تشخیص حملات شبکه‌های کامپیوتری با یادگیری ماشین و تحلیل داده‌های جریان ترافیک

هادی ویسی[1*]، سیدهادی موسوی[2]، محمد خوانساری[3]
۱- استادیار- دانشکده علوم و فنون نوین- دانشگاه تهران- تهران- ایران
h.veisi@ut.ac.ir
۲-دانشجوی دکتری- دانشکده علوم و فنون نوین- دانشگاه تهران- تهران- ایران
s.hadi.mousavi@ut.ac.ir
۳- استادیار- دانشکده علوم و فنون نوین- دانشگاه تهران- تهران- ایران
s.hadi.mousavi@ut.ac.ir

چکیده: با گسترش کاربردهای فناوری اطلاعات، هر روزه خدمات ب**ی**شتری برروی بستر شبکه‌های کامپیوتری ارائه می‌گردد که به همین نسبت تهدیدات امنیتی این سامانه‌ها با اهداف خراب‌کارانه و **ی**ا تجاری توسعه یافته است. یکی از روش‌هایی که می‌توان از پیچیدگی تحلیل کل ترافیک کم کرد، تحلیل خلاصه داده‌های مربوط به جریان ترافیک به جای کل ترافیک می‌باشد NetFlow. از استانداردهای تولید داده‌های جریان ترافیک است که داده‌های خلاصه از جریان‌های ترافیک شبکه را به صورت خودکار توسط مسیریاب‌ها و سوئیچ‌های سیسکو تولید می‌نماید. در این مقاله رویکرد مبتنی بر یادگیری ماشین برای تحلیل ترافیک و دسته‌بندی آن به منظور شناسایی ترافیک‌های مربوط به حملات و انجام اقدامات پیشگیرانه، ارائه شده‌است. برای این کار، از الگوریتم‌های مختلف یادگیری ماشین شامل بیز ساده (Naive Bayes) ، ماشین بردار پشتیبان (SVM) و درخت تصمیم بیز (NBTree) برای مدل‌سازی داده‌های خلاصه جریان ترافیک استفاده شده است. برای ارزیابی روش‌های ارائه شده از مجموعه داده KDDcup99 استفاده شده است که قبل از استفاده در الگوریتم‌های مربوطه، ویژگی‌های مربوط به خلاصه جریان ترافیک از آن استخراج شده (۷ ویژگی) و الگوریتم‌های دسته‌بندی مذکور هم بر روی همان ویژگی‌ها و هم بر روی همه ویژگی‌های موجود در داده‌ها (۴۱ ویژگی) اجرا شده‌اند. متوسط دقت دسته‌بندی برای دسته‌های مختلف (۲۲ دسته حمله و یک دسته ترافیک نرمال) نشان می‌دهد که استفاده از ۷ ویژگی کارایی را زیاد تغییر نمی‌دهد اما محاسبات را به میزان چشمگیری کاهش می‌دهد. متوسط دقت روش‌ها بیشتر از ۹۷٪ بوده و در بهترین حالت (روش SVM با ۴۱ ویژگی)، متوسط دقت بیشتر از ۹۹٪ است.

واژه‌های کلیدی: تشخیص حملات شبکه، داده‌های جریان ترافیک ، دسته‌بندی، یادگیری ماشین.

تاریخ ارسال مقاله: ۹۹/۰۳/۰۶
تاریخ پذیرش مقاله: ۹۹/۰۴/۰۹
نام نویسنده مسئول: هادی ویسی*

۱- مقدمه

در شبکه‌های بزرگ که پهنای باند شبکه بالا بوده و تعداد گره‌های شبکه نیز بالاست و یا در مراکز توزیع‌کنندگان دسترسی به اینترنت، استفاده از سامانه‌های تشخیص نفوذ (IDS)[1] برخط، نیازمند منابع سخت‌افزاری بسیار بالاست و عملا در شبکه‌های بزرگ درسطح یک بنگاه و یا کشور این روش غیرقابل استفاده می‌باشد. راه‌حل پیشنهادی در این شرایط استخراج خلاصه‌ای از ترافیک شبکه و تحلیل آن به منظور شناسایی حملات و نفوذها به شبکه می‌باشد. یک دسته از داده‌های خلاصه‌ای که از ترافیک شبکه می‌توان استخراج کرد، داده‌های جریان‌های شبکه می‌باشد. امروزه استفاده از داده‌های جریان شبکه کاربردهای متعددی از جمله نظارت بر میزان مصرف منابع شبکه و شناسایی گلوگاه‌های آن، اقدامات قضایی و سایرکاربردهای امنیتی از قبیل شناسایی حملات پیدا کرده است[۱،۲].

یکی از استانداردهایی که امروزی در رابطه با داده‌های جریان‌های ترافیک شبکه، رایج است، استاندارد NetFlow می‌باشد[۳]. این استاندارد که توسط شرکت سیسکو ارائه شده است، برروی کلیه محصولات و تجهیزات شبکه‌ای سیسکو نیز پیاده‌سازی شده است. سوئیچ‌ها و مسیریاب‌های سیسکو می‌توانند به نحوی پیکربندی شوند تا اطلاعات جریان‌های مبادله شده در شبکه را به صورت برخط به آدرس دلخواه ارسال کرده تا تحلیل‌ها و ذخیره سازی این داده‌ها در سرورهای مخصوص به آن، صورت پذیرد. داده‌های NetFlow دو دسته کاربرد مهم و اساسی دارند، یکی استفاده به منظور مشاهده وضعیت استفاده از منابع و زیرساخت‌های شبکه و دیگری به منظور اقدامات جرم‌شناسی می‌باشد. کاربرد دیگری که در این مقاله مورد بررسی قرار می‌دهیم، کاربرد امنیتی به منظور شناسایی حملات و نفوذ در شبکه می‌باشد. این کاربرد اخیرا محبوبیت بیشتری پیدا کرده است [۴]. سامانه‌های تشخیص نفوذی که به صورت برخط حملات و نفوذ به شبکه را تشخیص می‌دهند، نیازمند تحلیل برخط ترافیک شبکه هستند. در شبکه‌های بزرگ تحلیل برخط نیازمند منابع سخت‌افزاری سنگینی است و در شبکه‌های بزرگ‌تر در مقیاس WAN[2] امری نسبتا غیرممکن خواهد بود. لذا می‌توان با پذیرش درصد معقولی از خطا به جای تحلیل کل ترافیک، داده‌های جریان ترافیک را مورد ارزیابی قرار داد تا حملات و کاربردهای موجود در شبکه را شناسایی کرد. در [۵] پژوهشگران ضمن ارائه یک چارچوب مقیاس‌پذیر از داده‌های Netflow به منظور تشخیص سرورهای کنترل و فرماندهی در شبکه بات استفاده کرده‌اند.

تحلیل داده‌های جریان ترافیک شبکه یکی از موضوع‌های پژوهشی مهم است و اخیرا تحقیقات گسترده‌ای در این حوزه انجام شده است. Vargas و همکاران از یک شبکه بیزین برای دسته‌بندی ترافیک ناهنجار با استفاده از تحلیل داده های جریان ترافیک پرداخته است تا با استفاده از آن بتواند کرم های فعال درشبکه و حملات جست و جوی فراگیر[3] را شناسایی کند[۶]. در[۷] ترافیک غیرنرمال و یک‌طرفه با استفاده از داده‌های جریان ترافیک شناسایی شده است و [۸]به دسته‌بندی ترافیک و شناسایی ترافیک شبکه‌های نقطه به نقطه با استفاده از الگوریتم دسته‌بندی ماشین بردار پشتیبان (SVM)[4] در داده‌های جریان ترافیک پرداخته است. پژوهشگران و توسعه‌دهندگان در[۹] به دنبال توسعه یک سامانه تشخیص نفوذ مبتنی بر داده‌های جریان شبکه به جای تحلیل کل ترافیک هستند. در[۱۰] با استخراج یک سری قواعد موجود در ترافیک نرمال درشبکه سازمانی خاص، روشی برای شناسایی ترافیک غیرنرمال با استفاده از داده‌های جریان ترافیک ارائه شده است.

در این مقاله، رویکرد مبتنی بر یادگیری ماشین برای استفاده از داده‌های جریان ترافیک به منظور شناسایی انواع مختلف حملات ارائه شده است. به توجه به محدودیت داده آموزشی برای الگوریتم‌های دسته‌بندی، در این کار از مجموعه داده KddCup99 [11], [12] استفاده شده است و ویژگی‌های مربوط به جریان ترافیک از آن استخراج شده است. سپس استفاده از الگوریتم‌های دسته‌بندی SVM، بیز ساده (NB[5]) و درخت تصمیم بیز ساده (NBTree) به منظور دسته‌بندی ترافیک مبتنی بر این ویژگی‌ها مورد ارزیابی قرار گرفته است.

ساختار مقاله در ادامه به این صورت می‌باشد. ابتدا در بخش دوم به معرفی ادبیات موضوع، به ویژه بررسی استاندارد Netflow و ساختار داده‌های آن به عنوان یک استاندارد رایج در رابطه با داده‌های جریان ترافیک پرداخته می‌شود و پس از آن کاربردهای عمومی داده‌های جریان ترافیک را مورد بحث قرار خواهیم داد. سپس، در بخش سوم ایده پیشنهادی استخراج و استفاده از داده‌های جریان ترافیک به منظور تشخیص نفوذ را بیان خواهیم کرد که در آن روش‌های جمع‌آوری داده جریان ترافیک را نیز مرور می‌کنیم. پس از آن، در بخش چهارم دادگان و نتایج ارائه

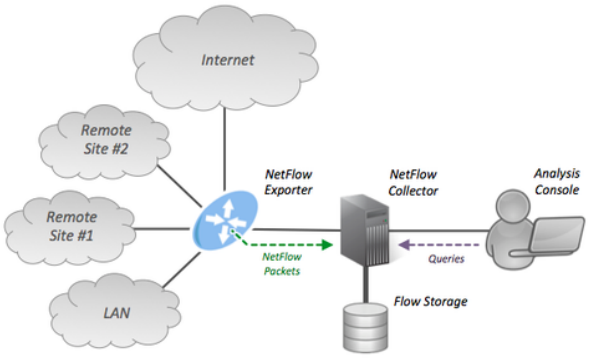
[1] Intrusion Detection System
[2] Wide-Area Network (WAN)
[3] Brute force
[4] Support Vector Machine (SVM)
[5] Naïve Bayes

می‌شود. به منظور ارزیابی ایده پیشنهادی از مجموعه داده KDDCUP99 استفاده شده که ویژگی‌های آن در بخش بعدی بیان شده است. نهایتاً بخش پنجم به تحلیل و جمع‌بندی کار صورت گرفته می‌پردازد.

۲- ادبیات موضوع

با توجه به اینکه مبنای این پژوهش استفاده از استاندارد Netflow است، در این بخش به معرفی آن و ساختار داده‌های آن می‌پردازیم. همان‌گونه که قبلا بیان شد،NetFlow استاندارد تولید داده‌های جریان ترافیک شبکه توسط محصولات سیسکو می‌باشد. یک مسیریاب و یا سوئیچ سیسکو می‌تواند این اطلاعات را از داده‌هایی که مبادله می‌نماید، استخراج نموده و در قالب رکوردهای Netflow به سرور مشخصی جهت ذخیره‌سازی و تحلیل بیشتر داده‌ها ارسال کند[۳]. شکل(۱)، معماری کلی این استاندارد را نشان می‌دهد. استاندارد Netflow سه مولفه اساسی، تولید رکوردها از ترافیک و ارسال آن‌ها، جمع‌آوری و ذخیره‌سازی داده‌های Netflow و تحلیل داده‌ها را در بر می‌گیرد.

مسیریاب‌ها و سوئیچ‌هایی که از این پروتکل پشتیبانی می‌کنند، می‌توانند اطلاعات آماری درخصوص بسته‌های IP مبادله شده بر روی کلیه رابط‌های پیکره‌بندی شده خود را جمع‌آوری کرده و درقالب رکوردهای Netflow به جمع‌آوری کننده ارسال کنند. به علاوه، بسته‌های نرم‌افزاری همچون F-Probe نیز توانایی استخراج داده‌های جریانی از ترافیک رسیده به کارت‌شبکه را دارا هستند[۱۳].



شکل ۱ - معماری NetFlow (Cisco, 2019)

ساختار رکوردها در استانداردهای مختلف تفاوت‌های جزئی دارد اما مهم‌ترین فیلدهای موجود شامل بازه زمانی مربوط به جریان، برچسب زمانی شروع جریان، IP مبدا و مقصد جریان، پروتکل لایه انتقال مربوط به مبدا و مقصد و همچنین شماره پورت‌های مبدا و مقصد (برای پروتکل TCP/UDP در لایه انتقال)، تعداد بسته‌های IP مبادله شده و حجم کل داده‌های مربوط به آن جریان می‌باشد. نحوه تولید این رکوردها به این صورت است که اولین بسته IPای که بین دو گره در شبکه مبادله می‌شود، یک رکورد برای آن ایجاد می‌شود. پس از آن کلیه بسته‌های IP که فیلدهای مبدا، مقصد، شماره پورت مبدا و شماره پورت مقصد یکسانی داشته باشند، در این رکورد تجمیع شده و تعداد بسته‌ها و حجم داده مبادله شده و مدت‌زمان جریان به ازای هر بسته به‌روز رسانی می‌شود. اگر فاصله زمانی بین دو بسته مربوط به یک جریان از یک مدت‌زمان مشخص بیشتر باشد (معمولا ۱۵ ثانیه)، برای بسته‌های بعدی رکورد جدیدی ایجاد خواهد شد. ضمن این که حداکثر بازه زمانی برای یک جریان نیز معمولا ۳۰ دقیقه است.

طبق این روش به ازای هر ارتباط[1] بین برنامه کاربردی مبدا و مقصد دو جریان ایجاد خواهد شد، یک جریان حاوی اطلاعات آماری مربوط به جریان رفت و دیگری مربوط به جریان برگشت. ابزارهایی برای تجمیع داده‌های دو جریان به صورت نرم‌افزاری وجود دارد که[۹] از جمله آنهاست. در این ابزار، علاوه بر تبدیل رکوردهای یک‌طرفه به دوطرفه، می‌توان سرور و کلاینت هر جریان را نیز با استفاده از الگوریتم‌های یادگیری ماشین و روش‌های اکتشافی، شناسایی کرد. البته استاندارد IPFIX[14] که یک استاندارد دیگر برای ذخیره‌سازی داده‌های جریان شبکه است، و در برخی از تجهیزات مسیریابی قرار داده‌شده است، نیز تولید داده‌های جریان ترافیک به صورت دوطرفه انجام می‌دهد.

۲-۱- کاربردهای داده‌های جریان ترافیک

پس از جمع‌آوری داده‌های جریان ترافیک، می‌توان برای اهداف مختلف از این داده‌ها استفاده کرد. از مهم‌ترین کاربردهای این داده‌ها شناسایی وضعیت استفاده از منابع شبکه و میزان ترافیک بین لینک‌های مختلف به منظور خطایابی و برنامه‌ریزی منابع سخت‌افزاری در آینده، اهداف جرم‌شناسی[2] و امنیتی نام برد.

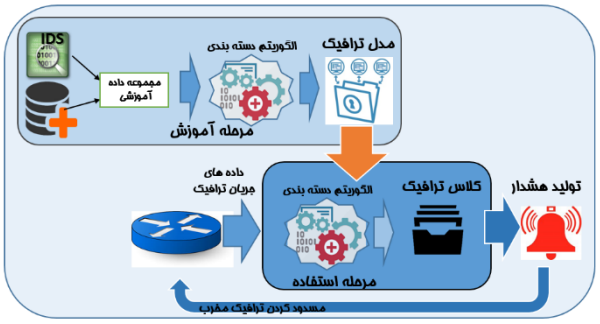
مدیران شبکه سازمانی می‌توانند با پیکربندی تجهیزات شبکه خود این داده‌ها را جمع‌آوری کرده و دائما برای نظارت بر میزان استفاده از پهنای باند و عملکرد تجهیزات و زیرساخت شبکه خود از این داده‌ها استفاده کنند. ابزارهای متعددی به منظور مصورسازی و بازیابی اطلاعات در این داده‌ها وجود دارد که از جمله ابزارهای متن باز موجود و رایج در این زمینه می‌توان به Nfsen اشاره کرد[۱۵].

[1] Connection
[2] Forensics

به طور کلی داده‌های NetFlow دو دسته کاربرد مهم و اساسی دارند، یکی استفاده به منظور مشاهده وضعیت استفاده از منابع و زیرساخت‌های شبکه و دیگری به منظور اقدامات جرم‌شناسی است. استفاده از داده‌های جریان ترافیک برای اهداف جرم‌شناسی به منظور شناسایی مهاجم و یا مجرمین می‌باشد. از آن‌جایی که نگهداری کل داده‌های مبادله شده در شبکه کار غیرممکنی است، می‌توان با نگهداری بخشی از داده‌های که به صورت خلاصه حاوی اطلاعات مربوط به ترافیک هستند، در کاربردهای جرم‌شناسی از آن استفاده کرد. به عنوان مثال، می‌توان با ابزارهای پرس‌وجوی موجود مانند Nfdump [۱۶] و یا [۱۷] ISILK برروی داده‌های جمع‌آوری شده پرس‌وجویی اجرا کرد که کل جریان‌های مربوط به یک IP خاص در بازه زمانی مشخص را استخراج نمود. ضمن این که حجم این داده‌ها علیرغم خلاصه بودن، برای شبکه‌های بزرگ بسیار زیاد خواهد شد. به صورت معمول حجم داده‌های جریان ترافیک حدود یک درصد کل داده‌های ترافیک مربوطه می‌باشد.

۳- روش پیشنهادی برای تشخیص حملات

در این مقاله استفاده از یادگیری ماشین برای تشخیص حملات شبکه از روی جریان ترافیک پیشنهاد شده است. در این رویکرد، الگوریتم‌های یادگیری ماشین از روی نمونه داده‌های موجود برای جریان ترافیک، قادر به استخراج خودکار الگوی رفتاری حالت نرمال شبکه و حالت‌های حمله هستند. برای این کار، از یادگیری با ناظر[1] استفاده می‌کنیم که در آن لازم است نمونه داده‌های دارای برچسب[2] مربوط به دسته‌های مختلف (نرمال و انواع حمله) باشند. با استفاده از داده‌های جریان ترافیک، الگوریتم‌ها آموزش داده شده و الگوی ترافیک مربوط به حملات مختلف را استخراج می‌کنند. مراحل کار شامل دو فاز آموزش و استفاده است که در فاز آموزش، مدل ترافیک از روی داده‌ها توسط الگوریتم‌های دسته‌بندی[3] استخراج می‌شود و در فاز استفاده از این مدل برای تشخیص نوع رفتار شبکه استفاده می‌شود. اجزای روش پیشنهادی در شکل(۲) نشان داده شده‌است. برای آموزش سیستم می‌توان از داده‌های برچسب‌گذاری شده مشابه مجموعه داده KDDCUP99 و یا داده‌های استخراجی با روش‌های دیگر استفاده کرد. پس از آموزش سیستم با داده‌های برچسب‌گذاری شده، از مدل استخراج شده می‌توان برای تشخیص حملات متعدد بهره گرفت. برای الگوریتم‌های یادگیری می‌توان از روش‌های مختلف دسته‌بندی مانند ماشین بردار پشتیبان، درخت تصمیم[4] ، شبکه عصبی[5] (به صورت ساده یا یادگیری عمیق[6]) بهره برد[۱۸]. در صورتی که حجم داده آموزشی زیاد باشد، روش‌های مبتنی بر شبکه عصبی، به ویژه شبکه‌های عمیق کارایی بالایی خواهند داشت اما در صورت عدم دسترسی با داده با حجم بالا، روش‌های کلاسیک یادگیری ماشین اولویت دارند. از این رو، در این مقاله از سه روش یادگیری بیز ساده[۱۹] ، SVM [۲۰]. درخت تصمیم بیز ساده[۲۱] برای این منظور استفاده شده است. در فاز استفاده، پس از شناسایی سریع مبتنی بر داده‌های جریان ترافیک، سیستم می‌تواند، هشدارهای مربوطه را تولید نموده و یا ترافیک مربوطه را به صورت خودکار مسدود نماید.



شکل ۲ - مراحل اجرای کار شناسایی حملات و دسته‌بندی ترافیک با استفاده از داده‌های جریان ترافیک در روش پیشنهادی

دسته‌بند بیز ساده، یک روش دسته‌بندی احتمالاتی بر پایه این فرض است که همهٔ ویژگی‌ها مستقل از همدیگر بوده هستند. بیز ساده به‌رغم ساده‌تر بودن آن نسبت به سایر دسته بندها، نتایج قابل قبولی را در شرایط داده کم نشان داده است.

[1] Supervised Learning
[2] Label
[3] Classification
[4] Decision Tree
[5] Neural Network
[6] Deep Learning

استنتاج بیزی[1]، که دسته‌بند بیز ساده یک نمونهٔ نمونهٔ ساده از آن است، بر پایه قانون بیز است که احتمال‌های شرطی و پیشین[2] را به هم مرتبط می‌سازد. به بیان دقیق‌تر، نشان می‌دهد که چگونه احتمال شرطی یک رویداد می‌تواند بر اساس احتمال اولیه‌اش و احتمال شرطی معکوس، محاسبه گردد. به‌دلیل اینکه قانون بیز احتمال خطا را به حداقل می‌رساند، از این نظر این قانون بهینه است؛ زیرا برای هر ترافیک در شبکه، دستهٔ با بالاترین احتمال شرطی را انتخاب می‌کند، بنابراین میزان خطای تعلق یک سند به یک دسته به کمترین مقدار می‌رسد. با توجه به قانون بیز داریم:

$$P\,(Class|NetFlow) = \frac{P\,(NetFlow|Class)\,P\,(Class)}{P\,(NetFlow)} \quad (۱)$$

مخرج این کسر یک عدد ثابت است و تأثیری بر نتیجه ندارد، بنابراین می‌توان آن را حذف کرد. درنهایت دسته‌ای انتخاب می‌شود که احتمال P (Class | NetFlow) بالاتری داشته باشد. لذا برای محاسبهٔ این بیشینه احتمال پسین[3]، فرمول به شکل زیر خواهد بود (c بیانگر دسته و f بیانگر جربان ترافیک شبکه است):

$$c_{NB} = \underset{c\,\in C}{\operatorname{argmax}} P\,(c|f) = \underset{c\,\in C}{\operatorname{argmax}} \frac{P\,(f|c)\,P\,(c)}{P\,(f)} = \underset{c\,\in C}{\operatorname{argmax}} P\,(f|c)\,P\,(c) \quad (۲)$$

با فرض نمایش جربان ترافیک f به‌صورت بردار ویژگی [x1, x2, …, xn]، داریم:

$$c_{BN} = \underset{c\,\in C}{\operatorname{argmax}} P\,(f|c)\,P\,(c) = \underset{c\,\in C}{\operatorname{argmax}} P\,(x_1, x_2, \dots, x_n|c)\,P(c) \quad (۳)$$

و این‌که ویژگی‌ها در بیز ساده مستقل از همدیگر هستند (ویژگی به شرط دسته P (x_i | C_j)):

$$P\,(x_1, x_2, \dots, x_n|c) = P\,(x_1|c) \times P\,(x_2|c) \times \dots \times P\,(x_n|c) \quad (۴)$$

یعنی:

$$c_{NB} = \underset{c\,\in C}{\operatorname{argmax}} P\,(x_1, x_2, \dots, x_n|c) \quad (۵)$$

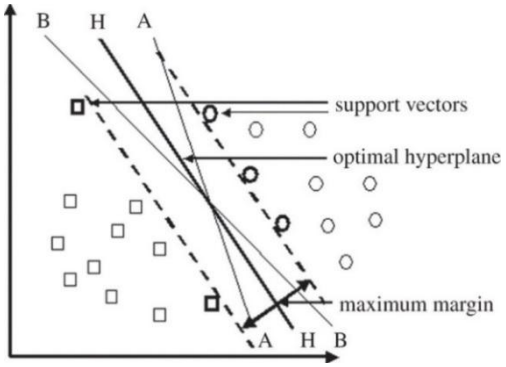
بنابراین:

$$c_{NB} = \underset{c_j\,\in C}{\operatorname{argmax}} P\,(c_j) \prod_{i=1}^{n} P\,(x_i|c_j) \quad (۶)$$

که P (c_j) احتمال دستهٔ jام است (احتمال پیشین) و P (x_i | c_j) نیز احتمال ویژگی iام به شرط دستهٔ jام است (احتمال رخداد این ویژگی در این دسته) و ارن دو احتمال از روی داده های آموزشی محاسبه می‌شوند.

طبق قانون بیز ساده اگر یکی از احتمال‌های P (x_i | c_j) صفر باشد، کل احتمال برای آن دسته صفر خواهد بود. برای رفع مشکل صفر شدن احتمال‌ها از هموارسازی[4] استفاده می‌کنیم. همچنین، به‌دلیل اینکه مقدار احتمال هر ویژگی کوچک‌تر از یک است و ضرب تعداد زیادی عدد کوچک‌تر از یک، صفر خواهد شد، برای رفع این مشکل از لگاریتم استفاده می‌شود. درنهایت دسته‌ای به‌عنوان پاسخ مناسب برای جریان ترافیک داده شده انتخاب خواهد شد که لگاریتم احتمال بیشتری داشته باشد.

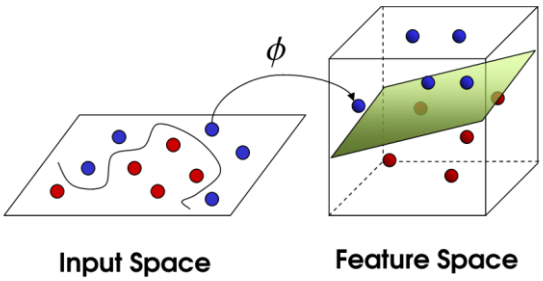
ماشین بردار پشتیبان را می‌توان نوع پیشرفته‌ای از جداساز خطی دانست که در آن دسته‌ها (در اینجا انواع حملات و حالت نرمال شبکه) توسط یک ابرصفحه[5] یا ترکیبی از ابرصفحات در یک فضای چندبعدی از هم تفکیک می‌شوند به گونه‌ای که فاصله ابرصفحه از نزدیک‌ترین نقاط دسته‌ها بیشنه است. به نزدیک‌ترین نقاط دسته‌ها به مرز تصمیم‌گیری اصطلاحاً بردارهای پشتیبان گفته می‌شود. ماشین بردار پشتیبان از نظر قابلیت تعمیم[6] به فضاهای با بعد بالاتر و تعداد دسته‌های بیشتر مناسب بوده و به بیش‌برازش[7] نیز چندان حساس نیست[20]. در شکل ۳ مفهوم بردارهای پشتیبان و مرز جداساز مشاهده می‌شود. همان‌طور که در تصویر مشخص است، در ابتدا بردارهای پشتیبان معین شده و سپس یک صفحه (در اینجا خط) به عنوان معادله جداساز انتخاب می‌شود به گونه‌ای که این خط دارای دورترین فاصله نسبت به بردارهای پشتیبان است.



شکل ۳: مرز تصمیم‌گیری در ماشین بردار پشتیبان

[1] Bayesian
[2] Marginal (Prior)
[3] Maximum a Posteriori (MAP)
[4] Smoothing
[5] Hyperplane
[6] Generalization
[7] Overfitting

هنگامی که دسته‌ها از نظر خطی قابل جداسازی نباشند از روشی استفاده می‌شود که به ترفند هسته[1] معروف است. در این روش بردارهای ویژگی را به فضایی با بعد بالاتر می‌برند به گونه‌ای که در این فضا قابل جداسازی به صورت خطی باشند. معادله هسته باید پیش از استفاده از SVM تعیین شده باشد. این مفاهیم در شکل زیر واضح‌تر بیان شده که در آن Φ تابع هسته است. برای این منظور، در SVM، از هسته‌های متفاوتی همچون RBF[2]، Linear و Polynomial استفاده می‌شود [۲۰].

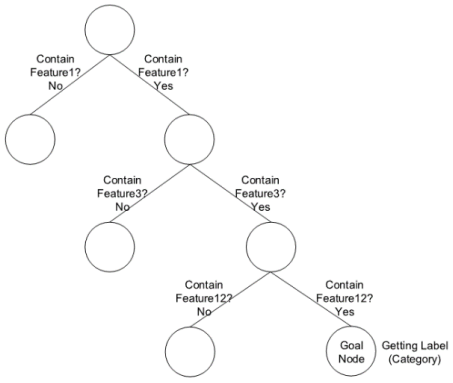


شکل ۴: نحوه‌ی کار تابع هسته در SVM

درخت تصمیم روشی است برای دسته‌بندی قواعد حاکم بر داده‌ها است که این کار را به‌وسیلهٔ ساختار درختی چندشاخه‌ای که به‌صورت بازگشتی[3] داده را تقسیم می‌کند، انجام می‌دهد[22]. می‌دهد[22]. هر شاخه از درخت تصمیم نمایانگر یک قانون است که ارتباط ویژگی‌ها (گره‌های داخلی) را آزمایش و برچسب دسته را در گرهٔ پایانی پیش‌بینی می‌کند. هر گره پایانی، بیانگر دسته‌های خروجی است و هر گره میانی نمایانگر ویژگی‌های استخراج شده از داده می‌باشد. هر گره میانی، از یک سطح آستانه استفاده می‌کند تا داده‌ها را به زیرشاخه‌های پایین‌تر تقسیم کند. این فرایند زمانی متوقف می‌گردد که هر گره پایانی شامل یک دسته از داده‌ها باشد. درخت تصمیم تولید شده، درختی است که برگ‌های آن دسته‌های مختلف و گره‌های میانی، ویژگی‌ها و حالات مختلف آن‌ها را نشان می‌دهد. فرض‌های اصلی الگوریتم پایه درخت تصمیم به صورت زیر است:

- درخت به صورت بالا پایین و بازگشتی ساخته می‌شود.
- در آغاز تمام مجموعه آموزشی در ریشه درخت قرار دارند.
- فرض می‌کنیم ویژگی‌ها مقادیر گسسته دارند.
- در هر مرحله، ویژگی‌ها به صورت بازگشتی بر حسب اهمیت آنها انتخاب شده و داده‌ها بر اساس آنها بخش‌بندی می‌شوند (شکل ۵).
- اهمیت ویژگی‌ها در هر مرحله بر اساس یک شاخص بهره اطلاعاتی[4] مانند انتروپی[5] انتخاب می‌شوند.

در این مقاله از روش درخت تصمیم بیز ساده[۲۱] استفاده کرده‌ایم که ترکیب درخت تصمیم و بیز ساده است. در این روش، گره‌های میانی درخت با همان روش‌های رایج درخت تصمیم ایجاد می‌شوند[۲۲] اما در برگ‌های درخت (معادل دسته‌ها) از بیز ساده برای دسته‌بندی استفاده می‌کند.



شکل ۵: نحوه عملکرد درخت تصمیم

۳-۱- جمع‌آوری داده جریان ترافیک

کلیه تحقیقات مرتبط انجام شده در حوزه تحلیل داده‌های ترافیک شبکه با دسترسی به ترافیک واقعی یک سازمان و استخراج داده‌های جریان ترافیک و اجرای روش‌های مختلف، تحلیل‌های خود را انجام داده‌اند. در پژوهش جاری با توجه به عدم دسترسی به داده‌های ترافیک واقعی، روش‌های متعددی برای حل مساله داده، بررسی شد. از آنجایی که هدف این پژوهش تحلیل داده‌های جریان ترافیک به منظور شناسایی برخی از حملات و رفتارهای غیرنرمال در شبکه، با استفاده از روش‌های یادگیری ماشین بود، نیازمند داده‌های ترافیک برچسب خورده برای الگوریتم‌های دسته‌بندی باناظر بودیم. لذا نیازمند حداقل دو دسته ترافیک (ترافیک نرمال و ترافیک حمله) بودیم. ترافیک نرمال و حمله به صورت مجزا مربوط به شبکه‌های مختلف در اینترنت در قالب فایل‌های PCAP وجود دارد (PCAP, 2018). اما برای این که روش دسته‌بندی یک روش معتبر باشد، داده‌های مربوط به کلاس‌های مختلف می‌بایست همگن و مربوط به یک شبکه باشند. به علاوه، ترافیک حمله را

[1] Kernel Trick
[2] Radial Basis Function (RBF)
[3] Recursively
[4] Information Gain
[5] Entropy

می‌توان به صورت مصنوعی با استفاده از ابزارهای تست نفوذ همچون BACKTRACK [۲۳]. و یا ابزارهای ارزیابی سامانه‌های تشخیص نفوذ مانند PYTBULL [۲۴]. تولید کرد. اما باز برای دسته ترافیک نرمال نیازمند ترافیک واقعی هستیم. در کلیه تحقیقات انجام شده نیز از ترافیک واقعی برای دسته نرمال استفاده شده است[۱].

با توجه به عدم دسترسی به ترافیک واقعی، راه دیگر پیش‌رو استفاده از مجموعه داده‌های استاندارد است. مجموعه داده‌های استانداردی که در دسترس هستند، برخی از فیلدهای مربوط به جریان داده ترافیک را در بر می‌گیرند، لذا می‌توان با درنظر گرفتن سطحی از خطا از مجموعه داده‌های استانداردی که بخشی از فیلدهای مربوط به جریان ترافیک را دارند، استفاده کرد. یکی از معتبرترین مجموعه داده‌های مربوط به امنیت شبکه، مجموعه داده‌های KDDCUP99 [۱۲] است. این مجموعه داده که از داده‌های یک شبکه نظامی استخراج شده است، در پروژه‌ای در دانشگاه MIT جمع‌آوری شده است. هدف از تولید این داده‌ها ارزیابی تحقیقات در حوزه تشخیص نفوذ است. در مسابقات داده‌کاوی سال ۱۹۹۹ بخشی از داده‌های این پروژه در اختیار عموم قرار گرفته است که در این پژوهش از آن استفاده می‌شود.

۴- نتایج ارزیابی‌ها

در این بخش ابتدا به معرفی مجموعه داده مورد استفاده پرداخته می‌شود و سپس نتایج به کارگیری روش‌های بیان شده در این مقاله برای تشخیص حملات ارائه شده است.

۴-۱- مجموعه داده KDDCUP99

در این مقاله از داده‌های KDDCUP99 استفاده کرده‌ایم که داده‌های اصلی ترافیک خام حدود ۵ گیگابایت فایل فشرده شده PCAP بوده که شامل ترافیک مربوط به ۷ هفته از شبکه شبیه‌سازی شده می‌باشد[۱۱،۱۲]. کل داده‌ها حدود ۵۰ میلیون رکورد، دسته‌بندی شده است که در این پژوهش به دلیل محدودیت‌های منابع از نصف این داده‌ها استفاده شده است و ۱۰٪ آن به عنوان داده آزمون استفاده شده است. هر رکورد شامل ۴۱ فیلد می‌باشد که از این بین تنها ۷ فیلد آن مرتبط با داده‌های جریان ترافیک می‌باشد و در عمل بدست آوردن سایر فیلدها در شبکه واقعی نیازمند تحلیل ترافیک خام و یا نصب ابزارهای خاص برروی گره‌های شبکه خواهد بود. لذا در عمل در شبکه‌های بزرگ استخراج سایر ویژگی‌ها امری بسیار پیچیده و زمان‌بر خواهد بود. در جدول ۱ ویژگی‌های پایه‌ای مربوط به ارتباطات TCP آمده است. با توجه به این که کلیه این ویژگی‌ها از داده‌های جریان ترافیک قابل استخراج می‌باشند، از این مجموعه داده و ویژگی‌های مدت زمان ارتباط، پروتکل لایه انتقال، سرویس مربوطه در لایه کاربرد، حجم داده ارسال شده و دریافت شده و بیت مشخص کننده وجود یا عدم وجود ارتباط در ۲ ثانیه اخیر بین دو گره، به عنوان ویژگی‌های قابل استخراج از داده‌های جریان ترافیک در نظر گرفته شده و در الگوریتم‌های دسته‌بندی مورد استفاده قرار گرفته‌اند.

داده‌های جریان ترافیک معمولا شامل دو رکورد، یکی شامل ویژگی‌های مربوط به جریان رفت و دیگری مربوط به جریان برگشت، برای هر ارتباط می‌باشند. البته یک‌طرفه بودن اطلاعات جریان ترافیک در استاندارد NetFlow وجود دارد که برای جبران این مساله و تجمیع رکوردهای دوطرفه در یک رکورد یک طرفه دو راهکار اساسی وجود دارد. یکی استفاده از استاندارد IPFIX است که به صورت خودکار در برخی از مسیریاب‌ها و تجهیزات قابل پیکره بندی می‌باشد. روش دیگر تجمیع رکوردهای مربوط به جریان‌های رفت و برگشت می‌باشد. به عنوان مثال می‌توان با در نظر گرفتن زوج آدرس IP و شماره پورت مبدا و مقصد و رکوردهایی که در یک بازه زمانی مشخص با این فیلدهای یکسان مبادله شده‌اند، را تجمیع نمود. به عنوان یک نمونه از این موضوع، در [۲۵] روشی برای این کار ارائه شده است. در[۹] نیز روش دیگری برای این کار ارائه شده‌است.

جدول (۱): ویژگی‌های پایه‌ای ارتباطات TCP در مجموعه داده مورد استفاده

نام ویژگی	نوع متغیر	توضیح
duration	پیوسته	مدت زمان ارتباط برحسب ثانیه
protocol_type	گسسته	نوع پروتکل(tcp, udpو..)
service	گسسته	سرویس ارائه شده در مقصد (http, telnet,…)
src_bytes	پیوسته	تعداد بایت داده ارسال شده از مبدا به مقصد
dst_bytes	پیوسته	تعداد بایت ارسال شده از مقصد به مبدا
land	گسسته	یک اگراخیرا ارتباطی از/به میزبان و پورت یکسانی بوده است و در غیر اینصورت صفر
flag	گسسته	وضعیت نرمال یا خطا در ارتباط
wrong_fragment	پیوسته	تعداد قطعات اشتباه در طول ارتباط
urgent	پیوسته	تعداد بسته‌های با برچسب اورژانسی در طول ارتباط

در داده‌های KDDCUP99 علاوه بر دسته نرمال، چهار نوع حمله وجود دارد که هر کدام از حملات شامل تعدادی زیر دسته دیگر است. جزئیات این دسته‌ها و تعداد نمونه‌های آنها در جدول ۲ نشان داده شده است[۲۶]. در مجموع یک دسته نرمال و ۲۲

دسته حمله در داده‌ها وجود دارد. همانطور که مشخص است عمده داده‌ها مربوط به دو دسته نرمال و حملات DOS هستند.

جدول ۲- دسته‌های مختلف مجموعه داده KDDCUP99 و تعداد آنها

Class	Sub-Classes	Samples
NORMAL		97278 (19.6911%)
U2R	buffer_overflow, loadmodule, multihop, perl, rootkit	52 (0.0105%)
R2L	ftp_write, guess_passwd, imap, phf, spy, warezclient, warezmaster	1126 (0.2279%)
DOS	back, land, neptune, pod, smurf, teardrop	391458 (79.2391%)
PRB	ipsweep, nmap, portsweep, satan	4107 (0.8313%)

۴-۲- نتایج

برای ارزیابی کارایی روش‌های استفاده شده از معیار دقت[1] استفاده شده است که بیانگر درصد نمونه‌هایی است که دسته آنها به درستی تشخیص داده شده است. در پیاده‌سازی‌ها از نرم‌افزار وکا[2] [۲۷] استفاده شده است و الگوریتم‌های دسته‌بندی بیز ساده (Naive Bayes)، ماشین بردار پشتیبان (SVM) با روش[3] SMO [۲۸] با هسته چندجمله‌ای و درخت تصمیم بیز (NBTree) برروی داده‌های اصلی با همه ۴۱ ویژگی اجرا شد. دقت دسته‌بندی برای هر سه الگوریتم و همه ۲۳ دسته در شکل ۶ نشان داده شده است. محور افقی دسته‌های مختلف ترافیک (نرمال و تعداد ۲۲ حمله) و محور عمودی درصد دقت دسته‌بندی را نمایش می‌دهد. همان‌گونه که در نمودار قابل مشاهده است، همه الگوریتم‌ها متوسط دقت دسته‌بندی نزدیک به هم داشته‌اند اما کارایی آنها برای شناسایی حملات مختلف با هم متفاوت است و روش بیز ساده کارایی کمتری دارد. متوسط وزنی دقت دسته‌بندی برای الگوریتم SVM که بهتر از بقیه است، برابر با ۹۹.۹ درصد می‌باشد. توجه شود که عمده داده‌ها مربوط به دو دسته نرمال و حملات DOS هستند که دقت الگوریتم‌ها برای آنها بالاست. اگر میانگین غیر وزن دار گرفته شود متوسط دقت‌ها پایین‌تر خواهند بود.

در شکل ۷ دقت دسته‌بندی برای الگوریتم‌های مختلف دسته‌بندی برای حالتی که از ۷ ویژگی انتخابی بیان شده استفاده کنیم، نشان داده شده است. در این شکل تنها تعدادی از حملات به همراه دسته نرمال نشان داده شده است و انتهای محور افقی میانگین وزن‌دار دقت روی همه دسته‌ها آورده شده است. در این حالت هم متوسط دقت نزدیک به دقت حالت ۴۱ ویژگی است. بهترین دقت در این حالت متعلق به NBTree با دقت ۹۹.۸ درصد است.

۴-۳- تحلیل نتایج

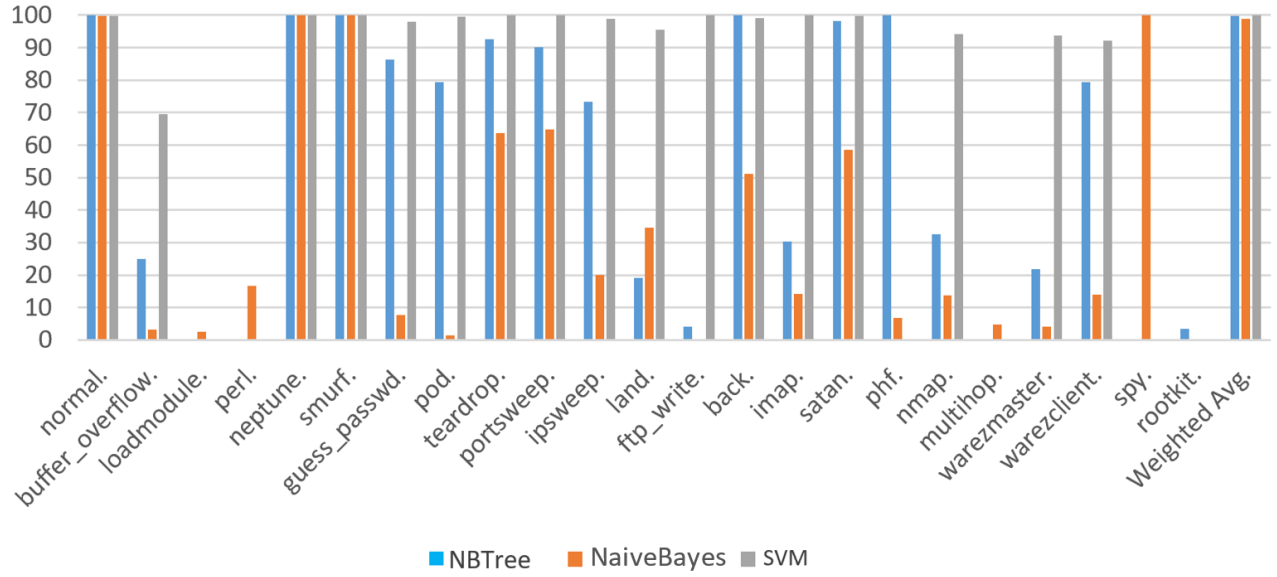
ارزیابی‌ها نشان می‌دهد کاهش بعد از ۴۱ به ۷، تاثیر چندان قابل ملاحظه‌ای در دقت الگوریتم‌های دسته‌بندی نداشته، اما سرعت اجرای الگوریتم‌ها به صورت قابل ملاحظه‌ای کاهش پیدا کرده است. با توجه به نتایج این ارزیابی، می‌توان از داده‌های خلاصه جریان ترافیک، با دقت قابل قبولی ترافیک نرمال و حملات را شناسایی نمود. البته دقت تشخیص با توجه به ماهیت برخی از حملات متفاوت بوده است. به عنوان مثال حملاتی همچون rootkitها وspy-wareها چون پس از نفوذ در سیستم میزبان، عملات خرابکارانه را انجام می‌دهند و تراکنش‌های شبکه‌ای قابل ملاحظه‌ای ندارند، از طریق خلاصه اطلاعات Netflow نمی‌توان به درستی این دسته از حملات را شناسایی نمود. برای این حمله‌ها استفاده از کلیه ۴۱ ویژگی نیز بهبود چندانی در دسته‌بندی حاصل ننموده است، و یکی از دلایلی که الگوریتم دسته‌بندی در هیچ حالتی نتوانسته است دسته‌بندی را به درستی انجام دهد. می‌تواند ناشی از کمبود داده آموزشی در این مجموعه داده باشد. از بین کلیه رکوردهای داده‌ای (حدود ۵ میلیون)، تنها حدود ۱۰۰ رکورد مربوط به حملات rootkit، .Spy، .perl، loadmodule، ftp_write، phf. و multihub بوده‌اند که این مساله نیز به نوبه خود در بالا بودن خطای دسته‌بندی تاثیرگذار است. مقایسه میانگین وزنی دقت دسته‌بندی در هر دو حالت نیز صحت این مساله را تایید می‌نماید. میانگین دقت دسته‌بندی برای حالت ۷ ویژگی و ۴۱ ویژگی برای سه الگوریتم‌ها NaiveBayes، SVM و NBTree در جدول ۳ خلاصه شده است.

جدول (۳): میانگین وزن‌دار درصد دقت سه الگوریتم یادگیری ماشین در تشخیص ۲۳ دسته (۲۲ حمله و دسته نرمال) با ۷ ویژگی انتخابی و ۴۱ ویژگی

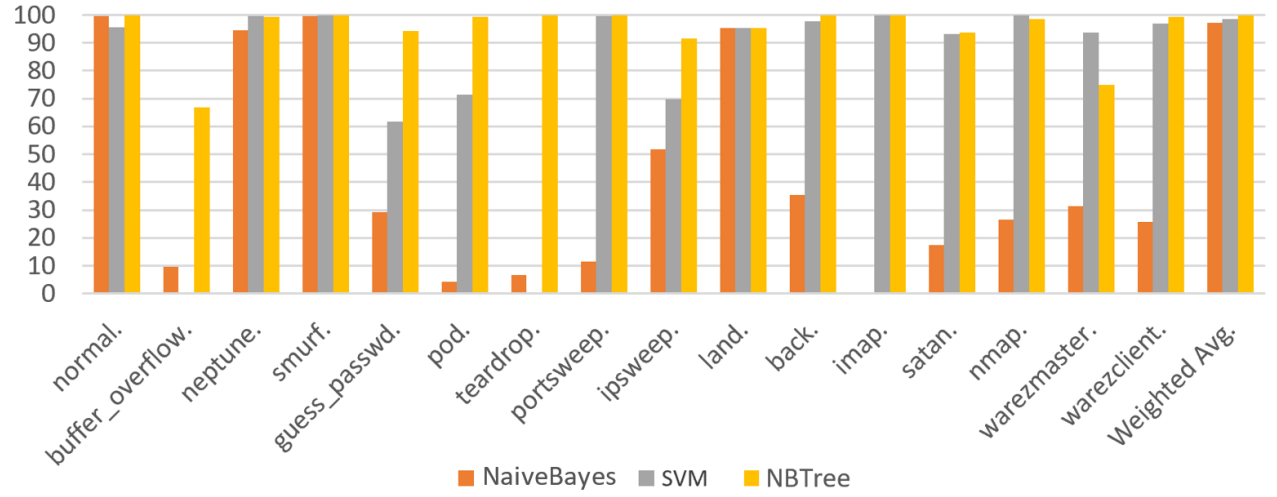
روش	۷ ویژگی	۴۱ ویژگی
NaiveBayes	۹۷.۱	۹۸.۹
SVM	۹۸.۶	**۹۹.۹**
NBTree	۹۹.۸	۹۹.۷

با توجه به نتایج ارزیابی اطلاعات خلاصه جریان ترافیک از بین ۲۳ دسته ترافیک سالم و حمله، این روش‌ها قادر به شناسایی شناسایی ۱۳ کلاس ترافیک سالم و حمله، با دقتی بالاتر از ۹۰ درصد هستند. با افزایش نمونه‌های آموزش، می‌توان برخی از حملات دیگر را نیز که رفتارشان از طریق ترافیک شبکه آنها قابل شناسایی باشد، شناسایی نمود.

[1] Accuracy
[2] Weka
[3] Sequential Minimal Optimization (SMO)



شکل ۶ – دقت دسته‌بندی الگوریتم‌های مورد بررسی با ۴۱ ویژگی در همه ۲۳ دسته (۲۲ نوع حملات شبکه و یک دسته نرمال). ستون آخر میانگین وزن‌دار دقت روی همه دسته‌ها است



شکل ۷ – دقت دسته‌بندی الگوریتم‌های مورد بررسی با ۷ ویژگی در دسته نرمال و حملات مختلف شبکه (ستون آخر میانگین وزن‌دار دقت روی همه دسته‌ها است

۵- نتیجه‌گیری و جمع‌بندی

در این مقاله به استفاده از روش‌های یادگیری ماشین برای تشخیص حملات شبکه‌ها پرداختیم و نشان داده شده که الگوریتم‌های یادگیری ماشین برای تشخیص ترافیک‌های غیرنرمال و حملات، کارایی قابل قبولی دارند. در شناسایی نفوذ فرض شده بود که تنها به اطلاعات جریان شبکه (NetFlow) دسترسی داریم و کلیه ترافیک شبکه و ویژگی‌های آن را در اختیار نداریم. این فرض در شبکه‌های بزرگ و یا شبکه‌های کوچکی که منابع محدودی برای تشخیص نفوذ در اختیار دارند، برقرار می‌باشد. در چنین شرایطی می‌توان از الگوریتم‌های یادگیری ماشین برای دسته‌بندی و شناسایی الگوی حملات استفاده کرد.

در این پژوهش، ۲۲ دسته از حملات مهم مورد بررسی قرار گرفت و از الگوریتم‌های دسته‌بندی NaiveBeyes، SVM و NBTree برای این کار استفاده شد. نتایج دسته‌بندی نشان داد هرچند متوسط دقت دسته‌بندی برای کلیه این الگوریتم‌ها در شرایط مختلف بیش از ۹۷ درصد است اما این روش‌ها قادر به تشخیص همه حملات نیستند، هم به دلیل کمبود داده و هم به دلیل نوع عملکرد حملات. بررسی الگوریتم‌های دیگر یادگیری ماشین به ویژه روش‌های یادگیری عمیق و

الگوریتم‌هایی که حافظه‌دار هستند و سابقه جریان را نیز در نظر می‌گیرند، از جمله مسیرهای پژوهشی پیش رو برای این حوزه هستند.

علاوه براین از آنجایی که استخراج ویژگی‌های مورد نیاز از ترافیک در شبکه‌های بزرگ سربار پردازشی نسبتا بالایی دارند، در شبکه‌های بزرگ که با داده‌های جریان شبکه با نرخ بالا تولید شده و حجم بالایی را دارند، می‌توان از رویکردهای مبتنی بر کلان داده و یادگیری عمیق برای جمع آوری این داده‌ها و استخراج ویژگی استفاده کرد. در پژوهش‌های آتی سعی داریم الگوریتم پیشنهادی را در شبکه های با مقیاس بزرگ و با استفاده از رویکردهای کلان داده پیاده‌سازی نماییم.

مراجع

[1] C.-F. Tsai, Y.-F. Hsu, C.-Y. Lin, and W.-Y. Lin, "Intrusion detection by machine learning: A review," *Expert Syst. Appl.*, vol. 36, no. 10, pp. 11994–12000, 2009.

[2] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *J. Netw. Comput. Appl.*, vol. 36, no. 1, pp. 16–24, 2013.

[3] "Cisco Systems NetFlow Services Export Version 9." https://www.ietf.org/rfc/rfc3954.txt (accessed Jun. 22, 2020).

[4] M. F. Umer, M. Sher, and Y. Bi, "Flow-based intrusion detection: Techniques and challenges," *Comput. Secur.*, vol. 70, pp. 238–254, 2017.

[5] S. H. Mousavi, M. Khansari, and R. Rahmani, "A fully scalable big data framework for botnet detection based on network traffic analysis," *Inf. Sci.*, vol. 512, pp. 629–640, 2020.

[6] M. J. Vargas-Muñoz, R. Martínez-Peláez, P. Velarde-Alvarado, E. Moreno-García, D. L. Torres-Roman, and J. J. Ceballos-Mejía, "Classification of network anomalies in flow level network traffic using Bayesian networks," in *2018 International Conference on Electronics, Communications and Computers (CONIELECOMP)*, 2018, pp. 238–243.

[7] E. Glatz and X. Dimitropoulos, "Classifying internet one-way traffic," in *Proceedings of the 2012 Internet Measurement Conference*, 2012, pp. 37–50.

[8] D. Rossi and S. Valenti, "Fine-grained traffic classification with netflow data," in *Proceedings of the 6th international wireless communications and mobile computing conference*, 2010, pp. 479–483

[9] "Berthier, Robin, Michel Cukier, Matti Hiltunen, Dave Kormann, Gregg Vesonder, and Dan Sheleheda. "Nfsight: netflow-based network awareness tool." In Proceedings of LISA'10: 24th Large Installation System Administration Conference, vol. 119. 2010.

[10] R. Vaarandi, "Detecting anomalous network traffic in organizational private networks," in *2013 IEEE International Multi-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, 2013, pp. 285–292.

[11] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, 2009, pp. 1–6.

[12] "KDD-CUP-99 Task Description." https://kdd.ics.uci.edu/databases/kddcup99/task.html (accessed Jun. 22, 2020).

[13] "NetFlow probes: fprobe and fprobe-ulog." http://fprobe.sourceforge.net/ (accessed Jun. 22, 2020).

[14] E. Boschi, L. Mark, J. Quittek, M. Stiemerling, and P. Aitken, "IP flow information export (IPFIX) implementation guidelines," *RFC 5153 Informational Internet Eng. Task Force*, 2008.

[15] "NfSen - Netflow Sensor." http://nfsen.sourceforge.net/ (accessed Jun. 22, 2020).

[16] "NFDUMP." http://nfdump.sourceforge.net/ (accessed Jun. 22, 2020).

[17] "iSiLK." https://tools.netsa.cert.org/isilk/ (accessed Jun. 22, 2020).

[18] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006

[19] Y. Zhang, G. Yan, and S. He, "Optical Fiber Spectrometer based on Smartphone Platform for Refractive Index Sensing Application," in *Asia Communications and Photonics Conference*, 2016, pp. AF3C–3.

[20] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998

[21] R. Kohavi, "Scaling up the accuracy of naive-bayes classifiers: A decision-tree hybrid.," in *Kdd*, 1996, vol. 96, pp. 202–207

[22] S. R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst. Man Cybern.*, vol. 21, no. 3, pp. 660–674, 1991

[23] "BackTrack Linux - Penetration Testing Distribution." https://www.backtrack-linux.org/ (accessed Jun. 22, 2020).

[24] "pytbull - IDS/IPS Testing Framework - home." http://pytbull.sourceforge.net/ (accessed Jun. 22, 2020)

[25] I. Nikolaev, "Network Service Anomaly Detection," *Czech Tech. Univ. Prague*, 2014

[26] M. Saniee Abadeh and J. Habibi, "A hybridization of evolutionary fuzzy systems and ant colony optimization for intrusion detection," *ISeCure-ISC Int. J. Inf. Secur.*, vol. 2, no. 1, pp. 33–46, 2010.

[27] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: an update," *ACM SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, 2009

[28] J. Platt, "Fast training of support vector machines using sequential minimal optimization. Advances in Kernel Methods—Support Vector Learning (pp. 185–208)," *AJ MIT Press Camb. MA*, 1999.