیاس وسیع شرایطی را ایجاد می‌کنند که رایانش ابری به طور موثر قادر به کنترل کارآمد آن نیست. به دلیل امکان واکنش سریع در نزدیکی مولفه‌های لبه، بکارگیری رایانش مه برای کاربردهای اینترنت اشیاء رشد سریعی پیدا کرد. از مزایای رایانش مه می‌توان به کاهش پهنای باند مصرفی و همچنین کاهش تاخیر در شبکه اشاره کرد. سیستم ماشین خودکار هوشمند، ماشین‌های فروش هوشمند و سیستم تراشه هوشمند نمونه‌های عملی کاربرد رایانش مه در اینترنت اشیاء است. مفهوم رایانش مه یک دیدگاه جدید است که اینترنت اشیاء را قادر می‌سازد تا کاربردهای خود را در لبه شبکه اجرا کند [۱].

رایانش مه جایگزینی برای رایانش ابری نیست بلکه گسترش‌دهنده رایانش ابری است و آن را با مفهوم دستگاه‌های هوشمندی که می‌توانند بر روی لبه شبکه کار کنند تکمیل می‌کند. در واقع رایانش مه دروازه‌ای بین رایانش ابری و اینترنت اشیاء است. مه به عنوان یک گسترش از

★از کمیته علمی هجدهمین کنفرانس بین‌المللی انجمن رمز ایران برای داوری این مقاله تشکر می‌شود.
*نویسنده مسئول
آدرس‌های رایانامه: seyedomid.azarkasb@email.kntu.ac.ir (سیدامید آذرکسب)، khasteh@kntu.ac.ir (سیدحسین خواسته)، sedighian@kntu.ac.ir (سعید صدیقیان کاشی)

ابر است از این رو اجتناب‌ناپذیر است که برخی از چالش‌های امنیتی رایانش ابری در آن ادامه پیدا نکند.

در حالی که برخی از روش‌های موجود در زمینه رایانش ابری می‌توانند بسیاری از مسائل امنیتی و حریم خصوصی را در رایانش ابری حل کنند، رایانش مه چالش‌های امنیتی جدیدی را به خاطر ویژگی‌های متمایزش، از جمله محدودیت در بکارگیری منابع، به همراه دارد. این چالش‌ها در انطباق رایانش مه بر اینترنت اشیاء تاثیر می‌گذارد. گره‌های مه به کاربر اجازه می‌دهند تا برخی از داده‌ها را بدون نیاز به ارسال آنها به مرکز داده ابری مورد پردازش قرار دهد. در حالی که مراکز داده به منابع فراوانی مانند پردازنده‌ها، انرژی و حافظه مجهز هستند، ولی دستگاه‌های مجهز به این منابع فراوان نیستند. این امر بدین معنی است که روش‌های معمولی و سنتی برای پیشگیری از نفوذ در سطح مه مناسب نیستند، زیرا این روش‌ها باعث ایجاد تاخیر و یا مصرف انرژی بیشتری خواهند شد. بنابراین نیاز به یک سیستم امنیتی قوی وجود دارد که با مصرف مقدار کمی از منابع، لایه مه را از حمله‌ها محافظت نماید. روش‌های تشخیص نفوذ، سوء رفتار و یا وسایل مخرب اینترنت اشیاء را تشخیص داده و به دیگران در شبکه اطلاع می‌دهند تا اقدامات لازم انجام پذیرد. ماهیت محیط‌های اینترنت اشیاء تشخیص حملات را در سطح جهانی با مشکل مواجه می‌سازد.

محل خدمات رایانش ابر در اینترنت است، و محل خدمات رایانش مه در لبه شبکه محلی می‌باشد. به عبارتی امنیت رایانش مه را می‌توان تعریف کرد ولی در رایانش ابری نمی‌توان آن را تعریف کرد [۲]. اما می‌توان با تمرکز بر روی گره‌های لایه مه امنیت را در سطح ساده‌تری برقرار نمود. از طرفی مزایا و قابلیت‌های ارزشمند ماشین بردار پشتیبان، محققان و پژوهشگران را به سوی استفاده از آن جهت تشخیص نفوذ سوق داده است. از جمله این قابلیت‌ها می‌توان به طراحی دسته‌بندی کننده با حداکثر تعمیم، رسیدن به بهینه سراسری، تعیین خودکار ساختار و مکان‌شناسی بهینه برای طبقه‌بندی کننده، مدل کردن توابع تمایز غیرخطی با استفاده از هسته‌های غیرخطی و مفهوم حاصلضرب داخلی در فضاهای هیلبرت، و همچنین سهولت در کار با داده‌های با ابعاد بالا اشاره‌کرد.

طبق نتیجه‌گیری تحقیقات و آزمایش‌های انجام گرفته، ماشین بردار پشتیبان بیشترین و پرکاربردترین الگوریتم‌ها، تکنیک‌ها و روش‌های یادگیری ماشین استفاده شده برای مقالات امنیتی اینترنت اشیاء است که هم برای تشخیص نفوذ و هم برای احراز هویت مورد استفاده قرار گرفته است [۳]. از اینرو ما نیز در این مقاله از تکنیک ماشین بردار پشتیبان جهت نجات گره مه از نفوذگرها استفاده می‌کنیم. روش‌های مبتنی بر شبکه‌های عصبی، بیزین، درخت تصمیم و ژنتیک به ترتیب در رده‌های بعدی هستند [۴]. ایده اصلی ماشین بردار پشتیبان انتخاب یک جداکننده یکه است، به طوری که حاشیه جداکننده دو دسته را به حداکثر برساند. ماشین بردار پشتیبان دقت تشخیص مناسبی را بدون سربار محاسبه زیاد ارائه می‌دهد.

۲ پیشینه پژوهشی

مرجع [۵] برای بالابردن کارایی ماشین بردار پشتیبان یک‌دسته‌ای [1] برای تشخیص بی‌نظمی بدون ناظر، ماشین بردار پشتیبان مقاوم را معرفی می‌کند. ایده اصلی آن، کم اثر کردن نقش داده‌های پرت در مرز تصمیم‌گیری است. [۶] با آموزش سیستم مبتنی بر ماشین بردار پشتیبان یک دسته‌ای به وسیله رفتارهای نرمال، رفتارهای ناهنجار را جستجو می‌کند. [۷] با این فرض که داده نرمال با غیرنرمال مخلوط است، از ماشین بردار پشتیبان مقاوم برای حل مسأله بیش برازش [2] در داده‌های دارای بی‌نظمی استفاده کرده است. آزمایشات آنها نشان داد که تشخیص نفوذ بر پایه مدل پردازش متن، نرخ اخطار مثبت غلط زیادی را تولید می‌کند و در عمل به سختی می‌توان آن را به کار بست. [۸] ماشین بردار پشتیبان یک دسته‌ای لانه‌ای [3] را برای تشخیص نفوذ پیشنهاد می‌کند. در این روش به جای برچسب‌گذاری با معیارهایی مانند دقت میانگین هندسی، از اطلاعات دورترین و نزدیک‌ترین همسایگان هر نمونه استفاده می‌شود. نتایج تجربی نشان می‌دهد که این روش عملکرد بهتری نسبت به روش پایه‌ای آن داشته است. ماشین‌های بردار پشتیبان همواره با شبکه‌های عصبی مورد قیاس قرار گرفته است. [۹] ماشین بردار پشتیبان را با شبکه‌های عصبی مقایسه کرده است و نشان داده است که ماشین بردار پشتیبان با وزن‌دهی tf-idf بهترین جواب را می‌دهد، در حالی که شبکه عصبی با وزن‌دهی ساده بدترین جواب را تولید می‌کند. همچنین نشان می‌دهد که ماشین بردار پشتیبان با هسته گوسی نتایج بهتری را نسبت به شبکه‌های عصبی RBF ارائه می‌دهد. [۱۰] با استفاده از ترکیب انواع شبکه‌های عصبی و ماشین‌های بردار پشتیبان با هسته RBF، برای تقویت سیستم تشخیص‌نفوذ اقدام نموده است.

پیدایش و فراگیر شدن رایانش ابری و رایانش مه در دهه اخیر و ضرورت برقراری امنیت در آن، محققان و پژوهشگران را به سمت ارائه روش‌های تشخیص نفوذ در این بسترها سوق داده است. [۱۱] ماشین بردار پشتیبان باینری مبتنی بر تراکم را پیشنهاد می‌کند. ایده اصلی بر اساس تراکم هر کلاس در مجموعه داده‌ها است. داده‌های آموزشی به یک توالی باینری تبدیل می‌شوند. با توجه به این توالی مدل آموزشی رفتار سیستم به دست می‌آید. برای اجرا، از چارچوب محاسباتی موازی نگاشت کاهش [4] در Hadoop که توسط Apache برای پردازش داده‌های توزیع‌شده در مقیاس بزرگ به صورت قابل اعتماد ارائه شده است، استفاده می‌کند. [۱۲] با معرفی ماشین بردار پشتیبان بهینه‌شده، از ترکیب الگوریتم ژنتیک و ماشین بردار پشتیبان برای بالا بردن توانایی تعمیم طبقه‌بندی و شناسایی حملات انکار سرویس [5] در رایانش ابری استفاده کرده است. [۱۳] با تمرکز بر یادگیری عمیق یک سیستم ترکیبی طراحی کرده است که از یک رمزگذار خودکار انقباضی [6] انباشته برای کاهش ابعاد ویژگی‌ها و از الگوریتم طبقه‌بندی ماشین بردار پشتیبان برای تشخیص حملات

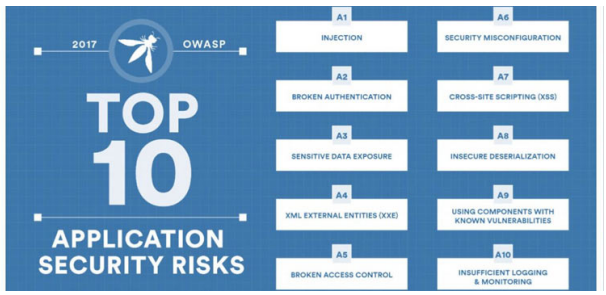
[1] one-class SVM [2] over fitting [3] nested set SVM [4] map reduce parallel computational framework [5] denail of service [6] stacked autoencoder

مخرب [1] استفاده می‌کند. [۱۴] از ماشین بردار پشتیبان برای طبقه‌بندی داده‌های شبکه به رفتار عادی و رفتار حمله و همچنین حذف ویژگی‌های بی‌ربط و زائد استفاده می‌کند. و در نهایت سیستم تشخیص تهاجمی را مبتنی بر ماشین بردار پشتیبان و بهره اطلاعاتی [2] معرفی می‌کند. [۷] ترکیبی از خوشه‌بندی فازی و ماشین بردار پشتیبان را برای بهبود دقت سیستم تشخیص در محیط رایانش ابری ارائه می‌دهد. در [۱۵] از ماشین بردار پشتیبان برای طبقه‌بندی اتصالات شبکه استفاده می‌شود. ماهیت ترافیک‌ها به مدیر شبکه اطلاع داده می‌شود تا هرگونه ارتباط متجاوز را به شبکه قطع و مسدود کند. در ادامه از بهینه‌سازی ازدحام ذرات باینری [3] برای انتخاب مرتبط‌ترین ویژگی‌های شبکه و از بهینه‌سازی ازدحام ذرات استاندارد برای تنظیم پارامترهای کنترلی ماشین بردار پشتیبان استفاده می‌شود. [۱۶] طرح تشخیص نفوذ ماشین بردار پشتیبان را مبتنی بر همکاری ابر و مه ارائه می‌دهد. در این طرح، از روش آنالیز مؤلفه اصلی [4] برای کاهش ابعاد، از بین بردن همبستگی بین ویژگی‌ها و همچنین کاهش زمان آموزش استفاده می‌شود. سرور ابر، از یک ماشین بردار پشتیبان بهینه‌شده توسط الگوریتم بهینه‌سازی ازدحام ذرات برای تکمیل عملیات آموزش مجموعه داده و دستیابی به طبقه‌بندی بهینه استفاده می‌کند. سپس نتایج به دست آمده به گره مه ارسال شده و عملیات شناسایی حمله در گره مه انجام می‌گردد. [۱۷] استفاده از IP را به همراه ماشین بردار پشتیبان جهت یک سیستم تشخیص نفوذ مبتنی بر شبکه پیشنهاد می‌دهد. استفاده از IP علاوه بر صرفه‌جویی درصد بالایی در مصرف حافظه، امکان ردیابی منبع واقعی بسته‌ها را در صورت وقوع حمله فراهم می‌کند. همانطور که دیدیم، به طور خلاصه تمام پیشینه‌های پژوهشی عنوان شده در بالا و سایر موارد عنوان نشده، دلالت بر نقش بسیار پررنگ استفاده از ماشین بردار پشتیبان در مباحث امنیتی اینترنت اشیاء، رایانش ابری و رایانش مه دارد.

۳ امنیت در مه

امنیت مه اگر به خطر بیافتد به طور مستقیم بر امنیت و اعتماد همه برنامه‌ها و کاربران تاثیر می‌گذارد. حساسیت گره‌های مه به دلیل محدودیت منابع آنها، اغلب نسبت به سرورهای ابری مانند دستگاه‌های IoT بیشتر است [۱۸]. همچنین سطوح حمله برای گره‌های مه وسیع‌تر است، زیرا آنها مستعد تزریق اطلاعات ناقص و بدافزار، دستکاری سرویس و نشت اطلاعات هستند. حمله به گره‌های مه می‌تواند خطرناک‌تر از حمله به دستگاه‌های IoT باشد، زیرا آنها معمولا اطلاعات خصوصی و نگرانی‌های مربوط به حفظ حریم خصوصی، و اعتماد به روابط را با تعداد بیشتری از گره‌های دیگر و موارد از راه دور را در اختیار دارند. ماهیت الگوی مه به طور طبیعی منجر به افزایش تهدیدهای ساختارهای سرکش و غیر قابل اعتماد می‌شود [۱۹]؛ زیرا آنها توان محاسباتی کمتری داشته و نسبت به ابر در مکان نزدیک‌تری به مهاجم قراردارند [۱۶]. پردازش مبتنی بر مه، یک

[1] malicious [2] information gain [3] binary particular swarm optimization
[4] principal component analysis



شکل ۱. آخرین گزارش بیشترین آسیب‌پذیری‌های وب در سال ۲۰۱۷

سیستم محاسباتی هوشمند است که در آن گره‌های موجود در مه می‌توانند به صورت مستقل درخواست‌های محاسباتی و پردازشی دستگاه‌های انتهایی موجود را پاسخ دهند و همچنین می‌توانند برای همکاری، به همدیگر متصل شوند. مدیریت و رویه‌های همکاری در گره‌های مه برای اعمال مدیریت و کنترل، اعمال می‌شوند. همکاری بین گره‌های مه می‌تواند از طریق ارتباطات راه دور یا ارتباطات محلی بین آنها انجام پذیرد [۲۰].

حملات وب، حملاتی هستند که از نقاط آسیب‌پذیری موجود در وب برای گذر از سیاست‌های امنیتی برنامه‌های کاربردی وب استفاده می‌کنند. حملات وب از پروتکل HTTP و یا HTTPS استفاده می‌کنند. پروتکل HTTP از درگاه [5] ۸۰ و پروتکل HTTPS از درگاه ۴۴۳ استفاده می‌کنند. به طور معمول حملات وب از این دو درگاه برای گذر از سیاست‌های اعمال شده در وب استفاده می‌کنند [۲۱]. اساساً همه آن چیزی که یک نفوذگر وب نیاز دارد، مرورگر وبی و اتصال اینترنتی است. آخرین گزارش رسمی منتشر شده در رابطه با فراوانی حملات وبی مربوط به سایت OWASP می‌باشد. طبق این گزارش، ده عدد از بیشترین آسیب‌پذیری‌های وب در سال ۲۰۱۷ به‌صورت شکل ۱ می‌باشد.

همانطورکه در شکل ۱ دیده می‌شود، حملات تزریق رخنه [6]، زیان‌بارترین حمله از نوع حملات وبی است. بر این اساس ما نیز در این مقاله، برای انجام آزمایش‌ها، حملات نوع تزریق را مورد توجه قرار می‌دهیم که در ادامه به معرفی آنها می‌پردازیم.

۱.۳ حملات تزریق رخنه

مهم‌ترین این نوع تزریق‌ها که تزریق SQL است، روشی است که یک نفوذی، یک برنامه کاربردی را در لایه پایگاه داده به اجرا می‌گذارد. تزریق هنگامی اتفاق می‌افتد که داده فراهم شده توسط کاربر به عنوان قسمتی از دستور یا پرس و جو به یک مفسر فرستاده می‌شود. مهاجمان مفسر را فریب داده، به اجرای دستورات برنامه‌ریزی نشده وا می‌دارند. تزریق رخنه به مهاجمان اجازه به وجود آوردن، خواندن، به‌روزرسانی یا پاک‌کردن هر داده دلخواه در دسترس برنامه کاربردی را می‌دهد. در بدترین حالت، این رخنه‌ها به مهاجمان اجازه می‌دهد برنامه کاربردی و سیستم‌های تحت آن را در خطر اکتشاف قرار داده، حتی از دیواره‌های آتش تودرتوی عمیق عبور کنند. همه چارچوب‌های برنامه‌های کاربردی وب که از مفسرها استفاده

[5] port [6] injection flows

می‌کنند یا پردازش‌های دیگر را احضار می‌کنند، نسبت به حملات تزریق آسیب‌پذیر هستند [۲۲]. انواع مختلف آسیب‌پذیری‌های تزریق عبارتند از [۲۳]:

پالایش کاراکترهای escape به طور نادرست: این‌نوع از آسیب‌پذیری تزریق، هنگامی اتفاق می‌افتد که ورودی کاربر از کاراکترهای escape پالوده نشده باشد. این کار موجب دستکاری‌های اساسی در دستورات اجرا شده بر روی پایگاه داده توسط کاربران نهایی برنامه کاربردی می‌شود.

مدیریت اشتباه چاپ: این نوع از تزریق هنگامی اتفاق می‌افتد که فیلد تهیه شده توسط کاربر به طور قوی نوشته نشده یا برای محدودیت‌های چاپ مورد امتحان قرار نگرفته باشد. این اتفاق هنگامی رخ می‌دهد که یک فیلد عددی در دستور مورد استفاده قرار گیرد، اما برنامه‌نویس هیچ بررسی انجام ندهد که تعیین کند داده تهیه شده توسط کاربر عددی است.

تزریق‌کور[1]: تزریق کور هنگامی استفاده می‌شود که یک برنامه کاربردی وب نسبت به تزریق آسیب‌پذیر است، ولی نتایج تزریق قابل مشاهده برای مهاجم نیست؛ صفحه‌ای که دارای آسیب‌پذیری ممکن است صفحه نشان‌دهنده داده نباشد، اما نتایج دستور منطقی تزریق‌شده به داخل دستورات قانونی فراخوانی شده برای آن صفحه را نشان دهد. این نوع حمله می‌تواند بسیار وقت‌گیر باشد، زیرا برای هر بیت بازیابی‌شده یک دستور جدید باید ارائه شود. ابزارهای زیادی وجود دارد که می‌تواند این حملات را، هنگامی که مکان آسیب‌پذیری و اطلاعات هدف مشخص باشد، به طور خودکار درآورد. انواع حملات تزریق‌کور بدین شرح هستند:

(۱) **پاسخ‌های شرطی:** یک نوع از تزریق کور است که پایگاه داده را مجبور می‌کند یک دستور منطقی را در صفحه یک برنامه کاربردی معمولی اجرا کند.

(۲) **خطاهای شرطی:** این نوع از تزریق‌کور با اجبار پایگاه داده به اجرای یک دستور، باعث خطا می‌شود. در واقع دستور یاد شده اگر دستور Where درست باشد، باعث خطا می‌شود.

(۳) **تأخیرهای زمانی:** تأخیرهای زمانی نوعی از تزریق‌کور هستند که باعث‌می‌شوند موتور پرس‌وجوی مداوم طولانی یا یک دستور تاخیر زمان را وابسته به منطق تزریق‌شده اجرا کند. سپس مهاجم می‌تواند مدت زمانی که صفحه بارگذاری شده را اندازه‌گیری کند تا تعیین کند که آیا دستور تزریق شده صحیح است یا خیر.

۴ جزئیات پیاده‌سازی

ویژگی‌ها و بردار ویژگی‌های استخراج شده در مرحله پیش پردازش، از مقادیر پارامترهای ارسال‌شده به هر گره به دست می‌آید که در ادامه معرفی می‌گردد.

جدول ۱. معادل مبنای ۱۶ کاراکترهای نفوذ

کاراکتر	--	=	%	;	!	.	)	(	'
معادل مبنای ۱۶	۲D۲D	۳D	۲۵	۳b	۲۱	۲e	۲۸	۲۹	۲۷

۱.۴ استخراج ویژگی‌ها

میانگین طول مقادیر پارامترهای درخواست: یکی از ویژگی‌های مهم استخراج شده، میانگین طول مقادیر پارامترهای درخواست می‌باشد. در حملات تزریق به دلیل استفاده مکرر از کاراکترهای خاص، طول مقادیر پارامترهای درخواست افزایش می‌یابد، در حالی که در درخواست‌های نرمال، طول مقادیر پارامترهای درخواست از یک مقدار مشخص تجاوز نمی‌کند. بنابراین می‌توان میانگین طول مقادیر پارامترهای درخواست را به عنوان یک ویژگی مهم برای تمایز بین درخواست‌های نرمال و تهاجم در نظر گرفت.

میانگین توزیع کاراکتری پارامترهای درخواست: برای ویژگی میانگین توزیع کاراکتری پارامترهای درخواست از کاراکترهای ۳۳ تا ۹۶ و از ۱۲۳ تا ۱۲۶ جدول اسکی یعنی جمعاً ۶۸ کاراکتر استفاده شده است. مجموعه کاراکترهای انتخاب شده، کاراکترهای قابل چاپ هستند که اغلب در یک متن نوشتاری به کار می‌روند. کاراکترهای ۹۷ تا ۱۲۲ حروف a تا z به صورت کوچک هستند و چون این حروف به صورت بزرگ از شماره ۶۵ تا ۹۰ یعنی در مجموعه گفته شده وجود دارند، به‌منظور کاهش بعد، حذف شده‌اند. با این شرط، در صورت مشاهده کاراکترها به‌صورت کوچک، معادل بزرگ آنها در نظر گرفته می‌شود. برای هر کاراکتر، درصد حضور آن کاراکتر نسبت به طول مقدار پارامتر محاسبه می‌شود. سپس برای هر کاراکتر میانگین این درصد حضورها در کل پارامترهای یک درخواست، محاسبه می‌شود و به عنوان یک فیلد به بردار ویژگی اضافه می‌گردد.

میانگین توزیع معادل مبنای ۱۶ کاراکترهای خاص: گاهی نفوذگران به جای استفاده از کاراکتر، از معادل مبنای ۱۶ آن کاراکتر استفاده می‌کنند. این اتفاق معمولاً در مورد کاراکترهایی می‌افتد که جهت نفوذ استفاده می‌شوند. اگر چه استفاده از معادل مبنای ۱۶ یک کاراکتر به تنهایی به‌معنای نفوذ نیست، ولی معمولاً نشان‌دهنده نوعی ناهنجاری است. این ناهنجاری به این معناست که چون به طور معمول یک کاربر عادی از کاراکترهای طبیعی استفاده می‌کند، استفاده از معادل مبنای ۱۶ کاراکتر می‌تواند نشان دهنده نوعی رفتار ناهنجار باشد. برای تشخیص چنین ناهنجاری، معادل مبنای ۱۶ کاراکترهای مهمی که به طور عمده در این حمله استفاده می‌شود، نیز در نظر گرفته شده است. استفاده از مبنای ۱۶ برای دیگر کاراکترها معمول نیست. از کاراکترهای جدول ۱ برای نفوذ استفاده می‌شود که معادل مبنای ۱۶ آنها نوشته شده است.

حضور کلمات کلیدی در درخواست: مشکل اصلی در حمله تزریق، استفاده از پرس و جوها از طریق وب برای دریافت یا تغییر اطلاعات حساس و محرمانه ذخیره شده در پایگاه داده می‌باشد. این پرس و جوها می‌تواند هم در URL و هم در بدنه یک درخواست قرارگیرد برای جلوگیری از اجرای این حمله، نباید به کاربر اجازه استفاده از دستورات را در URL

[1] Blind SQL Injection

جدول ۲. مثالی از اختصاص بیت به حضور کلمات کلیدی در یک درخواست

Select	Insert	Update	Delete	Execute	From	Where	AND	OR	Having
۱	۰	۰	۰	۰	۱	۱	۰	۰	۱

و یا در بدنه داد. نکته‌ای که در اینجا باید به آن اشاره کرد این است که این ویژگی در کل درخواست در نظر گرفته شده و در هر مقدار پارامتر به طور جداگانه محاسبه نشده است. علت این کار این است که معمولاً این گونه کلمات در یک یا بیشتر از مقادیر پارامترها مشاهده نمی‌شود و در عمل میانگین گرفتن از آن، احتمال حضور یک کلمه کلیدی را کاهش می‌دهد. بنابراین ویژگی حضور کلمات کلیدی برای کل درخواست محاسبه شده است. بدین منظور مهم‌ترین کلمات کلیدی استفاده شده در حمله تزریق به صورت زیر در نظر گرفته می‌شود.

$$Keywords\_SQLInjection = \{\text{Select Insert Update Delete Execute Where AND OR Having}\}$$

به ازای هر کلمه کلیدی یک بیت در نظر گرفته می‌شود، به طوری که در صورت حضور آن کلمه کلیدی عدد ۱ و در صورت عدم حضور آن، عدد ۰ به مقدار بیت داده می‌شود. با کنار هم قرار گرفتن این بیت‌ها در مبنای ۲ یک عدد به دست می‌آید. این عدد به عنوان مقدار ویژگی حضور کلمات کلیدی به بردار ویژگی اضافه می‌شود. برای‌مثال اگر در یک حمله تزریق از کلمات select، from، wher e و having استفاده شده باشد، مقدار این فیلد به صورت جدول ۲ می‌شود.

عدد به دست آمده $(1000011001)_2$ است که با تبدیل آن به مقدار دهدهی عدد ۵۳۷ به دست می‌آید. این عدد به عنوان ویژگی حضور کلمات کلیدی به بردار ویژگی اضافه می‌شود. بدیهی است با جابجا کردن ترتیب کلمات کلیدی، اعداد مختلفی به دست می‌آید. ترتیب این کلمات به صورت نشان داده شده در جدول ۲ است. علت این انتخاب، ترتیب تقریبی استفاده از این کلمات در پرس و جوها می‌باشد. به طوری که دریک پرس و جو، ابتدا کلماتی مثل select، update و غیره قرار می‌گیرد، سپس کلماتی مثل where و در پایان کلماتی مثل having وارد می‌شود. این ترتیب برای به دست آوردن یک مقدار دودویی منطقی با اختصاص بیت به هر کلمه رعایت شده است. برای هر کلمه از این مجموعه نیز یک بیت در نظر گرفته شده، در صورت ظهور آن کلمه مقدار بیت عدد ۱ و در غیر این صورت مقدار بیت ۰ می‌گردد. سپس این مقدار دودویی به یک عدد دهدهی تبدیل می‌شود. عدد دهدهی به دست آمده از این تبدیل، به عنوان مقدار ویژگی حضور کلمات کلیدی به بردار ویژگی اضافه می‌گردد.

ترتیب دوتایی کاراکترها: ویژگی ترتیب کاراکترها به این معنی است که به نحوی بتوان ترتیب کاراکترها را به صورت یک مقدار عددی به بردار ویژگی اضافه کرد. این ویژگی برای مقادیر پارامترهای یک درخواست به طوری که ترکیب‌های دوتایی کاراکترها در مقدار یک پارامتر در نظر گرفته شود، محاسبه می‌شود. مثلاً اگر مقدار guessme از پارامتر password را در نظر بگیرید، ترکیب‌های gu، ue، es، ss، sm و me باید در نظر گرفته شوند، به این معنا که مثلا در مقدار یک پارامتر، ترتیب کاراکتری g قبل از u دیده شده است. برای به دست آوردن چنین ترتیبی می‌توان به هر ترکیب دوتایی کاراکترها در مقدار پارامتر یک بیت یا پرچم اختصاص داد. در صورت ظهور یک ترتیب کاراکتری مشخص، مقدار بیت ۱ و در غیر این صورت ۰ می‌شود. همانطور که در پیشتر بیان شد با توجه به این که در توزیع کاراکتری مقدار پارامتر از ۶۸ کاراکتر استفاده شده است، برای ترتیب‌های دوتایی با توجه به اینکه تکرار دو کاراکتر یکسان پشت سر هم حذف می‌شود، $68 * 67$ یعنی ۴۵۵۶ حالت یا بیت امکان ظهور دارد. برای پیاده‌سازی ۴۵۵۶ حالت، با احتساب این که هر عدد صحیح دارای ۳۲ بیت است، نیاز به اضافه کردن ۱۴۳ عدد به بردار ویژگی است. اضافه‌کردن ۱۴۳ عدد به بردار ویژگی باعث ایجاد سربار محاسباتی زیادی می‌شود و عملاً غیرممکن است. برای حل این مشکل تعداد ظهور هر ترتیب کاراکتری در کلیه درخواست‌ها محاسبه شده، و در صورتی که آن ترتیب هیچگاه ظاهر نشده باشد حذف می‌گردد. سپس برای ترتیب‌هایی که حداقل یک بار ظاهر شده باشند، ترتیب‌هایی که تعداد ظهور آنها از میانگین تعداد ظهور کلیه ترتیب‌ها کمتر باشد نیز حذف می‌شوند. به این ترتیب تعداد حالات تقریباً به یک سوم یعنی ۱۶۰۰ حالت کاهش می‌یابد. برای پیاده‌سازی این ۱۶۰۰ حالت با احتساب اینکه هر عدد صحیح نیاز به ۳۲ بیت دارد، از ۵۰ عدد ۳۲ بیتی استفاده شده است. بنابراین ۵۰ عدد ۳۲ بیتی به نحوی که هر بیت نماینده یک ترتیب دوتایی از کاراکترها است، به بردار ویژگی اضافه می‌گردد.

ترتیب پارامترها: ترتیب پارامترها در احضارهای قانونی برنامه‌های سمت سرور، حتی هنگامی که برخی پارامترها حذف شوند، معمولا حفظ می‌گردد، در حالی که این ترتیب در حملات تزریق لزوماً وجود ندارد. ویژگی ترتیب پارامترها به همین منظور به بردار ویژگی اضافه گردیده است. برای این منظور از دو عدد صحیح ۳۲ بیتی یعنی جمعاً ۶۴ بیت استفاده شده است. به ازای هر ترتیب دوتایی ظاهرشده از مقادیر پارامترها یک بیت در نظر گرفته می‌شود، در صورت ظهور آن ترتیب دوتایی مقدار بیت ۱ و در غیر این صورت مقدار بیت ۰ می‌گردد. تعداد کل حالت‌ها همواره یکی از تعداد کل پارامترها کمتر است و انتخاب ۶۴ بیت نیز به همین منظور صورت گرفته است. بنابراین به ازای هر درخواست، دو عدد به بردار ویژگی اضافه می‌گردد که هربیت در این دو عدد، نشان‌دهنده ظهور یا عدم ظهور یک ترتیب از پارامترهای آن درخواست است.

۲.۴ استخراج بردار ویژگی‌ها

با توجه به توضیحات داده شده، بردار ویژگی‌های استخراج شده به صورت جدول ۳ در می‌آید. همانطورکه در جدول ۳ دیده می‌شود، بردار ویژگی‌های به دست آمده برای حملات تزریق دارای ۱۳۲ فیلد می‌باشد.

۳.۴ ملاحضات فنی پیاده‌سازی

استفاده از مجموعه داده‌ای که دارای خصوصیات جامعیت، صحت و ترافیک لازم را باشد، یک مسأله اساسی در آموزش و تست سیستم تشخیص نفوذ است. بهترین انتخاب، استفاده از داده‌های جمع‌آوری شده

جدول ۳. بردار ویژگی‌های استخراج شده

ترتیب پارامترها	ترتیب دوتایی کاراکترها	حضور کلمات کلیدی در درخواست	میانگین توزیع معادل مبنای ۱۶ کاراکترهای خاص	میانگین توزیع کاراکتری پارامترهای درخواست	میانگین طول مقادیر پارامترهای درخواست	ویژگیهای استخراج شده
۲	۵۰	۱	۱۰	۶۸	۱	تعداد فیلدها

از سرورهای وبی است که قرار است مورد محافظت قرار گیرند. اما این داده‌ها معمولاً به دلایل امنیتی نمی‌تواند در اختیار محققان برای مقایسه نتایج الگوریتم‌های مختلف قرار گیرد. این مسأله باعث گردیده برخی محققان از داده‌های باز ولی کمتر واقعی، و یا از مجموعه داده‌های بسته ولی شبه واقعی استفاده کنند. معروف‌ترین مجموعه داده باز مورد استفاده برای تشخیص نفوذ، مجموعه داده‌های DARPA/MIT می‌باشد. از طرفی این مجموعه داده بدون انتقاد نیست. مشکلات برشمرده‌شده ما را بر آن داشت تا خود نسبت به تولید و ساخت مجموعه داده اقدام نماییم. برای این منظور تلاش کردیم با بکارگیری ابزارها و روش‌هایی که در ادامه توضیح داده می‌شود، مجموعه داده‌ای بسازیم که دارای سه اصل جامعیت، صحت و ترافیک لازم باشد. در این خصوص از نرم‌افزار WebInspecct استفاده شده است. WebInspect دقیق‌ترین و جامع‌ترین راه حل تشخیص آسیب‌پذیری برنامه‌های کاربردی و سرورهای وب می‌باشد. با استفاده از WebInspect مدیران و کاربران می‌توانند به آسانی و به سرعت برنامه‌های کاربردی و سرورهای وب خود را برای پیدا کردن آسیب‌پذیری مورد بررسی قرار دهند. نحوه تعیین آسیب‌پذیری توسط این نرم‌افزار بدین گونه است که هدف تعیین‌شده را با حملات متعدد مورد تهاجم قرار می‌دهد و در صورتی که آسیب‌پذیری تشخیص داده شد، آن را گزارش می‌دهد. مهم‌ترین ویژگی‌های این نرم‌افزار بدین شرح است: امکان اجرای پردازش پویش و ممیزی به صورت مجزا و همزمان، گزارش‌گیری سازمان‌یافته، کنترل حمله به صورت دستی، ارائه خلاصه وضعیت سیستم، امکان تغییر و تصحیح سیاست‌های پیمایش، صفحه نمایش مشاهده ترافیک و امکان انتخاب حملات مختلف وبی.

برای تهیه داده حمله، یک برنامه کاربردی با خصوصیات ویژه را به عنوان هدف حمله در نظرگرفته، و با استفاده از ابزار تهاجم قوی WebInspect به آن حمله می‌کنیم. فایل‌های ثبت شده در سرور نشان‌دهنده رکوردهایی هستند که می‌توان با اطمینان به آنها برچسب حمله زد. برای این کار ابتدا سیستم شخصی خود را با استفاده از XAMMP تبدیل به یک سرور وب می‌کنیم. XAMMP یک بسته نرم‌افزاری است‌که بر روی کامپیوتر، سرور آپاچی و پایگاه داده MySQL را نصب می‌کند. سپس از یک برنامه کاربردی وب به نام Evilboard به عنوان هدف حمله استفاده می‌کنیم. در Evilboard کاربران می‌توانند برای یکدیگر پیام فرستاده و با هم گفتگو کنند. به طور خلاصه دلایل انتخاب آن به عنوان هدف حمله شامل موارد زیر است:

جدول ۴. میزان داده‌های تولید شده

میزان داده حمله	میزان داده نرمال
۶۳۹۴۲۸	۷۵۶۳۴۷

- قابلیت ارتباط کاربران با یکدیگر به عنوان یک انجمن است و همان طور که قبلاً گفتیم این گونه برنامه‌ها محل‌های مناسبی برای حملات تزریق هستند.
- تعداد زیاد اتصالات این برنامه که در نتیجه قابلیت پویش بیشتر توسط WebInspect را برای ما ایجاد می‌کند.

۵ پیاده‌سازی، ارزیابی سیستم و تحلیل نتایج

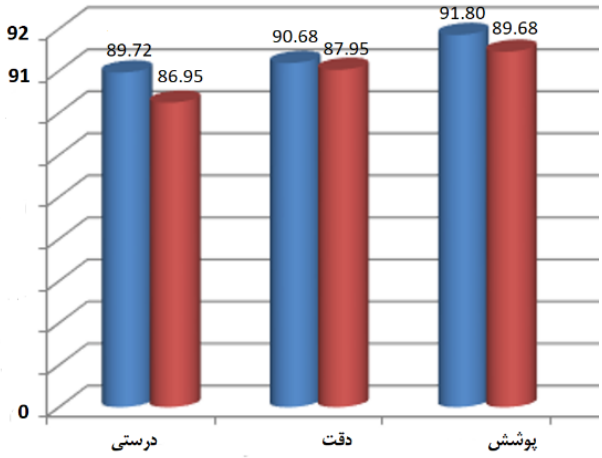
۱.۵ پیاده‌سازی

جهت پیاده‌سازی سیستم پیشنهادی، بر روی گره‌ای که تبدیل به سرور وب شده است، Evilboard را نصب می‌کنیم و در WebInspect هدف حمله را Evilboard معرفی می‌کنیم. برای تهیه یک مجموعه داده مناسب ابتدا با استفاده از حالت پویش در WebInspect کلیه اتصالات Evilboard را مورد پویش قرار می‌دهیم، سپس فایل ثبت‌شده را پاک کرده از حالت ممیزی برای فرستادن تهاجم به Evilboard استفاده می‌کنیم، به طوری که نوع حمله را مشخص کرده باشیم. در این حالت می‌توان مطمئن بود که فایل ثبت به دست آمده فقط دارای رکوردهایی با برچسب حمله می‌باشد. برای تهیه داده نرمال از یک شبکه محلی که بر روی سرور گره آن Evilboard نصب شده، استفاده شده است. کاربران این شبکه محلی افراد انتخاب شده‌ای هستند که از طریق این برنامه با هم در ارتباط می‌باشند. ترافیک ایجاد شده بر روی این شبکه در طول یک هفته جمع‌آوری شده و به عنوان داده نرمال مورد استفاده قرار گرفت. از آنجایی که این شبکه با هیچ شبکه خارجی مثل اینترنت و غیره در ارتباط نبوده است، می‌توان اطمینان داشت که هیچ داده حمله‌ای درآن وجود ندارد. از مجموعه داده‌های حمله و نرمال به دست آمده، ده درصد آن به عنوان داده تست و بقیه به عنوان داده‌آموزشی مورد استفاده قرار گرفته است. جدول ۴ میزان داده‌های به دست آمده را نشان می‌دهد.

پس از تهیه مجموعه داده، مرحله پیش پردازش با استفاده از نرم‌افزار Matlab بر روی داده‌های نرمال و داده حملات، انجام گرفته و ویژگی‌های مورد نیاز برای تشخیص حملات تزریق استخراج می‌گردد، به طوری که به ازای هر درخواست یک بردار ویژگی بر اساس ویژگی‌های تعریف شده به دست می‌آید. این بردارهای ویژگی به مؤلفه تحلیلگر داده می‌شود. در مؤلفه تحلیلگر، داده‌های به دست آمده از مرحله قبل برای آموزش به SVMLight داده می‌شود. SVMLight یک پیاده‌سازی از ماشین بردار پشتیبان به زبان C می‌باشد. این بسته نرم‌افزاری را می‌توان در یونیکس و همچنین محیط‌های دیگر از جمله ویندوز اجرا کرد. برای استفاده از این بسته نرم‌افزاری باید فایل‌های اجرایی آن را ساخته، فرمت داده‌های آموزش و تست را به صورتی درآورد که قابل استفاده توسط نرم‌افزار

جدول ۵. نتایج به دست آمده برای هسته‌های مختلف

نوع هسته	درستی	پوشش	دقت
RadialBasisFuncion $K(x,y)=\exp(-1/2\sigma^2 2\|x-y\|^2)$	89.72	91.80	90.68
Gaussian $k(x,y)=\exp(-(x-y)^2/\delta^2)$	86.95	89.68	87.95



شکل ۲. نتایج به دست آمده برای هسته‌های مختلف

باشد. با استفاده از برنامه LightDataAgent به راحتی می‌توان داده‌های آموزشی را به فرمت مورد نیاز SVMLight تبدیل کرد. یکی از مهمترین ویژگی‌های SVMLight، امکان انتخاب هسته‌های مختلف برای ماشین بردار پشتیبان است.

۲.۵ ارزیابی

برای ارزیابی سیستم پیشنهادی، از معیارهای درستی[1]، دقت[2] و پوشش[3] تعریف شده به کمک ماتریس ابهام[4]، استفاده شده است. نتایج به دست آمده به ازای هسته‌های مختلف ماشین بردار پشتیبان به صورت جداگانه در جدول ۵ و نمودار شکل ۲ نشان داده شده است. مقادیر نشان داده شده در این جدول، همگی با مقادیر پارامترهای پیش‌فرض به دست آمده‌اند. برای هسته RBF مقدار پیش‌فرض پارامتر گاما برابر با $1/k$ است که k تعداد ویژگی‌هاست.

برای هسته RBF آزمایشات با مقادیر مختلف پارامتر گاما انجام گرفته است. نتایج به دست آمده در جدول ۶ مشاهده می‌شود. برای هسته گوسی نیز آزمایشاتی با مقادیر مختلف پارامتر پهنای باند در هسته گوسی انجام شده است که در جدول ۷ دیده می‌شود. بهترین نتایج به دست آمده مربوط به مقدار ۰٫۵ برای پارامتر گاما می‌باشد و همان طور که مشاهده می‌شود با افزایش مقدار گاما نتیجه به دست آمده بهبود یافته است. همچنین مقایسه نتایج به دست آمده با مقادیر مختلف پارامتر پهنای باند در هسته گوسی نشان داد که بهترین نتیجه مربوط به مقدار ۲ برای این پارامتر است که در جدول ۷ مشاهده می‌شود.

در ادامه نمودارهای ۳، ۴ و ۵ دقت، پوشش و درستی برای مقادیر

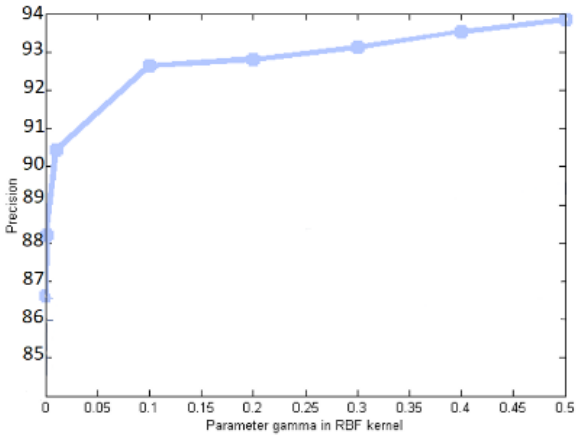
[1]Accuracy [2]Precision [3]Recall [4]Confusion Matrix

جدول ۶. نتایج به دست آمده با مقادیر مختلف پارامتر گاما برای هسته RBF

مقدار پارامتر گاما در هسته RBF	درستی	پوشش	دقت
0.0001	87.58	87.96	86.87
0.001	88.66	89.71	88.46
0.01	89.72	91.80	90.68
0.1	92.68	94.77	92.89
0.2	92.86	94.96	92.96
0.3	92.94	94.91	93.37
0.4	93.49	95.56	93.79
0.5	93.88	95.72	93.99

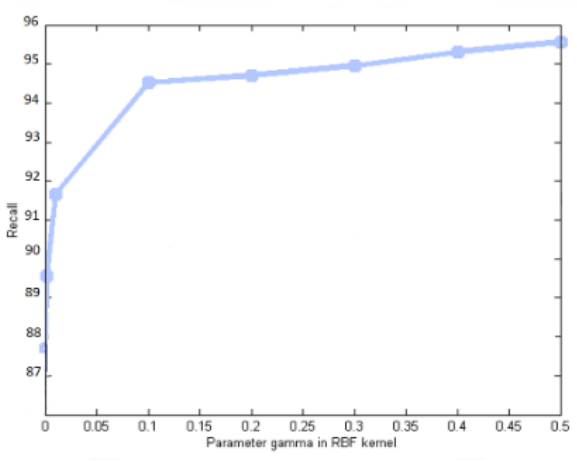
جدول ۷. نتایج به دست آمده با مقادیر مختلف پارامتر پهنای باند

مقدار پارامتر گاما در هسته گوسی	درستی	پوشش	دقت
0.0001	81.40	85.43	83.71
0.001	85.56	86.67	85.96
0.01	86.45	88.48	86.95
0.1	87.67	89.71	88.87
0.5	87.78	89.86	88.95
1	87.99	89.93/	88.98
1.3	87.99	89.94	88.96
1.5	88.37	90.26	89.37
1.8	88.56	90.51	89.56
2	88.68	90.78	89.64

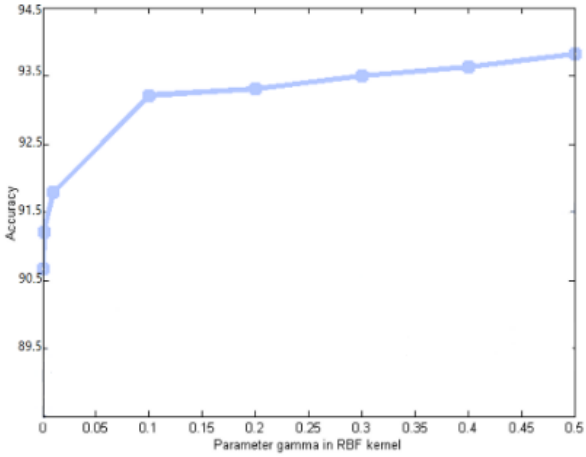


شکل ۳. نتایج معیار دقت به ازای مقادیر مختلف پارامتر گاما برای هسته RBF

مختلف گاما نشان داده شده است. نمودارهای مربوط به مقادیر مختلف پارامتر پهنای باند نشان نیز تقریباً از همین روند متابعت می‌کنند که به علت محدودیت در تعداد صفحات مقاله از آوردن آنها خودداری می‌شود.



شکل ۴. نتایج معیار پوشش به ازای مقادیر مختلف پارامتر گاما برای هسته RBF



شکل ۵. نتایج معیار درستی به ازای مقادیر مختلف پارامتر گاما برای هسته RBF

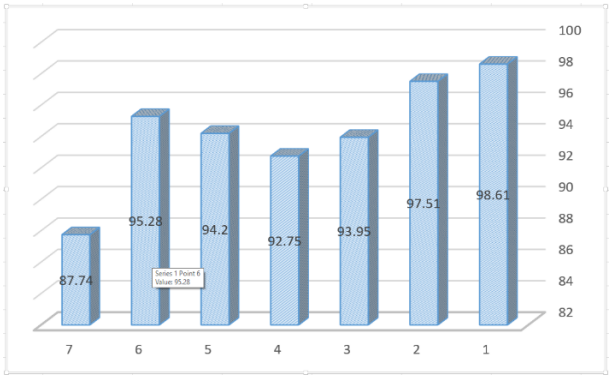
۳.۵ تحلیل نتایج

همانطور که در جداول ۶ و ۷ دیده می‌شود، بهترین نتایج به دست آمده جهت معیارهای دقت، پوشش و درستی مربوط به هسته RBF مربوط به هسته گوسی است. در ادامه جهت ارزیابی بهتر روش پیشنهادی، برخی از انواع مختلف ماشین‌های بردار پشتیبانی که پیشتر در رابطه با آنها صحبت شد مورد توجه قرار گرفت. قضاوت صحیح نیاز به آن دارد که همه روش‌ها از یک مجموعه داده یکسان استفاده کرده باشند و برای تشخیص حملات یکسانی به کار رفته باشند. لذا کلیه روش‌ها بر اساس الگوریتم ارائه شده در مراجع‌شان پیاده‌سازی و با داده‌های یکسانی مورد آزمایش قرار گرفت. همچنین جهت انجام آزمایش‌های در محیط رایانش ابری و رایانش مه از مجازی‌سازی استفاده شد. با این رویکرد سعی شد سه اصل جامعیت، صحت و برخوداری از ترافیک لازم رعایت شود. میانگین دقت حاصل از روش پیشنهادی در مقایسه با میانگین دقت حاصل از سایر روش‌ها در جدول ۸ ارائه گردیده است.

همانطورکه درجدول ۸ و نمودار ۶ دیده می‌شود روش پیشنهادی در مقایسه با سایر روش‌ها نتیجه قابل قبولی ارائه داده است. نتایج حاصله

جدول ۸. مقایسه میانگین دقت حاصل از روش پیشنهادی با میانگین دقت حاصل از سایر روش‌های مبتنی بر انواع ماشین‌های بردار پشتیبان

نوع روش	میانگین دقت
۱-روش پیشنهادی	۹۵٫۲۸
۲-ترکیب ماشین بردار پشتیبان، آنالیز مولفه اصلی و بهینه‌سازی ازدحام ذرات در رایانش مه [۱۶]	۹۴٫۲۰
۳-ترکیب ماشین بردار پشتیبان با یادگیری عمیق در رایانش‌ابری [۱۳]	۹۲٫۷۵
۴-ترکیب ماشین بردار پشتیبان با بهره اطلاعاتی در رایانش‌ابری [۱۴]	۹۳٫۹۵
۵-ماشین بردار پشتیبان یک دسته‌ای در رایانش‌ابری [۶]	۹۷٫۵۱
۶-ترکیب ماشین بردار پشتیبان با خوشه‌بندی فازی در رایانش ابری [۲۴]	۹۸٫۶۱
۷- استفاده از ماشین بردار پشتیبان به همراه لحاظ کردن IP وارد شوندگان به سیستم در یک سیستم تشخیص نفوذ مبتنی بر شبکه [۱۷]	۸۷٫۷۴



شکل ۶. مقایسه میانگین دقت حاصل از روش پیشنهادی با میانگین دقت حاصل از سایر روش‌های مبتنی بر انواع ماشین‌های بردار پشتیبان

حاکی از عملکرد خوب سیستم پیشنهادی است. دلیل آن هم استخراج مناسب ویژگی‌ها و بردار ویژگی‌های ارائه شده می‌باشد. همچنین بکارگیری هسته مناسب و مقادیر متناسب حاصله از نتیجه آزمایش‌های انجام گرفته برای مقادیر پارامترهای وابسته، از دیگر دلایل بالا بودن کارایی روش پیشنهادی می‌باشد. به دلیل استفاده از یک پایگاه داده واقعی و شرایط یکسان برای تمام روش‌ها، می‌توان صراحتاً گفت که نتایج به دست آمده به واقعیت نزدیک است و ارزیابی دقیقی انجام گرفته است.

۶ بحث و نتیجه‌گیری

همان طور که برای کاهش پهنای باند مصرفی و همچنین کاهش تاخیر در شبکه، از رایانش ابری به سمت رایانش مه پیش می‌رویم برقراری امنیت گره‌های مه به یک مسئله جدی تبدیل می‌شود. مضاف بر این، گره‌های مه به‌دلیل محدودیت در منابع‌شان و همچنین به دلیل نزدیکی‌شان به مهاجمان، بیشتر در معرض خطر آسیب‌پذیری مهاجمان قراردارند. قطع به یقین می‌توان گفت پیاده‌سازی روشی که بتواند گره‌های مه را از خطر مهاجمان نجات دهد می‌تواند امنیت را در کل سیستم برقرار نماید. امنیت

مه، امنیت ابر است. از طرفی نجات گره مه آسان‌تر است، زیرا که از منابع محدودتری نسبت به ابر برخوردار است و کنترل و برقرارای امنیت در آن، پیچیدگی‌های برقراری امنیت در رایانش ابری را ندارد. کاهش پیچیدگی معماری سیستم در مه، کلید موفقیت برنامه‌های کاربردی IoT است. در این مقاله جهت نجات گره‌های مه ازدست مهاجمان، استفاده از تکنیک ماشین بردار پشتیبان مورد توجه قرار گرفت. علت توجه به آن نیز سهولت در کار با داده‌های با ابعاد بالا، طراحی دسته‌بندی کننده با حداکثر تعمیم، رسیدن به بهینه سراسری تابع هزینه، تعیین خودکار ساختار و مکان‌شناسی بهینه برای طبقه‌بندی‌کننده می‌باشد. طبق مستندات ارائه شده در مقالات معتبر، ماشین بردار پشتیبان از پرکاربردترین و کاراترین الگوریتم‌های یادگیری ماشین بکارگرفته شده در مباحث امنیتی رایانش‌های اخیر می‌باشد. به دلیل رعایت اصول جامعیت، صحت، و برخورداری از ترافیک به‌روز، از نرم‌افزار WebInspect با هدف قراردادن Evilboard استفاده شد. مهمترین کار در پیش‌پردازش، استخراج ویژگی‌های مناسب است. ویژگی‌های استخراج‌شده شامل میانگین طول مقادیر پارامترهای درخواست، میانگین توریع کاراکتری پارامترهای درخواست، میانگین توزیع معادل مبنای ۱۶ کاراکترهای خاص، حضور کلمات کلیدی، ویژگی ترتیب دوتایی کاراکترها و ویژگی ترتیب پارامترها بودند. با توجه به‌ویژگی‌های استخراج‌شده، بردار ویژگی‌ها به دست آمد. برای پیاده‌سازی مولفه تحلیلگر از SVMLight استفاده شد. که در آن با استفاده از LightDataAgent داده‌ها به فرمت خوانا توسط SVMLight تبدیل، و برای تعیین دسته مناسب به SVMLight داده شد. در مؤلفه پاسخ‌دهنده، در صورتی که خروجی دریافتی از تحلیلگر نشانگر یک حمله باشد، اطلاعات مربوط به ترافیک موردنظر همراه با خروجی حاصله از تحلیلگر، ثبت و اخطار مناسب تولید می‌گردد.

برای ارزیابی سیستم پیشنهادی، معیارهای دقت، درستی و پوشش با استفاده از اجزای ماتریس ابهام معرفی گردید. نتایج به دست آمده به ازای هسته‌های مختلف ماشین بردار پشتیبان به صورت جداگانه مورد ارزیابی قرار گرفت. بهترین نتایج به دست آمده شامل معیارهای دقت، پوشش و درستی مربوط به هسته RBF است. مقایسه نتایج به دست آمده با مقادیر مختلف پارامتر گاما در هسته RBF نشان داد که بهترین نتیجه مربوط به مقدار ۰٫۵ برای این پارامتر است. همچنین مقایسه نتایج به دست آمده با مقادیر مختلف پارامتر پهنای باند در هسته گوسی نشان داد که بهترین نتیجه مربوط به مقدار ۲ برای این پارامتر است. در پایان جهت ارزیابی بیشتر روش پیشنهادی، میانگین دقت حاصل از روش پیشنهادی با میانگین دقت حاصل از بکارگیری انواع روش‌های ماشین‌های بردار پشتیبان مقایسه گردید. نتایج به دست آمده بیانگر کارایی بالاتر روش پیشنهادی در مقایسه با سایر روش‌ها می‌باشد. دلیل آن هم تعریف ویژگی‌های مناسب جهت استخراج ویژگی‌ها و در پی آن استخراج موثر بردار ویژگی‌ها و استفاده توامان امکانات موجود در رایانش مه می‌باشد. کاهش تعداد حالات از ۴۵۵۶ حالت به ۱۶۰۰ حالت نیز از دیگر دلایل بالا رفتن کارایی روش پیشنهادی می‌باشد. با این کار سربار محاسباتی به شدت کاهش می‌یابد.

مراجع

[1] Xingshuo An, Xianwei Zhou, Xing Lü, Fuhong Lin, and Lei Yang. Sample selected extreme learning machine based intrusion detection in fog computing and mec. *Wireless Communications and Mobile Computing*, 2018, 2018.

[2] Luis M Vaquero and Luis Rodero-Merino. Finding your way in the fog: Towards a comprehensive definition of fog computing. *ACM SIGCOMM computer communication Review*, 44(5):27–32, 2014.

[3] Darko Andročec and Neven Vrček. Machine learning for the internet of things security: a systematic. In *13th International Conference on Software Technologies*, volume 4120, pages 563–570, 2018.

[4] Basant Subba, Santosh Biswas, and Sushanta Karmakar. Enhancing performance of anomaly based intrusion detection systems through dimensionality reduction using principal component analysis. In *2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pages 1–6. IEEE, 2016.

[5] Mennatallah Amer, Markus Goldstein, and Slim Abdennadher. Enhancing one-class support vector machines for unsupervised anomaly detection. In *Proceedings of the ACM SIGKDD workshop on outlier detection and description*, pages 8–15, 2013.

[6] Ahmed M Mahfouz, Abdullah Abuhussein, Deepak Venugopal, and Sajjan G Shiva. Network intrusion detection model using one-class support vector machine. In *Advances in Machine Learning and Computational Intelligence*, pages 79–86. 2020.

[7] Wenjie Hu, Yihua Liao, and V Rao Vemuri. Robust anomaly detection using support vector machines. In *Proceedings of the international conference on machine learning and applications*. Los Angeles, California, USA, 2003.

[8] Quoc Thong Nguyen, Kim Phuc Tran, Philippe Castagliola, Truong Thu Huong, Minh Kha Nguyen, and Salim Lardjane. Nested one-class support vector machines for network intrusion detection. In *2018 IEEE Seventh International Conference on Communications and Electronics (ICCE)*, pages 7–12. IEEE, 2018.

[9] Rung-Ching Chen and Su-Ping Chen. Intrusion detection using a hybrid support vector machine based on entropy

model. *Future Generation Computer Systems*, 85:235–249, 2018.

[20] PeiYun Zhang, MengChu Zhou, and Giancarlo Fortino. Security and trust issues in fog computing: A survey. *Future Generation Computer Systems*, 88:16–27, 2018.

[21] Naing Naing Kyaw. Analysis and simulation of hyper text transfer protocol at the application layer of the internet. *International Journal of Scientific and Research Publications*, 9(1):78–84, 2019.

[22] Inyong Lee, Soonki Jeong, Sangsoo Yeo, and Jongsub Moon. A novel method for sql injection attack detection based on removing sql query attribute values. *Mathematical and Computer Modelling*, 55(1-2):58–68, 2012.

[23] Chandershekhar Sharma and SC Jain. Analysis and classification of sql injection vulnerabilities and attacks on web applications. In *2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014)*, pages 1–6. IEEE, 2014.

[24] Aws Naser Jaber and Shafiq Ul Rehman. Fcm–svm based intrusion detection system for cloud computing environment. *Cluster Computing*, 23(4):3221–3231, 2020.

and tf-idf. *International Journal of Innovative Computing, Information, and Control (IJICIC)*, 4(2):413–424, 2008.

[10] AM Chandrashekhar and K Raghuveer. Fortification of hybrid intrusion detection system using variants of neural networks and support vector machines. *International Journal of Network Security & Its Applications*, 5(1):71–90, 2013.

[11] Mingyuan Yu, Shuhang Huang, Qing Yu, Yan Wang, and Jiaquan Gao. A density-based binary svm algorithm in the cloud security. *International Journal of Security and Its Applications*, 9(7):153–162, 2015.

[12] M Mayuranathan, M. Murugan, and V. Dhanakoti. An intrusion detection system using optimized svm for detecting ddos in cloud. In *International Journal of Scientific & Technology Research*, volume 8, 2019.

[13] Wenjuan Wang, Xuehui Du, Dibin Shan, Ruoxi Qin, and Na Wang. Cloud intrusion detection method based on stacked contractive auto-encoder and support vector machine. *IEEE transactions on cloud computing*, 2020.

[14] Emmanuel Mugabo and Qiu-Yu Zhang. Intrusion detection method based on support vector machine and information gain for mobile cloud computing. *Int. J. Netw. Secur.*, 22(2):231–241, 2020.

[15] Mahmoud M Sakr, Medhat A Tawfeeq, and Ashraf B El-Sisi. Network intrusion detection system based pso-svm for cloud computing. *International Journal of Computer Network and Information Security*, 11(3):22–29, 2019.

[16] Ruizhong Du, Xiaoyan Liang, and Junfeng Tian. Support vector machine intrusion detection scheme based on cloud-fog collaboration. In *International Conference on Security and Privacy in New Computing Environments*, pages 321–334, 2020.

[17] Pynbianglut Hadem, Dilip Kumar Saikia, and Soumen Moulik. An sdn-based intrusion detection system using svm with selective logging for ip traceback. *Computer Networks*, 191:108015, 2021.

[18] Keke Gai, Meikang Qiu, Zhong Ming, Hui Zhao, and Longfei Qiu. Spoofing-jamming attack strategy using optimal power distributions in wireless smart grid networks. *IEEE Transactions on Smart Grid*, 8(5):2431–2439, 2017.

[19] Riccardo Rapuzzi and Matteo Repetto. Building situational awareness for network threats in fog/edge computing: Emerging paradigms beyond the security perimeter

MonadiAfta-Bianual Scientific Journal for Cyberspace Security
2022, Volume 19, Number 3 (pp. 83–93)
https://monadi.isc.org.ir

Presented at the ISCISC 2021 in University of Isfahan, Isfahan, Iran

Inventing a Method to Save the Fog Node from Attacks★

Seyed Omid Azarkasb *, Seyed Hossein Khasteh and Saeed Sedighian Kashi

Software Engineering Faculty, K.N. Toosi University of Technology, Tehran, Iran

ARTICLE INFO.

Keywords:
fog computing
cloud computing
intrusion detection
SQL injection flaw attacks
support vector machines
internet of things

dor: 20.1001.1.24763047.1401.11.1.10.7

Type: research paper

ABSTRACT

Fog is a cloud that closes to the ground. The components of fog and cloud complement each other. These components provide mutually beneficial interdependent services for communication, processing, control, and storage across the network. Attacking the fog nodes are as important as attacking the cloud. Since the fog node has more limited resources, it is more targeted by intruders. In addition, fog nodes are more attractive to attackers because they have less computing power and are located closer to the attacker than the cloud. But the key point is that access to limited resources makes it easier to save the fog node because the fog does not have the complexities of the cloud, and it is easy to run an intrusion detection system on it. In this article, focusing on the resource limitation in the fog node, we will invent a method to save the fog node. In the proposed method, the support vector machines (SVMs) technique is used. Among the advantages of using the support vector machine, we can mention not being trapped in local optima, solving the over fitting problem, and ease of working with high-dimensional data. Based on the research, support vector machine is the most widely used machine learning method for Internet of Things security articles in the literature. In this article, in order to conduct tests, according to published global statistics, the most important category of web attacks, i.e. SQL injection attacks, is considered. The average detection accuracy is obtained and the results of the evaluations indicate the acceptable efficiency of the proposed method.

★ The ISCISC 2021 Program Committee effort is highly acknowledged for reviewing this paper.
* Corresponding author
Email addresses: seyedomid.azarkasb@email.kntu.ac.ir (Seyed Omid Azarkasb), khasteh@kntu.ac.ir (Seyed Hossein Khasteh), sedighian@kntu.ac.ir (Saeed Sedighian Kashi)