Active learning to detect DDoS attack using ranked features

2019 july

Univirsity of Colorado

31 citations

استخراج ویژگی[1] به منظور تقسیم بندی[2] ترافیک در سامانه‌های تشخیص نفوذ برای شناسایی حملات منع خدمت(بی نظمی‌ها) در شبکه‌های پهن‌باند(که حجم زیاد و متنوعی از داده در حال تبادل می‌باشد) با چالش همراه می‌باشد. این چالش‌ها به دلیل الگوهای متغیر این گونه حملات و سختی تشخیص آنها در کنار ترافیک معمولی می‌باشد. پورت‌های غیر استاندارد،پورتهای تبدیلی و NAT این تقسیم بندی را دشوار می‌کنند. ترافیک رمز شده نیز استخراج ویژگی را با مشکل مواجه می‌کند. بلادرنگ بودن تقسیم بندی ترافیک در مواجه با ترافیک دایم درحال تغییر نیازمند برقراری تعادل در دقت،کارایی و هزینه می‌باشد. در این مقاله یک الگوریتم رتبه بندی تجمعی موازی برای رتبه بندی ویژگیهای دیتاست(دیتاست‌های امروزی high-dimensional هستند یعنی تعداد زیادی ویژگی دارند) ارایه می‌دهیم. به منظور تقسیم بندی مقرون به صرفه ترافیک ارایه می‌دهیم. همچنین در مورد اهمیت یادگیری فعال[3] برای انتخاب نمونه های مناسب توسط یک ماژول خبره[4] به روش بدون نظارت[5] برای آموزش یک طبقه بندی کننده باینری SVM[6] برای تشخیص ترافیک حمله DDoS نیز بحث می کنیم. روش ما کمترین تعداد نمونه برای آموزش را انتخاب می‌کند. همچنین دقت بالا و زمان اجرای کم درقبال ترافیک‌های بالا برای دسته بندی ارایه می‌دهد.

در تقسیم بندی ترافیک، روشهای مختلف یادگیری از انواع نظارتی و غیرنظارتی وجود دارد. در روشهای نظارتی، دقت تقسیم بندی به کیفیت نمونه‌های برچسب گذاری شده برای یادگیری مدل بستگی دارد. در روشهای نظارتی، برچسب گذاری نمونه‌ها در حالت بلادرنگ معمولاً سخت و پرهزینه می‌باشد، زیرا بایستی بسته‌های در حال ورود را به طور پیوسته برچسب گذاری کرد. یادگیری فعال که در دسته یادگیری‌های نیمه نظارتی می‌باشد، سعی میکند مجموعه مربوط به آموزش را، تا حد امکان کوچک در نظر گرفته می‌شود تا از افزونگی جلوگیری کند. نمونه‌های برچسب گذاری نشده را به منظور انتخاب الگوهای باارزشتر بررسی می‌کند و بر اساس تصمیم یک استاد[7] یا خبره[8] که به نمونه‌های انتخابی برچسب گذاری نشده، برچسب می‌زند، دایماً مجموعه آموزشی را به‌روز می‌کند. بر اساس معیارتنوع[9]، سعی میکنیم حداقل نمونه‌ها را انتخاب کرده و ویژگی‌های آنها را درنظر می‌گیریم، تا کارایی تقسیم کننده[10] افزایش یابد.

ویژگی‌های مهم روش ما در مقایسه با روش‌های دیگر:

- برخلاف یک رویکرد بسته‌بندی، روش‌ما متکی به یک تقسیم کننده ثابت نمی‌باشد.
- رویکرد ما به دلیل استفاده از یک رتبه‌بندی کننده موازی از نظر هزینه موثر می‌باشد. برخلاف روش‌های دیگر که متمرکز بودند و تنها روی دیتاست‌های کوچک می‌توانستند عمل کنند.

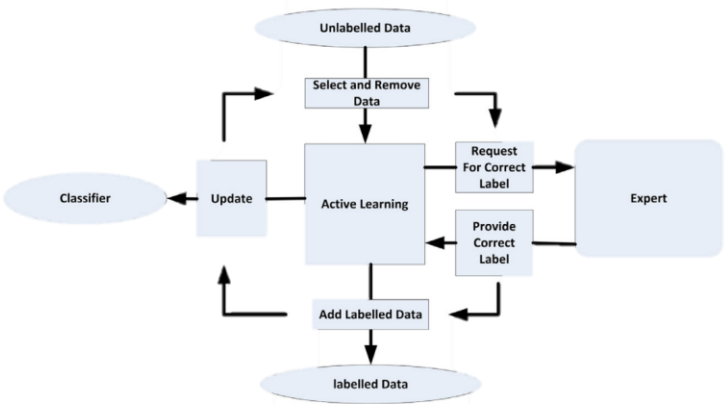
---

[1] Feature selection
[2] Classification
[3] Active learning
[4] Expert module
[5] unsupervised
[6] Support Vector Machine
[7] Supervised
[8] Expert
[9] Diversirt criterion
[10] Classifier

- رویکرد ما دقت بالایی خواهد داشت.

تا به امروز روش‌های مختلفی برای دسته بندی ترافیک پیشنهاد شده است:

- مبتنی بر پورت: سریع اما اطمینان بالایی ندارد.
- مبتنی بر پیلود(DPI): امضاهای اپلیکیشن استفاده می‌کند و بسیار دقیق می باشد. اما همانطور که می‌دانیم این روش‌های چالشهایی به همراه خواهد داشت.
- روش‌های آماری: از هدرهای لایه اینترنت برای شناسایی و دسته‌بندی جریان‌های مختلف استفاده می‌کند
- روش‌های یادگیری ماشین: در روشهای نظارتی، دقت تقسیم کننده به تعداد داده‌های آموزشی و کیفیت آنها وابستگی زیادی دارد.



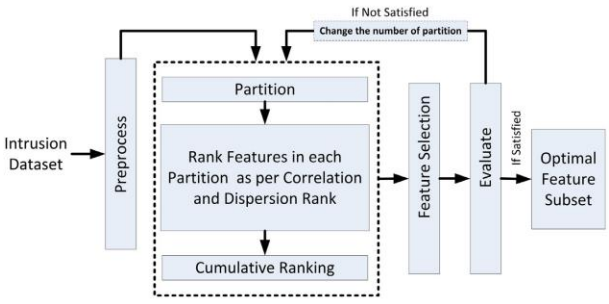
استخراج و طبقه بندی ویژگی‌ها با استفاده از محاسبات موازی: دلایلی که از استخراج کننده موازی ویژگی استفاده می‌کنیم

- یک داده واحد بسیار بزرگ داریم
- داده ممکن است در مجموعه‌های مختلفی در مکان‌های مختلفی باشد
- در سیستم‌های بلادرنگ داده‌ها با نرخ بالایی در حال ورود باشند
- ممکن است دیتاست بزرگ نباشد، اما استفاده از روشهای استخراج ویژگی موازی به انتخاب روش موثرتر می‌انجامد.

استخراج ویژگی بدین معنی می‌باشد که بهترین زیرمجموعه از ویژگی‌ها را از بین تمام زیرمجموعه‌ها انتخاب کنیم. یک زیرمجموعه کارا و موثر میتواند زمان آموزش و تست سیستم تشخیص نفوذ را بهبود ببخشد. همچنین می‌تواند به سبکی آن و امکان استفاده در محیطهای بلادرنگ بیانجامد. ویژگی‌ها را به سه روش می‌توان رتبه بندی کرد:

- Wrapper
- Filter
- Embedded

Parallel Cumulative Rank(PCR):

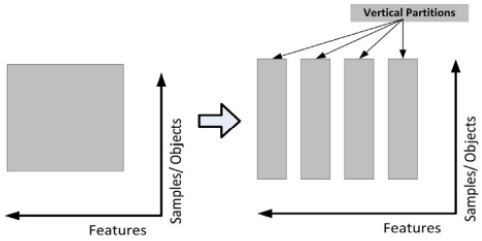


از سه task اصلی زیر تشکیل شده‌است:

Feature ranking: تمامی خصیصه‌های یک پارتیشن از دیتاست را رتبه بندی می‌کنیم. رتبه بندی به معنای شایع بودن و عدم افزونگی آن خصیصه می‌باشد. سپس یک رتبه‌بندی تجمعی با ترکیب رتبه‌بندی‌های

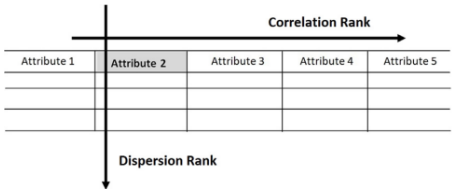
جدا برای هر پارتیشن به دست می‌آید. هر چه رتبه بندی بالاتر باشد، احتمال شامل شدن در زیرمجموعه ویژگی برای طبقه‌بندی نیز افزایش می‌یابد. این بخش خودش شامل سه زیر بخش می‌باشد

1. پیش‌پردازش: برای ساخت نمونه اولیه پنج‌تا خصیصه ترافیک شبکه‌ای هر نمونه را استخراج می‌کنیم: آدرس آی پی و پورت مبدا و مقصد و همچنین طول فریم. مشاهده میکنیم که برای هر تعداد خصیصه، فریمورک ما مقیاس پذیر می‌باشد. همچنین دقت را با تغییر تعداد پارتیشن هر دیتاست بررسی می‌کنیم.
2. پارتیشن بندی دیتاست: برای انجام محاسبات موازی توزیع‌شده، دیتاست را به صورت عمودی تقسیم می‌کنیم. تعداد پارتیشن‌ها را تغییر می‌دهیم و رتبه هر خصوصیت را در هر پارتیشن محاسبه می‌کنیم. همچنین زمان اجرا را هم در یک محیط ترتیبی و یک محیط موازی بررسی می‌کنیم.



Vertical Partition

3. رتبه‌بندی موازی: هر پارتیشن بر روی یک ماشین با استفاده از GPU ها بدون همپوشانی پردازش می‌شود. برای هر پارتیشن به صورت جدا رتبه بندی را با دو معیار ارتباط و افزونگی (رتبه بندی همبستگی[11] و رتبه بندی پراکندگی[12] و در نهایت رتبه بندی تجمعی) انجام میدهیم. منظور از رتبه بندی پراکندگی به معنای پراکندگی مقادیر یک خصیصه می‌باشد.



رتبه بندی تجمعی که از ترکیب دو معیار قبلی به دست می‌آید، میزان ارتباط اون خصیصه را در بین همه پارتیشن‌ها بهتر تعیین می‌کند. باید دقت شود که رتبه بندی یک مجموعه از ویژگی‌ها برای یک دیتاست ثابتَ بسته به تعداد پارتیشن‌ها می‌تواند تغییر یابد که در نتیجه دقت طبقه بندی کننده نیز تغییر پیداخواهد کرد. محاسبات موازی که در تنظیم تعداد پارتیشن‌های مجازی می‌تواند کمک کنند، منجر به بهبود دقت طبقه‌بندی کننده منجر خواهند شد.

Feature selection: با دانستن یک مقدارآستانه، سه خصیصه مناسب برای تقسیم بندی را انتخاب می‌کنیم.

Classification: پارتیشنها را ۱۰۰.۰۰۰ بسته‌ای درنظر می‌گیریم، با استفاده از رتبه‌بندی خصیصه، سه تا خصیصه را اولی را در نظر می‌گیریم و از این خصیصه‌ها برای طبقه‌بندی استفاده می‌کنیم. از پنج‌تا الگوریتم طبقه‌بندی مختلف استفاده می‌کنیم: KNN، Linear Discriminant، Logistic Regression، Boosting، Complex Decision Tree

برای ارزیابی PCR از محیطها و دیتاست‌های زیر استفاده می‌کنیم:

- دیتاست‌ها: MIT-DARPA، CAIDA-2007، ISCX و TUDDoS. با استفاده از T-Shark و editcap سه دسته بسته‌های ۵۰.۰۰۰،۱۰۰.۰۰۰ و ۱.۰۰۰.۰۰۰ از هر دیتاست را، پردازش می‌کنیم. در هرگروه نیز ۶۰ درصد ترافیک نرمال و بقیه ۴۰ درصد ترافیک DDoS می‌باشد. آدرس‌های آی‌پی را نیز به مقادیر دسیمال تبدیل می‌کنیم.
- ایستگاه‌های کاری[13]: پردازنده ۲.۳۰ گیگاهرتز، ۶۴ گیگ حافظه و windows 10 64-bit. برای پردازش موازی نیز از کارت گرافیک NVIDIA Quadro K620 با حافظه گرافیکی ۳۴ گیگ و ۲ گیگ مخصوص ویدیو و ۳۸۴ هسته استفاده می‌کنیم.

[11] Correlation
[12] Dispersion
[13] Workstation

- پلتفرم: از Matlab 2016 برای اجرای آزمایش‌ها استفاده می‌کنیم. ابزار مخصوص پردازش موازی دارد. ۱۰ تا محیط اجرایی متلب را به صورت موازی روی یک ماشین با استفاده از این ابزار اجرا می‌کنیم.

نتایج: زمانی که به صورت موازی خصوصیت‌ها را رتبه بندی می‌کنیم، در یک تعداد پارتیشن ثابت، نسبت به حالت ترتیبی زمان اجرا بسیار کاهش می‌یابد. همین مورد نیز برای تعداد worker ها نیز برقرار می‌باشد. اما تغییر تعداد پارتیشن‌ها بهبودی در این مورد نخواهد داشت.

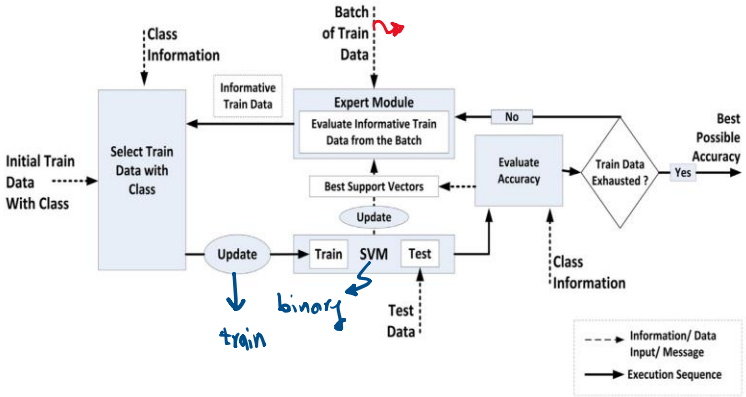
برای دقت دسته‌بندی، به نکات زیر می‌رسیم:

- به انتخاب صحیح ویژگی‌ها بستگی دارد
- پارتیشن بندی خوب دیتاست، با احتمال زیادی می‌تواند به انتخاب مجموعه ویژگی خوب کمک کند
- انتخاب تعداد worker مناسب نیز روی سرعت کار تاثیر می‌دارد اما روی دقت دسته‌بندی تاثیر ندارد.
- پارتیشن بندی نامناسب می‌تواند سرعت اثر بسیار منفی داشته‌باشد
- پارتیشن بندی و انتخاب تعداد نود مناسب، منجر به انتخاب مجموعه ویژگی مناسب و در نتیجه افزایش دقت دسته بندی کننده می‌انجامد.

دسته‌بندی ترافیک با یادگیری فعال:

فرض می‌کنیم حجم زیاد و متنوعی از ترافیک داریم اما در عین حال منابع‌مان کافی نمی‌باشد. با استفاده از دسته‌بندی کالکشنی[14] سعی می‌کنیم ترافیک را به دو دسته مجاز و غیرمجاز تقسیم‌بندی کنیم. به معنای دسته‌بندی یک سری از اشیا مرتبط با هم با استفاده از تمام اطلاعاتی هست که داریم. برای این کار لازم است که در ابتدا، برچسب‌های تعیین کننده کلاس را برای توزیع اولیه از ترافیک نمونه داشته‌باشیم و سپس از این برچسب‌ها در دور بعدی دسته‌بندی استفاده کنیم. لذا برای دسته‌بندی کل ترافیک، یک سری نمونه‌ها بایستی با استفاده از اطلاعات موجود برچسب‌گذاری شوند و کلاس متناظر آن‌ها (از بین چندین کلاس) تعیین شود. پس مسله اصلی، تعیین اون نمونه‌های اولیه می‌باشد. بدیهی ترین راه استفاده از نودهای تصادفی که برچسب دارند، می‌باشد یا نودهایی که کل ترافیک را تخمین می‌زنند. این رویکرد ما یادگیری افزایشی یا نتیجه‌گیری فعال[15] می‌باشد.

یادگیری فعال: نوع خاصی از یادگیری نیمه نظارتی. به کاربر یا آن شخص خبره برای برچسپ گذاری داده‌ها، کویری می‌زند(دایماً در حال تعامل). از آنجا که یادگیرنده نمونه‌های آزمایشی را انتخاب می‌کند، تعداد آنها اغلب اوقات از تعداد مورد نیاز در یادگیری‌های نظارتی کمتر می‌باشد.



[14] Collective Classification
[15] Active inference

استفاده از دیتاست‌های گفته‌شده و انتخاب آن پنج ویژگی گفته‌شده و به کمک دو ابزار editcap و Tshark . آدرس‌های آی‌پی باید به دسیمال تبدیل شوند.

Data components : چندین نوع داده داریم:

تمام برچسب‌های داده‌های مربوط به train و test را در pool جمع کرده‌ایم و تناظر یک به یک بین داده‌ها و برچسب‌ها را نیز داریم. هر زمان که داده‌های train برای طبقه بندی‌کننده SVM لازم بود بروز شود، برچسب‌های train به سیستم خبره داده می‌شود. Best Possible Support Vector شامل یک مجموعه از support vector های دایماً در حال آپدیت می‌باشد. بعد از اتمام داده‌های آموزشی، این کامپوننت شامل وکتورهای پشتیبانی برای طبقه‌بندی داده‌های تست می‌باشد.

Process Components:

- Selection of training data: یک سری نمونه‌های تصادفی را از train انتخاب می‌کنیم. و با یادگیری از طریق اینها، n SVM تا support vector تولید می‌دهد. با استفاده از این وکتورها، ماژول خبره نمونه‌های train با برچسب‌های متناظرشان از آن pool برچسب‌ها، به درخواست این کامپوننت فراهم می‌کند. 3xn تا نمونه از داده‌های train در سیستم خبره در هر دور ارزیابی می‌شوند. نمونه‌های مناسب توسط این کامپوننت در حین آموزش SVM به روز می‌شوند.
- Expert Module: نمونه‌هایی که ارزیابی می‌کند، سه برابر support vector های تولید شده توسط SVM می‌باشد، سیستم خبره این داده‌های آموزشی را به منظور پیداکردن نمونه‌های موردنیاز طبق استراتژی‌ای که در بعد راجع بهش صحبت می‌کنیم، پردازش می‌کند. نمونه‌های داده و کلاس انتخاب شده با مجموعه داده آموزشی پیشین، به منظور آموزش بیشتر در دور بعد ادغام می‌شوند که در نتیجه منجر به دستیابی به support vector و تعیین مرزهای دقیق‌تر خواهد شد.

در آزمایش‌های مختلف با تغییر تعداد داده‌های نمونه‌های اولیه آموزشی، تعداد نمونه‌های آموزشی در batch، مجموع نمونه داده‌های آموزشی و دقت طبقه‌بندی کننده را برای هر ۴ تا دیتاست اندازه می‌گیریم. نتایجی که می‌گیریم بدین صورت می‌باشد:

- طبقه‌بندی کننده مبتنی بر یادگیری فعال برای طبقه‌بندی ترافیک بسیار موثر می‌باشد.
- دقت طبقه‌بندی کننده به تعداد نمونه‌های batch، تعداد نمونه‌های اولیه، مجموع تعداد نمونه‌های داده آموزشی بستگی دارد.
- تعداد دورهای مناسب برای شناسایی بهترین support vector نیز وابسته به متغیرهای بالا می‌باشد.
- برای یک مجموعه نمونه آموزشی، میتوان بهترین support vector را با استفاده از یادگیری افزایشی برای بهبود دقت طبقه‌بندی برای نمونه داده تست انتخاب کرد
- با استفاده ازماژول خبره که به صورت غیرنظارتی عمل می‌کند، می توان نمونه‌های آموزنده را انتخاب کرد
- یادگیری فعال برای دیتاست‌های مختلف استحکام و دقت بالایی دارا می‌باشد