تاریخ دریافت مقاله: ۹۵/۰۹/۱۸
تاریخ پذیرش مقاله: ۹۵/۱۱/۳۰

طبقه‌بندی ترافیک شبکه با استفاده از الگوریتم جنگل تصادفی بهبودیافته

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

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

چکیده

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 به‌منظور 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 ناظراست، اعمال کرده و آن را جنگل تصادفی بهبودیافته نامگذاری کرده‌ایم. تغییرات داده شده به صورت انتخاب‌های پارامتری در رابطه با معیار مناسب 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

* نویسندهٔ مسئول

افزایش اســت. ســهم قابل توجهی از ترافیــک اینترنت را برنامه‌های نظیربه‌نظیر، شــکل می‌دهند. وبگاه‌های مختلف چندرســانه‌ای و پروتکل‌های معمــول مانند HTTP، ایمیل، انتقال فایل و غیره نیز ســهم زیادی در این ترافیک دارند. بنابراین، چگونگی شناســایی ترافیک شبکه به یک مسئله قابل توجه و پیچیده تبدیل شده است.

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

روش‌هــای اولیه طبقه‌بندی ترافیــک اینترنت مبتنی بر بازرسی بسته‌های جریان بودند. روش مبتنی بر شماره درگاه[1]، شــماره درگاه در سرآیند بسته‌ها را با شماره‌های درگاه ثبت‌شده توسط «مرجع شــماره‌های اختصاص داده‌شده اینترنت» [۳] برای برنامه‌های خاص، مقایســه می‌کند. این روش روی جریان‌هــای با شــماره درگاه پویا قابل اجرا نیست. روش طبقه‌بندی مبتنی بر payload، داده‌های مربوط به کاربر را بــرای پیدا کردن هر گونه الگوی مرتبط با یک برنامه خاص جســتجو می‌کند. این روش روی بازرســی داده‌هــای کاربر تکیه دارد و درنتیجــه، باعث نقص حریم خصوصی کاربر می‌شــود. روش مبتنی بر رفتار میزبان، مســتقل از بازرسی بســته‌های جریان، با نظارت بر همه جریان‌های ارســالی یا دریافتی روی میزبان‌های شــبکه، می‌تواند ترافیک ایجادشده توســط برنامه‌ها را طبقه‌بندی نماید. این روش مبتنی بــر این فرض (معمولا غیر واقعی) اســت که یک میزبان در هر لحظــه تنها یک برنامه را اجرا می‌کند؛ اما در واقعیت کاربران بیش از یک برنامه را به‌طور هم‌زمان استفاده می‌کنند [۴].

در ســال‌های اخیــر، به طبقه‌بنــدی ترافیک اینترنت بر اســاس روش‌هــای یادگیری ماشــین توجه زیادی شده است. تکنیک‌های یادگیری ماشــین، با استفاده از مجموعه ویژگی‌هــای آماری جریان به‌طــور 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 ماشــین برای طبقه‌بندی ترافیک شــبکه اســت که بتواند جریان‌هــا را با شــمارهٔ درگاه پویا و بدون وابســتگی به بازرســی Payload بســته‌ها و بدون بررســی رفتارهای میزبان روی شبکه، طبقه‌بندی نماید. علاوه بر این، الگوریتم پیشــنهادی، باید نسبت به ســایر الگوریتم‌های طبقه‌بندی موجود روی ترافیک شبکه، دقت شناسایی بالاتری را روی مجموعه داده ترافیک واقعی اینترنت ارائه دهد.

براین اســاس، در این مقاله الگوریتم یادگیری ماشین جنــگل تصادفــی بهبودیافته برای بهبود دقت شناســایی جریان‌های ترافیک پیشنهاد می‌شود. این الگوریتم با انجام تغییراتــی روی جنگل تصادفی ســاده به دســت می‌آید. تغییرات داده شده شــامل انتخاب پارامترهای مناسب در رابطــه با نحوه وزن دهی به درختــان هنگام رای گیری و انتخاب معیار بهره اطلاعاتی برای ساخت درختان تصمیم می‌باشد.

بــرای ارزیابی، روش پیشــنهادی را بــا الگوریتم‌های درخت تصمیم C4.5، ماشین بردار پشتیبان، شبکه عصبی پرسپترون، الگوریتم نزدیک‌ترین همسایه، الگوریتم بیزین ساده و جنگل تصادفی ســاده مقایسه خواهیم کرد. برای به دســت آوردن مجموعه داده ترافیک واقعی، از مجموعه ردیابــی بســته‌های ترافیک اینترنت UNIBS [۵] اســتفاده کرده‌ایم.

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

1-port

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

۲- انواع روش‌های طبقه‌بندی ترافیک و کارهای مرتبط

تکنیک‌هـــای مختلفی بـــرای طبقه‌بندی ترافیک شـــبکه اینترنت اســـتفاده شـــده اســـت. ما در این بخش هر یک از تکنیک‌های موجود همراه با برخی کارهای مرتبط با آن‌ها را آورده‌ایم.

۱،۲. روش مبتنی بر درگاه

شـــناخته‌ترین و قدیمی‌ترین روش مورد استفاده برای طبقه‌بندی ترافیک اینترنت، تطبیق شـــمارهٔ درگاه است. در این روش، از شـــمارهٔ درگاه مقصد در سرآیند لایه انتقال بســـته برای شناســـایی ترافیک استفاده می‌شـــود. مقدار شمارهٔ درگاه با لیست شماره درگاه‌های تعیین‌شده، برای شناسایی بســـته جاری مقایسه می‌شـــود. اما برنامه‌های جدید بخصـــوص برنامه‌های نظیر بـــه نظیر از روش‌های مختلف برای پنهان کردن خود به‌منظور فرار از تشخیص اســـتفاده می‌کنند. آن‌ها از درگاه‌هـــای پویا و یا درگاه‌های دیگر برنامه‌های شناخته‌شـــده در اتصالاتشـــان استفاده می‌کنند. این وضعیت باعث می‌شـــود که روش شناســـایی مبتنی بر درگاه روشـــی کم اثر و حتـــی در بعضی مواقع و روی برنامه‌های ناشـــناخته کاملاً بی‌اثر باشد. این روش حداکثر می‌تواند ۷۰ درصد ترافیک اینترنت را تشخیص دهد [۶]. همچنین، در [۷] نیز نشان داده‌شده که این روش حدود ۳۰ تا ۷۰ درصد ترافیک اینترنت را نمی‌تواند شناسایی کند.

۲،۲. روش مبتنی بر بازرسی Payload

این روش بر اســـاس تجزیه‌وتحلیل Payload بسته‌های جریان است که در آن محتوای بسته‌ها را برای پیدا کردن امضای برنامه‌های شناخته‌شده جستجو می‌کند. محتوای بسته اطلاعات زیادی دارد که می‌توان با تجزیه‌وتحلیل آن به شناسایی دقیق‌تری دست یافت.

ســـن [۲] و همـــکاران [۸] بـــا بازرســـی محتـــوای هـــر یـــک از بســـته‌های ترافیـــک امضاهـــای ســـطح برنامه را بـــرای برنامه‌هـــای P2P تولید می‌کنند. ایـــن روش به‌طور قابل‌توجهی می‌تواند تخمیـــن ترافیک P2P را بیش از آنچه روش مبتنـــی بر درگاه ارائه می‌داد، بهبود ببخشـــد. مور[۳] و همـــکاران [۹] با اســـتفاده از روش بازرســـی Payload نشان دادند که طبقه‌بندی‌های بر اساس شمارهٔ درگاه‌های شناخته‌شـــده ناکارآمد هســـتند و مقدار زیادی از ترافیک را یا به صورت ناشـــناس یا به اشتباه طبقه‌بندی می‌کنند. درحالی‌که روش پیشنهادی ایشان، هیچ جریانی را به‌عنوان جریان ناشـــناس 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].این‌برنامه‌به‌طورمتوالی‌هربسته‌راباجدول‌درهم‌سازی[۵] که جریان‌های طبقه‌بندی‌شـــده قبلی در آن ذخیره شـــده اســـت، مقایســـه 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 بازررســـی Payload بســـیار دقیق است، اما چند مشکل اساسی دارد. مهم‌تر از همه این‌که نمی‌توان آن را روی بسته‌های رمزگذاری شده اعمال کرد. تجزیه‌وتحلیل مستقیم داده Payload باعث نقض سیاست‌های حفظ حریم خصوصی ســـازمانی یا نقـــض 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 اســـت که برنامه‌های مختلف

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

الگوهـــای اتصال متفاوتـــی دارند. روش مبتنـــی بر رفتار میزبـــان، ارتباطات میان میزبان‌های خاص در یک شـــبکه را شناســـایی می‌کند و الگوی ارتباط یک میزبان خاص با الگوی رفتار فعالیت‌های متفاوت مقایسه می‌شود. این روش Payload بسته را برای طبقه‌بندی ترافیک استفاده نمی‌کند. درنتیجه می‌تواند بســـته‌ها با محتوای رمزگذاری شده را شناســـایی کند. بیشـــتر تکنیک‌های مبتنی بر رفتار میزبان فرض می‌کنند که میزبان‌های تحت نظارت در یک لحظه تنها از یک برنامه اســـتفاده می‌کنند و در واقعیت، این وضعیت ممکن است هرگز اتفاق نیفتد. بیشتر کاربران از برنامه‌های زیادی به‌طور هم‌زمان اســـتفاده 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 ســـاده کرده اســـت، اما یک چالش بزرگ برای استفاده از آن است [۱۱].

کاراجیانیـــس[6] و همـــکاران [۱۲] یک روش شناســـایی جریان‌های ترافیک P2P بر اســـاس الگوهای اتصال جریان ترافیـــک P2P ارائه کرده‌اند. این روش قادر به شناســـایی جریان‌هایی اســـت که در تجزیه‌وتحلیل Payload از دست رفته‌انـــد و حدود ۱۰٪ بیشـــتر از تجزیه‌وتحلیـــل Payload جریان‌هـــای P2P را شناســـایی می‌کنـــد. کاراجیانیـــس و همـــکاران همچنین رفتار میزبان را در ســـه ســـطح مورد بررسی قرار دادند: سطح اجتماعی، سطح کاربردی و سطح برنامه. آن‌ها توانســـتند به دقت شناسایی ترافیک بیش از ۸۰ درصد دســـت یابند. این روش قادر به شناسایی دقیق برنامه‌ها نیســـت که این می‌تواند بـــرای 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 توســـط ایلیفوتو و همـــکاران در [۱۴] ارائه شـــده است. آن‌ها با اســـتفاده از گراف‌های پراکندگی، ترافیک تعاملات گســـترده شـــبکه را نمایش داده‌اند. این گراف قادر به تشـــخیص رفتارهایی از شبکه است که در میان برنامه‌های P2P مشترک بوده و با دیگر برنامه‌ها تفاوت دارد.

تکنیک‌هـــای مبتنی بـــر میزبان قادر به شناســـایی نوع یک برنامه هســـتند، امـــا نمی‌توانند انـــواع زیربرنامه‌های آن را طبقه‌بنـــدی کننـــد. به‌عنوان‌مثـــال، آن‌هـــا می‌توانند جریان‌های P2P را شناســـایی کنند، امـــا تعیین نوع برنامه P2P (به‌عنوان‌مثـــال eMule یا بیت‌تورنت) که این جریان را تولید کرده، دشـــوار اســـت [۱۲]. این روش محدودیت‌های دیگری نیز دارد. برخی الگوهای رفتاری برنامه را نمی‌توان به‌آســـانی کشف کرد به مقدار حافظه و جریان‌های زیادی برای همـــه میزبان‌ها نیاز دارد تا بتوانـــد الگوی اتصال را کشف کند.

۴،۲. روش مبتنی بر یادگیری ماشین

یادگیـــری ماشـــین[7] مجموعـــه‌ای از تکنیک‌هـــا برای داده‌کاوی و کشف دانش است که الگوهای ساختاری مفید در داده‌ها را جســـتجو می‌کند. در ســـال ۱۹۹۴ برای اولین بار از روش‌های یادگیری ماشـــین برای مطالعه طبقه‌بندی جریان در تشـــخیص نفوذ استفاده شـــد [۱۵]. پس از سال ۲۰۰۴، با افزایش سریع مقیاس و پیچیدگی شبکه، برخی از محدودیت‌های روش‌های موجود شناسایی شد و محققان از الگوریتم‌های یادگیری ماشـــین برای طبقه‌بندی استفاده کردنـــد [۱۶]. ورودی الگوریتم‌های یادگیری ماشـــین یک مجموعه داده از نمونه‌ها اســـت. هر نمونه از مجموعه داده توســـط 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- الگوریتم‌هـــای یادگیری ماشـــین دو مرحله دارند. مرحله اول، آموزش اســـت که یادگیری انجام می‌شـــود و در آن مجموعه داده آموزش بررســـی می‌شود و یک مدل طبقه‌بندی بر اســـاس آن ســـاخته می‌شـــود. مرحله دوم، آزمایش اســـت و مدلی که در مرحله آموزش ساخته شده

6-Karagiannis

7-Machin Learning

برای رده‌بندی نمونه‌های جدید مشـــاهده نشـــده استفاده می‌شود.

ایـــن تکنیک طبقه‌بندی بـــرای غلبه بـــر محدودیت‌های روش‌های قبلی ارائه شده است. هر جریان، با مجموعه‌ای از ویژگی‌هـــای آماری و مقادیر مرتبـــط با آن‌ها توصیف می‌شـــود. یک ویژگی آماری روی بســـته‌های یک جریان محاسبه می‌شود، مانند متوســـط طول بسته و یا انحراف معیار زمان رسیدن بســـته. الگوریتم‌های یادگیری ماشین روی ترافیک شبکه را می‌توان به سه دسته تقسیم کرد.

۲،۴،۱. الگوریتم‌های یادگیری با نظارت ۲. الگوریتم‌های یادگیری بـــدون نظـــارت ۳. الگوریتم‌هـــای یادگیری نیمه نظارتی.

۱،٤،۲. الگوریتم‌های یادگیری با نظارت

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

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

ژانگ[8] و همکاران در [۱۸]، جریان‌های وابسته به هم را تعیین کرده و با اســـتفاده از آن‌ها، مدل مجموعه جریان‌ها را می‌ســـازند. اگر جریان‌های مشاهده‌شـــده در یک بازه زمانی خاص، درگاه مقصد، IP مقصد و پروتکل لایه انتقال یکسانی داشته باشـــند، آن‌ها به عنوان جریان‌های وابسته تعیین می‌شـــوند. در مرحله اول، یک طبقه‌بند بیزین ساده منفـــرد احتمالات رده را بـــرای هر جریـــان تولید می‌کند. در مرحلـــه دوم، پیش‌بینی‌هـــای بیزین‌های ســـاده جریان جمع‌آوری می‌شـــود تا رده نهایی برای مجموعه جریان‌ها را تعیین کند. روش پیشنهادی آن‌ها با روش‌های طبقه‌بندی مثل بیزین ســـاده، درخت تصمیم و نزدیک‌ترین همســـایه مقایسه شده است.

در مقالـــه [۱۹]، جمونـــا[9] و اواردز[10] روش‌های مختلف یادگیری ماشـــین بـــا نظـــارت ماننـــد C4.5، بیز ســـاده، 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 مقایســـه با الگوریتم‌های دیگر به عملکرد بهتری دست می‌یابد.

۲،٤،۲. الگوریتم‌های یادگیری بدون نظارت

روش‌های یادگیـــری بدون نظارت، در مرحله یادگیری خود بـــه هیچ دانـــش از قبل تعیین‌شـــده‌ای نیـــاز ندارند. روش‌های بدون نظارت به عنوان روش‌های خوشـــه‌بندی شناخته شده‌اند. این روش‌ها به یک مجموعه داده آموزش با برچسب نیاز نداشته، گروه‌های طبیعی (خوشه‌ها) را در داده‌ها کشـــف می‌کنند و روی کشف الگوهای موجود در داده‌ها تمرکز دارند. نمونه‌ها را بر اســـاس میزان شباهت ویژگی‌هـــای آن‌ها که توســـط یـــک رویکـــرد اندازه‌گیری فاصله تعریف می‌شـــود مانند فاصله اقلیدسی به گروه‌ها خوشه‌بندی می‌کنند.

مـــک گرگـــور[11] و همـــکاران در ســـال ۲۰۰۴ از روش خوشـــه‌بندی EM بـــرای طبقه‌بنـــدی جریان‌ها اســـتفاده کرده‌انـــد[۲۰]. هـــدف پیدا کـــردن مجموعه خوشـــه‌ها با بیشترین شـــباهت از داده‌های آموزش اســـت. این روش جریان‌های شـــبکه را به گروه‌های مختلف از برنامه‌ها بر اساس ویژگی‌های قابل مشاهده مشابه جریان‌ها گروه‌بندی

8-Zhang

9-Jamuna
10-Ewards
11-McGregor

می‌کند. بااین‌حال، نویسندگان دقت خوشه‌بندی را ارزیابی نمی‌کنند.

در مقاله [۲۱] زانـدر[12] و همکارانش از روش یادگیری ماشـین بدون نظارت Autoclass برای طبقه‌بندی ترافیک و شناسایی برنامه‌ها استفاده کردند. الگوریتم انتخاب ویژگی SFS به‌منظور پیدا کردن بهترین مجموعه ویژگی‌ها انتخاب شده است. دقت روش برای برخی از برنامه‌های نزدیک به ۹۵ درصد می‌رسـد ولی میانگین دقت روی تمام برنامه‌ها ۸۶/۵ درصد است.

ینـگ کیو[13] و همکاران الگوریتم K-means را در سـال ۲۰۰۷ برای شناسـایی ترافیک شـبکه بـه کار برده‌اند و به‌منظور بهبود دقت طبقه‌بندی، الگوریتم انتخاب ویژگی و تبدیل لگاریتم را اضافه کرده‌انـد[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 برچسـب زدن خوشـه‌ها و نگاشـت آن‌ها به برنامه‌ها مهم اسـت. همچنین بررسـی چگونگی برچسب زدن باید با برنامه‌های جدید که شناسایی می‌شـوند، به‌روزرسانی شـود و پیچیدگی محاسباتی و هزینه‌های برچسب زدن نیز باید بهینه باقی بماند [۲۳].

۳,٤,۲. الگوریتم‌های یادگیری نیمه نظارتی

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

یـک روش نیمـه نظارتـی توسـط شریواسـتاو[14] و تیواری[15] در سـال ۲۰۱۰ بـرای طبقه‌بندی جریان‌های پیاده‌سـازی شده اسـت [۲۴]. مرحلهٔ آموزش روش نیمه نظارت شـامل دو مرحله خوشـه‌بندی و طبقه‌بندی است. الگوریتم خوشـه‌بندی K-Means مجموعـه آموزش داده متشـکل از جریان‌های بـا برچسـب و جریان‌های بدون برچسب را به خوشه‌های مشابه گروه‌بندی می‌کند. پس از خوشـه‌بندی داده‌های آموزشی، از جریان‌های با برچسب برای نگاشت خوشـه‌ها به رده‌های شناخته‌شده استفاده می‌شود. در مرحله آزمایش، از نمونه‌های غیرآموزشی و بدون برچسب استفاده می‌شـود. نتایج آزمایش برای این روش در مقایسـه با طبقه‌بند SVM به دقت ۹۵/۶۵ درصد دسـت می‌یابد، در حالی‌که دقت روش طبقه‌بند SVM روی مجموعه داده آزمایشی ۷۶ درصد است.

ژانگ و همکارانش در سال ۲۰۱۲ استفاده از جریان‌های وابسـته در هر دو مرحله آموزش و آزمایش را پیشـنهاد دادند[۲۵]. در مرحله آموزش اسـتفاده از جریان وابسـته، مجموعه داده تحت نظارت را گسـترش می‌دهد. در مرحله آزمایش، جریان‌های وابسته شناسایی می‌شوند و با ترکیب پیش‌بینی‌های فـردی روی هر یـک از جریان‌ها، مجموعه جریان‌های وابسته طبقه‌بندی می‌شوند. دقت طبقه‌بندی برای روش پیشنهادی به صورت میانگین حدود ۹۶ درصد است.

بااین‌حـال، روش‌های یادگیری ماشـین نیمه نظارتی ممکن اسـت در فرایند یادگیری‌شان مخصوصا زمانی که تعداد داده‌های آموزشـی برچسـب 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 به‌هیچ‌وجـه با داده‌هـای کاربـران و همچنین اطلاعات ارتباطی بین میزبان‌ها سروکار ندارد.

در روش پیشـنهادی از یادگیـری بانظارت اسـتفاده می‌کنیـم، چراکه در الگوریتم‌های یادگیـری بدون نظارت هزینه زیادی برای برچسـب زدن نمونه‌ها نیاز است و در

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

الگوریتم‌هـــای یادگیری نیمه نظارتی نیز همیشـــه تعدادی خوشـــه در مرحله آموزش به وجود می‌آیند که هیچ نمونه برچســـب داری در آن‌هـــا قرار نمی‌گیـــرد. همچنین، نتایج الگوریتم‌های با نظارت روی طبقه‌بندی شبکه بهتر است.

روش پیشـــنهادی ما بهبودیافته 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 نســـبت واقعی تعداد جریان‌ها روی ترافیک استفاده می‌کنیم و ثابت می‌کنیم که روش پیشنهادی روی جریان‌ها با تعداد کم نیز بســـیار خوب و با فاصله از سایر الگوریتم‌ها عمل می‌کند. الگوریتم پیشـــنهادی، با تغییراتی نسبت به جنگل تصادفی ساده آن را بهبود می‌دهد. بنابراین، برای درک بهتر آن، ابتدا مفهوم روش جنگل تصادفی ساده را تشـــریح می‌کنیم و در ادامه، تغییرات صورت گرفته بر روی آن و فرآینـــد تولید روش الگوریتـــم جنگل تصادفی بهبودیافته را بیان خواهیم کرد.

۱،۳. الگوریتم جنگل تصادفی ساده

الگوریتـــم جنـــگل تصادفی یـــک الگوریتـــم گروهی با مجموعه‌ای از درختان تصمیم است. دقت طبقه‌بندی روش جنگل تصادفی با ساخت مجموعه‌ای از درختان و رأی‌گیری بین آن‌ها برای به دســـت آوردن رده‌ای با بیشترین تعداد رأی، پیشـــرفت‌های قابل‌توجهی داشته اســـت. دو ویژگی مهـــم در ســـاخت جنگل‌هـــای تصادفـــی، روش بگینگ[16] [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[17] در سال ۱۹۹۶ مطرح شد. ایـــن روش یک فرا الگوریتم بر مبنای مفاهیم خود راه‌انداز[18] وترکیب،برای‌بهبودیادگیری‌ماشین‌است.الگوریتم‌های‌گروهی در یادگیری ماشین، چند یادگیرنده ضعیف را ترکیب می‌کنند تا به یک یادگیرنده قوی دست یابند. این روش از بیش برازش[19] داده‌ها جلوگیری می‌کند. در بگینگ، نتایج خوب زمانی تولید می‌شـــود که طبقه‌بندهای پایه جـــزء الگوریتم‌های یادگیری ناپایدار باشند (مانند درخت تصمیم‌گیری یا شبکه عصبی)، به‌طوری‌که تغییرات کوچک در داده‌های آموزشـــی منجر به تغییرات عمده‌ای در مدل ساخته‌شـــده توســـط آن الگوریتم شود.

فرآینـــد الگوریتـــم بگینگ بدین شـــرح می‌باشـــد؛ یک مجموعه آموزش D به‌انـــدازه m را در نظر بگیرید. بگینگ با نمونه‌گیـــری یکنواخت و با جایگزینـــی نمونه‌ها از D، n مجموعه آموزشی جدید D_i با اندازه اولیه m تولید می‌کند. نمونه‌گیـــری با جایگزینی این امـــکان را می‌دهد که در هر D_i بعضی از نمونه‌ها امکان تکرار داشـــته باشـــند. این نوع نمونه‌گیری به‌عنـــوان نمونه‌گیری خودراه‌انداز شـــناخته می‌شـــود. خروجـــی ترکیب n مدل بـــا میانگین‌گیری برای رگرســـیون و رأی‌گیری برای طبقه‌بندی به دست می‌آید. بنابراین، با استفاده از نمونه‌گیری دوباره وتولید مجموعه داده‌های مختلف، تنوع مورد نیاز حاصل خواهد شد.

۲،۱،۳. ویژگی‌های تصادفی

خصوصیت ویژگی‌های تصادفی، بدین‌صورت است که در هر گره از ساخت هر درخت، به‌تصادف 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 بیشـــترین بهره اطلاعاتی برای رشـــد درخت انتخاب می‌شـــود. تعداد این ویژگی‌ها کمتر از تعداد ویژگی‌های اصلی است. هر درخت در جنگل تصادفی با اســـتفاده از الگوریتم درخـــت تصمیم کارت[20] و با حداکثر اندازه و بدون هرس رشـــد می‌کند. بریمن در هر گره از تعداد $\lfloor log_2 M \rfloor + 1$ ویژگی اســـتفاده کرده است که در آن M تعداد کل ویژگی‌های ورودی است [۲۸].

۳،۱،۳. تعریف جنگل تصادفی

جنگل تصادفی یـــک طبقه بند مجموعه‌ای متشـــکل از طبقه‌بندهای درخت تصمیم اســـت. هـــر طبقه بند برای هر

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

نمونه ورودی به صورت $h(x\ ,\ \Theta_k)$ اســت، که x یک نمونــه ورودی و Θ_k مجموعه آمــوزش برای درخت kام اســت. Θها مستقل از یکدیگر ولی با توزیع یکسان هستند. برای هــر نمونه x، هر درخت یک پیش‌بینــی را برای ردهٔ نمونه x ارائه می‌دهد و در نهایت رده‌ای با بیشــترین تعداد رأی درختــان روی ورودی x به عنوان ردهٔ نمونه انتخاب می‌شود. این فرآیند را جنگل تصادفی می‌نامند [۲۸].

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

۴،۱،۳.تخمین خارج از کیسه

فــرض کنید هــر طبقه‌بند با مجموعه آموزشــی جدید بــا روش درخت تصمیم ســاخته می‌شــود. بــا توجه به مجموعــه آموزش Θ و با روش خودراه‌انداز مجموعه‌های آموزشی Θ_k تشکیل می‌شوند. سپس طبقه‌بندهای 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 نیســت، نمونه‌های خارج از کیسهٔ طبقه‌بند kام نامیده می‌شود.

در هــر مجموعــه آموزش بــه دســت‌آمده از روش خودراه‌انــداز، نمونه‌هــای OOB در حــدود یک‌ســوم از نمونه‌های مجموعه آموزش اصلی اســت که در مجموعه آموزش قرار نمی‌گیرند. رابطه ۱ شــیوه تخمین ردهٔ نمونه OOB را روی جنگل نشــان می‌دهد. برای به دست آوردن ردهٔ 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 ردهٔ نمونه در نظر گرفته می‌شود.

(۱)

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

(۲)

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

کــه K تعــداد درختــان، c نشــان‌دهنده رده، $h_k(x)$ پیش‌بینــی درخــت kام روی نمونه x را نشــان می‌دهند و OOB_k مجموعــه نمونه‌های OOB درخت kام می‌باشــند. رابطه ۲ نشــان می‌دهد که مقدار تابع شاخص I یک خواهد بود اگــر x در مجموعه نمونه‌های درخــت kام قرار دارد (عضو مجموعــه آموزش درخت kام نیســت) و همچنین درخــت k ام نمونــه x را به ردهٔ c طبقه‌بنــدی کند. در غیر این‌صورت، مقدار تابع شاخص صفر می‌شود.

برای به دســت آوردن تخمیــن نمونه‌های OOB روی جنــگل از $error_k(OOB)$ در رابطه ۳ اســتفاده می‌کنیم که خطای طبقه‌بندی جنــگل روی نمونه‌های OOB درخت kام می‌باشد.

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

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

N تعــداد همه نمونه‌های مجموعــه آموزش اصلی، x_i نمونــه i ام روی مجموعه آمــوزش اصلی، y_i ردهٔ واقعی x_i، $y(x_i)$ ردهٔ پیش‌بینی‌شــده بــرای x_i برحســب رابطه ۱ اســت. در رابطه ۴ مقدار تابــع 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 جنــگل تصادفی، برای خطا یک کــران بالا در نظر گرفته می‌شود. دو پارامتر برای این کران بالا اندازه‌گیری می‌شــود. پارامتر اول، قدرت طبقه‌بندهای فردی و پارامتر دوم، وابستگی بین درختان جنگل است.

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

در رابطه s،۵ قــدرت طبقه‌بندهای فردی در جنگل و $\bar{\rho}$

21-Strength
22-Correlation

وابســـتگی بین درختان جنگل را نشـــان می‌دهد. بر اساس این رابطه برای خطای*PE هرچه مقدار s بیشـــتر و مقدار $\bar{\rho}$ کمتر باشد، میزان خطا نیز کمتر خواهد بود. نسبت $\frac{\bar{\rho}}{s^2}$ برای جنگل تصادفی در درک عملکرد آن، یک راهنمای مفید است. این نسبت، تقســـیم 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 ردهٔ نمونه آزمایشی برای جنگل تصادفی ســـاده از پیش‌بینی تمام درختان جنگل اســـتفاده می‌شـــود. برای تعیین ردهٔ نمونه آزمایشی از روابط ۶ و ۷ استفاده می‌کنیم.

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

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

۲،۳. جنگل تصادفی بهبودیافته

در ایـــن مقاله، برای بهبـــود روش جنگل تصادفی، یک تغییـــر در مرحله آموزش و یک تغییر در مرحله طبقه‌بندی به جنگل تصادفی ســـاده اعمال می‌کنیم تا الگوریتم جنگل تصادفی بهبودیافته را به دست آوریم.

در مرحلـــه آمـــوزش، در جنـــگل تصادفـــی ســـاده برای ســـاخت هـــر درخـــت از معیـــار Gini Index (معیار ســـاخت درخت Cart) بـــرای انتخاب بهتریـــن ویژگی در هر گره اســـتفاده می‌شـــود. اما در اینجا، برای به دســـت آوردن طبقه‌بندهـــای قوی‌تـــر از معیار بهـــرهٔ اطلاعات[۲۳] 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 بـــا تعداد کمتر (با تخمین بالا) دقت پایینی دارد. برای این مشـــکل، ما یک راه‌حل پیشنهاد داده‌ایـــم. در مرحلـــه آموزش و پس از ســـاخت درخت، با اســـتفاده از نمونه‌های OOB آن درخت، دقت طبقه‌بندی را فقط روی آن درخت به دســـت می‌آوریم و دقت درخت را به عنـــوان وزن آن درخت در نظـــر می‌گیریم و در مرحله طبقه‌بنـــدی به‌منظور بهبود پیش‌بینـــی ردهٔ نهایی، از وزن هر درخت به‌دســـت‌آمده برای طبقه‌بندی نمونه آزمایشی اســـتفاده می‌کنیم. این باعث می‌شـــود که درختان با دقت بالاتر وزن بیشتری در طبقه‌بندی داشته باشند.

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

۱،۲،۳. معیار بهرهٔ اطلاعات

بهـــرهٔ اطلاعـــات Information Gain(S,A) بـــرای یـــک ویژگی نظیر A نسبت به مجموعه نمونه‌های S در رابطه ۸ تعریف می‌شـــود که در آن Values(A) مجموعه همه مقدار ویژگی‌هـــای A بـــوده و S_V زیرمجموعه‌ای از S اســـت که برای آن ویژگی A دارای مقدار v است. $|S_v|$ تعداد نمونه‌ها بـــا مقدار v روی ویژگـــی A در مجموعه S و $|S|$ تعداد کل نمونه‌ها روی مجموعه S است.

$$Information\ Gain\ (S\ ,\ A) = Entropy(S) - \sum_{v\in Values(A)}\left[\frac{|S_v|}{|S|}\ Entropy(S)\right] \quad (۸)$$

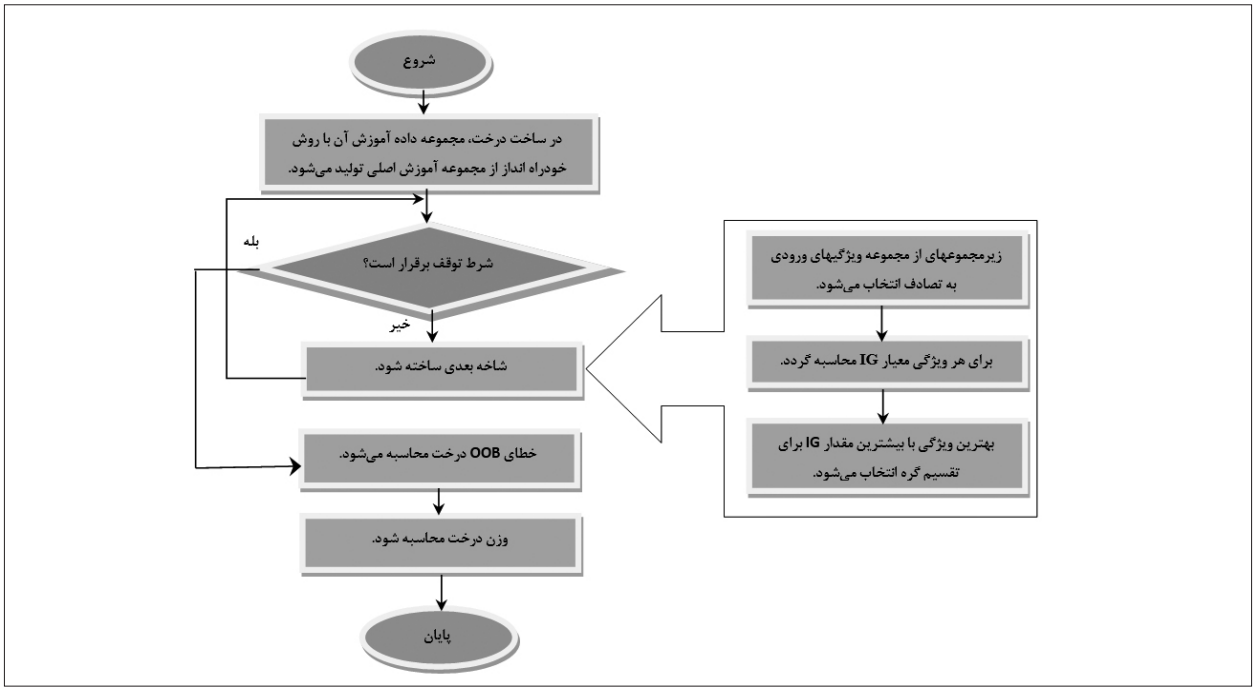
$$entropy(S) = \sum_{i=1}^{numclass} -p_i\ log_2\ p_i \quad (۹)$$

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

۲،۲،۳. مرحله آموزش IRF

در فرآینـــد ایجاد جنگل، در ســـاخت هر درخت و برای تقســـیم هر گره، یک زیرمجموعه از ویژگی‌های ورودی را به‌تصادف انتخاب می‌کنیم و برای به دست آوردن بهترین ویژگی در گره برای ایجاد شـــاخه، از معیار بهرهٔ اطلاعات اســـتفاده می‌شـــود. هر درخت روی جنگل مجموعه نمونه OOB مخصوص بـــه خود را دارد. دقت هـــر درخت را با استفاده از نمونه‌های OOB آن درخت به دست می‌آوریم و آن را به‌عنـــوان وزن آن درخت در نظر می‌گیریم. وزن هر درخت را از رابطه ۱۰ محاسبه می‌کنیم.

23-Information Gain



شکل ۱: روند کلی الگوریتم جنگل تصادفی بهبودیافته برای هر درخت

$$W_k = \frac{\sum_{i=1}^{N} I(h_k(x_i)=y_i \,,\, (x_i\,,y_i)\in OOB_k)}{\sum_{i=1}^{N} I((x_i\,,y_i)\in OOB_k)} \qquad (۱۰)$$

$$I(h_k(x_i)=y_i \;,\; (x_i\;,\;y_i)\in OOB_k) = \begin{cases} 1\; h_k(x_i)=y_i \\ 0\; h_k(x_i)\neq y_i \end{cases} \qquad (۱۱)$$

N تعداد همه نمونه‌های مجموعه آموزش اصلی، x_i نمونه iام 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 ردهٔ نمونه‌های آزمایشی بر اساس وزن درختان جنگل پیش‌بینی می‌شود. برای به دست آوردن ردهٔ هر نمونه همه درختان جنگل شرکت دارند و از رابطه ۱۲ استفاده می‌کنیم که براساس وزن درختان به‌دست‌آمده در مرحله آموزش است. برای هر ردهٔ c مجموعه وزن درختانی که ردهٔ c را برای نمونه x پیش‌بینی می‌کنند به دست می‌آوریم و سپس رده‌ای با بیشترین میانگین وزن روی درختان به عنوان ردهٔ آزمایشی در نظر گرفته می‌شود. در رابطه زیر y(x) ردهٔ پیش‌بینی‌شده توسط جنگل تصادفی بهبودیافته برای نمونه x است، K تعداد درختان و W_k وزن درخت kام می‌باشد.

$$y(x) = arg\,max_c \left\{ \frac{1}{K} \sum_{k=1}^{K} W_k, I(h_k(x)=c) \right\} \qquad (۱۲)$$

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،٤. 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 طبقه‌بندی‌شده تقسیم بر تعداد کل جریان‌ها می‌باشد.

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

$$\text{f\_measure} = \frac{2 * \text{Precision} * \text{recall}}{\text{precision} + \text{recall}} \qquad (۱۳)$$

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

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

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

۲،٤. مجموعه داده

به‌منظور ارزیابی عملکرد روش طبقه‌بندی روی داده‌های واقعی ترافیک از مجموعه داده UNIBS استفاده می‌کنیم که از ردیابی جریان‌های اینترنت در دانشگاه برشیا جمع‌آوری‌شده 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،٤. پیش‌پردازش

با استفاده از ۵تایی مشخصه جریان (شمارهٔ 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 ردهٔ Skype، Mail، Web و برنامه‌های P2P تعریف می‌کنیم. جزئیات در جدول ۱ گزارش شده است.

٤،٤. مجموعه ویژگی‌ها و شرایط آزمایش

۲۲ نوع ویژگی را برای هر جریان به‌صورت 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

به‌کاربرده 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 شمارهٔ درگاه مبدأ و شمارهٔ 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 وتحلیل 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، به‌صورت

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

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

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

پررنگ 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 به‌دست‌آمده 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 پس‌ازآن 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 به‌غیراز NB روی افزایش تعداد نمونه‌های جریان تقریباً روند صعودی دارند و مقادیر بهبود می‌یابند. اما الگوریتم NB روند تغییرات ثابتی ندارد و حتی به‌صورت 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. به‌طور خاص روش IRF یک‌روند 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 ۲/۵۳ 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 همه 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

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

الگوریتم / تعداد داده‌ها	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

الگوریتم / تعداد داده‌ها	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

الگوریتم / 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

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 قابل‌توجه و بیش از هر نوع برنامه دیگری تحقیقات و روش‌های مختلف برای شناسایی برنامه‌های P2P انجام‌شده 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. درحالی که 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

جدول۸: نتایج الگوریتم‌های مختلف بر اساس معیار F-measure روی تعداد نمونه‌های مختلف برای ردهٔ P2P

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

برنامه‌های P2P تعلق دارند.

جدول ۸ نشـــان‌دهنده نتایج الگوریتم‌ها بر اســـاس معیار F_measure برای ردهٔ P2P با افزایش تعداد نمونه‌های جریان اســـت. با توجه بـــه مقادیر این جدول می‌توان نتیجه گرفت که روش پیشـــنهادی IRF عملکرد بسیار مناسبی روی ردهٔ 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 قســـمت‌های مختلف مانند تخصیص منابع، بهبود کیفیت خدمات ســـرویس، و تشـــخیص برنامه‌های مخرب می‌شـــود. بنابراین، با توجه به اهمیت شناســـایی جریان‌های ترافیک اینترنت، ما در این مقاله مســـئله طبقه‌بندی ترافیک اینترنـــت را مطرح کردیم و الگوریتم جدیدی را برای حل این مســـئله پیشنهاد دادیم. روش پیشنهادی جنگل تصادفی بهبودیافته، یک روش یادگیری ماشـــین بانظارت گروهی است که از مجموعه‌ای از درختان تصمیم ســـاخته می‌شـــود. برای هر درخت روی این جنگل مجموعه آموزش جدیدی با روش خودراه‌اندازی ساخته می‌شود و نمونه‌های باقی‌مانـــده از مجموعه آموزش اصلی که در مجموعه آموزش درخت وجـــود ندارند، در یک مجموعه به نام OOB برای ارزیابی درخت قرار می‌گیرند. در ســـاخت هر درخت و در تقسیم هر گره، انتخاب ویژگی از زیرمجموعـــه تصادفـــی از ویژگی‌های اصلی انجام می‌شـــود. با اســـتفاده از نمونه‌های OOB دقت درخت روی نمونه‌های آموزش را به دست آورده و به‌عنوان وزن درخت قرار می‌گیرد و در مرحلـــه آزمایش با تغییر در نوع رأی‌گیری روی درختان بر اساس وزن آن‌ها ردهٔ نمونهٔ آزمایشی مشخص می‌شود. سپس عملکرد روش پیشنهادی را با روش‌هایی که در تحقیقات قبلی دقت خوبی را گـــزارش کرده بودنـــد، روی مجموعه ترافیک جریان‌هـــای واقعـــی UNIBS با مقایســـه کردیم. نتایج مقایســـه‌ها و ارزیابی‌ها نشـــان می‌دهد که روش پیشـــنهادی نسبت به ســـایر روش‌ها دقت شناسایی بالاتری دارد و در شناسایی برنامه‌ها با تعداد نمونه‌های آموزش کم عملکرد قابل قبولی را ارائه می‌دهد. دقت طبقه‌بندی آن روی هرکدام از برنامه‌هـــای مجموعه داده نســـبت به ســـایر الگوریتم‌ها بالاتر است و میانگین دقت طبقه‌بندی روی همهٔ 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., "MINETRAC: 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-

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] Antoniades, 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 INFOCOM 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 Ewards, 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 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.