

تاریخ دریافت مقاله: ۹۵/۰۹/۱۸

تاریخ پذیرش مقاله: ۹۵/۱۱/۳۰

طبقه‌بندی ترافیک شبکه با استفاده از الگوریتم جنگل تصادفی بهبودیافته

زهرا امینی خویی

کارشناس ارشد دانشکده مهندسی - گروه کامپیوتر - دانشگاه کردستان - سنندج - ایران
پست الکترونیکی: zamini69@gmail.com

علیرضا عبدالله پوری*

استادیار دانشکده مهندسی - گروه کامپیوتر - دانشگاه کردستان - سنندج - ایران
پست الکترونیکی: Abdollahpour@uok.ac.ir

چکیده

برای ساخت درختان تصمیم و نحوه وزن دهی به درختان هنگام رای گیری می‌باشد. برای ارزیابی و مقایسه روش پیشنهادی از شش طبقه‌بند دیگر یادگیری ماشین شامل: الگوریتم شبکه عصبی چندلایه، ماشین بردار پشتیبان، درخت تصمیم C4.5، الگوریتم نزدیک‌ترین همسایه، بیز ساده و جنگل تصادفی ساده استفاده کرده‌ایم. نتایج آزمایش‌ها بر روی مجموعه داده جریان‌های ترافیک واقعی UNIBS نشان می‌دهد که روش پیشنهادی عملکرد بهتری نسبت به سایر الگوریتم‌ها ارائه می‌دهد و میانگین دقت طبقه‌بندی روی همه برنامه‌ها به ۹۸/۷۵ درصد می‌رسد.

واژه‌های کلیدی: طبقه‌بندی ترافیک شبکه، الگوریتم جنگل تصادفی بهبودیافته، یادگیری ماشین

۱- مقدمه

با رشد سریع تعداد کاربران و برنامه‌های مختلف در طول چند سال اخیر، ترافیک اینترنت به شدت در حال

با افزایش سریع تعداد کاربران اینترنت و ظهور برنامه‌های جدید، ترافیک اینترنت به شدت در حال رشد است. در نتیجه، شناسایی برنامه‌ها در شبکه به امر پیچیده‌ای تبدیل شده است. از طرف دیگر، طبقه‌بندی جریان‌ها نقش مهمی در امنیت و مدیریت شبکه دارد. تکنیک‌های سنتی طبقه‌بندی ترافیک به بازرسی مستقیم بسته‌های جریان وابسته بودند. اخیراً، با توجه به محدودیت‌های روش‌های قبلی، از الگوریتم‌های یادگیری ماشین به منظور بهبود دقت طبقه‌بندی جریان‌های ترافیک، استفاده شده است. الگوریتم‌های یادگیری ماشین با استفاده از اطلاعات آماری جریان‌ها روی بسته‌ها، توانایی بالایی در طبقه‌بندی جریان‌های ترافیک شبکه دارند. در این مقاله، ما تغییراتی را در الگوریتم جنگل تصادفی که جزء الگوریتم‌های با نظراست، اعمال کرده و آن را جنگل تصادفی بهبودیافته نامگذاری کرده‌ایم. تغییرات داده شده به صورت انتخاب‌های پارامتری در رابطه با معیار مناسب

* نویسنده مسئول

افزایش است. سهم قابل توجهی از ترافیک اینترنت را برنامه‌های نظیر به نظیر، شکل می‌دهند. وبگاه‌های مختلف چندرسانه‌ای و پروتکل‌های معمول مانند HTTP، ایمیل، انتقال فایل و غیره نیز سهم زیادی در این ترافیک دارند. بنابراین، چگونگی شناسایی ترافیک شبکه به یک مسئله قابل توجه و پیچیده تبدیل شده است.

علاوه بر این، طبقه‌بندی ترافیک شبکه کاربرد اساسی در سیستم‌های مدیریت، طراحی و امنیت شبکه دارد [۱]. هدف از طبقه‌بندی ترافیک، ایجاد ارتباط بین بسته‌های یک جریان با یک سرویس یا برنامه خاص است که آن‌ها را تولید کرده است [۲].

روش‌های اولیه طبقه‌بندی ترافیک اینترنت مبتنی بر بازرسی بسته‌های جریان بودند. روش مبتنی بر شماره درگاه، شماره درگاه در سرآیند بسته‌ها را با شماره‌های درگاه ثبت شده توسط «مرجع شماره‌های اختصاص داده شده اینترنت» [۳] برای برنامه‌های خاص، مقایسه می‌کند. این روش روی جریان‌های با شماره درگاه پویا قابل اجرا نیست. روش طبقه‌بندی مبتنی بر payload، داده‌های مربوط به کاربر را برای پیدا کردن هر گونه الگوی مرتبط با یک برنامه خاص جستجو می‌کند. این روش روی بازرسی داده‌های کاربر تکیه دارد و در نتیجه، باعث نقص حریم خصوصی کاربر می‌شود. روش مبتنی بر رفتار میزبان، مستقل از بازرسی بسته‌های جریان، با نظارت بر همه جریان‌های ارسالی یا دریافتی روی میزبان‌های شبکه، می‌تواند ترافیک ایجاد شده توسط برنامه‌ها را طبقه‌بندی نماید. این روش مبتنی بر این فرض (معمولا غیر واقعی) است که یک میزبان در هر لحظه تنها یک برنامه را اجرا می‌کند؛ اما در واقعیت کاربران بیش از یک برنامه را به‌طور هم‌زمان استفاده می‌کنند [۴].

در سال‌های اخیر، به طبقه‌بندی ترافیک اینترنت بر اساس روش‌های یادگیری ماشین توجه زیادی شده است. تکنیک‌های یادگیری ماشین، با استفاده از مجموعه ویژگی‌های آماری جریان به‌طور خودکار

الگوهای ساختاری موجود در انتقال داده‌های جریان را کشف می‌کنند. این روش می‌تواند مشکلاتی مانند شماره درگاه پویا، عدم حفظ حریم خصوصی کاربران و فرض عدم اجرای هم‌زمان چند برنامه روی یک میزبان را رفع نماید.

هدف از این پژوهش، ارائه یک روش مبتنی بر یادگیری ماشین برای طبقه‌بندی ترافیک شبکه است که بتواند جریان‌ها را با شماره درگاه پویا و بدون وابستگی به بازرسی Payload بسته‌ها و بدون بررسی رفتارهای میزبان روی شبکه، طبقه‌بندی نماید. علاوه بر این، الگوریتم پیشنهادی، باید نسبت به سایر الگوریتم‌های طبقه‌بندی موجود روی ترافیک شبکه، دقت شناسایی بالاتری را روی مجموعه داده ترافیک واقعی اینترنت ارائه دهد.

براین اساس، در این مقاله الگوریتم یادگیری ماشین جنگل تصادفی بهبود یافته برای بهبود دقت شناسایی جریان‌های ترافیک پیشنهاد می‌شود. این الگوریتم با انجام تغییراتی روی جنگل تصادفی ساده به دست می‌آید. تغییرات داده شده شامل انتخاب پارامترهای مناسب در رابطه با نحوه وزن دهی به درختان هنگام رای گیری و انتخاب معیار بهره‌آطلاعاتی برای ساخت درختان تصمیم می‌باشد.

برای ارزیابی، روش پیشنهادی را با الگوریتم‌های درخت تصمیم C4.5، ماشین بردار پشتیبان، شبکه عصبی پرسپترون، الگوریتم نزدیک‌ترین همسایه، الگوریتم بیزین ساده و جنگل تصادفی ساده مقایسه خواهیم کرد. برای به دست آوردن مجموعه داده ترافیک واقعی، از مجموعه ردیابی بسته‌های ترافیک اینترنت UNIBS [۵] استفاده کرده‌ایم.

ادامه مقاله به این صورت سازمان‌دهی شده است. در بخش دوم، روش‌های مختلف موجود برای طبقه‌بندی ترافیک شبکه و کارهای مرتبط بیان شده است. در بخش سوم روش پیشنهادی و انواع ویژگی‌ها و پارامترهای آن مورد بحث قرار گرفته شده است. آزمایش‌ها و تنظیمات

انجام شده روی الگوریتم‌ها در بخش چهارم آورده شده است. در بخش پنجم نتایج آزمایش‌ها و خروجی الگوریتم‌ها با یکدیگر مقایسه شده است. در نهایت، نتیجه‌گیری در بخش ششم ارائه شده است.

۲- انواع روش‌های طبقه‌بندی ترافیک و کارهای مرتبط
تکنیک‌های مختلفی برای طبقه‌بندی ترافیک شبکه اینترنت استفاده شده است. ما در این بخش هر یک از تکنیک‌های موجود همراه با برخی کارهای مرتبط با آن‌ها را آورده‌ایم.

۱،۲. روش مبتنی بر درگاه

شناخته‌ترین و قدیمی‌ترین روش مورد استفاده برای طبقه‌بندی ترافیک اینترنت، تطبیق شماره درگاه است. در این روش، از شماره درگاه مقصد در سرآیند لایه انتقال بسته برای شناسایی ترافیک استفاده می‌شود. مقدار شماره درگاه با لیست شماره درگاه‌های تعیین شده، برای شناسایی بسته جاری مقایسه می‌شود. اما برنامه‌های جدید بخصوص برنامه‌های نظیر به نظیر از روش‌های مختلف برای پنهان کردن خود به منظور فرار از تشخیص استفاده می‌کنند. آن‌ها از درگاه‌های پویا و یا درگاه‌های دیگر برنامه‌های شناخته شده در اتصالاتشان استفاده می‌کنند. این وضعیت باعث می‌شود که روش شناسایی مبتنی بر درگاه روشی کم اثر و حتی در بعضی مواقع و روی برنامه‌های ناشناخته کاملاً بی‌اثر باشد. این روش حداکثر می‌تواند ۷۰ درصد ترافیک اینترنت را تشخیص دهد [۶]. همچنین، در [۷] نیز نشان داده شده که این روش حدود ۳۰ تا ۷۰ درصد ترافیک اینترنت را نمی‌تواند شناسایی کند.

۲،۲. روش مبتنی بر بازرسی Payload

این روش بر اساس تجزیه و تحلیل Payload بسته‌های جریان است که در آن محتوای بسته‌ها را برای پیدا کردن امضای برنامه‌های شناخته شده جستجو می‌کند. محتوای

بسته اطلاعات زیادی دارد که می‌توان با تجزیه و تحلیل آن به شناسایی دقیق‌تری دست یافت.

سن^۲ و همکاران [۸] با بازرسی محتوای هر یک از بسته‌های ترافیک امضاها را بر سطح برنامه را برای برنامه‌های P2P تولید می‌کنند. این روش به‌طور قابل توجهی می‌تواند تخمین ترافیک P2P را بیش از آنچه روش مبتنی بر درگاه ارائه می‌داد، بهبود ببخشد. مور^۳ و همکاران [۹] با استفاده از روش بازرسی Payload نشان دادند که طبقه‌بندی‌های بر اساس شماره درگاه‌های شناخته شده ناکارآمد هستند و مقدار زیادی از ترافیک را یا به صورت ناشناس یا به اشتباه طبقه‌بندی می‌کنند. در حالی که روش پیشنهادی ایشان، هیچ جریانی را به عنوان جریان ناشناس طبقه‌بندی نمی‌کند و رده‌ای که اصلاً در روش شماره درگاه شناسایی نشد را شناسایی می‌کند.

Appmon یک برنامه برای طبقه‌بندی برنامه‌های ترافیک شبکه است که توسط آنتونیادیز^۴ و همکاران ارائه شده است [۱۰]. این برنامه به‌طور متوالی هر بسته را با جدول درهم‌سازی^۵ که جریان‌های طبقه‌بندی شده قبلی در آن ذخیره شده است، مقایسه می‌کند و در صورت تطبیق با جریانی از جدول، بسته به آن جریان طبقه‌بندی می‌شود و در غیر این صورت، در سطح بعدی پردازش، از ردیاب بازرسی بسته برای ردیابی پروتکل‌ها در سطح برنامه استفاده می‌شود. روش طبقه‌بندی مبتنی بر بازرسی Payload بسیار دقیق است، اما چند مشکل اساسی دارد. مهم‌تر از همه این‌که نمی‌توان آن را روی بسته‌های رمزگذاری شده اعمال کرد. تجزیه و تحلیل مستقیم داده Payload باعث نقض سیاست‌های حفظ حریم خصوصی سازمانی یا نقض حقوق قانونی خصوصی می‌شود. همچنین به حافظه و زمان بسیار زیادی برای پردازش بسته‌های شبکه نیاز دارد.

۳،۲. روش مبتنی بر رفتار میزبان

ایده اصلی این روش این است که برنامه‌های مختلف

2-Sen
3-Moore
4-Antoniades
5-Hash table

الگوهای اتصال متفاوتی دارند. روش مبتنی بر رفتار میزبان، ارتباطات میان میزبان‌های خاص در یک شبکه را شناسایی می‌کند و الگوی ارتباط یک میزبان خاص با الگوی رفتار فعالیت‌های متفاوت مقایسه می‌شود. این روش Payload بسته را برای طبقه‌بندی ترافیک استفاده نمی‌کند. در نتیجه می‌تواند بسته‌ها با محتوای رمزگذاری شده را شناسایی کند. بیشتر تکنیک‌های مبتنی بر رفتار میزبان فرض می‌کنند که میزبان‌های تحت نظارت در یک لحظه تنها از یک برنامه استفاده می‌کنند و در واقعیت، این وضعیت ممکن است هرگز اتفاق نیفتد. بیشتر کاربران از برنامه‌های زیادی به‌طور هم‌زمان استفاده می‌کنند. روش‌های مبتنی بر رفتار میزبان ممکن است ترافیک برنامه‌های دیگر را به برنامه هدف طبقه‌بندی کنند. این پیش‌فرض، در واقع تحقیق تکنیک‌های مبتنی بر میزبان را ساده کرده است، اما یک چالش بزرگ برای استفاده از آن است [۱۱].

کاراجیانیس^۶ و همکاران [۱۲] یک روش شناسایی جریان‌های ترافیک P2P بر اساس الگوهای اتصال جریان ترافیک P2P ارائه کرده‌اند. این روش قادر به شناسایی جریان‌هایی است که در تجزیه و تحلیل Payload از دست رفته‌اند و حدود ۱۰٪ بیشتر از تجزیه و تحلیل Payload جریان‌های P2P را شناسایی می‌کند. کاراجیانیس و همکاران همچنین رفتار میزبان را در سه سطح مورد بررسی قرار دادند: سطح اجتماعی، سطح کاربردی و سطح برنامه. آن‌ها توانستند به دقت شناسایی ترافیک بیش از ۸۰ درصد دست یابند. این روش قادر به شناسایی دقیق برنامه‌ها نیست که این می‌تواند برای برنامه‌هایی که از لحاظ نظری از گروه‌های متفاوتی هستند اما رفتار مشابهی دارند یک مشکل باشد [۱۳].

یک چارچوب مبتنی بر گراف برای طبقه‌بندی ترافیک P2P توسط ایلیفوتو و همکاران در [۱۴] ارائه شده است. آن‌ها با استفاده از گراف‌های پراکنده، ترافیک تعاملات گسترده شبکه را نمایش داده‌اند. این گراف قادر به تشخیص رفتارهایی از شبکه است که در میان

برنامه‌های P2P مشترک بوده و با دیگر برنامه‌ها تفاوت دارد.

تکنیک‌های مبتنی بر میزبان قادر به شناسایی نوع یک برنامه هستند، اما نمی‌توانند انواع زیربرنامه‌های آن را طبقه‌بندی کنند. به‌عنوان مثال، آن‌ها می‌توانند جریان‌های P2P را شناسایی کنند، اما تعیین نوع برنامه P2P (به‌عنوان مثال eMule یا بیت‌تورنت) که این جریان را تولید کرده، دشوار است [۱۲]. این روش محدودیت‌های دیگری نیز دارد. برخی الگوهای رفتاری برنامه را نمی‌توان به‌آسانی کشف کرد به مقدار حافظه و جریان‌های زیادی برای همه میزبان‌ها نیاز دارد تا بتواند الگوی اتصال را کشف کند.

۴.۲. روش مبتنی بر یادگیری ماشین

یادگیری ماشین^۷ مجموعه‌ای از تکنیک‌ها برای داده‌کاوی و کشف دانش است که الگوهای ساختاری مفید در داده‌ها را جستجو می‌کند. در سال ۱۹۹۴ برای اولین بار از روش‌های یادگیری ماشین برای مطالعه طبقه‌بندی جریان در تشخیص نفوذ استفاده شد [۱۵]. پس از سال ۲۰۰۴، با افزایش سریع مقیاس و پیچیدگی شبکه، برخی از محدودیت‌های روش‌های موجود شناسایی شد و محققان از الگوریتم‌های یادگیری ماشین برای طبقه‌بندی استفاده کردند [۱۶]. ورودی الگوریتم‌های یادگیری ماشین یک مجموعه داده از نمونه‌ها است. هر نمونه از مجموعه داده توسط مقادیر ویژگی‌های آن توصیف می‌شود. در زمینه شبکه بسته‌های متوالی از یک جریان یک نمونه جریان را به وجود می‌آورند. هر نمونه جریان مجموعه‌ای از ویژگی‌ها است که روی جریان تعریف می‌شوند.

۱- الگوریتم‌های یادگیری ماشین دو مرحله دارند. مرحله اول، آموزش است که یادگیری انجام می‌شود و در آن مجموعه داده آموزش بررسی می‌شود و یک مدل طبقه‌بندی بر اساس آن ساخته می‌شود. مرحله دوم، آزمایش است و مدلی که در مرحله آموزش ساخته شده

6-Karagiannis

7-Machin Learning

برای رده‌بندی نمونه‌های جدید مشاهده نشده استفاده می‌شود.

این تکنیک طبقه‌بندی برای غلبه بر محدودیت‌های روش‌های قبلی ارائه شده است. هر جریان، با مجموعه‌ای از ویژگی‌های آماری و مقادیر مرتبط با آن‌ها توصیف می‌شود. یک ویژگی آماری روی بسته‌های یک جریان محاسبه می‌شود، مانند متوسط طول بسته و یا انحراف معیار زمان رسیدن بسته. الگوریتم‌های یادگیری ماشین روی ترافیک شبکه را می‌توان به سه دسته تقسیم کرد.

۲،۴،۱. الگوریتم‌های یادگیری با نظارت ۲. الگوریتم‌های یادگیری بدون نظارت ۳. الگوریتم‌های یادگیری نیمه نظارتی.

۱،۴،۲. الگوریتم‌های یادگیری با نظارت

روش‌های یادگیری بانظارت، مبتنی بردانش از پیش تعریف‌شده هستند. این الگوریتم‌ها در مرحله آموزش، نمونه‌های از پیش طبقه‌بندی‌شده (متشکل از ویژگی‌ها و برچسب مرتبط با آن‌ها) را به‌عنوان ورودی می‌گیرند و قوانین طبقه‌بندی ایجاد می‌شود. در مرحله طبقه‌بندی تلاش می‌کنند تا برچسب نمونه‌های بدون برچسب را پیش‌بینی کنند.

مور و همکارش در سال ۲۰۰۵ الگوریتم انتخاب ویژگی FCBF را برای از بین بردن ویژگی‌های زائد استفاده کرده‌اند و سپس برای طبقه‌بندی ترافیک یک طبقه‌بند بیز با تخمین هسته تولید کردند [۱۷]. در طبقه‌بند بیز ساده دقت ۶۵ درصد به دست می‌آید، درحالی‌که روش الگوریتم بیز مبتنی بر هسته همراه با روش انتخاب ویژگی به دقت بیش از ۹۵ درصد منجر می‌شود.

ژانگ^۸ و همکاران در [۱۸]، جریان‌های وابسته به هم را تعیین کرده و با استفاده از آن‌ها، مدل مجموعه جریان‌ها را می‌سازند. اگر جریان‌های مشاهده‌شده در یک بازه زمانی خاص، درگاه مقصد، IP مقصد و پروتکل لایه انتقال یکسانی داشته باشند، آن‌ها به عنوان جریان‌های وابسته

تعیین می‌شوند. در مرحله اول، یک طبقه‌بند بیزین ساده منفرد احتمالات رده را برای هر جریان تولید می‌کند. در مرحله دوم، پیش‌بینی‌های بیزین‌های ساده جریان جمع‌آوری می‌شود تا رده نهایی برای مجموعه جریان‌ها را تعیین کند. روش پیشنهادی آن‌ها با روش‌های طبقه‌بندی مثل بیزین ساده، درخت تصمیم و نزدیک‌ترین همسایه مقایسه شده است.

در مقاله [۱۹]، جمونا^۹ و اواردز^{۱۰} روش‌های مختلف یادگیری ماشین با نظارت مانند C4.5، بیز ساده، نزدیک‌ترین همسایه و شبکه عصبی RBF برای طبقه‌بندی ترافیک مبتنی بر جریان را استفاده نموده‌اند و عملکرد آن‌ها را با یکدیگر مقایسه کرده‌اند. الگوریتم درخت تصمیم C4.5 با دقت ۹۳/۳۳ درصد در مقایسه با الگوریتم‌های دیگر به عملکرد بهتری دست می‌یابد.

۲،۴،۲. الگوریتم‌های یادگیری بدون نظارت

روش‌های یادگیری بدون نظارت، در مرحله یادگیری خود به هیچ دانش از قبل تعیین‌شده‌ای نیاز ندارند. روش‌های بدون نظارت به عنوان روش‌های خوشه‌بندی شناخته شده‌اند. این روش‌ها به یک مجموعه داده آموزش با برچسب نیاز نداشته، گروه‌های طبیعی (خوشه‌ها) را در داده‌ها کشف می‌کنند و روی کشف الگوهای موجود در داده‌ها تمرکز دارند. نمونه‌ها را بر اساس میزان شباهت ویژگی‌های آن‌ها که توسط یک رویکرد اندازه‌گیری فاصله تعریف می‌شود مانند فاصله اقلیدسی به گروه‌ها خوشه‌بندی می‌کنند.

مک گرگور^{۱۱} و همکاران در سال ۲۰۰۴ از روش خوشه‌بندی EM برای طبقه‌بندی جریان‌ها استفاده کرده‌اند [۲۰]. هدف پیدا کردن مجموعه خوشه‌ها با بیشترین شباهت از داده‌های آموزش است. این روش جریان‌های شبکه را به گروه‌های مختلف از برنامه‌ها بر اساس ویژگی‌های قابل مشاهده مشابه جریان‌ها گروه‌بندی

9-Jamuna
10-Ewards
11-McGregor

8-Zhang

می‌کند. با این حال، نویسندگان دقت خوشه‌بندی را ارزیابی نمی‌کنند.

در مقاله [۲۱] زاندر^{۱۲} و همکارانش از روش یادگیری ماشین بدون نظارت Autoclass برای طبقه‌بندی ترافیک و شناسایی برنامه‌ها استفاده کردند. الگوریتم انتخاب ویژگی SFS به منظور پیدا کردن بهترین مجموعه ویژگی‌ها انتخاب شده است. دقت روش برای برخی از برنامه‌های نزدیک به ۹۵ درصد می‌رسد ولی میانگین دقت روی تمام برنامه‌ها ۸۶/۵ درصد است.

ینگ کیو^{۱۳} و همکاران الگوریتم K-means را در سال ۲۰۰۷ برای شناسایی ترافیک شبکه به کار برده‌اند و به منظور بهبود دقت طبقه‌بندی، الگوریتم انتخاب ویژگی و تبدیل لگاریتم را اضافه کرده‌اند [۲۲]. نتایج آزمایش‌ها با الگوریتم K-means بر روی مجموعه داده‌های مختلف نشان می‌دهد که دقت این روش با افزایش تعداد خوشه‌ها بهبود می‌یابد. تبدیل لگاریتم دقت را حداقل ۱۰٪ بهبود می‌بخشد و زمانی که K برابر با ۸۰ است، دقت به ۹۰٪ می‌رسد.

هنگام ارزیابی الگوریتم یادگیری ماشین بدون نظارت در چارچوب عملیاتی، روش برچسب زدن خوشه‌ها و نگاشت آن‌ها به برنامه‌ها مهم است. همچنین بررسی چگونگی برچسب زدن باید با برنامه‌های جدید که شناسایی می‌شوند، به‌روزرسانی شود و پیچیدگی محاسباتی و هزینه‌های برچسب زدن نیز باید بهینه باقی بماند [۲۳].

۳.۴.۲. الگوریتم‌های یادگیری نیمه نظارتی

یک الگوریتم یادگیری ماشین نیمه نظارتی، حدوسط یادگیری ماشین تحت نظارت و یادگیری ماشین بدون نظارت است. یادگیری نیمه نظارتی قادر به استفاده از مقدار کمی داده‌های آموزشی برچسب دار و تعداد زیادی داده‌ها بدون برچسب است.

یک روش نیمه نظارتی توسط شریواستاو^{۱۴} و تیواری^{۱۵} در سال ۲۰۱۰ برای طبقه‌بندی جریان‌های

پایه‌سازی شده است [۲۴]. مرحله آموزش روش نیمه نظارت شامل دو مرحله خوشه‌بندی و طبقه‌بندی است. الگوریتم خوشه‌بندی K-Means مجموعه آموزش داده متشکل از جریان‌های با برچسب و جریان‌های بدون برچسب را به خوشه‌های مشابه گروه‌بندی می‌کند. پس از خوشه‌بندی داده‌های آموزشی، از جریان‌های با برچسب برای نگاشت خوشه‌ها به رده‌های شناخته‌شده استفاده می‌شود. در مرحله آزمایش، از نمونه‌های غیرآموزشی و بدون برچسب استفاده می‌شود. نتایج آزمایش برای این روش در مقایسه با طبقه‌بند SVM به دقت ۹۵/۶۵ درصد دست می‌یابد، در حالی که دقت روش طبقه‌بند SVM روی مجموعه داده آزمایشی ۷۶ درصد است.

ژانگ و همکارانش در سال ۲۰۱۲ استفاده از جریان‌های وابسته در هر دو مرحله آموزش و آزمایش را پیشنهاد دادند [۲۵]. در مرحله آموزش استفاده از جریان وابسته، مجموعه داده تحت نظارت را گسترش می‌دهد. در مرحله آزمایش، جریان‌های وابسته شناسایی می‌شوند و با ترکیب پیش‌بینی‌های فردی روی هر یک از جریان‌ها، مجموعه جریان‌های وابسته طبقه‌بندی می‌شوند. دقت طبقه‌بندی برای روش پیشنهادی به صورت میانگین حدود ۹۶ درصد است. با این حال، روش‌های یادگیری ماشین نیمه نظارتی ممکن است در فرایند یادگیری‌شان مخصوصاً زمانی که تعداد داده‌های آموزشی برچسب دار کم هستند، گمراه شوند.

۳. روش پیشنهادی

روش پیشنهادی که یک الگوریتم یادگیری ماشین است، تنها از اطلاعات موجود در سرآیند بسته‌ها استفاده می‌کند و به هیچ وجه با داده‌های کاربران و همچنین اطلاعات ارتباطی بین میزبان‌ها سروکار ندارد.

در روش پیشنهادی از یادگیری بانظارت استفاده می‌کنیم، چراکه در الگوریتم‌های یادگیری بدون نظارت هزینه زیادی برای برچسب زدن نمونه‌ها نیاز است و در

12-Zander
13-Yingqiu
14-Shrivastav
15-Tiwari

الگوریتم‌های یادگیری نیمه نظارتی نیز همیشه تعدادی خوشه در مرحله آموزش به وجود می‌آیند که هیچ نمونه برچسب داری در آن‌ها قرار نمی‌گیرد. همچنین، نتایج الگوریتم‌های با نظارت روی طبقه‌بندی شبکه بهتر است. روش پیشنهادی ما بهبود یافته روش جنگل تصادفی است و نشان خواهیم داد نسبت به دیگر روش‌های یادگیری ماشین با نظارت، دقت شناسایی بالاتری ارائه خواهد داد. ما در اینجا از نسبت واقعی تعداد جریان‌ها روی ترافیک استفاده می‌کنیم و ثابت می‌کنیم که روش پیشنهادی روی جریان‌ها با تعداد کم نیز بسیار خوب و با فاصله از سایر الگوریتم‌ها عمل می‌کند. الگوریتم پیشنهادی، با تغییراتی نسبت به جنگل تصادفی ساده آن را بهبود می‌دهد. بنابراین، برای درک بهتر آن، ابتدا مفهوم روش جنگل تصادفی ساده را تشریح می‌کنیم و در ادامه، تغییرات صورت گرفته بر روی آن و فرآیند تولید روش الگوریتم جنگل تصادفی بهبود یافته را بیان خواهیم کرد.

۱.۱.۳. الگوریتم جنگل تصادفی ساده

الگوریتم جنگل تصادفی یک الگوریتم گروهی با مجموعه‌ای از درختان تصمیم است. دقت طبقه‌بندی روش جنگل تصادفی با ساخت مجموعه‌ای از درختان و رأی‌گیری بین آن‌ها برای به دست آوردن رده‌ای با بیشترین تعداد رأی، پیشرفت‌های قابل توجهی داشته است. دو ویژگی مهم در ساخت جنگل‌های تصادفی، روش بگینگ^{۱۶} [۲۶] و انتخاب تصادفی در هر گره است. در ادامه، ویژگی‌ها و خصوصیات جنگل تصادفی ساده توضیح داده شده‌اند.

۱.۱.۳. روش بگینگ

روش بگینگ توسط لئو بریمن^{۱۷} در سال ۱۹۹۶ مطرح شد. این روش یک فرا الگوریتم بر مبنای مفاهیم خود راه‌انداز^{۱۸} و ترکیب، برای بهبود یادگیری ماشین است. الگوریتم‌های گروهی در یادگیری ماشین، چند یادگیرنده ضعیف را ترکیب می‌کنند تا به یک یادگیرنده قوی دست یابند. این روش از بیش برآزش^{۱۹}

16- Bagging
17- Leo Breiman
18- Bootstrapping
19- Overfitting

داده‌ها جلوگیری می‌کند. در بگینگ، نتایج خوب زمانی تولید می‌شود که طبقه‌بندهای پایه جزء الگوریتم‌های یادگیری ناپایدار باشند (مانند درخت تصمیم‌گیری یا شبکه عصبی)، به طوری که تغییرات کوچک در داده‌های آموزشی منجر به تغییرات عمده‌ای در مدل ساخته شده توسط آن الگوریتم شود.

فرآیند الگوریتم بگینگ بدین شرح می‌باشد؛ یک مجموعه آموزش D به اندازه m را در نظر بگیرید. بگینگ با نمونه‌گیری یکنواخت و با جایگزینی نمونه‌ها از D ، n مجموعه آموزش جدید D_i با اندازه اولیه m تولید می‌کند. نمونه‌گیری با جایگزینی این امکان را می‌دهد که در هر D_i بعضی از نمونه‌ها امکان تکرار داشته باشند. این نوع نمونه‌گیری به عنوان نمونه‌گیری خودراه‌انداز شناخته می‌شود. خروجی ترکیب n مدل با میانگین‌گیری برای رگرسیون و رأی‌گیری برای طبقه‌بندی به دست می‌آید. بنابراین، با استفاده از نمونه‌گیری دوباره و تولید مجموعه داده‌های مختلف، تنوع مورد نیاز حاصل خواهد شد.

۲.۱.۳. ویژگی‌های تصادفی

خصوصیت ویژگی‌های تصادفی، بدین صورت است که در هر گره از ساخت هر درخت، به تصادف یک گروه کوچک از ویژگی‌های ورودی انتخاب می‌شود و برای تقسیم گره به جای جستجو از میان همه ویژگی‌ها، از میان ویژگی‌های این زیرگروه، بهترین ویژگی با بیشترین بهره اطلاعاتی برای رشد درخت انتخاب می‌شود. تعداد این ویژگی‌ها کمتر از تعداد ویژگی‌های اصلی است. هر درخت در جنگل تصادفی با استفاده از الگوریتم درخت تصمیم کارت^{۲۰} و با حداکثر اندازه و بدون هرس رشد می‌کند. بریمن در هر گره از تعداد $1 + \log_2 M$ ویژگی استفاده کرده است که در آن M تعداد کل ویژگی‌های ورودی است [۲۸].

۳.۱.۳. تعریف جنگل تصادفی

جنگل تصادفی یک طبقه بند مجموعه‌ای متشکل از طبقه‌بندهای درخت تصمیم است. هر طبقه بند برای هر

20-CART

(۲)

$$I(h_k(x) = c, x \in OOB_k) = \begin{cases} 1 & h_k(x) = c, x \in OOB_k \\ 0 & \text{Otherwise} \end{cases}$$

که K تعداد درختان، c نشان‌دهنده رده، $h_k(x)$ پیش‌بینی درخت k ام روی نمونه x را نشان می‌دهند و OOB_k مجموعه نمونه‌های OOB درخت k ام می‌باشند. رابطه ۲ نشان می‌دهد که مقدار تابع شاخص I یک خواهد بود اگر x در مجموعه نمونه‌های درخت k ام قرار دارد (عضو مجموعه آموزش درخت k ام نیست) و همچنین درخت k ام نمونه x را به رده c طبقه‌بندی کند. در غیر این صورت، مقدار تابع شاخص صفر می‌شود.

برای به دست آوردن تخمین نمونه‌های OOB روی جنگل از $error_k(OOB)$ در رابطه ۳ استفاده می‌کنیم که خطای طبقه‌بندی جنگل روی نمونه‌های OOB درخت k ام می‌باشد.

$$I((y_i, x_i) \in OOB_k) = \begin{cases} 1 & (x_i, y_i) \in OOB_k \\ 0 & (x_i, y_i) \notin OOB_k \end{cases} \quad (۳)$$

$$error_k(OOB) = \frac{\sum_{i=1}^N I(y(x_i) \neq y_i : (x_i, y_i) \in OOB_k)}{\sum_{i=1}^N I((x_i, y_i) \in OOB_k)} \quad (۴)$$

N تعداد همه نمونه‌های مجموعه آموزش اصلی، x_i نمونه i ام روی مجموعه آموزش اصلی، y_i رده واقعی x_i ، $y(x_i)$ رده پیش‌بینی شده برای x_i برحسب رابطه ۱ است. در رابطه ۴ مقدار تابع I یک خواهد بود، اگر نمونه (x_i, y_i) متعلق به مجموعه OOB درخت k باشد و در غیر این صورت، صفر است.

۵.۱.۳. قدرت^{۲۱} و وابستگی^{۲۲}

در جنگل تصادفی، برای خطای یک کران بالا در نظر گرفته می‌شود. دو پارامتر برای این کران بالا اندازه‌گیری می‌شود. پارامتر اول، قدرت طبقه‌بندی فردی و پارامتر دوم، وابستگی بین درختان جنگل است.

$$PE^* \leq \bar{\rho}(1 - s^2)/s^2 \quad (۵)$$

در رابطه ۵، s قدرت طبقه‌بندی فردی در جنگل و $\bar{\rho}$

نمونه ورودی به صورت $h(x, \theta_k)$ است، که x یک نمونه ورودی و θ_k مجموعه آموزش برای درخت k ام است. θ ها مستقل از یکدیگر ولی با توزیع یکسان هستند. برای هر نمونه x ، هر درخت یک پیش‌بینی را برای رده نمونه x ارائه می‌دهد و در نهایت رده‌ای با بیشترین تعداد رأی درختان روی ورودی x به عنوان رده نمونه انتخاب می‌شود. این فرآیند را جنگل تصادفی می‌نامند [۲۸].

الگوریتم جنگل تصادفی می‌تواند دقت پیش‌بینی را نسبت به درخت طبقه‌بند فردی افزایش دهد. در درخت فردی با تغییرات کوچک در مجموعه آموزش بی‌ثباتی به وجود می‌آید که باعث اختلال در دقت پیش‌بینی در نمونه آزمایشی می‌شود. اما گروهی بودن الگوریتم جنگل تصادفی باعث سازگاری با تغییرات می‌شود و بی‌ثباتی را از بین می‌برد.

۴.۱.۳. تخمین خارج از کیسه

فرض کنید هر طبقه‌بند با مجموعه آموزشی جدید با روش درخت تصمیم ساخته می‌شود. با توجه به مجموعه آموزش θ و با روش خودراه‌انداز مجموعه‌های آموزشی θ_k تشکیل می‌شوند. سپس طبقه‌بندی درخت $h(x, \theta_k)$ ساخته می‌شود و از هر درخت برای پیش‌بینی رده رأی‌گیری می‌شود. نمونه‌های آموزش در مجموعه داده آموزش اصلی که در مجموعه آموزش طبقه‌بند k نیست، نمونه‌های خارج از کیسه طبقه‌بند k ام نامیده می‌شود.

در هر مجموعه آموزش به دست آمده از روش خودراه‌انداز، نمونه‌های OOB در حدود یک‌سوم از نمونه‌های مجموعه آموزش اصلی است که در مجموعه آموزش قرار نمی‌گیرند. رابطه ۱ شیوه تخمین رده نمونه OOB را روی جنگل نشان می‌دهد. برای به دست آوردن رده نمونه باید ابتدا پیش‌بینی درختانی که مجموعه آموزش آن‌ها حاوی نمونه نیست، جمع‌آوری شود و سپس رده‌ای با بیشترین میانگین رأی روی پیش‌بینی‌های درختان جنگل به عنوان رده نمونه در نظر گرفته می‌شود.

(۱)

$$y(x) = \arg \max_c \left(\frac{1}{K} \sum_{k=1}^K I(h_k(x) = c, x \in OOB_k) \right)$$

وابستگی بین درختان جنگل را نشان می‌دهد. بر اساس این رابطه برای خطای PE^* هرچه مقدار S بیشتر و مقدار \bar{p} کمتر باشد، میزان خطا نیز کمتر خواهد بود. نسبت $\frac{\bar{p}}{S^2}$ برای جنگل تصادفی در درک عملکرد آن، یک راهنمای مفید است. این نسبت، تقسیم وابستگی بر مربع قدرت است و هر چه کوچکتر باشد، عملکرد جنگل بهتر و خطا نیز کمتر است.

۶.۱.۳. پیش‌بینی رده توسط جنگل تصادفی ساده

برای به دست آوردن رده نمونه آزمایشی برای جنگل تصادفی ساده از پیش‌بینی تمام درختان جنگل استفاده می‌شود. برای تعیین رده نمونه آزمایشی از روابط ۶ و ۷ استفاده می‌کنیم.

$$y(x) = \arg \max_c \left\{ \frac{1}{K} \sum_{k=1}^K I(h_k(x) = c) \right\} \quad (۶)$$

$$I(h_k(x) = c) = \begin{cases} 1 & h_k(x) = c \\ 0 & \text{Otherwise} \end{cases} \quad (۷)$$

۲.۳. جنگل تصادفی بهبودیافته

در این مقاله، برای بهبود روش جنگل تصادفی، یک تغییر در مرحله آموزش و یک تغییر در مرحله طبقه‌بندی به جنگل تصادفی ساده اعمال می‌کنیم تا الگوریتم جنگل تصادفی بهبودیافته را به دست آوریم.

در مرحله آموزش، در جنگل تصادفی ساده برای ساخت هر درخت از معیار Gini Index (معیار ساخت درخت Cart) برای انتخاب بهترین ویژگی در هر گره استفاده می‌شود. اما در اینجا، برای به دست آوردن طبقه‌بندهای قوی‌تر از معیار بهره اطلاعات^{۳۳} استفاده می‌کنیم.

در مرحله طبقه‌بندی، جنگل تصادفی ساده تمرکز روی پیش‌بینی رده بر اساس اکثریت آرا است و ضریب مشارکت همه درختان با دقت‌های مختلف در رأی‌گیری یکسان است. بنابراین، اغلب برای رده‌ای با تعداد کمتر (با تخمین بالا) دقت پایینی دارد. برای این مشکل، ما یک راه‌حل پیشنهاد داده‌ایم. در مرحله آموزش و پس از ساخت درخت، با

23-Information Gain

استفاده از نمونه‌های OOB آن درخت، دقت طبقه‌بندی را فقط روی آن درخت به دست می‌آوریم و دقت درخت را به عنوان وزن آن درخت در نظر می‌گیریم و در مرحله طبقه‌بندی به منظور بهبود پیش‌بینی رده نهایی، از وزن هر درخت به دست آمده برای طبقه‌بندی نمونه آزمایشی استفاده می‌کنیم. این باعث می‌شود که درختان با دقت بالاتر وزن بیشتری در طبقه‌بندی داشته باشند.

برای درک بهتر روش پیشنهادی، الگوریتم ساخت هر درخت در جنگل به شکل روندنما در شکل ۱ آورده شده است.

۱.۲.۳. معیار بهره اطلاعات

بهره اطلاعات $Information\ Gain(S, A)$ برای یک ویژگی نظیر A نسبت به مجموعه نمونه‌های S در رابطه ۸ تعریف می‌شود که در آن $Values(A)$ مجموعه همه مقدار ویژگی‌های A بوده و S_v زیرمجموعه‌ای از S است که برای آن ویژگی A دارای مقدار v است. $|S_v|$ تعداد نمونه‌ها با مقدار v روی ویژگی A در مجموعه S و $|S|$ تعداد کل نمونه‌ها روی مجموعه S است.

(۸)

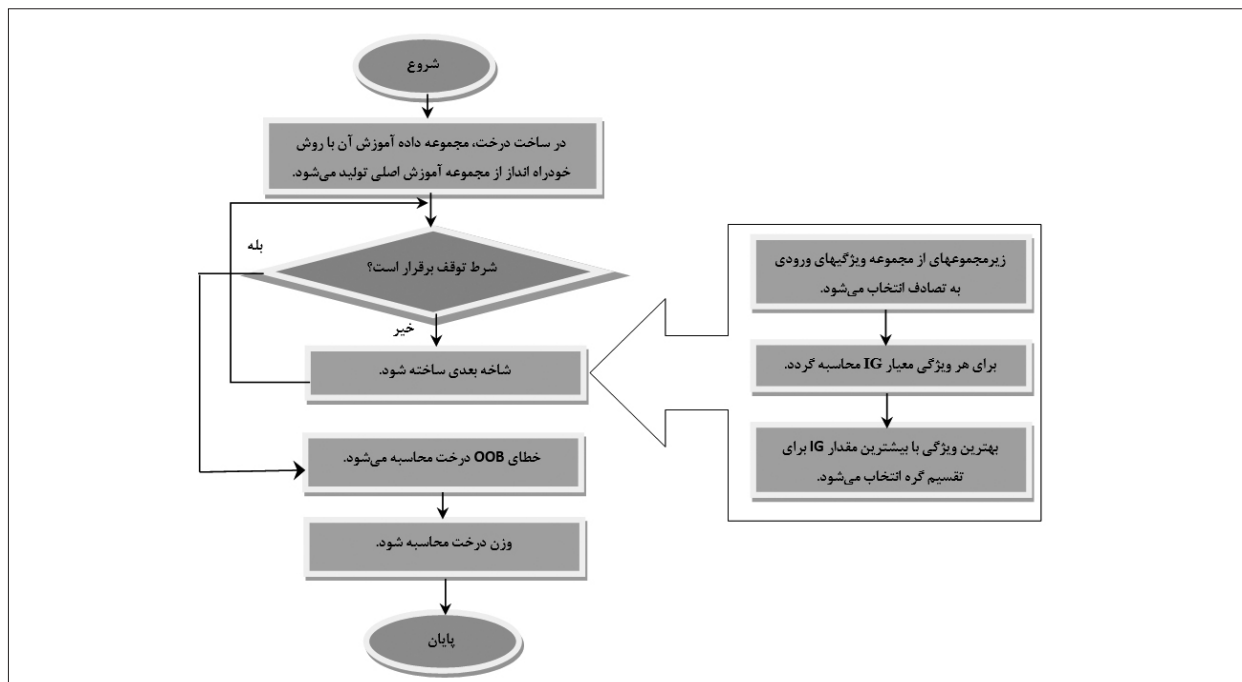
$$Information\ Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} Entropy(S_v)$$

$$entropy(S) = \sum_{i=1}^{numclass} -p_i \log_2 p_i \quad (۹)$$

آنتروپی از رابطه ۹ محاسبه می‌شود. p_i نسبت نمونه‌ها در S که به رده نام تعلق دارد، می‌باشد.

۲.۲.۳. مرحله آموزش IRF

در فرآیند ایجاد جنگل، در ساخت هر درخت و برای تقسیم هر گره، یک زیرمجموعه از ویژگی‌های ورودی را به تصادف انتخاب می‌کنیم و برای به دست آوردن بهترین ویژگی در گره برای ایجاد شاخه، از معیار بهره اطلاعات استفاده می‌شود. هر درخت روی جنگل مجموعه نمونه OOB مخصوص به خود را دارد. دقت هر درخت را با استفاده از نمونه‌های OOB آن درخت به دست می‌آوریم و آن را به عنوان وزن آن درخت در نظر می‌گیریم. وزن هر درخت را از رابطه ۱۰ محاسبه می‌کنیم.



شکل ۱: روند کلی الگوریتم جنگل تصادفی بهبود یافته برای هر درخت

پیش‌بینی شده توسط جنگل تصادفی بهبود یافته برای نمونه x است، K تعداد درختان و W_k وزن درخت k ام می‌باشد.

$$(12)$$

$$y(x) = \arg \max_c \left\{ \frac{1}{K} \sum_{k=1}^K W_k, I(h_k(x) = c) \right\}$$

۴. آزمایش‌ها و تحلیل نتایج

در این بخش تعریف معیارهای ارزیابی، شرایط آزمایش‌ها، تنظیم مجموعه داده و پارامتر الگوریتم‌ها در پیاده‌سازی آورده شده است.

۱.۴. معیارهای ارزیابی

از معیارهای Accuracy و F-measure طبق تعاریف زیر برای ارزیابی و مقایسه عملکرد روش پیشنهادی با سایر الگوریتم‌ها استفاده می‌شود.

معیار Accuracy: نسبت تعداد جریان‌های صحیح طبقه‌بندی شده تقسیم بر تعداد کل جریان‌ها می‌باشد.

معیار F-measure: این معیار ترکیبی از معیارهای Precision و Recall است که از رابطه ۱۳ محاسبه می‌شود.

$$f_measure = \frac{2 * Precision * recall}{precision + recall} \quad (13)$$

$$W_k = \frac{\sum_{i=1}^N I(h_k(x_i) = y_i, (x_i, y_i) \in OOB_k)}{\sum_{i=1}^N I((x_i, y_i) \in OOB_k)} \quad (10)$$

$$(11)$$

$$I(h_k(x_i) = y_i, (x_i, y_i) \in OOB_k) = \begin{cases} 1 & h_k(x_i) = y_i \\ 0 & h_k(x_i) \neq y_i \end{cases}$$

N تعداد همه نمونه‌های مجموعه آموزش اصلی، x_i نمونه نام روی مجموعه آموزش اصلی، y_i رده واقعی x_i و تابع شاخص I در صورت کسر رابطه ۱۰، از رابطه ۱۱ به دست می‌آید.

۳.۲.۳. مرحله آزمایش IRF

در این مرحله برای ارزیابی الگوریتم از مجموعه داده آزمایشی استفاده می‌کنیم که رده نمونه‌های آزمایشی بر اساس وزن درختان جنگل پیش‌بینی می‌شود. برای به دست آوردن رده هر نمونه همه درختان جنگل شرکت دارند و از رابطه ۱۲ استفاده می‌کنیم که براساس وزن درختان به دست آمده در مرحله آموزش است. برای هر رده c مجموعه وزن درختانی که رده c را برای نمونه x پیش‌بینی می‌کنند به دست می‌آوریم و سپس رده‌ای با بیشترین میانگین وزن روی درختان به عنوان رده آزمایشی در نظر گرفته می‌شود. در رابطه زیر $y(x)$ رده

جدول ۱: انواع برنامه‌ها و نوع پروتکل‌ها و تعداد جریان‌ها در هر برنامه

برنامه‌ها	پروتکل‌ها	تعداد جریان‌ها
Web	HTTP و HTTPS	۴۸۹۹
Mail	POP3، IMAP4، SMTP، SSL	۵۱۰
Skype	Skype	۳۲۲
P2P	eDonkey، Bittorrent	۲۲۶۹

ما در تمامی آزمایش‌ها برای ارزیابی از روش 10-fold cross validation استفاده می‌کنیم.

۲.۴. مجموعه داده

به منظور ارزیابی عملکرد روش طبقه‌بندی روی داده‌های واقعی ترافیک از مجموعه داده UNIBS استفاده می‌کنیم که از ردیابی جریان‌های اینترنت در دانشگاه برشیا جمع‌آوری شده است. ترافیک شامل برنامه‌های وب (پروتکل‌های HTTP و HTTPS)، پست الکترونیکی (پروتکل‌های POP3، IMAP4، SMTP و SSL)، اسکایپ، برنامه‌های P2P (bittorrent و eDonkey) و پروتکل‌های دیگر مانند SSH، FTP و MSN است.

۳.۴. پیش‌پردازش

با استفاده از ۵ تایی مشخصه جریان (شماره درگاه مبدأ، شماره درگاه مقصد، نشانی IP مبدأ، نشانی IP مقصد، پروتکل لایه انتقال) جریان‌های روی بسته‌ها را به دست می‌آوریم. در انتها، برای مجموعه داده خود یک زیرمجموعه داده با ۸۰۰۰ جریان به صورت تصادفی انتخاب می‌کنیم.

توزیع برنامه‌ها در مجموعه داده به توزیع و تعداد برنامه‌ها روی مجموعه داده اصلی UNIBS نزدیک است. ما مجموعه داده را روی چهار نوع رده Web، Mail، Skype و برنامه‌های P2P تعریف می‌کنیم. جزئیات در جدول ۱ گزارش شده است.

۴.۴. مجموعه ویژگی‌ها و شرایط آزمایش

۲۲ نوع ویژگی را برای هر جریان به صورت دوطرفه به دست می‌آوریم. در جدول ۲ جزئیات ویژگی‌های

جدول ۲: نوع ویژگی‌ها و تعداد آن‌ها روی مجموعه داده

نوع ویژگی‌ها	تعداد
تعداد بسته‌های ارسالی در هر جهت	۲
حجم ارسالی در هر جهت	۲
کمترین، بیشترین، میانگین و انحراف معیار زمان بین رسیدن بسته‌ها در هر جهت	۸
کمترین، بیشترین، میانگین و انحراف معیار اندازه بسته در هر جهت	۸
شماره درگاه مبدأ و شماره درگاه مقصد	۲

به کار برده شده برای هر جریان آمده است. ۲۰ نوع ویژگی، ویژگی‌های آماری روی اندازه بسته، زمان رسیدن بسته و تعداد بسته‌ها تعریف شده است [۷-۱۰]. دو ویژگی شماره درگاه مبدأ و شماره درگاه مقصد نیز در مجموعه ویژگی‌ها قرار دارد.

پارامترهای الگوریتم‌های مختلف در پیاده‌سازی را بر اساس جدول ۳ تنظیم می‌کنیم. برای بررسی عملکرد الگوریتم‌ها نسبت به افزایش تعداد نمونه جریان‌ها در مجموعه داده آموزش، ما مجموعه اولیه را به ۸ مجموعه تقسیم می‌کنیم. این ۸ مجموعه با مقادیر ۱۰۰۰ تا ۸۰۰۰ نمونه جریان در هر مجموعه داده تولید شده‌اند. در هر مجموعه داده نسبت جریان‌ها مشابه با نسبت اصلی جریان‌ها روی ترافیک است. کد تمامی الگوریتم‌ها با زبان برنامه‌نویسی متلب پیاده‌سازی شده است.

۵. تحلیل نتایج

در این قسمت خروجی الگوریتم‌های مختلف روی معیارهای Accuracy و F-measure را آورده‌ایم و نتایج را با یکدیگر مقایسه نموده و تجزیه و تحلیل کرده‌ایم.

۱.۵. معیار Accuracy

جدول ۴ نتایج الگوریتم‌های مختلف را روی مجموعه جریان‌ها بر اساس معیار Accuracy نمایش می‌دهد. بیشترین مقدار در تمام الگوریتم‌ها روی تعداد نمونه جریان‌های مختلف در هر ستون جدول، به صورت

جدول ۴: نتایج الگوریتم‌های مختلف بر اساس معیار Accuracy روی تعداد نمونه‌های متفاوت

تعداد داده‌ها	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰
C4.5	۹۶/۷۸	۹۵/۴۶	۹۶/۴۷	۹۶/۷۲	۹۷/۱۲	۹۶/۸۷	۹۷/۳۳	۹۷/۰۴
NN	۸۱/۴۵	۸۴/۸۹	۸۷/۳۲	۸۸/۱۳	۸۹/۴۹	۹۰/۲۱	۹۱/۰۲	۹۱/۷۹
SVM	۸۲/۷۲	۷۹/۶۲	۸۲/۶۱	۸۰/۶۱	۸۱/۳۹	۸۲/۲۱	۸۳/۶۷	۸۴/۴۹
MLP	۸۷/۴۱	۹۱/۱۳	۹۲/۷۷	۹۲/۲۱	۹۳/۸۸	۹۳/۸۸	۹۴/۹۷	۹۵/۲۹
NB	۹۰/۲۸	۹۱/۷۵	۹۲/۱۰	۹۲/۲۳	۹۲/۳۶	۹۲/۲۳	۹۱/۶۷	۹۰/۸۶
RF	۹۴/۷۹	۹۵/۵۰	۹۷/۲۷	۹۷/۴۵	۹۷/۸۶	۹۷/۵۲	۹۸/۳۱	۹۸/۱۸
RF_IG	۹۵/۷۲	۹۶/۶۴	۹۷/۵۷	۹۷/۷۰	۹۷/۸۸	۹۸/۲۲	۹۸/۳۱	۹۸/۷۳
IRF	۹۶/۲۱	۹۶/۸۴	۹۷/۶۷	۹۷/۷۷	۹۷/۹۴	۹۸/۳۰	۹۸/۴۰	۹۸/۷۵

جدول ۳: تنظیم پارامتر الگوریتم‌های مختلف

الگوریتم	پارامتر	تنظیم
C4.5	معیار بهره اطلاعات	Gain Ratio
MLP	نرخ یادگیری نرخ شتاب تعداد تکرار	۰/۳ ۰/۲ ۵۰۰
NB	هسته تخمین احتمال پسین	هسته تخمین چگالی
NN	K نوع فاصله	۱ فاصله اقلیدسی
SVM	تابع هسته	RBF
RF	تعداد درختان تعداد ویژگی‌ها معیار بهره اطلاعات	۵۰ ۴ Gini Index
IRF	تعداد درختان تعداد ویژگی‌های معیار بهره اطلاعات	۵۰ ۴ Information Gain

بهبود می‌یابند. اما الگوریتم NB روند تغییرات ثابتی ندارد و حتی به صورت نزولی عمل می‌کند. اما برای سایر الگوریتم‌ها، روند صعودی‌شان نشان‌دهنده این است که زمانی که تعداد داده‌های آموزش بیشتر می‌شود، یادگیری بهتر صورت می‌گیرد و دقت طبقه‌بندی بیشتر خواهد شد. به طور خاص روش IRF یک روند کاملاً صعودی دارد و با افزایش تعداد نمونه‌ها از ۱۰۰۰ تا ۸۰۰۰ نمونه، مقدار دقت طبقه‌بندی آن از ۹۶/۲۱ به ۹۸/۷۵ رسیده است و حدود ۲/۵۲ درصد بهبود و افزایش دقت داشته است.

۲.۵. ارزیابی معیار F-measure روی رده‌های مختلف

ردهٔ وب بیشترین تعداد جریان را روی ترافیک اینترنت دارد و بیش از ۶۰ درصد جریان‌ها متعلق به برنامه‌های ردهٔ وب هستند. بنابراین، طبقه‌بندی و شناسایی درست این جریان‌ها روی دقت طبقه‌بندی الگوریتم‌ها بسیار مؤثر است. جدول ۵ نتایج معیار F-measure برای ردهٔ وب روی الگوریتم‌های متفاوت را نشان می‌دهد. روش پیشنهادی IRF در همهٔ تعداد نمونه‌های جریان به بهترین نتایج دست‌یافته است و با افزایش تعداد نمونه‌های جریان روی یک روند صعودی حرکت می‌کند و بیشترین مقدار آن، نزدیک به ۹۹ درصد است.

ردهٔ اسکایپ کمترین تعداد جریان را روی مجموعه داده ترافیک دارد. نتایج معیار F-measure برای ردهٔ

پرنرگ و زیرخطدار، نمایش داده شده است. الگوریتم RF_IG، الگوریتم جنگل تصادفی ساده به همراه معیار بهرهٔ اطلاعات می‌باشد. مقادیر الگوریتم RF_IG دقت شناسایی الگوریتم جنگل تصادفی ساده را بهبود می‌بخشد ولی روش پیشنهادی IRF با تغییر روی مرحله طبقه‌بندی الگوریتم RF_IG روی معیار دقت، دوباره بهبود نشان می‌دهد. مقادیر به دست آمده توسط این الگوریتم نسبت به الگوریتم‌های RF_IG و RF بیشتر است. روش پیشنهادی IRF در میانگین دقت شناسایی برنامه‌های مختلف، بالاتر از همه الگوریتم‌های مقایسه قرار می‌گیرد و با ۹۸/۷۵ درصد دقت شناسایی روی ۸۰۰۰ نمونه جریان بهترین و بیشترین مقدار را نشان می‌دهد.

الگوریتم SVM با مقادیر کمتر از ۹۰ درصد بدترین نتایج را ارائه می‌دهد و پس از آن به ترتیب الگوریتم‌های NB، NN و MLP قرار دارند که عملکرد پایینی دارند. الگوریتم C4.5 نتایج بهتری دارد و در فاصله نزدیک‌تری به نتایج بهینه قرار می‌گیرد.

همه الگوریتم‌ها به‌غیر از NB روی افزایش تعداد نمونه‌های جریان تقریباً روند صعودی دارند و مقادیر

جدول ۵: نتایج الگوریتم‌های مختلف بر اساس معیار F-measure روی تعداد نمونه‌های مختلف برای ردهٔ وب

تعداد داده‌ها / الگوریتم	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰
C4.5	۹۴/۷۸	۹۵/۴۶	۹۶/۴۷	۹۶/۷۲	۹۷/۱۲	۹۶/۸۷	۹۷/۳۳	۹۷/۰۴
NN	۸۱/۴۵	۸۴/۸۹	۸۷/۳۲	۸۸/۱۳	۸۹/۴۹	۹۰/۲۱	۹۱/۰۲	۹۱/۷۹
SVM	۸۲/۷۲	۷۹/۶۲	۸۲/۶۱	۸۰/۶۱	۸۱/۳۹	۸۲/۲۱	۸۳/۶۷	۸۴/۴۹
MLP	۸۷/۴۱	۹۱/۱۳	۹۲/۷۷	۹۲/۲۱	۹۳/۸۸	۹۳/۸۸	۹۴/۹۷	۹۵/۲۹
NB	۹۰/۲۸	۹۱/۷۵	۹۲/۱۰	۹۲/۲۳	۹۲/۳۶	۹۲/۲۳	۹۱/۶۷	۹۰/۸۶
RF	۹۴/۷۹	۹۵/۵۰	۹۷/۲۷	۹۷/۴۵	۹۷/۸۶	۹۷/۵۲	۹۸/۳۱	۹۸/۱۸
RF_IG	۹۵/۷۲	۹۶/۶۴	۹۷/۵۷	۹۷/۷۰	۹۷/۸۸	۹۸/۲۲	۹۸/۳۱	۹۸/۷۳
IRF	۹۶/۲۱	۹۶/۸۴	۹۷/۶۷	۹۷/۷۷	۹۷/۹۴	۹۸/۳۰	۹۸/۴۰	۹۸/۷۵

جدول ۶: نتایج الگوریتم‌های مختلف بر اساس معیار F-measure روی تعداد نمونه‌های مختلف برای ردهٔ پست الکترونیکی

تعداد داده‌ها / الگوریتم	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰
C4.5	۶۹/۲۵	۷۳/۴۷	۷۷/۶۱	۷۹/۲۷	۸۱/۳۷	۸۰/۲۸	۸۲/۶۹	۸۰/۸۹
NN	۱۹/۰۸	۲۹/۲۱	۳۵/۵۷	۳۸/۰۸	۴۲/۸۴	۴۵/۶۲	۵۰/۱۲	۵۳/۵۰
SVM	۳۰/۹۲	۲۷/۲۹	۳۳/۴۰	۳۱/۰۸	۳۴/۲۶	۳۵/۰۷	۳۸/۳۰	۳۹/۹۲
MLP	۴۵/۵۴	۴۹/۴۵	۵۱/۶۱	۵۳/۶۸	۵۶/۳۵	۵۷/۹۱	۶۲/۸۵	۶۵/۴۷
NB	۴۱/۱۵	۴۸/۷۳	۴۹/۸۵	۵۰/۰۸	۵۴/۱۱	۵۱/۵۳	۵۰/۰۸	۴۸/۹۰
RF	۶۵/۸۷	۶۸/۵۱	۸۰/۵۷	۸۲/۲۵	۸۵/۰۲	۸۲/۸۰	۸۷/۶۶	۸۷/۱۸
RF_IG	۶۸/۷۴	۷۶/۳۰	۸۳/۰۸	۸۳/۵۳	۸۵/۲۷	۸۷/۵۱	۸۸	۹۰/۷۸
IRF	۷۲/۵۸	۷۶/۹۹	۸۳/۶۳	۸۴/۳۸	۸۵/۱۱	۸۷/۸۲	۸۸/۶۹	۹۱/۰۳

جدول ۷: نتایج الگوریتم‌های مختلف بر اساس معیار F-measure روی تعداد نمونه‌های مختلف برای ردهٔ اسکایپ

تعداد داده‌ها / الگوریتم	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰
C4.5	۴۸/۳۷	۴۹/۵۰	۵۹/۹۹	۶۰/۹۲	۶۳/۰۷	۶۳/۴۸	۶۸/۰۳	۶۵/۰۱
NN	۱۱/۳۴	۱۶/۳۱	۱۸/۶۸	۱۸/۷۱	۲۱/۸۵	۲۱/۱۷	۲۶/۲۹	۲۹/۲۶
SVM	۷/۷۳	۱۵/۵۵	۱۵/۷۰	۱۶/۷۲	۲۰/۸۸	۱۸/۷۷	۲۲/۳۴	۲۱/۱۴
MLP	۲۸/۶۷	۳۵/۹۰	۳۹/۳۶	۳۷/۰۶	۳۹/۴۹	۴۳/۷۴	۴۸/۷۵	۴۹/۰۶
NB	۱۷/۹۰	۲۸/۵۳	۲۷/۷۳	۳۲/۴۸	۲۹/۴۹	۳۲/۱۷	۳۲/۲۹	۲۸/۱۲
RF	۳۳/۲۱	۴۷/۵۳	۶۶/۴۶	۶۴/۷۷	۷۱/۷۱	۶۶/۷۲	۷۶/۵۶	۷۵/۴۷
RF_IG	۴۶/۷۹	۵۶/۱۵	۶۷/۵۶	۶۸/۳۰	۶۹/۲۹	۷۶/۴۳	۷۷/۹۸	۸۲/۶۷
IRF	۴۹/۵۵	۵۵/۵۹	۶۸/۹۵	۶۸/۱۰	۷۰/۳۲	۷۶/۹۴	۷۸/۹۸	۸۳/۱۰

اسکایپ روی تعداد نمونه‌های مختلف در جدول ۷ نشان می‌دهد که الگوریتم‌های SVM، NB، NN، MLP و C4.5 نتایج بسیار پایین و در بسیاری از موارد کمتر از ۵۰ درصد دارند که نشان می‌دهد این الگوریتم‌ها روی تعداد نمونه‌های کم بسیار ضعیف عمل می‌کنند. درحالی‌که روش پیشنهادی IRF نتایج خوبی به دست آورده است و بالاتر از سایر الگوریتم‌ها قرار گرفته است.

امروزه، برنامه‌های P2P مهم‌ترین برنامه‌های حاضر در ترافیک شبکه هستند. اهمیت برنامه‌های P2P چندین علت عمده دارد: از نظر حجم، اطلاعات زیادی از طریق این برنامه‌ها مبادله می‌شود، به صورتی که در مجموعه داده مورد استفاده بیش از ۸۶ درصد حجم اطلاعات متعلق به این گروه می‌باشد. همچنین این جریان‌ها زمان‌های اتصال طولانی دارند و از رمزگذاری اطلاعات، فرار از شناسایی شدن و شماره‌های درگاه پویا استفاده می‌کنند. به‌طور قابل توجه و بیش از هر نوع برنامه دیگری تحقیقات و روش‌های مختلف برای شناسایی برنامه‌های P2P انجام شده است که نشان از تأثیر و اهمیت جریان‌های این نوع برنامه روی ترافیک دارد. بنابراین ارائه یک روش که عملکرد مناسبی در طبقه‌بندی این برنامه داشته باشد، ضروری به نظر می‌رسد.

یکی از نقاط قوت کار ما استفاده از تعداد جریان‌ها بر اساس نسبت واقعی آن‌ها روی ترافیک است که نشان می‌دهیم با همین نسبت واقعی روش پیشنهادی دقت شناسایی بالایی را به دست می‌آورد. درحالی‌که در اکثر کارهای صورت گرفته بخصوص در شناسایی برنامه‌های P2P تعداد هر یک از برنامه‌ها را در مجموعه داده برابر در نظر می‌گیرند. در مجموعه داده ما حدود ۲۸ درصد از تعداد جریان‌های ترافیک به

جدول ۸: نتایج الگوریتم‌های مختلف بر اساس معیار F-measure روی تعداد نمونه‌های مختلف برای رده P2P

تعداد داده‌ها	۱۰۰۰	۲۰۰۰	۳۰۰۰	۴۰۰۰	۵۰۰۰	۶۰۰۰	۷۰۰۰	۸۰۰۰
C4.5	۴۸/۳۷	۴۹/۵۰	۵۹/۹۹	۶۰/۹۲	۶۳/۰۷	۶۳/۴۸	۶۸/۰۳	۶۵/۰۱
NN	۱۱/۳۴	۱۶/۳۱	۱۸/۶۸	۱۸/۷۱	۲۱/۸۵	۲۱/۱۷	۲۶/۲۹	۲۹/۲۶
SVM	۷/۷۳	۱۵/۵۵	۱۵/۷۰	۱۶/۷۲	۲۰/۸۸	۱۸/۷۷	۲۲/۳۴	۲۱/۱۴
MLP	۲۸/۶۷	۳۵/۹۰	۳۹/۳۶	۳۷/۰۶	۳۹/۴۹	۴۳/۷۴	۴۸/۷۵	۴۹/۰۶
NB	۱۷/۹۰	۲۸/۵۳	۲۷/۷۳	۳۲/۴۸	۲۹/۴۹	۳۲/۱۷	۳۲/۲۹	۲۸/۱۲
RF	۳۳/۲۱	۴۷/۵۳	۶۶/۴۶	۶۴/۷۷	۷۱/۷۱	۶۶/۷۲	۷۶/۵۶	۷۵/۴۷
RF_IG	۴۶/۷۹	۵۶/۱۵	۶۷/۵۶	۶۸/۳۰	۶۹/۲۹	۷۶/۴۳	۷۷/۹۸	۸۲/۶۷
IRF	۴۹/۵۵	۵۵/۵۹	۶۸/۹۵	۶۸/۱۰	۷۰/۳۲	۷۶/۹۴	۷۸/۹۸	۸۳/۱۰

برنامه‌های P2P تعلق دارند.

جدول ۸ نشان‌دهنده نتایج الگوریتم‌ها بر اساس معیار F-measure برای رده P2P با افزایش تعداد نمونه‌های جریان است. با توجه به مقادیر این جدول می‌توان نتیجه گرفت که روش پیشنهادی IRF عملکرد بسیار مناسبی روی رده برنامه‌های مهم P2P دارد و بیشترین دقت شناسایی با حدود ۹۸ درصد توسط این الگوریتم به دست می‌آید.

۶. نتیجه‌گیری

شناسایی جریان جاری روی ترافیک اینترنت روی جنبه‌های مختلف شبکه مانند مدیریت، امنیت و غیره تأثیر زیادی دارد. همچنین تشخیص جریان‌های ترافیک باعث برنامه‌ریزی صحیح در قسمت‌های مختلف مانند تخصیص منابع، بهبود کیفیت خدمات سرویس، و تشخیص برنامه‌های مخرب می‌شود. بنابراین، با توجه به اهمیت شناسایی جریان‌های ترافیک اینترنت، ما در این مقاله مسئله طبقه‌بندی ترافیک اینترنت را مطرح کردیم و الگوریتم جدیدی را برای حل این مسئله پیشنهاد دادیم. روش پیشنهادی جنگل تصادفی بهبود یافته، یک روش یادگیری ماشین با نظارت گروهی است که از مجموعه‌ای از درختان تصمیم ساخته می‌شود. برای هر درخت روی این جنگل مجموعه آموزش جدیدی با روش خودراه‌اندازی ساخته می‌شود و نمونه‌های باقی‌مانده از مجموعه آموزش اصلی که در مجموعه آموزش درخت وجود ندارند، در یک مجموعه به نام OOB برای ارزیابی درخت قرار می‌گیرند. در ساخت هر درخت و در تقسیم هر گره، انتخاب ویژگی

از زیرمجموعه تصادفی از ویژگی‌های اصلی انجام می‌شود. با استفاده از نمونه‌های OOB دقت درخت روی نمونه‌های آموزش را به دست آورده و به‌عنوان وزن درخت قرار می‌گیرد و در مرحله آزمایش با تغییر در نوع رأی‌گیری روی درختان بر اساس وزن آن‌ها رده نمونه آزمایشی مشخص می‌شود. سپس عملکرد روش پیشنهادی را با روش‌هایی که در تحقیقات قبلی دقت خوبی را گزارش کرده بودند، روی مجموعه ترافیک جریان‌های واقعی UNIBS با مقایسه کردیم. نتایج مقایسه‌ها و ارزیابی‌ها نشان می‌دهد که روش پیشنهادی نسبت به سایر روش‌ها دقت شناسایی بالاتری دارد و در شناسایی برنامه‌ها با تعداد نمونه‌های آموزش کم عملکرد قابل قبولی را ارائه می‌دهد. دقت طبقه‌بندی آن روی هر کدام از برنامه‌های مجموعه داده نسبت به سایر الگوریتم‌ها بالاتر است و میانگین دقت طبقه‌بندی روی همه برنامه‌ها به ۹۸/۷۵ درصد می‌رسد.

مراجع

- [1] Zhang, J., Chen, C., Xiang, Y., Zhou, W., and Xiang, Y., "Internet Traffic Classification by Aggregating Correlated Naive Bayes Predictions," IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY, vol. 8, pp. 5-15, JANUARY 2013.
- [2] Casas, P., Mazel, J., and Owezarski, P., "MINE-TRAC: Mining Flows for Unsupervised Analysis & Semi-Supervised Classification," presented at the Proceedings of the 23rd International Teletraffic Congress, San Francisco, 2011.
- [3] IANA, "Internet Assigned Numbers Authority," <http://www.iana.org/assignments/port-numbers>.
- [4] Xue, Y., Wang, D., and Zhang, L., "Traffic Classification: Issues and Challenges," presented at the International Conference on Computing, Networking and Communications, San Diego, CA 2103.
- [5] <http://netweb.ing.unibs.it/~ntw/tools/traces/>. (Last visited December 7th, 2016)
- [6] Moore, A. W., and Papagiannaki, K., "Toward the Accurate Identification of Network Applications." vol. 3431, C. Dovrolis, Ed., ed: Springer Ber-

traffic classification and application identification using machine learning,” in *Local Computer Networks*, 2005. 30th Anniversary. 2005, pp. 250-257.

[22] Yingqiu, L., Wei, L., and Yunchun, L., “Network traffic classification using k-means clustering,” in *Computer and Computational Sciences*, 2007. IMSCCS 2007, pp. 360-365.

[23] Nguyen, T. T., and Armitage, G., “A survey of techniques for internet traffic classification using machine learning,” *Communications Surveys & Tutorials*, IEEE, vol. 10, pp. 56-76, 2008.

[24] Shrivastav, A., and Tiwari, A., “Network traffic classification using semi-supervised approach,” in *Machine Learning and Computing (ICMLC)*, 2010, pp. 345-349.

[25] Zhang, J., Chen, C., Xiang, Y. and Zhou, W., “Semi-supervised and compound classification of network traffic,” *International Journal of Security and Networks*, vol. 7, pp. 252-261, 2012.

[26] Breiman, L. “Bagging predictors,” *Machine learning*, vol. 24, pp. 123-140, 1996.

[27] Dietterich, T. G. “An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization,” *Machine learning*, vol. 40, pp. 139-157, 2000.

[28] Breiman, L. “Random forests,” *Machine learning*, vol. 45, pp. 5-32, 2001.

lin Heidelberg, 2005, pp. 41-54.

[7] Madhukar, A., and Williamson, C., “A Longitudinal Study of P2P Traffic Classification,” presented at the 14th IEEE International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2006.

[8] Sen, S., Spatscheck, O., and Wang, D., “Accurate, scalable in-network identification of p2p traffic using application signatures,” presented at the 13th international conference on World Wide Web, New York, USA, 2004.

[9] Moore, A. W. and Papagiannaki, K., “Toward the accurate identification of network applications,” in *Passive and Active Network Measurement*, ed: Springer, 2005, pp. 41-54.

[10] Antoniadis, D., Polychronakis, M., Antonatos, S., Markatos, E. P., Ubik, S., and Øslebø, A., “Appmon: An application for accurate per application network traffic characterization,” presented at the BroadBand Europe, Geneva, Switzerland, 2006.

[11] Xue, Y., Wang, D., and Zhang, L., “Traffic Classification: Issues and Challenges,” presented at the 2013 International Conference on Computing, Networking and Communications (ICNC), San Diego, CA 2013.

[12] Karagiannis, T., Broido, A., Faloutsos, M., and claffy, K., “Transport Layer Identification of P2P Traffic,” presented at the 4th ACM SIGCOMM conference on Internet measurement New York, USA, 2004.

[13] Karagiannis, T., Papagiannaki, K., and Faloutsos, M., “BLINC: Multilevel Traffic Classification in the Dark,” *Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA 2005.

[14] Iliofotou, M., Kim, H. c., Faloutsos, M., Mitzenmacher, M., Pappu, P., and Varghese, G., “Graph-based P2P Traffic Classification at the Internet Backbone,” presented at the IN-FOCOM Workshops 2009, Rio de Janeiro

[15] Frank, J., “Machine learning and intrusion detection: Current and future directions,” presented at the 17th National Computer Security Conference, Washington, DC, USA, 1994.

[16] Hu, B., and Shen, Y., “Machine Learning Based Network Traffic Classification: A Survey,” *Journal of Information & Computational Science*, pp. 3161-3170, 2012.

[17] Moore, A. W., and Zuev, D., “Internet traffic classification using bayesian analysis techniques,” in *ACM SIGMETRICS Performance Evaluation Review*, 2005, pp. 50-60.

[18] Zhang, J., Chen, C., Xiang, Y., Zhou, W., and Xiang, Y., “Internet traffic classification by aggregating correlated naive bayes predictions,” *Information Forensics and Security*, IEEE Transactions on, vol. 8, pp. 5-15, 2013.

[19] Jamuna, A., and Edwards, S.E, “Efficient Flow based Network Traffic Classification using Machine Learning,” *International Journal of Engineering Research and Applications (IJERA)*, vol. 3, pp. 1324-1328, 2013.

[20] McGregor, A., Hall, M., Lorier, P., and Brunskill, J., “Flow clustering using machine learning techniques,” in *Passive and Active Network Measurement*, ed: Springer, 2004, pp. 205-214.

[21] Zander, S., Nguyen, T., and Armitage, G. “Automated