مقاله پژوهشی

بهبود تشخیص ناهنجاری بات‌نت‌های حوزه اینترنت اشیای مبتنی بر انتخاب ویژگی پویا و پردازش‌های ترکیبی

بشری پیشگو و احمد اکبری ازیرانی

چکیده: پیچیده‌شدن کاربردهای دنیای واقعی خصوصاً در حوزه‌های اینترنت اشیا، ریسک‌های امنیتی متنوعی را برای این حوزه به همراه داشته است. بات‌نت‌های این حوزه به عنوان گونه‌ای از حملات امنیتی پیچیده شناخته می‌شوند که می‌توان از ابزارهای یادگیری ماشین، به منظور شناسایی و کشف آنها استفاده نمود. شناسایی حملات مذکور از یک سو نیازمند کشف الگوی رفتاری بات‌نت‌ها از طریق پردازش‌های دسته‌ای و با دقت بالا بوده و از سویی دیگر می‌بایست همانند پردازش‌های جریانی، به لحاظ عملیاتی بلادرنگ عمل نموده و وفق‌پذیر باشند. این مسئله، اهمیت بهره‌گیری از تکنیک‌های پردازش ترکیبی دسته‌ای و جریانی را با هدف تشخیص بات‌نت‌ها، بیش از پیش آشکار می‌سازد. از چالش‌های مهم این پردازش‌ها می‌توان به انتخاب ویژگی‌های مناسب و متنوع جهت ساخت مدل‌های پایه و نیز انتخاب هوشمندانه مدل‌های پایه جهت ترکیب و ارائه نتیجه نهایی اشاره نمود. در این مقاله به ارائه راهکاری مبتنی بر ترکیب روش‌های یادگیری جریانی و دسته‌ای با هدف تشخیص ناهنجاری بات‌نت‌ها می‌پردازیم. این راهکار از یک روش انتخاب ویژگی پویا که مبتنی بر الگوریتم ژنتیک بوده و به طور کامل با ماهیت پردازش‌های ترکیبی سازگار است، بهره می‌گیرد و ویژگی‌های مؤثر در فرایند پردازش را در طول زمان و وابسته به جریان ورودی داده‌ها به صورت پویا تغییر می‌دهد. نتایج آزمایش‌ها در مجموعه داده‌ای مشتمل بر دو نوع بات‌نت شناخته‌شده، بیانگر آن است که رویکرد پیشنهادی از یک سو با کاهش تعداد ویژگی‌ها و حذف ویژگی‌های نامناسب موجب افزایش سرعت پردازش‌های ترکیبی و کاهش زمان تشخیص بات‌نت می‌گردد و از سویی دیگر با انتخاب مدل‌های مناسب جهت تجمیع نتایج، دقت پردازش را افزایش می‌دهد.

کلیدواژه: **انتخاب ویژگی پویا، تشخیص ناهنجاری بات‌نت‌ها، اینترنت اشیا، پردازش‌های ترکیبی دسته‌ای و جریانی.**

۱- مقدمه

آشکارشدن کاربردهای جدید در حوزه اینترنت اشیا و بهره‌گیری از این بسترها در زمینه‌های گوناگون، زمینه‌ساز کشف هرچه بیشتر آسیب‌پذیری‌های این حوزه و اعمال حملات پیچیده امنیتی با بهره‌گیری از آسیب‌پذیری‌های موجود خواهد بود. بات‌نت‌ها و بدافزارهای مبتنی بر اینترنت اشیا مانند Mirai و Bashlite که با بهره‌گیری از حملات منع سرویس توزیع‌شده به اهداف خویش در حوزه شبکه IoT دست می‌یابند، در زمره تهدیدات موجود می‌باشند [۱] و [۲].

این مقاله در تاریخ ۱۶ آبان ماه ۱۴۰۰ دریافت و در تاریخ ۱۷ بهمن ماه ۱۴۰۰ بازنگری شد.

بشری پیشگو، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، (email: boshra.pishgoo@student.iust.ac.ir).

احمد اکبری ازیرانی (نویسنده مسئول)، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران، ایران، (email: akbari@iust.ac.ir).

تا کنون رویکردهای گوناگونی برای مقابله با این حملات ارائه شده که در میان آنها، تکنیک‌های یادگیری ماشین یکی از مهم‌ترین تکنیک‌های موجود محسوب می‌شوند [۳] و [۴]. از آنجا که در بستر اینترنت اشیا، حجم داده‌های جمع‌آوری شده بسیار بالاست و عملاً داده‌های گردآوری شده، مرجع تشخیص بات‌نت‌ها مبتنی بر تکنیک‌های یادگیری ماشین هستند، لذا به منظور دستیابی به نتایجی دقیق باید از بسترهای تحلیل کلان‌داده به منظور اجرای تکنیک‌های مذکور بهره‌برداری نمود.

بسترهای تحلیل کلان‌داده از منظر پلتفرم‌های پردازشی به دو دسته پردازش‌های دسته‌ای و جریانی قابل تفکیک می‌باشند. پردازش‌های دسته‌ای دارای دقت بالاتری هستند اما فرایند یادگیری این روش‌ها معمولاً زمان‌بر بوده و قادر به یادگیری داده‌های اخیر نیستند. بنابراین فرایند کشف الگوی داده‌ها از طریق تکنیک‌های یادگیری دسته‌ای امکان‌پذیر می‌باشد اما این تکنیک‌ها به دلیل نیاز به زمان بالای یادگیری، نمی‌توانند به صورت وفق‌پذیر[1] عمل نموده و بلادرنگ به شناسایی الگوهای جدید بپردازند. در مقابل پردازش‌های جریانی به صورت بلادرنگ عمل نموده و فرایند آموزش آنها سریع و تدریجی می‌باشد. از این رو هم سرعت بالایی در آموزش و یادگیری دارند و هم قادر به تأثیرپذیری از داده‌های اخیر در فرایند آموزش هستند. البته دقت آنها معمولاً کمتر از تکنیک‌های پردازش دسته‌ای می‌باشد.

شناسایی و تشخیص بات‌نت‌ها که هدف مقاله حاضر است، از یک سو نیازمند کشف الگوی رفتارهای هنجار و ناهنجار مبتنی بر حجم وسیع داده‌های پیشین می‌باشد و از سوی دیگر می‌بایست وفق‌پذیر بوده و به لحاظ عملیاتی به صورت بلادرنگ عمل نماید [۵] و [۶]. لذا با ترکیب هوشمندانه و مناسب دو نوع پردازش دسته‌ای و جریانی در قالب پردازش ترکیبی، می‌توان مزایای هر دو روش را با یکدیگر تجمیع نمود و به پردازش‌هایی دست یافت که به صورت بلادرنگ و با سرعت بالا، قادر به انجام محاسبات دقیق بر روی حجم بالای داده هستند.

از سوی دیگر با توجه به گسترش حسگرها و تکنولوژی‌های مرتبط با عملیات جمع‌آوری و رکوردکردن خروجی حسگرها در حوزه بات‌نت‌های اینترنت اشیا، مجموعه دادگان تولیدی عمدتاً از ابعاد و دانه‌بندی بسیار بالایی برخوردار است [۷]. این امر منجر به ایجاد مقادیر زیادی اطلاعات تکراری و نامرتبط در کلان‌داده‌ها می‌گردد که علاوه بر آن که حجم بالایی را اشغال می‌کنند، کارایی و دقت محاسبات را نیز تحت تأثیر قرار می‌دهند. همچنین وجود این داده‌های تکراری و نامرتبط، پیچیدگی زمانی اجرای تکنیک‌های هوشمند تشخیص بات‌نت‌ها را افزایش می‌دهد و منجر به کاهش سرعت عملیات می‌گردد که همین مسئله، تشخیص بلادرنگ

1. Adaptive

باتنت‌ها را با اختلال مواجه می‌سازد.

بر این اساس، از چالش‌های مهم پردازش‌های ترکیبی مبتنی بر بستر کلان‌داده‌ها و با هدف تشخیص ناهنجاری، می‌توان به انتخاب ویژگی‌های مناسب و متنوع جهت ساخت مدل‌های پایه و نیز انتخاب هوشمندانه مدل‌های پایه جهت ترکیب و ارائه نتیجه نهایی اشاره نمود. تا کنون در حوزه پردازش‌های دسته‌ای کلان‌داده‌ها، روش‌های انتخاب ویژگی متفاوتی ارائه گردیده است [۸] و [۹]. این روش‌ها عمدتاً بر یادگیری توزیع‌شده و بخش‌بندی‌کردن داده‌ها به صورت عمودی (مبتنی بر توزیع ویژگی) [۱۰]، افقی (مبتنی بر توزیع نمونه‌ها) [۱۱] یا ترکیب هر دو [۱۲] تمرکز می‌نمایند و به سازگارساختن تکنیک‌های انتخاب ویژگی سنتی با توازی‌گرایی و ساختار توزیع‌شده زیرساخت‌های کلان‌داده [۱۳] می‌پردازند. همچنین دسته دیگری از پژوهش‌ها کوشیده‌اند تا روش‌های انتخاب ویژگی سنتی را با استفاده از تکنیک مپ- ردیوس و فریم‌ورک هادوپ، پیاده‌سازی نموده و امکان اجرای موازی و توزیع‌شده آن را بر بستر کلان‌داده فراهم کنند [۱۴] تا [۱۶]. از سوی دیگر، تکنیک‌های انتخاب ویژگی در حوزه پردازش‌های جریانی کلان‌داده بیشتر بر ارائه الگوریتم‌هایی کارا و سریع تحت عنوان انتخاب ویژگی جریانی یا برخط [۱۷] تا [۱۹]، تمرکز داشته‌اند و کمتر بر سازگاری با زیرساخت‌های پردازش جریانی کلان‌داده تأکید می‌نمایند [۲۰].

علی‌رغم وجود پژوهش‌های مذکور که به صورت جداگانه در حوزه راهکارهای انتخاب ویژگی دسته‌ای و جریانی صورت گرفته است، تا کنون راهکاری جهت ارائه یک تکنیک انتخاب ویژگی مناسب و سازگار با پردازش‌های ترکیبی ارائه نشده است. از آنجا که پردازش‌های ترکیبی به طور کلی با جریان داده‌ای سر و کار دارند، لذا بهره‌گیری از تکنیک‌های انتخاب ویژگی دسته‌ای برای آنها عملیاتی نمی‌باشد. از سویی دیگر، گرچه تکنیک‌های انتخاب ویژگی جریانی قابلیت به کارگیری در پردازش‌های ترکیبی را دارند، لیکن هیچ یک از روش‌های موجود از ظرفیت پردازش‌های ترکیبی برای انتخاب ویژگی استفاده نمی‌نمایند و به عبارت دیگر با ماهیت پردازش‌های ترکیبی سازگار نیستند. به عنوان نمونه، الگوریتم‌های تکاملی همواره به عنوان روش مناسبی برای انتخاب ویژگی در داده‌های دسته‌ای، مطرح بوده‌اند [۲۱] تا [۲۴]. اما از آنجا که این روش‌ها، مجموعه ویژگی‌های منتخب را با صرف هزینه زمانی بالا انتخاب می‌نمایند، لذا یا در تکنیک‌های انتخاب ویژگی جریانی از آنها بهره‌برداری نمی‌شود و یا بهره‌گیری از این تکنیک‌ها، بلادرنگ بودن و سرعت عمل تکنیک‌های انتخاب ویژگی جریانی را با اختلال مواجه می‌سازد.

در این مقاله به ارائه راهکاری با هدف تشخیص باتنت‌های اینترنت اشیا می‌پردازیم. این راهکار علاوه بر آن که شرایط انجام موازی پردازش‌های دسته‌ای و جریانی و بهره‌گیری از دقت تکنیک‌های پردازش دسته‌ای هم‌زمان با سرعت و بلادرنگ‌بودن پردازش‌های جریانی را فراهم می‌آورد، از یک روش انتخاب ویژگی پویا نیز مبتنی بر الگوریتم ژنتیک بهره می‌گیرد که به طور کامل با ماهیت پردازش ترکیبی سازگار بوده و از ظرفیت‌های ذاتی این پردازش‌ها در راستای انتخاب ویژگی‌های مؤثر استفاده می‌نماید. معماری مذکور، ویژگی‌های مؤثر در فرایند پردازش را در طول زمان و وابسته به جریان ورودی داده‌ها به صورت پویا تغییر می‌دهد. عملیات به‌روزرسانی مجموعه ویژگی‌ها، متناسب با جریان ورودی و مادامی که به سیستم وارد می‌شود، ادامه خواهد یافت.

مقاله حاضر به صورت زیر سازماندهی می‌شود. بخش ۲ پژوهش‌های پیشین مرتبط با موضوع مقاله را مرور می‌نماید. معماری راهکار پیشنهادی و مؤلفه‌های آن با جزئیات کامل در بخش ۳ شرح داده خواهد شد. در بخش ۴ نتایج ارزیابی راهکار از جنبه‌های مختلف و بر اساس معیارهای ارزیابی متفاوت ارائه و تحلیل و بررسی می‌شود. سرانجام بخش ۵ به جمع‌بندی مقاله و ذکر راهکارهای آتی خواهد پرداخت.

۲- پژوهش‌های پیشین

۲-۱ تشخیص ناهنجاری باتنت‌ها

تا کنون رویکردهای گوناگونی برای تشخیص و شناسایی باتنت‌ها ارائه شده که در میان آنها تکنیک‌های یادگیری ماشین یکی از مهم‌ترین تکنیک‌های موجود محسوب می‌شوند [۳] و [۴]. عملیات تشخیص باتنت مبتنی بر تکنیک‌های یادگیری در دسته‌های مختلف قابل تفکیک هستند [۳] و [۲۵]. این دسته‌بندی که شامل تکنیک‌های یادگیری نظارتی و بدون نظارت، شبکه‌های عصبی عمیق، یادگیری تقویتی عمیق، شبکه‌های پیچیده، هوش ازدحامی، تحلیل‌های آماری، رویکردهای توزیع‌شده و راهکارهای ترکیبی می‌باشد، به همراه مراجع مربوط در جدول ۱ نشان داده شده است. طبق این جدول، راهکار ترکیبی، یکی از راهکارهای شناسایی باتنت‌ها محسوب می‌شود که از جنبه‌های گوناگون به ترکیب تکنیک‌های متفاوت شناسایی باتنت‌ها با هدف افزایش دقت می‌پردازد. البته روش‌های پیشین بیشتر به ترکیب اهداف شناسایی مختلف مانند ترکیب شناسایی با تکنیک تشخیص ناهنجاری و تشخیص امضا [۲۶]، ترکیب عامل‌های متفاوت [۲۷] و یا ترکیب الگوریتم‌های یادگیری متفاوت نظیر شبکه عصبی و گراف [۲۸] پرداخته‌اند و به ترکیب راهکارهای یادگیری دسته‌ای و جریانی که ایده اصلی مقاله حاضر است، پرداخته نشده است.

۲-۲ انتخاب ویژگی پویا مبتنی بر بستر کلان‌داده

از آنجا که در بستر اینترنت اشیا، حجم داده‌های جمع‌آوری شده بسیار بالاست و عملاً داده‌های گردآوری شده، مرجع تشخیص باتنت‌ها مبتنی بر تکنیک‌های یادگیری ماشین هستند، لذا به منظور دستیابی به نتایجی دقیق باید از بسترهای تحلیل کلان‌داده به منظور اجرای تکنیک‌های مذکور بهره‌برداری نمود.

پردازش کلان‌داده در سال‌های اخیر با پیشرفت‌های چشم‌گیری هم در حوزه پردازش‌های دسته‌ای و هم در حوزه پردازش‌های جریانی مواجه بوده است. اما مفهوم پردازش‌های ترکیبی با ارائه تعریف معماری لامبدا در سال ۲۰۱۳ شکل گرفت [۵۲]. سپس این تعریف در مطالعات پژوهشی [۵۳] تا [۵۵] و صنعتی [۵۶] و [۵۷] متعدد به شیوه‌های متفاوت پیاده‌سازی شد. در کنار این تلاش‌ها، آمازون به ارائه ابزارهایی پرداخت که با یکپارچه‌سازی آنها، امکان تحقق معماری لامبدا مبتنی بر اسپارک استریمینگ [۵۸] مهیا می‌گردید. سپس گوگل با هدف فراهم‌سازی امکان پردازش ترکیبی، سرویس Google Cloud Dataflow [۵۹] را بر روی پلتفرم ابری خود ارائه نمود. پس از آن، [۶۰] پیشنهاد به کارگیری معماری لامبدا را با هدف حل مسایل کلان‌داده مبتنی بر تکنیک‌های یادگیری ماشین و داده‌کاوی مطرح کرد. این پیشنهاد به عنوان چالش جدیدی برای یادگیری ماشین بلادرنگ در پژوهش‌های متعدد مطرح شد [۶۰] تا [۶۲] و پس از آن در کاربردهای متنوعی از جمله اینترنت اشیا [۶۳] و [۶۴] و تشخیص ناهنجاری [۶۵] تا [۶۷] پیاده‌سازی گردید.

علاوه بر پژوهش‌های فوق، چارچوب Oryx [۶۸] و پس از آن نسخه توسعه‌یافته آن با عنوان Oryx ۲.۰ [۶۹] نیز به عنوان یک چارچوب متن‌باز با هدف تحقق معماری لامبدا برای پردازش‌های بلادرنگ با حجم

جدول ۱: دسته‌بندی تکنیک‌های تشخیص ناهنجاری بات‌نت‌ها مبتنی بر یادگیری ماشین.

ردیف	دسته‌بندی راهکارها	شرح روش	تکنیک‌ها	مراجع
۱	تکنیک‌های یادگیری نظارتی	در تکنیک‌های نظارتی، رفتار بات‌نت‌ها و نیز رفتار نرمال شبکه، به همراه برچسب تخصیص داده شده به آنها، به منظور آموزش سیستم به کار گرفته می‌شود و خروجی روش‌های مذکور، طبقه‌بندی‌ای خواهد بود که قادر به تفکیک رفتار نرمال از رفتار بات‌نت‌ها است.	درخت تصمیم	[۲۹]
			بیز ساده گاوسی	[۳۰]
			K نزدیک‌ترین همسایه	[۳۱]
			طبقه‌بند SVM	[۳۲]
۲	تکنیک‌های یادگیری بدون نظارت	روش‌های یادگیری بدون نظارت، بدون بهره‌گیری از برچسب حملات و رفتارهای هنجار، تنها مبتنی بر ذات داده‌های بدون برچسب گردآوری شده از رفتار بات‌نت‌ها، به خوشه‌بندی می‌پردازند.	روش DBSCAN	[۳۳]
			روش X-means	[۳۴]
۳	شبکه عصبی عمیق	ایده اصلی این روش‌ها استخراج ویژگی‌های ترافیک شبکه مبتنی بر مشابهت‌های مکانی و زمانی می‌باشد. این روش، ترافیک شبکه را به یک تصویر خاکستری و یا بردار ویژگی نگاشت می‌کند و با ارسال آن به یک شبکه عصبی، ویژگی‌های متمایزکننده مکانی و زمانی را از آن استخراج می‌نماید.	شبکه CNN	[۳۵]
			شبکه RNN	[۳۶]
			شبکه GAN	[۳۷]
			شبکه DNN	[۳۸]
			شبکه FDL	[۳۹]
			شبکه FNN	[۴۰]
۴	یادگیری تقویتی عمیق	این روش‌ها بر حل مسایل تصمیم‌گیری ترتیبی که در آن، عامل‌ها و یا تصمیم‌ها به منظور یادگیری شرایط متفاوت با محیط اطراف تعامل می‌کنند، متمرکز هستند و به همین دلیل می‌توانند در ترکیب با شبکه‌های عصبی عمیق، نقش مؤثری در استخراج ویژگی‌ها داشته باشند.	یادگیری تقویتی عمیق	[۴۱]
۵	شبکه‌های پیچیده	ارتباطات بات‌نت‌ها هم از لحاظ شباهت و هم از لحاظ پایداری، شکل می‌گیرد. اقدامات ارتباطی بات‌نت‌ها مبتنی بر مکانیزم‌های ضربان قلب، منجر به شکل‌گیری نوعی گراف همبستگی می‌شود و به همین دلیل شبکه‌های پیچیده می‌توانند مبتنی بر دو تکنیک گراف و انجمن‌کاوی، بات‌نت‌های شبکه را شناسایی نمایند.	گراف	[۴۲]
			انجمن‌کاوی	[۴۳]
۶	هوش ازدحامی	الگوریتم‌های بهینه‌سازی هوش ازدحامی، عمدتاً به شبیه‌سازی رفتار گروهی پرندگان، حشرات و ماهی‌ها و مانند آن که به صورت جمعی در جستجوی غذا هستند، می‌پردازند. ایده اصلی به کارگیری این تکنیک‌ها در راستای تشخیص بات‌نت‌ها، استفاده از رفتار بیولوژیک و مبتنی بر هیوریستیک برای جستجو و یافتن نقاط ناهنجار، استخراج ویژگی و ترکیب با طبقه‌بندها است.	تکنیک PSO	[۴۴]
			تکنیک GWO	[۴۵]
۷	تحلیل آماری	این روش‌ها عمدتاً مبتنی بر مدل‌های داده ایجادشده از آمار ویژگی‌ها و پارامترها عمل می‌نمایند تا موارد ناهنجار را پیدا نموده و وجود یا عدم وجود رفتار بات‌نت‌ها را تخمین بزنند.	آنتروپی اطلاعات	[۴۶]
			قدم تصادفی	[۴۷]
			جستجوی کوانتومی	[۴۸]
			رویکرد MTD	[۴۹]
۸	رویکردهای توزیع‌شده	هدف اصلی این روش‌ها، گردآوری حجم بالا و چندبعدی داده با هدف شناسایی دقیق‌تر بات‌نت‌ها است.	رویکرد SDN	[۵۰]
			زنجیره بلوکی	[۵۱]
۹	روش‌های ترکیبی	این روش‌ها عمدتاً بر ترکیب تکنیک‌های تشخیص بات‌نت با هدف افزایش دقت، استوار هستند که راهکار پیشنهادی در این مقاله در این دسته جای می‌گیرد.	چندبعدی	[۲۶]
			چندعامله	[۲۷]
			چندتکنولوژی	[۲۸]

بالا توسعه داده شد. البته تحقیقات موجود در حوزه پردازش‌های ترکیبی به معماری لامبدا محدود نگردید و راهکارهای دیگری نیز نظیر Liquid [۷۰]، چارچوب Summingbird [۷۱]، Kappa [۷۲] که با LinkedIn ارائه شده است، RADStack [۷۳] و HDBS [۷۴] به عنوان راهکارهای جایگزین برای پردازش‌های ترکیبی ارائه گردیدند.

یکی از چالش‌های مهم پردازش‌های ترکیبی، انتخاب ویژگی‌های مناسب و متنوع جهت ساخت مدل‌های پایه می‌باشد. تا کنون تحقیقات گسترده‌ای با هدف کاهش ابعاد کلان‌داده‌ها صورت گرفته است که کلیه روش‌های مرتبط با آن را می‌توان به دو دسته انتخاب ویژگی و استخراج ویژگی تفکیک نمود [۷۵]. تکنیک‌های انتخاب ویژگی با حذف ویژگی‌های غیر مرتبط و تکراری، به حفظ ویژگی‌های مرتبط و تأثیرگذار می‌پردازند؛ در حالی که تکنیک‌های استخراج ویژگی، فضای ویژگی‌های اولیه را به یک فضای ویژگی جدید با ابعاد کمتر تبدیل می‌نمایند [۷۶].

از آنجا که روش‌های انتخاب ویژگی، زیرمجموعه‌ای از ویژگی‌های واقعی و اصلی را حفظ می‌کنند، بهتر از تکنیک‌های استخراج ویژگی قادر به بازتاب معنای فیزیکی داده‌ها بوده و تفسیر و خوانایی مدل را به طرز مناسب‌تری ارائه می‌دهند [۷۷]. همچنین مدل‌های یادگیری جریانی ساخته‌شده مبتنی بر تکنیک‌های انتخاب ویژگی تنها نیاز به پردازش ویژگی‌های منتخب دارند، در حالی که در روش‌های استخراج ویژگی، به تمامی ویژگی‌ها جهت انتقال داده از فضای ویژگی اولیه به ابعاد پایین‌تر نیاز خواهد بود [۷۸] که این امر از میزان بلادرنگ‌بودن عملیات تشخیص تکنیک‌های جریانی می‌کاهد. این دلایل سبب می‌شوند که تکنیک‌های انتخاب ویژگی بیش از تکنیک‌های استخراج ویژگی در کاربردهای دنیای واقعی به کار گرفته شوند [۷۶] و [۷۷].

تکنیک‌های انتخاب ویژگی را از منظر ثابت یا متغیربودن مجموعه ویژگی‌های منتخب می‌توان در دو دسته ایستا و پویا تفکیک نمود. روش‌های انتخاب ویژگی ایستا بر استخراج یک زیرمجموعه ثابت از ویژگی‌های اولیه که تا حد امکان غیر تکراری و مرتبط هستند، تمرکز می‌نمایند و مجموعه ویژگی‌های منتخب در طول زمان تغییر نمی‌کنند [۷۵]، [۷۹] و [۸۰]. در مقابل، روش‌های پویا بر متغیربودن ویژگی‌ها و ابعاد فضای ویژگی در زمان‌های متفاوت [۱۹] و [۸۱] و یا برای داده‌هایی در گروه‌های متفاوت [۸۲] و [۸۳] تأکید می‌نمایند و بر این اساس، ارتباط

بین ویژگی‌ها را به صورت تدریجی، اندازه‌گیری و به‌روزرسانی می‌کنند [۸۴]. زیرا ممکن است یک ویژگی در زمان t با ویژگی دیگر وابسته باشد و در زمان $t+1$ وابسته نباشد.

پژوهش‌های محدودی در حوزه اعمال الگوریتم‌های انتخاب ویژگی پویا بر روی داده‌های دسته‌ای و غیر جریانی موجود است [۸۲] تا [۸۵]. در مقابل با توجه به ماهیت ذاتی داده‌های جریانی که با مفاهیمی نظیر ورود جریانی ویژگی‌ها یا داده‌ها، تغییر مداوم ابعاد فضای ویژگی و مفاهیمی نظیر وجود تغییر محتوا[1] و تغییر ویژگی[2] سر و کار دارند، روش‌های انتخاب ویژگی پویا، عمدتاً برای حل مسئله انتخاب ویژگی جریانی یا برخط [۱۷]، [۱۸] و [۸۴] به کار گرفته شده و در بسیاری از کاربردها، معادل یکدیگر در نظر گرفته می‌شوند. بر این اساس در سال ۲۰۰۳، روش grafting [۸۶] به عنوان نخستین روش در حوزه انتخاب ویژگی جریانی ارائه گردید. پس از آن در سال ۲۰۰۵ [۸۷]، ایده فضای ویژگی داینامیک ارائه شد و سپس روش Alpha investing [۸۸] مبتنی بر مفهوم p-value مطرح گردید.

پس از روش‌های اولیه، روش دیگری با عنوان OSFS [۸۹] ارائه شد که از تئوری اطلاعات و مفهوم Markov blanket برای حل مسئله انتخاب ویژگی جریانی استفاده می‌نمود. همچنین در سال ۲۰۱۳ نسخه به روز شده این الگوریتم با عنوان Fast-OSFS [۹۰]، کارایی عملیات انتخاب ویژگی جریانی را از طریق تفکیک مرحله تحلیل تکراری‌بودن الگوریتم به دو مرحله تحلیل تکراری‌بودن درونی و بیرونی، افزایش داد.

در سال ۲۰۱۱ تکنیکی برای انتخاب مجموعه‌ای از ویژگی‌ها مبتنی بر محاسبه تدریجی و به‌روزرسانی میزان ارتباط بین ویژگی‌های ماتریس دودویی مطرح شد [۹۱] و پس از آن الگوریتم طبقه‌بندی جریانی DX-Miner [۹۲] که شامل انتخاب ویژگی پویا مبتنی بر یک روش فیلتر نظارتی یا بدون نظارت بود، ارائه گردید. روش HEFT [۹۳] در سال ۲۰۱۲ از یک فیلتر مبتنی بر همبستگی سریع به عنوان یک فیلتر نظارتی برای انتخاب ویژگی‌های بااهمیت هر قطعه پنجره‌ای از داده‌های جریانی استفاده نمود. روش SAOLA [۹۴] نیز مبتنی بر تحلیل تئوری همبستگی بین ویژگی‌ها برای مقایسه دوبه‌دوی آنها به ارائه رویکرد برخط، مقیاس‌پذیر و دقیق انتخاب ویژگی جریانی در مجموعه دادگان چندبعدی پرداخت.

در حوزه مجموعه‌های سخت، در سال ۲۰۱۳ الگوریتم DIA-RED [۹۵] برای مدیریت ویژگی‌های جریانی ارائه گردید. همچنین الگوریتم OS-NRRSAR-SA [۹۶] نیز در سال ۲۰۱۶ مبتنی بر مجموعه‌های سخت و بدون نیاز به داشتن دانش اولیه نسبت به فضای ویژگی‌ها، به حل مسئله انتخاب ویژگی جریانی پرداخت و در سال ۲۰۱۸، روشی [۹۷] برای کنترل فضای ویژگی ناشناخته در مسئله انتخاب ویژگی جریانی با استفاده از مفهوم تحلیل ارزشمندی ویژگی‌ها در تئوری مجموعه‌های سخت ارائه شد.

مرجع [۹۸] در سال ۲۰۱۶ یک روش انتخاب ویژگی مبتنی بر عدم قطعیت متقارن (ر پایه تئوری اطلاعات است) ارائه داد. این روش در [۹۹] در سال ۲۰۱۹ با مفهوم انتخاب پویای عدم قطعیت متقارن برای داده‌های جریانی تحت عنوان الگوریتمی به نام DISCUSS توسعه یافت.

الگوریتم انتخاب ویژگی پویای DCFS [۱۰۰] در سال ۲۰۱۹ به عنوان روشی مبتنی بر همبستگی ارائه گردید. همچنین الگوریتم ABFS [۱۰۱] به عنوان یک الگوریتم تقویتی تطبیقی مبتنی بر ترکیبی از تقویت‌کننده‌ها و استامپ‌های تصمیم برای انتخاب ویژگی ارائه شد. به علاوه در زمینه یادگیری تقویتی، Xu و همکاران [۷۸] مسئله انتخاب ویژگی را به عنوان یک فرایند تصمیم‌گیری ترتیبی فرموله کرده و از طریق ترکیب عملیات انتخاب ویژگی با Q-learning به ارائه یک روش انتخاب ویژگی پویا پرداختند.

همچنین Sahmoud و همکارانش [۱۰۲] در سال ۲۰۱۹ به ارائه یک فریمورک عمومی با هدف طبقه‌بندی داده‌های جریانی پرداختند. این فریمورک وجود تغییر در ویژگی‌ها را از طریق مؤلفه تشخیص تغییر ویژگی خود احراز می‌کند و به محض تشخیص تغییر، یک الگوریتم تکاملی چندهدفه پویا به نام DFBFS که یک تکنیک انتخاب ویژگی مبتنی بر فیلتر پویا است، با هدف انتخاب مجموعه‌ای بهینه از ویژگی‌ها اجرا می‌شود.

۳- راهکار پیشنهادی

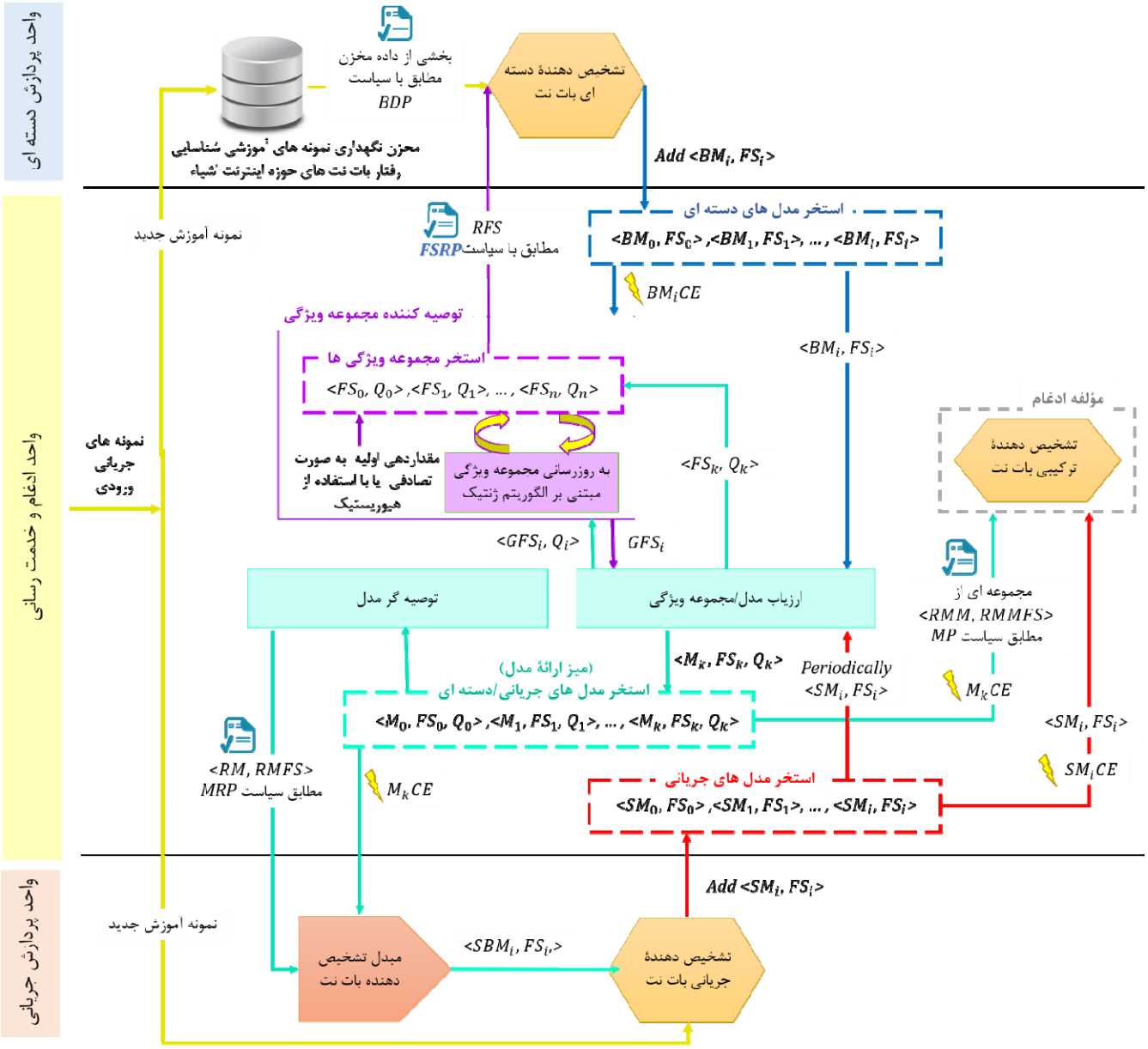
پیشنهاد مقاله حاضر ارائه روشی مبتنی بر اعمال انتخاب ویژگی پویا در ترکیب پردازش‌های دسته‌ای و جریانی با هدف تشخیص باتنت‌های حوزه اینترنت اشیا می‌باشد. شکل ۱ معماری مفهومی راهکار پیشنهادی را در قالب سه واحد پردازش دسته‌ای، پردازش جریانی و واحد ادغام و خدمت‌رسانی به تصویر می‌کشد. در ادامه به شرح سه واحد مذکور خواهیم پرداخت.

۳-۱ واحد پردازش دسته‌ای

وظیفه این واحد، ساخت مدل‌های دسته‌ای دقیق از روی حجم بالای داده‌های آموزشی واردشده به سیستم و مبتنی بر مجموعه ویژگی‌های متنوع تولیدی در لایه خدمت‌رسان می‌باشد. در حقیقت تمرکز اصلی این لایه بر افزایش دقت تشخیص‌دهنده باتنت‌ها استوار است. لذا با توجه به آن که داده‌های آموزشی به صورت جریانی به سیستم وارد می‌شوند، این لایه می‌بایست پیش از هر چیز به یک مخزن داده جهت ذخیره‌سازی داده‌های آموزشی واردشده به سیستم، مجهز باشد. این واحد به طور تکرارشونده، اقدام به ساخت مدل‌های دسته‌ای بر روی داده‌های ذخیره‌شده در مخزن داده‌ها می‌نماید. این مدل‌های دسته‌ای دارای دو ویژگی می‌باشند:

- هر مدل دسته‌ای مبتنی بر بخشی از داده‌های ذخیره‌شده در مخزن که از طریق سیاست داده‌های دسته‌ای[3] (BDP) تعیین می‌گردد، ساخته می‌شود. به عنوان مثال مطابق سیاست تعیین‌شده در این بخش، ساخت مدل دسته‌ای می‌تواند هر بار بر اساس کل داده‌های موجود در مخزن مرجع یا l داده اخیر موجود در مخزن مرجع صورت پذیرد. همچنین می‌توان این سیاست را به گونه‌ای تنظیم نمود که هر مدل دسته‌ای صرفاً مبتنی بر داده‌هایی که پس از ساخت مدل قبلی به مخزن مرجع افزوده شده‌اند، ساخته شود.
- هر مدل دسته‌ای مبتنی بر یک مجموعه ویژگی توصیه‌شده (RFS)[4] که توسط مؤلفه توصیه‌گر مجموعه ویژگی پیشنهاد می‌گردد، ساخته می‌شود. بنابراین مدل‌های دسته‌ای توسعه داده شده در این رویکرد، دارای مجموعه ویژگی‌های متفاوت می‌باشند. مؤلفه توصیه‌گر مجموعه ویژگی، یکی از مؤلفه‌های واحد ادغام و

1. Concept Drift
2. Feature Drift
3. Batch Data Policy
4. Recommended Feature Set



شکل ۱: معماری مفهومی راهکار پیشنهادی.

```
Each newTD is stored in MDB
BM0 = {Ø}
Add_to_Pool (<BM0, FS0>, BMP)
i=1
while (true)
    Batch_Data = BDP (MDB)
    FS = RFS
    BMi = BBD (Batch_Data, FSi)
    Add_to_Pool (<BMi, FSi>, BMP)
    Inc (i)
End
```

شکل ۲: شبه‌کد عملکرد واحد پردازش دسته‌ای.

خدمت‌رسانی می‌باشد که در بخش مربوط شرح داده خواهد شد.

- به محض اتمام فرایند ساخت یک مدل دسته‌ای، عملیات ساخت مدل دسته‌ای جدید مبتنی بر داده‌های ذخیره‌شده در مخزن و مجموعه ویژگی RFS تکرار می‌گردد. لذا مدل‌های دسته‌ای ایجادشده توسط این مؤلفه، به صورت متناوب بازسازی می‌شوند.

شکل ۲، شبه‌کد عملکرد واحد پردازش دسته‌ای را به اختصار نشان می‌دهد. در این شبه‌کد، *newTD* بیانگر داده آموزشی جدید واردشده به سیستم و *BBD*[1] مؤلفه تشخیص‌دهنده دسته‌ای بات‌نت است که می‌تواند شامل انواع طبقه‌بندهای رایج باشد، به شرط آن که تعریف طبقه‌بند جریانی متناسب با آن و نیز تعریف مؤلفه تبدیل‌کننده طبقه‌بند دسته‌ای به پایه طبقه‌بند جریانی امکان‌پذیر باشد. مؤلفه *BBD* پس از ساخت مدل دسته‌ای مناسب، این مدل و مجموعه ویژگی‌های متناسب با آن (*FS*) را از طریق تابع *Add_to_Pool* در استخر مدل‌های دسته‌ای (*BMP*) که در واحد ادغام و خدمت‌رسانی قرار دارد، اضافه می‌نماید.

۳-۲ واحد پردازش جریانی

دومین واحد پردازشی راهکار پیشنهادی که به صورت موازی با واحد پردازش دسته‌ای عمل می‌نماید، واحد پردازش جریانی نام دارد. وظیفه این واحد، پردازش جریان داده‌ها بر اساس بخشی از داده‌ها که به تازگی وارد شده‌اند، می‌باشد. مدل یادگیری این لایه به صورت تدریجی عمل کرده و با گذشت زمان تکمیل‌تر می‌شود. این لایه می‌تواند نتایج پردازش خود را به صورت بلادرنگ اعلام نماید. تمرکز اصلی لایه پردازش جریانی، بر

1. Batch BotNet Detector

```
while (true)
   if MkCE is received from BSMP
      FSi = RMFS
      SBMi = BD_Convertor (RM)
      SMi = Initialize_SM (SBMi, FSi)
      Add_to_Pool (<SMi, FSi>, SMP)
   if newTD is received
      SMi = SBD (newTD, SMi)
      Update_Pool (<SMi, FSi>, SMP)
End
```

شکل ۳: شبه‌کد عملکرد واحد پردازش جریانی.

افزایش سرعت پاسخگویی تشخیص‌دهنده ناهنجاری استوار است، لذا انجام عملیات پردازش دسته‌ای که نیازمند زمان بالایی است، در این واحد مناسب نمی‌باشد. بر این اساس، لایه پردازش جریانی، نیازی به ذخیره‌سازی داده‌های جریانی ورودی و ساخت مدل‌های پردازشی دقیق بر روی داده‌های ذخیره‌شده نخواهد داشت. در این واحد باید مؤلفه‌ای به منظور ساخت مدل‌های جریانی به عنوان مؤلفه تشخیص‌دهنده جریانی باتنت[1] (*SBD*) تعبیه گردد. مدل‌های جریانی (*SM*) خروجی این مؤلفه دارای ویژگی‌های زیر می‌باشند:

- هر مدل جریانی بر پایه یک مدل توصیه‌شده (*RM*)[2] که توسط واحد توصیه‌گر مدل پیشنهاد می‌گردد ساخته می‌شود و پس از آن، این مدل پایه به تدریج و با ورود داده‌های جریانی جدید، تکامل می‌یابد. مؤلفه توصیه‌گر مدل به منظور انتخاب یک مدل توصیه‌شده، استخر مدل‌های جریانی و دسته‌ای (*BSMP*)[3] یا همان میز ارائه مدل را که در بر گیرنده مدل‌هایی با مجموعه ویژگی‌های متنوع است، بر اساس معیارهایی جستجو می‌کند. بر این اساس مدل‌های جریانی در این رویکرد، بر پایه مدل توصیه‌شده و مجموعه ویژگی مربوط به آن مدل توصیه‌شده (*RMFS*)[4] تکامل می‌یابند. واحد توصیه‌گر مدل و استخر *BSMP* (میز ارائه مدل)، از مؤلفه‌های واحد ادغام و خدمت‌رسانی می‌باشند که در بخش مربوط شرح داده خواهند شد.
- پایه مدل جریانی (*SBM*)[5]، خروجی مؤلفه مبدل تشخیص‌دهنده باتنت[6] می‌باشد. مدل توصیه‌شده توسط مؤلفه توصیه‌گر مدل به مبدل وارد می‌شود و مبدل پس از انجام تغییرات ساختاری بر روی ساختمان داده مدل ورودی، آن را به عنوان پایه اولیه برای مدل جریانی بعدی تبدیل می‌کند. پایه مدل جریانی که به طور کامل با ساختار درخت‌های جریانی، سازگار شده است به عنوان ورودی به مؤلفه *SBD* وارد می‌شود و با ورود داده آموزشی جدید به تدریج تکامل می‌یابد.

شکل ۳، شبه‌کد عملکرد واحد پردازش جریانی را به اختصار نشان می‌دهد. در این شبه‌کد به محض دریافت رخداد M_kCE، مدل توصیه‌شده (*RM*) و مجموعه ویژگی مربوط به آن (*RMFS*) به تابع *BD_Convertor* وارد می‌شوند تا مدل توصیه‌شده را از طریق تغییر ساختمان داده، به پایه اولیه مدل جریانی بعدی SM_i تبدیل نمایند. سپس تابع *Initialize_SM*، مدل جریانی SM_i را بر پایه SBM_i و مبتنی بر مجموعه ویژگی *RMFS* ایجاد کرده و از طریق تابع *Add_to_Pool* در استخر مدل‌های جریانی (*SMP*)[7] که در واحد ادغام و خدمت‌رسانی قرار دارد، اضافه می‌نماید. همچنین با ورود هر داده آموزشی جدید، عملیات به‌روزرسانی تدریجی مدل جریانی SM_i از طریق مؤلفه *SBD* و متناسب با *newTD* انجام خواهد شد و پس از آن، مدل بروزرسانی شده از طریق تابع *Update_Pool* در استخر مدل‌های جریانی در لایه ادغام و خدمت‌رسانی قرار خواهد گرفت.

۳-۳ واحد ادغام و خدمت‌رسانی

سومین واحد مورد بررسی در راهکار پیشنهادی، واحد ادغام و خدمت‌رسانی نام دارد که مسئولیت‌های حساسی را عهده‌دار می‌باشد. انجام عملیات تولید مجموعه ویژگی‌های متنوع و انتخاب مجموعه ویژگی مناسب جهت ساخت مدل‌های دسته‌ای، ارزیابی مدل‌های پردازشی و انتخاب مدل پردازشی مناسب برای قرارگیری به عنوان پایه مدل جریانی و بارگذاری و ارائه سریع مدل‌های پردازشی تولیدشده در دو واحد پردازش دسته‌ای و جریانی جهت ترکیب سریع و مؤثر مدل‌ها در قالب الگوریتم تشخیص‌دهنده ترکیبی باتنت (HBD)[8]، از جمله مهم‌ترین وظایف این واحد است.

در این راستا، واحد ادغام و خدمت‌رسانی باید مجهز به برخی پایگاه داده‌هایی باشد که قابلیت خواندن و دسترسی‌پذیری سریع به داده‌ها را فراهم آورند. واحدهای پردازش دسته‌ای و جریانی، مدل‌های تولیدی خود را بر روی این پایگاه داده‌ها قرار می‌دهند و واحد ادغام و خدمت‌رسانی نیز مدل‌های قرارگرفته در این پایگاه داده‌های سریع را در مواقع لزوم، در اختیار مؤلفه ادغام (*MC*)[9] که مسئولیت ترکیب مدل‌های پردازشی در تشخیص‌دهنده ترکیبی نهایی را بر عهده دارد، می‌گذارد. در معماری پیشنهادی، از این پایگاه داده‌ها با عنوان استخر یاد شده است. در ادامه، ضمن معرفی این استخرها و نیز مؤلفه ادغام، به ذکر جزئیات عملکرد واحد خواهیم پرداخت. شکل ۴، شبه‌کد عملکرد این واحد را به اختصار نشان می‌دهد.

۳-۳-۱ استخر مدل‌های دسته‌ای

مدل‌های دسته‌ای مبتنی بر مجموعه ویژگی‌های متنوعی که توسط مؤلفه توصیه‌گر ویژگی پیشنهاد می‌گردد، ساخته می‌شوند. پس از تکمیل فرایند ایجاد یک مدل دسته‌ای، این مدل به همراه مجموعه ویژگی‌های به کار گرفته شده در آن $\langle BM_i, FS_i \rangle$، بر روی استخر مدل‌های دسته‌ای قرار می‌گیرد. به محض قرارگیری مدل بر روی استخر مدل‌های دسته‌ای، به طور هم‌زمان وقایع زیر به وقوع می‌پیوندند:

- از یک سو رخداد BM_iCE برای مؤلفه توصیه‌گر مجموعه ویژگی ارسال می‌شود تا این مؤلفه از پایان‌یافتن فرایند ساخت مدل دسته‌ای مطلع گردد و مجموعه ویژگی جدیدی را برای ساخت مدل دسته‌ای بعدی توسط تشخیص‌دهنده دسته‌ای باتنت (*BBD*) پیشنهاد نماید.
- از سوی دیگر مدل دسته‌ای ایجادشده و مجموعه ویژگی‌های به کار گرفته شده در آن $\langle BM_i, FS_i \rangle$، به منظور طی مراحل ارزیابی و کیفیت‌سنجی، به مؤلفه ارزیاب مدل[10] ارسال می‌شوند. این مؤلفه،

1. Stream BotNet Detector
2. Recommended Model
3. Batch/Stream Models Pool
4. Recommended Model Feature Set
5. Stream Based Model
6. BotNet Detector Convertor (BD Convertor)
7. Stream Models Pool
8. Hybrid BotNet Detector
9. Merging Component
10. Model Evaluator

```
k = 1
while (true)
   if new <BMi, FSi> add to BMP
      Send_Event (BMiCE to FSR)
      Mk = BMi
      FSk = FSi
      Qk = Evaluator (Mk, FSk)
      Add_to_Pool (<Mk, FSk, Qk>, BSMP)
      Update_Pool (<FSk, Qk>, FSP)
      inc (k)
   if new <SMi, FSi> add to SMP
      Send_Event (SMiCE to MC)
      Send <SMi, FSi> to MC
      After each period Time 'PT'
      Mk = SMi
      FSk = FSi
      Qk = Evaluator (Mk, FSk)
      Add_to_Pool (<Mk, FSk, Qk>, BSMP)
      Update_Pool (<FSk, Qk>, FSP)
      inc (k)
   if new <Mk, FSk, Qk> add to BSMP
      Send_Event (MkCE to BD Conventor)
      <RM, RMFS> = M_Recom (BSMP, MRP)
      Send <RM, RMFS> to BD Convertor
      Send_Event (MkCE to MC)
      <RMM, RMMFS> = M_Recom (BSMP, MP)
      Send a set of <RMM, RMMFS> to MC
   if FSR received BMiCE from BMP
      FSP = Genetic_Algorithm (FSP)
      RFS = FS_Recom (FSP, FSRP)
      Send RFS to BBD
End
```

شکل ۴: شبه‌کد عملکرد واحد ادغام و خدمت‌رسانی.

کیفیت مدل ورودی را بر اساس معیارهای متفاوتی اندازه‌گیری می‌نماید. برخی از این معیارها نظیر دقت مدل (در مقابل مجموعه‌ای از داده‌های آموزش که به منظور تست در نظر گرفته شده‌اند)، تعداد ویژگی‌های به کار گرفته شده در مدل و میزان داده‌هایی که مدل بر اساس آنها آموزش دیده است، در خصوص تمامی مدل‌ها قابل اندازه‌گیری است. اما برخی دیگر نظیر عمق درخت یا عمق مؤثر درخت و مواردی از این دست، با توجه به طبقه‌بند دسته‌ای و جریانی به کار گرفته شده در مؤلفه‌های *SBD* و *BBD* قابل تعریف خواهند بود. نهایتاً مؤلفه ارزیاب مدل/ مجموعه ویژگی، خروجی ارزیابی‌های خود را در قالب یک بردار کیفیت Q اعلام می‌نماید و مدل ارزیابی‌شده، مجموعه ویژگی به کار گرفته شده در مدل و مقادیر پارامترهای کیفیت مدل مورد نظر $\langle M_k, FS_k, Q_k \rangle$ را به استخر مدل‌های دسته‌ای و جریانی (میز ارائه مدل) وارد می‌نماید.

۳-۳-۲ استخر مدل‌های جریانی

هر مدل جریانی بر پایه یک مدل توصیه‌شده (*RM*) که توسط واحد توصیه‌گر مدل پیشنهاد می‌گردد، ساخته می‌شود و پس از آن، این مدل پایه به تدریج و با ورود داده‌های جریانی جدید، تکامل می‌یابد. در ابتدا، پایه مدل جریانی به همراه مجموعه ویژگی‌های به کار گرفته شده در آن، $\langle SM_i, FS_i \rangle$، بر روی استخر مدل‌های جریانی قرار می‌گیرد. از این پس، مدل جریانی پس از هر بار تکامل تدریجی، روی استخر مدل‌های جریانی به‌روزرسانی خواهد شد. به محض قرارگیری یا به‌روزرسانی مدل بر روی استخر مدل‌های جریانی، به طور هم‌زمان وقایع زیر به وقوع می‌پیوندند:

- از یک سو رخداد SM_iCE به همراه آخرین مدل جریانی قرارگرفته بر روی استخر *SMP* و مجموعه ویژگی‌های مربوط به آن $\langle SM_i, FS_i \rangle$ برای مؤلفه ادغام (*MC*) ارسال می‌شود تا این مؤلفه نسبت به به‌روزرسانی مدل جریانی مورد نیاز خود بر اساس مدلی که به تازگی تولید یا به‌روزرسانی شده، اقدام نماید. جزئیات عملکرد مؤلفه ادغام در بخش ۳-۳-۵ شرح داده خواهد شد.
- از سوی دیگر پس از گذشت بازه زمانی PT[1] (و یا به‌روزرسانی مدل جریانی با تعداد مشخصی داده آموزشی) مدل جریانی به‌روزرسانی شده و مجموعه ویژگی‌های به کار گرفته شده در آن $\langle SM_i, FS_i \rangle$، به منظور طی مراحل ارزیابی و کیفیت‌سنجی، به مؤلفه ارزیاب مدل ارسال می‌شوند. این مؤلفه، همان گونه که در خصوص ارزیابی مدل‌های دسته‌ای عنوان شد، کیفیت مدل ورودی را بر اساس معیارهای متفاوتی اندازه‌گیری و خروجی ارزیابی‌های خود را در قالب $\langle M_k, FS_k, Q_k \rangle$ به استخر مدل‌های دسته‌ای و جریانی (میز ارائه مدل) وارد می‌نماید.

۳-۳-۳ استخر مدل‌های دسته‌ای و جریانی (*BSMP*)

مدل‌های جریانی و دسته‌ای پس از قرارگیری بر روی استخرهای *BMP* و *SMP* و انجام مراحل کیفیت‌سنجی از طریق مؤلفه ارزیاب مدل، در قالب $\langle M_k, FS_k, Q_k \rangle$ به استخر مدل‌های دسته‌ای و جریانی وارد می‌شوند. از آنجا که کلیه مدل‌های ایجادشده بر روی این استخر، ذخیره و به سایر مؤلفه‌ها ارائه می‌گردند، به این استخر، میز ارائه مدل نیز گفته می‌شود. به محض قرارگیری یک مدل جدید بر روی میز ارائه مدل، به طور هم‌زمان وقایع زیر به وقوع می‌پیوندند:

- از یک سو رخداد M_kCE برای مؤلفه مبدل در واحد پردازش جریانی ارسال می‌شود و به این واحد اطلاع می‌دهد که باید نسبت به تغییر پایه مدل جریانی اقدام نماید. هم‌زمان مؤلفه توصیه‌گر مدل نیز، میز ارائه مدل را که دربرگیرنده مدل‌هایی با مجموعه ویژگی‌های متنوع است، بر اساس سیاست توصیه مدل (MRP)[2] جستجو می‌کند و از میان مدل‌های موجود، یکی از مدل‌ها را به همراه مجموعه ویژگی مربوط به آن مدل $\langle RM, RMFS \rangle$ در قالب مدل توصیه‌شده به مؤلفه مبدل وارد می‌نماید تا این مؤلفه پس از انجام تغییرات لازم بر روی مدل توصیه‌شده، آن را به عنوان پایه‌ای برای تکامل تدریجی مدل جریانی در *SBD* قرار دهد. نکته قابل توجه آن است که سیاست توصیه مدل، به طور مستقیم به پارامترهای ارزیابی ارائه‌شده توسط مؤلفه ارزیاب مدل وابسته است و بسته به سیاست در نظر گرفته شده، می‌تواند تابع یک یا چندمتغیره باشد. به عبارت دیگر، انتخاب یک مدل مناسب به عنوان پایه مدل جریانی، می‌تواند وابسته به یکی از پارامترهای ارزیابی مدل و یا ترکیبی از چند پارامتر ارزیابی‌شده باشد. به عنوان نمونه، در صورتی که ارزیابی مدل از طریق پارامترهای عمومی نظیر دقت مدل، تعداد ویژگی‌های به کار گرفته شده در مدل و میزان داده‌های مشارکت‌کننده در ساخت مدل صورت پذیرد، آن گاه سیاست توصیه مدل می‌تواند انتخاب مدلی با الف) بیشترین دقت، ب) کمترین تعداد ویژگی، ج) بیشترین داده دخیل در فرایند آموزش و د) بیشترین دقت حاصل با کمترین تعداد ویژگی و یا هر ترکیب دیگری از پارامترهای ارائه‌شده باشد.
- از سوی دیگر، رخداد M_kCE برای مؤلفه ادغام (*MC*) نیز ارسال می‌شود تا این مؤلفه نسبت به به‌روزرسانی مدل‌های ترکیبی مورد نیاز خود بر اساس وضعیت جدید میز ارائه خدمت و مبتنی بر سیاست ادغام (MP)[3]، اقدام نماید. هم‌زمان مؤلفه توصیه‌گر مدل

1. Periodic Time
2. Model Recommendation Policy
3. Merge Policy

نیز، میز ارائه مدل را که دربرگیرنده مدل‌هایی با مجموعه ویژگی‌های متنوع است، بر اساس سیاست ادغام جستجو می‌کند و از میان مدل‌های موجود، تعدادی از مدل‌ها را به همراه مجموعه ویژگی‌های مربوط به آن مدل‌ها $\langle RMM, RMMFS \rangle$ در قالب مجموعه مدل‌های توصیه‌شده برای ترکیب نهایی به مؤلفه ادغام وارد می‌نماید. جزئیات عملکرد مؤلفه ادغام و سیاست ادغام در بخش ۳-۳-۵ شرح داده خواهد شد.

۳-۳-٤ استخر مجموعه ویژگی‌ها

استخر مجموعه ویژگی‌ها، مسئولیت فراهم‌سازی مجموعه ویژگی‌های متنوع جهت دستیابی به انتخاب ویژگی پویا را که هدف اصلی مقاله حاضر است بر عهده دارد. این استخر حاوی مجموعه ویژگی‌های متنوعی می‌باشد که هر یک از آنها بیانگر حضور تعدادی از ویژگی‌ها و عدم حضور تعدادی دیگر در یک مجموعه هستند. اگر تعداد ویژگی‌های داده‌های ورودی را d فرض کنیم، آن گاه تعداد مجموعه ویژگی‌های ممکن برای قرارگیری در استخر، 2^d مجموعه خواهد بود. با توجه به بالابودن تعداد ویژگی‌های بسیاری از داده‌های دنیای واقعی، بدیهی است که استخر مجموعه ویژگی‌ها می‌بایست دارای ظرفیت محدود باشد و نمی‌تواند تمام مجموعه ویژگی‌های ممکن را شامل شود. این استخر برخلاف سایر استخرها که پیش از این شرح داده شد، در ابتدا مقداردهی اولیه می‌شود. عملیات مقداردهی اولیه استخر ویژگی‌ها می‌تواند مبتنی بر یکی از سه حالت زیر باشد:

- انتخاب کلیه مجموعه ویژگی‌های اولیه به صورت تصادفی
- انتخاب کلیه مجموعه ویژگی‌ها مبتنی بر هیوریستیک و تجربه افراد خبره که باید در خصوص ویژگی‌های هر مجموعه دادگان به صورت اختصاصی مورد استعلام قرار گیرد.
- ترکیبی از دو روش فوق به این معنا که کلیه مجموعه ویژگی‌های مورد تأیید افراد خبره در خصوص هر مجموعه دادگان (در صورت وجود افراد خبره) در استخر ویژگی‌ها قرار می‌گیرد و مابقی مجموعه ویژگی‌ها به صورت تصادفی به استخر اضافه می‌گردند تا ظرفیت آن تکمیل شود.

همچنین استخر ویژگی‌ها در کنار هر مجموعه ویژگی، کیفیت آن مجموعه را نیز از منظر مدلی که بر مبنای آن مجموعه ویژگی ساخته می‌شود، نگهداری می‌کند. البته روشن است که در هنگام مقداردهی اولیه استخر ویژگی‌ها، هنوز هیچ مدلی بر مبنای مجموعه ویژگی‌های موجود در استخر ایجاد نشده و بنابراین می‌توان برای محاسبه کیفیت مجموعه ویژگی در این زمان، وابسته به پارامترهای موجود برای کیفیت‌سنجی، بر مبنای برخی پیش‌فرض‌ها عمل نمود. مثلاً دقت تمامی مجموعه ویژگی‌ها در ابتدا می‌تواند برابر ۵۰٪ در نظر گرفته شود.

پس از مقداردهی اولیه استخر ویژگی‌ها، فعالیت این بخش زمانی آغاز می‌شود که رخداد BM_iCE به نشانه اتمام فرایند ساخت یک مدل دسته‌ای برای مؤلفه توصیه‌گر مجموعه ویژگی ارسال شود. در این زمان، محتویات استخر ویژگی‌ها مبتنی بر الگوریتم ژنتیک به‌روزرسانی می‌گردد. به این منظور استخر ویژگی‌ها به منزله جمعیت و هر مجموعه ویژگی در این استخر به منزله یک کروموزم با d ژن در نظر گرفته می‌شود که هر ژن بیانگر وجود (مقدار ۱) یا عدم وجود (مقدار ۰) آن ویژگی در کروموزوم مربوط است.

در این مرحله، مطابق اصول کلی الگوریتم‌های ژنتیک، میزان برازندگی کروموزم‌های جمعیت از طریق یک تابع برازش محاسبه می‌شود. این تابع، برازندگی هر مجموعه ویژگی (کروموزوم) را از منظر مدلی که بر مبنای آن مجموعه ویژگی ساخته می‌شود، محاسبه می‌نماید. بنابراین تابع برازش در این مسئله می‌تواند یکی از پارامترهای خروجی مؤلفه ارزیاب مدل/ مجموعه ویژگی و یا ترکیبی از پارامترهای خروجی به شکل مناسب باشد. به عنوان مثال می‌توان میزان برازندگی مجموعه ویژگی‌ها را متناسب با دقت مدل ساخته‌شده بر مبنای آن و یا ترکیبی از پارامترهای دقت و تعداد ویژگی‌های موجود در مجموعه ویژگی و یا هر ترکیب معتبر دیگر از پارامترهای خروجی مؤلفه ارزیاب مدل دانست. به منظور دستیابی به این هدف، کلیه مدل‌های واردشده به مؤلفه ارزیاب مدل، پس از طی مراحل کیفیت‌سنجی، باید بردار کیفیت Q را علاوه بر استخر مدل‌های دسته‌ای و جریانی، به استخر مجموعه ویژگی‌ها نیز در قالب $\langle FS_k, Q_k \rangle$ ارسال نمایند تا به این ترتیب، پارامترهای کیفیت را برای مجموعه ویژگی‌هایی که بر مبنای آنها مدل ساخته شده است، به‌روزرسانی نمایند.

پس از محاسبه برازندگی کلیه مجموعه ویژگی‌های موجود در استخر ویژگی‌ها، می‌توان به انتخاب کروموزم‌های برتر و اعمال عملگرهای الگوریتم ژنتیک نظیر تقاطع و جهش روی آنها پرداخت. همچنین می‌توان در صورت لزوم، به تولید مجدد کروموزم‌های جدید نیز اقدام نمود و به این ترتیب، جمعیت کروموزم‌ها به میزان یک نسل به‌روزرسانی می‌شود. لازم به ذکر است کلیه کروموزم‌هایی که طی این فرایند تولید می‌گردند (GFS)[1] باید جهت کیفیت‌سنجی به مؤلفه ارزیاب مدل/ مجموعه ویژگی ارسال شوند.

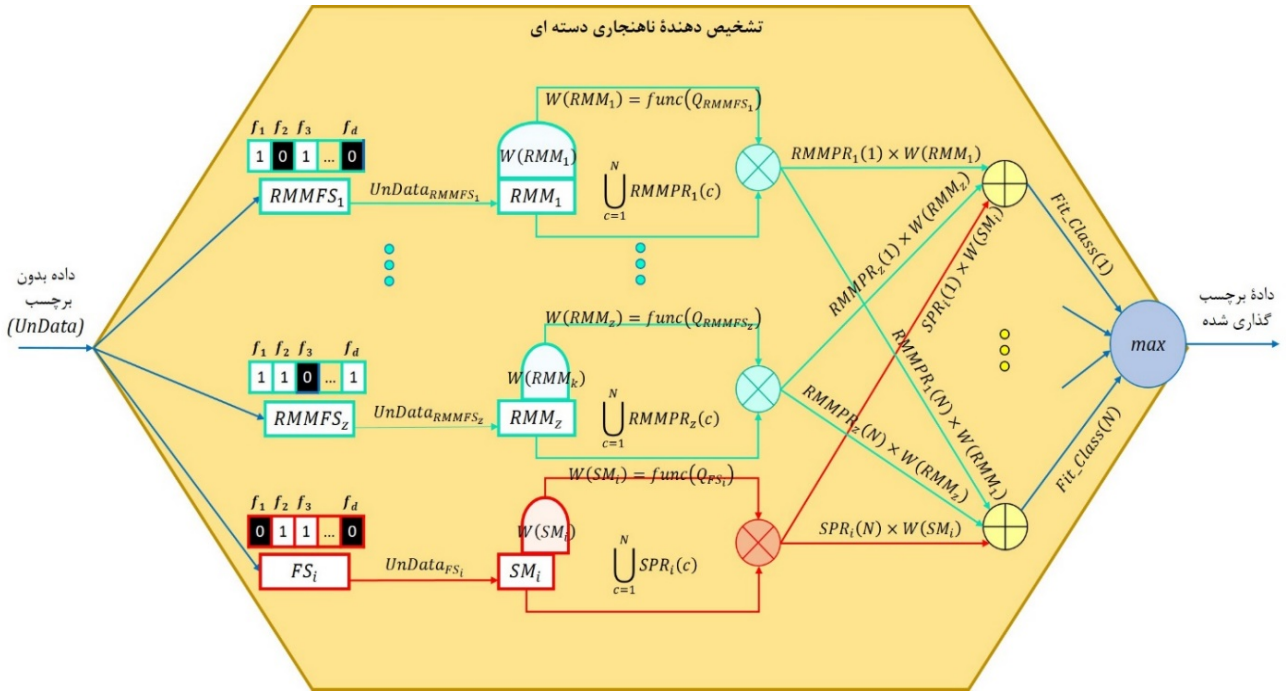
پس از اتمام فرایند به‌روزرسانی جمعیت کروموزم‌ها که در حقیقت همان به‌روزرسانی مجموعه ویژگی‌های موجود در استخر ویژگی‌ها است، مؤلفه توصیه‌گر مجموعه ویژگی، مبتنی بر سیاست توصیه مجموعه ویژگی (FSRP)[2]، اقدام به توصیه یک مجموعه ویژگی بهینه برای ساخت مدل دسته‌ای بعدی توسط تشخیص‌دهنده ناهنجاری دسته‌ای (BBD) می‌نماید. عملیات به‌روزرسانی جمعیت کروموزم‌ها یا همان استخر ویژگی‌ها برای نسل‌های بعد نیز به محض ساخت یک مدل دسته‌ای جدید، ادامه خواهد یافت تا پس از گذشت چندین نسل، جمعیت کروموزم‌ها به مجموعه ویژگی مناسب‌تر و بهینه‌تری همگرا گردد. البته لازم به ذکر است که ذات چارچوب ارائه‌شده، مبتنی بر انتخاب ویژگی پویا می‌باشد. به این معنا که حتی پس از همگراشدن عملیات انتخاب ویژگی به یک مجموعه ویژگی مشخص، ممکن است به دلایل گوناگون نظیر تغییر نوع داده، ورود نمونه‌های متنوع و جدید ناهنجاری، مجموعه ویژگی دیگری به عنوان مجموعه ویژگی توصیه‌شده برای ساخت مدل‌های دسته‌ای بعدی به کار گرفته شود.

۳-۳-۵ مؤلفه ادغام و الگوریتم HBD

مؤلفه ادغام در حقیقت بخش تجمیع‌کننده و نهایی راهکار پیشنهادی محسوب می‌شود و با ایجاد پیوستگی میان واحدهای پردازشی متفاوت، می‌کوشد تا عملیات تشخیص بات‌نت را با دقت و سرعت مطلوبی به انجام رساند. این مؤلفه، مجهز به یک تشخیص‌دهنده بات‌نت ترکیبی به نام HBD است که با ورود داده‌های جدید و ساخته‌شدن مدل‌های جریانی و دسته‌ای متفاوت، خود را به‌روزرسانی می‌نماید. عملیات به‌روزرسانی HBD در هنگام وقوع ۲ رخداد صورت می‌پذیرد:

- به محض قرارگیری یا به‌روزرسانی یک مدل جریانی (SM_i) بر روی استخر مدل‌های جریانی، رخداد SM_iCE به همراه آخرین

1. Generated Feature Set
2. Feature Set Recommendation Policy



شکل ۵: معماری داخلی مؤلفه ادغام.

مدل جریانی قرارگرفته روی استخر SMP و مجموعه ویژگی‌های مربوط به آن $\langle SM_i, FS_i \rangle$ برای مؤلفه ادغام (MC) ارسال می‌شود تا این مؤلفه نسبت به به‌روزرسانی مدل جریانی مورد نیاز خود بر اساس مدلی که به تازگی تولید یا به‌روزرسانی شده است، اقدام نماید.

- مدل‌های جریانی و دسته‌ای پس از قرارگیری بر روی استخرهای BMP و SMP و انجام مراحل کیفیت‌سنجی از طریق مؤلفه ارزیاب مدل، در قالب $\langle M_k, FS_k, Q_k \rangle$ به استخر مدل‌های دسته‌ای و جریانی وارد می‌شوند. به محض قرارگیری یک مدل جدید بر روی این استخر، رخداد $M_k CE$ برای مؤلفه ادغام (MC) ارسال می‌شود تا این مؤلفه نسبت به به‌روزرسانی مدل‌های ترکیبی مورد نیاز خود بر اساس وضعیت جدید میز ارائه مدل اقدام نماید. هم‌زمان مؤلفه توصیه‌گر مدل نیز، میز ارائه مدل را که دربرگیرنده مدل‌هایی با مجموعه ویژگی‌های متنوع است، بر اساس سیاست ادغام جستجو می‌کند و از میان مدل‌های موجود، تعدادی از مدل‌ها را به همراه مجموعه ویژگی‌های مربوط به آن مدل‌ها $\langle RMM, RMMFS \rangle$ در قالب مجموعه مدل‌های توصیه‌شده برای ترکیب نهایی به مؤلفه ادغام وارد می‌نماید.

شکل ۵، معماری داخلی الگوریتم HBD و نحوه عملکرد آن را نشان می‌دهد. مطابق این شکل، الگوریتم HBD بر اساس آخرین درخت جریانی موجود در استخر مدل‌های جریانی (SM_i) و تعداد z مدل دسته‌ای یا جریانی توصیه‌شده توسط واحد توصیه‌گر مدل یعنی RMM_1 تا RMM_z عمل می‌نماید. با توجه به این که کلیه مدل‌های جریانی و دسته‌ای فوق مبتنی بر مجموعه ویژگی‌های متنوع و نیز مجموعه دادگان آموزشی متفاوت ساخته شده‌اند، لذا این مجموعه عملاً دید بسیار مناسبی را نسبت به داده‌های جدید و پیشین در اختیار تشخیص‌دهنده ترکیبی HBD قرار می‌دهد.

به این ترتیب اصلی‌ترین واحد مؤلفه ادغام که الگوریتم HBD است، به مرور زمان از یک سو با ورود جریان داده‌های آموزشی جدید $(newTD)$[1] به‌روزرسانی می‌شود و از سوی دیگر، آمادگی کامل خود را برای برچسب‌گذاری جریان ورودی ناشناس $(UnData)$[2] حفظ می‌کند و ارتقا می‌دهد.

زمانی که یک داده ورودی ناشناس $(UnData)$ به سیستم وارد می‌شود، مستقیماً به مؤلفه ادغام هدایت می‌گردد و این مؤلفه (مطابق شکل ۵) داده ورودی را به منظور برچسب‌گذاری به الگوریتم HBD تحویل می‌دهد. الگوریتم HBD به صورت هم‌زمان، داده را به z درخت توصیه‌شده توسط مؤلفه توصیه‌گر مدل یعنی RMM_1 تا RMM_z و آخرین مدل جریانی موجود در استخر مدل‌های جریانی یعنی (SM_i) وارد می‌کند. اما از آنجایی که هر یک از مدل‌های مذکور دارای مجموعه ویژگی‌های متفاوتی می‌باشند، لذا الگوریتم HBD باید قبل از ورود داده به مدل‌های مذکور، آن را مطابق الگوی مربوط به مجموعه ویژگی خاص هر مدل، آماده‌سازی نماید. این بدان معناست که تنها مقادیر ویژگی‌هایی که در الگوی مجموعه ویژگی خاص هر مدل برابر ۱ می‌باشند، باید به آن مدل وارد شوند و مقادیر سایر ویژگی‌های داده ورودی حذف گردند.

مطابق (۱) خروجی این عملیات z ترکیب ویژگی متفاوت از داده ورودی یعنی $UnData_{RMMFS_1}$ تا $UnData_{RMMFS_z}$ می‌باشد که به لحاظ ترکیب ویژگی‌ها، امکان ورود به مدل‌های توصیه‌شده را دارا خواهند بود

$$\forall_{\substack{j=1:z \\ m=1:d}} UnData_{RMMFS_j}(f_m) = \begin{cases} UnData(f_m) & \text{if } RMMFS_j(f_m)=1 \\ Remove & \text{if } RMMFS_j(f_m)=0 \end{cases} \quad (1)$$

به این ترتیب هر مدل توصیه‌شده RMM_j به صورت مجزا، اقدام به برچسب‌گذاری احتمالی داده $UnData_{RMMFS_j}$ می‌نماید. مطابق (۲) با فرض وجود N کلاس متفاوت، خروجی این عملیات برای هر مدل RMM_j، به صورت آرایه N درایه‌ای $RMMPR_j$ خواهد بود که درایه

1. New Training Data
2. Unknown Data

cام آن بیانگر احتمال تخصیص برچسب کلاس c توسط مدل توصیه‌شده RMM_j به داده ورودی می‌باشد

$$\forall_{j=1:z} RMMPR_j(c) = prob(RMM_j(UnData_{RMMFS_j}), c) \quad (۲)$$

همچنین در خصوص مدل جریانی SM_i نیز عملیاتی مشابه با مدل‌های توصیه‌شده انجام گردیده و پس از آماده‌سازی $UnData$ مطابق الگوی مربوط به مجموعه ویژگی FS_i، برچسب‌گذاری احتمالی آن به صورت مجزا انجام می‌پذیرد. معادله (۳) فرایند آماده‌سازی داده $UnData$ را نشان می‌دهد. خروجی این عملیات داده $UnData_{FS_i}$ با ترکیب ویژگی‌هایی مطابق با FS_i می‌باشد که امکان ورود به مدل SM_i را دارا خواهد بود

$$\forall_{m=1:d} UnData_{FS_i}(f_m) = \begin{cases} UnData(f_m) & \text{if } FS_i(f_m)=1 \\ Remove & \text{if } FS_i(f_m)=0 \end{cases} \quad (۳)$$

همچنین (۴) خروجی عملیات برچسب‌زنی به داده $UnData_{FS_i}$ توسط مدل SM_i را به صورت آرایه N درایه‌ای SPR_i[1] نشان می‌دهد که درایه cام آن بیانگر احتمال تخصیص برچسب کلاس c توسط مدل جریانی SM_i به داده ورودی می‌باشد

$$SPR_i(c) = prob(SM_i(UnData_{FS_i}), c) \quad (۴)$$

پس از اتمام فرایند برچسب‌گذاری احتمالی داده ورودی توسط مدل‌های مختلف، باید عملیات ترکیب برچسب‌های تخصیص‌یافته بر اساس ارزش و وزن مربوط به هر مدل انجام پذیرد. معادله (۵)، عملیات ترکیب نتایج را بر اساس محاسبه تابع $Fit-Class$ نشان می‌دهد

$$\forall_{c=1:N} Fit-Class(UnData, c) = (W(SM_i)\times SPR_i(c)) + \sum_{j=i-k}^{i-1} (W(RMM_j)\times RMMPR_j(c)) \quad (۵)$$

در این معادله $W(SM_i)$ بیانگر وزن مربوط به مدل جریانی SM_i و $W(RMM_j)$ بیانگر وزن تخصیص داده شده به مدل‌های توصیه‌شده RMM_1 تا RMM_z است. این وزن‌ها در حقیقت تابعی از خروجی کیفیت‌سنجی و یا میزان برازندگی مجموعه ویژگی تخصیص‌یافته به هر مدل می‌باشند و به صورت (۶) و (۷) محاسبه می‌شوند

$$\forall_{j=1:z} W(RMM_j) = func(Q_{RMMFS_j}) \quad (۶)$$

$$W(SM_i) = func(Q_{FS_i}) \quad (۷)$$

که Q_{RMMFS_j} و Q_{FS_i} بیانگر خروجی کیفیت‌سنجی و یا میزان برازندگی مجموعه ویژگی‌های تخصیص‌یافته به مدل‌های توصیه‌شده و آخرین مدل جریانی می‌باشد و $func(.)$ بیانگر یک تابع نمونه است که می‌تواند مطابق سیاست ادغام (MP) تعریف شود.

سرانجام پس از محاسبه تابع $Fit-Class$ به ازای تمامی کلاس‌های c، برچسب نهایی داده ورودی از طریق (۸) مشخص خواهد شد. بر اساس این معادله، داده ورودی به کلاسی تخصیص می‌یابد که دارای بیشترین برازندگی بر اساس تابع $Fit-Class$ باشد

$$label = Index(\max_{c=1:N} Fit-class(UnData, c)) \quad (۸)$$

در انتها لازم به ذکر است که سیاست ادغام، عملکرد مؤلفه ادغام و الگوریتم تشخیص‌دهنده ترکیبی باتنت (HBD) را کنترل می‌نماید. این سیاست، مدل‌های مؤثر و میزان تأثیرگذاری آنها را از طریق وزن‌هایی که به هر مدل اختصاص می‌دهد، کنترل می‌کند. نحوه گزینش و انتخاب مدل‌های توصیه‌شده برای فرایند ادغام که مشابه با سیاست MRP می‌تواند حالات متفاوت داشته باشد، تعیین تعداد مدل‌های توصیه‌شده و انتخاب پارامتر z، روش وزن‌دهی به هر مدل و تعریف تابع $func(.)$ در (۶) و (۷)، شیوه برچسب‌گذاری داده‌های ورودی توسط مدل‌ها و نحوه تجمیع نتایج خروجی مدل‌ها، از جمله مواردی هستند که در تعریف این سیاست، نقش اساسی ایفا می‌نمایند که باید متناسب با نیاز و محدودیت‌های هر مسئله تعریف و انتخاب گردند. زیرا هر سیاست، عملاً تبعات متفاوتی را بر پیچیدگی زمانی و مکانی، میزان بلادرنگ‌بودن پاسخ‌دهی، حجم حافظه اشغال‌شده و مواردی از این دست خواهد داشت.

٤- ارزیابی راهکار پیشنهادی

در این بخش به ارزیابی راهکار پیشنهادی می‌پردازیم. بر این اساس نخست در بخش ۴-۱ مجموعه دادگان مورد استفاده را توصیف می‌کنیم، سپس به ذکر معیارهای ارزیابی گوناگون در بخش ۴-۲ می‌پردازیم و سرانجام نتایج آزمایش‌های تجربی را در بخش ۴-۳ ارائه خواهیم داد.

٤-١ مجموعه دادگان

برای ارزیابی روش پیشنهادی از مجموعه داده‌ای با عنوان N-BaIoT [۱۰۳] و [۱۰۴] استفاده می‌نماییم. این مجموعه، شامل داده‌های ترافیکی تولیدشده توسط ۹ ابزار تجاری اینترنت اشیا است که با استفاده از ۱۰ نوع حمله توسط دو نمونه از باتنت‌های متداول حوزه اینترنت اشیا با عنوان‌های Mirai و Bashlite [۱۰۵] آلوده شده‌اند. هدف ما در به کارگیری این مجموعه داده، ایجاد تمایز و تفکیک میان داده‌های ترافیکی هنجار و ناهنجار است. لذا در این آزمایش‌ها، کلیه رکوردهای مربوط به حملات مختلف را به عنوان ناهنجار و سایر رکوردها را به عنوان هنجار، برچسب‌گذاری نمودیم. این مجموعه داده شامل ۷۰۶۲۶۰۶ رکورد و ۱۱۵ ویژگی برای هر نمونه است که در اینجا از یک زیرمجموعه شامل ۸۵۵۹۳۲ رکورد برای ارزیابی نتایج استفاده شده است.

٤-٢ معیارهای ارزیابی

معیارهای ارزیابی مورد استفاده در این مقاله عبارت هستند از:

- قدرت تشخیص باتنت: این معیار بیانگر میزان دقت راهکار در تخصیص برچسب مناسب به جریان داده‌های ناشناس ورودی است و برای محاسبه این معیار، از ۲ معیار زیرمجموعه استفاده می‌نماییم:

۱) دقت: این معیار به طور میانگین، نسبت تعداد داده‌هایی را که به درستی برچسب خورده‌اند به تعداد کل داده‌های ورودی نشان می‌دهد.

۲) مساحت زیر نمودار ROC: این معیار مساحت زیر نمودار ROC را که محور افقی آن بیانگر نرخ مثبت اشتباه (FPR) و محور عمودی آن بیانگر نرخ مثبت صحیح (TPR) است، ارائه می‌دهد.

- پیچیدگی زمانی: این معیار از دو منظر قابل حصول است که بر این اساس به دو معیار زیرمجموعه قابل تفکیک می‌باشد:

۱) زمان برچسب‌گذاری: این معیار که "زمان پاسخ‌دهی" نیز نامیده می‌شود، بیانگر میانگین زمان مورد نیاز برای برچسب‌گذاری یک داده ورودی ناشناس می‌باشد و از این طریق دید مناسبی را نسبت

1. Stream Probabilistic Result

جدول ۲: نتایج آزمایش‌های مربوط به بررسی نوع طبقه‌بند و تأثیر انتخاب ویژگی و انتخاب هوشمندانه مدل بر عملکرد تشخیص‌دهنده باتنت مبتنی بر طبقه‌بند درخت تصمیم.

رویکردهای تشخیص باتنت	نوع طبقه‌بند			انتخاب ویژگی و مدل		قدرت تشخیص باتنت		تعداد ویژگی‌ها		پیچیدگی زمانی (میکروثانیه)		کد
	دسته‌ای	جریانی	ترکیبی	انتخاب ویژگی	انتخاب مدل	دقت	مساحت نمودار ROC	به طور میانگین	در انتهای آزمایش	زمان برچسب‌زنی	زمان به‌روزرسانی مدل	
غیر ترکیبی	✓	✗	✗	✗	✗	۰٫۹۲۵	۰٫۸۹۳	۱۱۵	۱۱۵	۱۱۲٫۱	۱۶۹۴٫۲ ثانیه	DTB۰۰
	✗	✓	✗	✗	✗	۰٫۹۴۹	۰٫۹۳۲	۱۱۵	۱۱۵	۸۵٫۴	۲۹۴٫۹	DTS۰۰
ترکیبی	✓	✓	✓	✗	✗	۰٫۹۸۸	۰٫۹۸۴	۱۱۵	۱۱۵	۱۲۲٫۶	۲۷۵	DTH۰۰
	✓	✓	✓	✓	✗	۰٫۹۷۱	۰٫۹۵۹	۲۲٫۲	۲۶	۲۰۱٫۲	۱۲۱٫۷	DTH۰۱
	✓	✓	✓	✗	✓	۰٫۹۹۶	۰٫۹۹۶	۱۱۵	۱۱۵	۱۶۰٫۸	۲۲۸٫۸	DTH۱۰
	✓	✓	✓	✓	✓	۰٫۹۹۹	۰٫۹۹۹	۷۲٫۲	۱۵	۱۹۳٫۷	۱۴۰٫۵	DTH۱۱

جدول ۳: نتایج آزمایش‌های مربوط به بررسی نوع طبقه‌بند و تأثیر انتخاب ویژگی و انتخاب هوشمندانه مدل بر عملکرد تشخیص‌دهنده باتنت مبتنی بر طبقه‌بند بیز ساده گاوسی.

رویکردهای تشخیص باتنت	نوع طبقه‌بند			انتخاب ویژگی و مدل		قدرت تشخیص باتنت		تعداد ویژگی‌ها		پیچیدگی زمانی (میکروثانیه)		کد
	دسته‌ای	جریانی	ترکیبی	انتخاب ویژگی	انتخاب مدل	دقت	مساحت نمودار ROC	به طور میانگین	در انتهای آزمایش	زمان برچسب‌زنی	زمان به‌روزرسانی مدل	
غیر ترکیبی	✓	✗	✗	✗	✗	۰٫۹۹۴	۰٫۹۹۵	۱۱۵	۱۱۵	۷۵۳٫۵۵	۴٫۱۴۷ ثانیه	NBB۰۰
	✗	✓	✗	✗	✗	۰٫۶۳۱	۰٫۶۰۷	۱۱۵	۱۱۵	۷۵۰٫۶۷	۵۱٫۶۲	NBS۰۰
ترکیبی	✓	✓	✓	✗	✗	۰٫۸۶۱	۰٫۸۳۹	۱۱۵	۱۱۵	۱۲۴۳٫۶	۳۷٫۴	NBH۰۰
	✓	✓	✓	✓	✗	۰٫۵۹۷	۰٫۶۶۱	۹٫۲	۹	۴۶۹٫۳	۱۹٫۷	NBH۰۱
	✓	✓	✓	✗	✓	۰٫۹۹۶	۰٫۹۹۷	۱۱۵	۱۱۵	۱۳۲۲	۳۴	NBH۱۰
	✓	✓	✓	✓	✓	۰٫۹۹۱	۰٫۹۹۲	۱۰	۱۰	۴۲۶٫۵	۱۹٫۸	NBH۱۱

به میزان بلادرنگ‌بودن راهکار پیشنهادی در اختیار قرار می‌دهد.

۲) زمان به‌روزرسانی: این معیار بیانگر میانگین زمان مورد نیاز برای آموزش تدریجی و به‌روزرسانی مدل یادگیری راهکار پیشنهادی، مبتنی بر ورود یک داده آموزشی جدید می‌باشد.

- تعداد ویژگی‌های منتخب: این معیار بیانگر تعداد ویژگی‌های انتخاب‌شده توسط راهکار می‌باشد که در دو حالت گزارش می‌شود:

۱) در حالت اول، تعداد ویژگی‌های منتخب نهایی پس از یادگیری راهکار پیشنهادی مبتنی بر کل جریان داده‌های ورودی گزارش می‌شوند که بیانگر تعداد ویژگی موجود در زیرمجموعه منتخب ویژگی‌ها است که راهکار انتخاب ویژگی پیشنهادی به آن همگرا شده است.

۲) در حالت دوم میانگین تعداد ویژگی‌های منتخب بعد از ورود هر داده آموزشی گزارش می‌شود و بیانگر آن است که راهکار پیشنهادی به طور میانگین با چه تعداد ویژگی، عملیات تشخیص باتنت را به انجام رسانده است.

۴-۳ نتایج آزمایش‌ها و تحلیل

در این مقاله برای اجرای آزمایش‌ها و ارزیابی نتایج مطابق معیارهای ارائه‌شده، از یک محیط شبیه‌سازی شده مبتنی بر یک ماشین مجازی با ۱۲ هسته CPU از نوع Intel (R) Xeon (R) E3-12xx v2 (Ivy Bridge, IBRS) با پردازنده ۲٫۶۹۴ GHz و حافظه RAM به میزان ۶۴ گیگابایت استفاده شده است. شبیه‌سازی پردازش‌های دسته‌ای و جریانی مبتنی بر نرم‌افزار Matlab انجام گرفته و موازی‌بودن اجرای پردازش‌های دسته‌ای و جریانی، صرفاً از طریق کنترل زمان اجرا، شبیه‌سازی گردیده است.

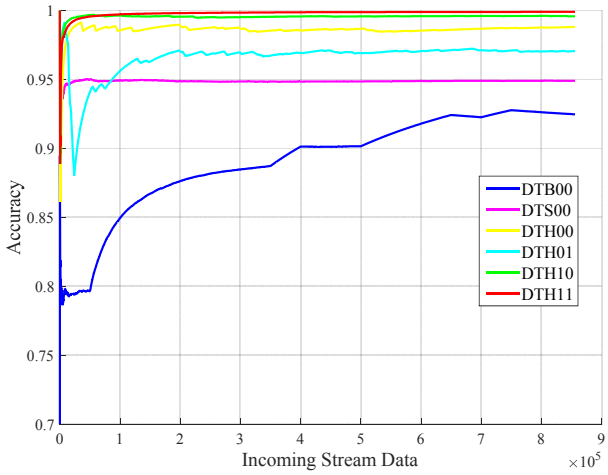
از آنجا که راهکار پیشنهادی مستقل از طبقه‌بند می‌باشد، مؤلفه‌های *BBD* و *SBD* می‌توانند از میان طبقه‌بندهای رایج دسته‌ای و جریانی جهت تشخیص باتنت‌ها انتخاب شوند؛ به شرط آن که مدل خروجی آنها با یکدیگر سازگار بوده و مؤلفه مبدل، با انجام تغییرات ساختاری بر روی ساختمان داده مدل ورودی، امکان تبدیل طبقه‌بند دسته‌ای به پایه طبقه‌بند جریانی را داشته باشد. بر این اساس به منظور ارزیابی راهکار پیشنهادی، از ۲ دسته طبقه‌بند استفاده می‌نماییم:

- دسته اول شامل طبقه‌بندهای مبتنی بر درخت تصمیم می‌باشد. در اینجا از الگوریتم *VFDT* به عنوان طبقه‌بند جریانی (*SBD*) و از *C*۴.۵ به عنوان طبقه‌بند دسته‌ای (*BBD*) استفاده شده است.
- دسته دوم شامل طبقه‌بندهای مبتنی بر بیز ساده گاوسی می‌باشد. در این آزمایش‌ها برای طبقه‌بند دسته‌ای از محاسبه میانگین و انحراف معیار هر کلاس مبتنی بر تمامی داده‌های موجود استفاده شده و برای طبقه‌بند جریانی، پارامترهای مذکور در پنجره زمانی اخیر تخمین زده شده‌اند.

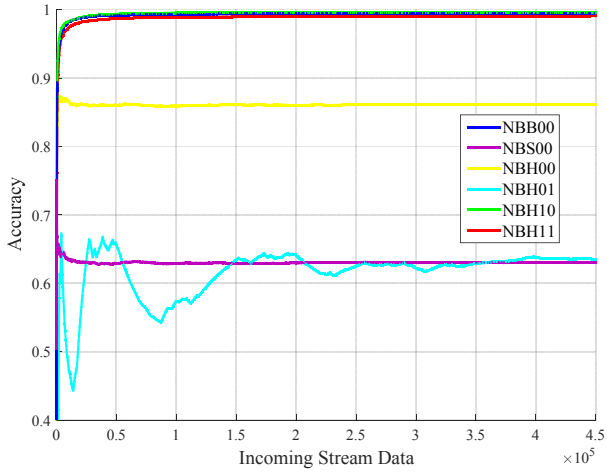
جدول ۲، نتایج ارزیابی راهکار پیشنهادی را مطابق معیارهای ارزیابی بخش ۴-۲ بر روی مجموعه دادگان N-BaIoT، مبتنی بر درخت تصمیم گزارش می‌کند. همچنین جدول ۳ به ارائه گزارش‌های مشابه در خصوص ارزیابی راهکار پیشنهادی مبتنی بر طبقه‌بند بیز ساده گاوسی می‌پردازد.

لازم به ذکر است که تمامی پارامترهای مورد نیاز برای پیاده‌سازی راهکار پیشنهادی مبتنی بر طبقه‌بندهای درخت تصمیم و بیز ساده گاوسی، در سطوح مختلف نظیر پارامترهای الگوریتم ژنتیک، پارامترهای طبقه‌بندها، پارامترهای مرتبط با سیاست‌های انتخاب و توصیه مجموعه ویژگی‌های منتخب و مدل‌های مناسب برای ترکیب نهایی در مؤلفه ادغام، در تمامی آزمایش‌ها با مقادیر یکسان در نظر گرفته شده‌اند تا از این جهت، مقایسه نتایج کسب‌شده امکان‌پذیر باشد.

در این راستا، سیاست BDP مبتنی بر l داده اخیر موجود در مخزن عمل می‌کند که در آن پارامتر l برابر با ۲۰۰۰ در نظر گرفته شده و در نتیجه مدل‌های دسته‌ای به طور متوالی مبتنی بر ۲۰۰۰ داده اخیر موجود در مخزن مرجع ساخته می‌شوند. سیاست *MP* نیز مبتنی بر ترکیب آخرین مدل جریانی موجود در استخر مدل‌های جریانی و ۵ مدل توصیه‌شده



شکل ۶: مقایسه دقت تشخیص باتنت در حالات شش‌گانه B۰۰، S۰۰، H۰۰، H۰۱، H۱۰ و H۱۱ با استفاده از طبقه‌بند درخت تصمیم.



شکل ۷: مقایسه دقت تشخیص باتنت در حالات شش‌گانه B۰۰، S۰۰، H۰۰، H۰۱، H۱۰ و H۱۱ با استفاده از طبقه‌بند بیز ساده گاوسی.

توسط واحد توصیه‌گر مدل ($z = ۵$) و با لحاظ‌کردن وزن برابر (۱/۶) برای هر یک از مدل‌ها، عمل می‌نماید.

مدل‌های جریانی به‌روزرسانی شده، پس از گذشت بازه زمانی *PT* (معادل به‌روزرسانی مدل جریانی با ۵۰۰۰ نمونه) به ارزیاب مدل وارد می‌شوند. مؤلفه ارزیاب، کیفیت مدل‌های ورودی را بر اساس معیار دقت مدل (در مقابل ۵۰۰۰ داده آموزش که به منظور تست در نظر گرفته شده‌اند)، اندازه‌گیری می‌نماید و بر این اساس سیاست‌های *MP* و *MRP* هر دو مبتنی بر همین معیار عمل کرده و به توصیه مدل‌های مناسب به ترتیب به واحدهای ادغام‌کننده مدل و مبدل مدل می‌پردازند. البته لازم به ذکر است که این معیار تنها در خصوص آزمایش‌های جداول ۲ و ۳ می‌باشد و در ادامه، معیارهای دیگر نیز مورد ارزیابی قرار خواهند گرفت.

عملیات به‌روزرسانی مجموعه ویژگی نیز که مبتنی بر الگوریتم ژنتیک است، با مقداردهی تصادفی جمعیت با تعداد ۲۰ کروموزوم (۲۰ زیرمجموعه از ویژگی‌ها) با نمایش دودویی[1] آغاز می‌گردد.

همچنین عملگر تقاطع مبتنی بر یک شکست[2] و با نرخ ۰/۸ و عملگر جهش مبتنی بر bit-flip و با نرخ ۰/۰۰۵ به ازای هر ژن پیاده‌سازی شده است. به علاوه از آنجا که مؤلفه ارزیاب تنها مبتنی بر معیار دقت عمل می‌نماید، لذا تابع برازش الگوریتم ژنتیک نیز بر همین مبنا خواهد بود و استخر مجموعه ویژگی‌ها نیز بر اساس معیار بالاترین دقت (به عنوان سیاست FSRP)، نسبت به توصیه مجموعه ویژگی‌های مناسب برای ساخت مدل‌های دسته‌ای اقدام می‌نماید.

همچنین در این آزمایش‌ها از روش ارزیابی Prequential [۱۰۶] استفاده شده که این روش ارزیابی، از ابتدا و به صورت ذاتی برای ارزیابی الگوریتم‌های یادگیری جریانی ارائه گردیده است. در این روش، هر داده ورودی به الگوریتم، نخست به وسیله مدلی که توسط داده‌های قبلی ساخته شده است، مورد ارزیابی قرار می‌گیرد و سپس خود آن داده، در فرایند آموزش مدل جدیدتر دخالت داده می‌شود. از این رو این روش اصطلاحاً interleaved test then train نامیده می‌شود.

همچنین در جداول مذکور، معیارهای ارزیابی، بسته به آن که عملیات انتخاب ویژگی و انتخاب هوشمندانه مدل فعال یا غیر فعال هستند و از طبقه‌بندهای جریانی و دسته‌ای به صورت منفرد و یا ترکیبی استفاده شده است، در شش حالت متفاوت گزارش شده‌اند. دو حالت نخست، صرفاً به بهره‌گیری از طبقه‌بندها منفرد دسته‌ای و جریانی، پیش از ترکیب بسنده نموده‌اند و لذا عملیات انتخاب ویژگی و انتخاب مدل نیز بر روی آن اعمال نشده است. اما چهار حالت انتهایی بر ترکیب طبقه‌بندهای دسته‌ای و جریانی تمرکز نموده و عملکرد رویکرد پیشنهادی را در حضور و عدم حضور عملیات انتخاب ویژگی و انتخاب مدل، بررسی نموده‌اند. به منظور افزایش قابلیت ارجاع نتایج ذکرشده در جداول ۲ و ۳ در حالت‌های شش‌گانه فوق، در مقابل هر یک از ردیف‌های جدول، یک کد پنج‌رقمی قرار داده شده است. دو رقم اول این کد بیانگر نوع طبقه‌بند مورد استفاده است که می‌تواند DT (برای طبقه‌بند درخت تصمیم) یا NB (برای طبقه‌بند بیز ساده گاوسی) باشد. رقم سوم بیانگر دسته‌ای‌بودن (B)، جریانی‌بودن (S) و یا ترکیبی‌بودن (H) رویکرد تشخیص باتنت است و سرانجام دو رقم انتهایی، فعال‌بودن یا غیر فعال‌بودن انتخاب ویژگی را مشخص می‌نمایند. به این ترتیب عملاً راهکارهای DTH۱۱ و NBH۱۱ را که هم از طبقه‌بند ترکیبی استفاده کرده و هم عملیات انتخاب ویژگی و انتخاب مدل در آن فعال است، می‌توان به عنوان راهکار پیشنهادی محسوب نمود.

همچنین به منظور اثبات آن که نتایج کسب‌شده از آزمایش‌ها، تصادفی نبوده و به لحاظ آماری معتبر می‌باشد، از تست فرضیه آماری مک‌نمار[3] [۱۰۷] استفاده گردیده است. این تست بیانگر آن است که آیا تفاوت میان دقت راهکار پیشنهادی در حالت‌های گوناگون، از نظر آماری معنادار می‌باشد یا خیر. به این منظور به صورت تصادفی، ۱۰۰۰۰ رکورد نمونه انتخاب گردید و تست مک‌نمار با مقدار آلفا برابر ۰/۰۵ به منظور مقایسه دقت حالت DTH۱۱ و NBH۱۱ (به عنوان روش پیشنهادی) با دقت سایر حالات، بر آن اعمال شد. مطابق نتایج کسب‌شده از این تست، p-value در بیشتر مواقع بسیار ناچیز و نزدیک به صفر می‌باشد و عملاً در تمامی موارد، فرضیه H۰ مربوط به تست مک‌نمار قابل ردکردن است. این امر بیانگر معتبربودن و وجود تفاوت آماری معنادار در نتایج آزمایش‌ها می‌باشد.

شکل ۶ میزان تأثیر فعال و غیر فعال بودن عملیات انتخاب ویژگی و انتخاب مدل و نیز بهره‌گیری از طبقه‌بندهای منفرد و یا ترکیبی را بر دقت تشخیص باتنت، مبتنی بر طبقه‌بند درخت تصمیم با یکدیگر مقایسه می‌نماید. شکل ۷ نیز نمودار مشابهی را در خصوص طبقه‌بند بیز به تصویر

1. Binary Representation
2. Single Point Crossover
3. McNemar's Statistical Hypothesis Test

می‌کشد. لازم به ذکر است از آنجا که در آزمایش‌های صورت‌گرفته، نتایج حاصل از معیارهای ارزیابی دقت و سطح زیر نمودار *ROC* تا حدود بسیار زیادی هم‌راستا و موافق یکدیگر هستند، لذا تحلیل‌ها و بحث‌های ارائه‌شده بر دقت تشخیص متمرکز شده‌اند.

مطابق نتایج ارائه‌شده در جداول و نمودارهای مذکور، دقت تشخیص بات‌نت در حالت H۱۱ که راهکار پیشنهادی می‌باشد، همواره بیشتر از حالت S۰۰ بوده و در مقایسه با دقت طبقه‌بند دسته‌ای غیر ترکیبی (حالت B۰۰) بزرگ‌تر و یا در سطح آن می‌باشد. این در حالی است که تعداد ویژگی‌ها در حالت H۱۱ به شدت کاهش یافته و این مسئله منجر به کاهش پیچیدگی زمانی این حالت نسبت به حالات S۰۰ و B۰۰ می‌گردد.

در خصوص مقایسه راهکارهای ترکیبی با یکدیگر، دقت تشخیص ناهنجاری در حالت H۱۱ که عملاً روش پیشنهادی محسوب می‌گردد، همواره بهتر از حالت H۰۰ بوده و تعداد ویژگی‌های آن نیز با بهره‌گیری از انتخاب ویژگی پویا به طور قابل ملاحظه‌ای کمتر از H۰۰ می‌باشد. به عبارت دیگر، از آنجا که H۰۰ تنها به ترکیب پردازش‌های دسته‌ای و جریانی بسنده می‌نماید، می‌توان راهکار پیشنهادی (H۱۱) را در حذف ویژگی‌های تکراری و نامرتبط پردازش‌های ترکیبی و در عین حال حفظ یا افزایش دقت این پردازش‌ها مؤثر دانست.

همچنین بررسی نتایج حالت H۰۱ بیانگر آن است که فعال‌سازی عملیات انتخاب ویژگی پویا بدون وجود انتخاب مدل هوشمند، گرچه قادر به کاهش تعداد ویژگی‌ها می‌باشد، اما نمی‌تواند دقت مطلوبی را در تشخیص ناهنجاری به همراه داشته باشد و همواره از دقت کمتری نسبت به سایر حالات ترکیبی برخوردار است. در مقابل، بررسی حالت H۱۰ نشان می‌دهد که بهره‌گیری از عملیات انتخاب مدل هوشمند در حضور تمام ویژگی‌ها (یعنی عدم استفاده از انتخاب ویژگی)، در بسیاری اوقات دارای دقت مطلوب و در سطح دقت H۱۱ می‌باشد اما طبیعتاً به دلیل بالابودن تعداد ویژگی‌ها، از پیچیدگی زمانی بیشتری نسبت به H۱۱ برخوردار است.

نتایج فوق، گویای مکمل‌بودن عملیات انتخاب ویژگی و انتخاب مدل در دستیابی به دقت و سرعت بالا در عملیات تشخیص بات‌نت معماری پیشنهادی می‌باشد. به بیانی دیگر، دقت تشخیص بات‌نت در حالت‌هایی که عملیات انتخاب مدل، فعال است (حالت‌های H۱۰ و H۱۱)، بیشتر از حالت‌هایی است که این عملیات غیر فعال می‌باشد و در مقابل، پیچیدگی زمانی در حالت‌هایی که عملیات انتخاب ویژگی پویا فعال است (حالت‌های H۰۱ و H۱۱)، اغلب اوقات کمتر از حالت‌هایی است که مبتنی بر تمامی ویژگی‌ها عمل می‌کنند. این بدان معناست که عملیات انتخاب ویژگی منجر به کاهش پیچیدگی زمانی و افزایش سرعت و عملیات انتخاب هوشمندانه مدل منجر به افزایش دقت تشخیص ناهنجاری می‌گردند. از این رو معماری پیشنهادی با تجمیع مناسب و سازگار این عملیات برای پردازش‌های ترکیبی دسته‌ای و جریانی، در مجموع از دقت و سرعت مناسب‌تری نسبت به سایر حالات برخوردار است.

البته در خصوص پیچیدگی زمانی، نمی‌توان از مکانیزم و نحوه عملکرد طبقه‌بندهای به کار گرفته شده در پردازش‌های دسته‌ای و جریانی صرف نظر نمود. به عبارت دیگر گاهی اوقات عملکرد یک طبقه‌بند به گونه‌ای است که پیچیدگی زمانی آن، رابطه مستقیم با تعداد ویژگی‌ها دارد و گاهی کاهش تعداد ویژگی‌ها، لزوماً منجر به بهبود پیچیدگی زمانی نمی‌گردد. به عنوان نمونه عملکرد طبقه‌بند بیز ساده گاوسی به دلیل نیاز به محاسبه میانگین و انحراف معیار برای تک‌تک ویژگی‌ها، به گونه‌ای است که هم زمان برچسب‌زنی و هم زمان به‌روزرسانی مدل، مستقیماً متناسب با تعداد ویژگی‌ها عمل می‌نمایند و لذا در جدول ۳، همواره پیچیدگی زمانی برای حالت‌هایی که دارای انتخاب ویژگی پویا هستند (حالت‌های H۰۱ و H۱۱) کمتر از حالت‌هایی است که مبتنی بر تمامی ویژگی‌ها عمل می‌کنند.

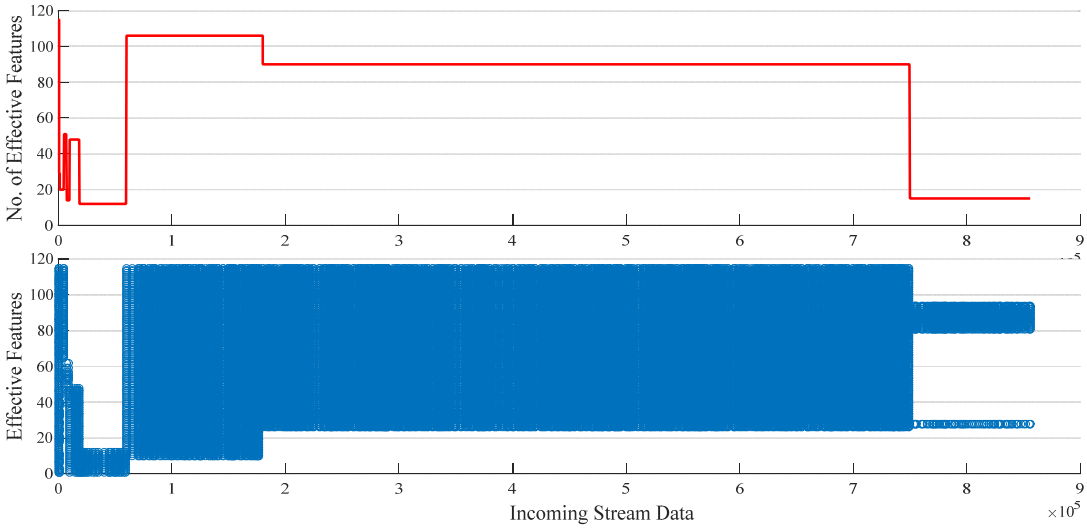
در مقابل، در خصوص طبقه‌بند درخت تصمیم، زمان برچسب‌زنی به یک داده ورودی جدید کاملاً مستقل از تعداد ویژگی‌های داده ورودی بوده و با عمق درخت رابطه مستقیم دارد. اما زمان به‌روزرسانی درخت تصمیم با یک داده آموزشی جدید، علاوه بر عمق درخت، به منظور به‌روزرسانی آمار مربوط به تک‌تک ویژگی‌ها در نود برگی که داده ورودی متعلق به آن است، وابسته به تعداد ویژگی‌ها نیز می‌باشد. از این رو در جدول ۲، همواره زمان به‌روزرسانی مدل برای حالاتی که دارای انتخاب ویژگی پویا هستند (حالت‌های H۰۱ و H۱۱)، کمتر از حالت‌هایی است که مبتنی بر تمامی ویژگی‌ها عمل می‌کنند اما زمان برچسب‌گذاری، برخی اوقات از این قاعده پیروی نمی‌کند.

مسئله فوق در مقایسه نتایج آزمایش‌های مربوط به DTH۱۰ و DTH۱۱ کاملاً مشهود است. در اینجا DTH۱۱ با بهره‌گیری از انتخاب ویژگی فعال به طور میانگین دارای ۷۲٫۲ ویژگی بوده و در انتها تنها مبتنی بر ۱۵ ویژگی عمل می‌کند، اما در DTH۱۰ انتخاب ویژگی غیر فعال بوده و همواره مبتنی بر ۱۱۵ ویژگی عمل می‌نماید. این در حالی است که مطابق بررسی‌های صورت‌گرفته، درخت جریانی به‌روزرسانی شده در DTH۱۱ پس از اتمام آزمایش‌ها، دارای عمق ۱۸ و عمق مؤثر ۸٫۳۷۵ می‌باشد، در حالی که عمق و عمق مؤثر درخت جریانی DTH۱۰ پس از اتمام آزمایش‌ها به ترتیب برابر ۵ و ۳٫۸۳۳ است. از این رو زمان برچسب‌زنی یک داده ورودی جدید نیز متناسب با عمق درخت، در DTH۱۰ کمتر از DTH۱۱ خواهد بود.

نکته دیگری که می‌بایست در آزمایش‌های صورت‌گرفته مد نظر قرار گیرد، مسئله پویابودن عملیات انتخاب ویژگی مبتنی بر الگوریتم ژنتیک می‌باشد. شکل ۸، این پویایی در انتخاب ویژگی‌های مؤثر را در هنگام اعمال راهکار پیشنهادی با بهره‌گیری از طبقه‌بند درخت تصمیم (حالت DTH۱۱) به تصویر می‌کشد. این تصویر شامل دو نمودار است: نمودار فوقانی، تعداد ویژگی‌های مؤثر و منتخب در فرایند تشخیص بات‌نت را در طول زمان و با ورود جریان داده‌ها نمایش می‌دهد و نمودار پایینی، شماره ویژگی‌های منتخب را تعیین می‌کند و نشان می‌دهد که کدام ویژگی‌ها منتخب هستند. این دو نمودار در کنار هم بیانگر آن هستند که الگوریتم انتخاب ویژگی پویای پیشنهادی، در هر زمان، کدام ویژگی‌ها را به عنوان ویژگی‌های مؤثر انتخاب نموده و این ویژگی‌های مؤثر مجموعاً چه تعداد می‌باشند.

مطابق این شکل، ویژگی‌های مؤثر به صورت پویا و در طول زمان، متناسب با نوع جریان داده ورودی تغییر می‌کنند و پس از مدتی به زیرمجموعه‌ای مشخص از ویژگی‌های منتخب و مؤثر همگرا می‌شوند. البته با توجه به پویابودن ذات الگوریتم، حتی پس از همگراشدن عملیات انتخاب ویژگی به یک مجموعه ویژگی مشخص، ممکن است به دلایل گوناگون نظیر تغییر نوع داده، ورود نمونه‌های متنوع و جدید ناهنجاری و مواردی از این دست، مجموعه ویژگی دیگری به عنوان مجموعه ویژگی منتخب و توصیه‌شده، برای تشخیص ناهنجاری نمونه‌های ورودی و ساخت مدل‌های دسته‌ای بعدی به کار گرفته شود.

نکته دیگری که باید در این آزمایش‌ها مد نظر قرار گیرد، بررسی تأثیر اتخاذ سیاست‌های توصیه مدل (*MRP*) متفاوت در کارایی راهکار پیشنهادی می‌باشد. همان گونه که پیش از این اشاره شد، سیاست‌های توصیه مدل، تابعی یک یا چندمتغیره از پارامترهای ارزیابی ارائه‌شده توسط مؤلفه ارزیاب مدل هستند. لذا به منظور بررسی تأثیر سیاست توصیه مدل



شکل ۸: عملکرد انتخاب ویژگی پویا در هنگام اعمال راهکار پیشنهادی بر جریان داده‌ها در حالت DTH۱۱.

جدول ۴: نتایج آزمایش‌های مربوط به بررسی تأثیر اعمال معیارهای متفاوت در تنظیم سیاست انتخاب مدل بر عملکرد تشخیص‌دهنده بات‌نت.

طبقه‌بند پایه	سیاست توصیه مدل	قدرت تشخیص بات‌نت		تعداد ویژگی‌ها		پیچیدگی زمانی (میکروثانیه)	
		دقت	مساحت نمودار ROC	به طور میانگین	در انتهای آزمایش	زمان برچسب‌زنی	زمان به‌روزرسانی مدل
درخت تصمیم	max_acc	۰٫۹۹۹	۰٫۹۹۹	۷۲٫۲	۱۵	۱۹۳٫۷	۱۴۰٫۵
	min_FNo	۰٫۹۲۱	۰٫۸۷۲	۷٫۳	۷	۲۰۵	۱۱۲
	min_age	۰٫۹۶	۰٫۹۴۶	۳۰٫۶	۲۵	۲۰۲٫۲	۱۲۷٫۸
بیز ساده گاوسی	max_acc	۰٫۹۹۱	۰٫۹۹۲	۱۰	۱۰	۴۲۶٫۵	۱۹٫۸
	min_FNo	۰٫۳۵۳	۰٫۴۹۲	۷	۷	۴۷۲٫۳	۱۹٫۴
	min_age	۰٫۶۲۹	۰٫۷۵۷	۹٫۵	۱۰	۴۶۸٫۳	۲۰٫۱

مناسب، سه پارامتر ارزیابی اولیه و عمومی (که در خصوص هر دو طبقه‌بند قابل محاسبه است) شامل: الف) دقت مدل، ب) تعداد ویژگی‌های به کار گرفته شده در مدل و ج) زمان ایجاد یا سن مدل را به عنوان پارامترهای ارزیابی مؤلفه ارزیاب مدل انتخاب نمودیم. جدول ۴ مبتنی بر این پارامترها، نتایج ارزیابی راهکار پیشنهادی را مبتنی بر سه سیاست: الف) توصیه مدلی با بیشترین دقت (max-acc)، ب) توصیه مدلی با کمترین تعداد ویژگی (min-FNo) و ج) توصیه جدیدترین مدل ایجادشده (min-age) گزارش می‌نماید.

به منظور قابل مقایسه‌بودن نتایج، کلیه پارامترهای موجود در اجرای راهکار پیشنهادی را مطابق آزمایش‌های مرتبط با حالت H۱۱ در جداول ۱ و ۲ تنظیم نموده و تنها *MRP* را مطابق سیاست‌های فوق تغییر دادیم. از آنجا که در جداول ۱ و ۲، سیاست توصیه مدل دقیقاً مشابه سیاست max-acc می‌باشد، لذا نتایج این سیاست در جدول ۴ عملاً مشابه نتایج مربوط به حالات H۱۱ در جداول ۲ و ۳ است. مطابق نتایج موجود در جدول ۴ با انتخاب سیاست توصیه مدل max-acc، دقت تشخیص بات‌نت بیشتر از سیاست‌های دیگر است که کاملاً قابل انتظار می‌باشد. همچنین با انتخاب سیاست توصیه مدل min-FNo، تعداد ویژگی‌های مؤثر نهایی و میانگین، در تمامی حالات، کمتر از سیاست‌های دیگر است. البته از آنجا که سیاست توصیه مدل min-FNo تمایل به انتخاب مدل‌هایی با کمترین تعداد ویژگی دارد و در شکل ایده‌آل، مدلی با یک یا حتی صفر ویژگی را انتخاب خواهد کرد (که عملاً غیر قابل قبول است)، لذا به منظور آن که نتایج ایجادشده از مقبولیت نسبی در دقت برخوردار باشند، در این آزمایش، محدودیتی برای کمترین تعداد ویژگی‌های منتخب (۷ ویژگی) اعمال نمودیم.

مطابق جدول ۴، تمام آزمایش‌هایی که مبتنی بر این سیاست صورت پذیرفته‌اند، در نهایت به کمترین تعداد ویژگی یعنی ۷ ویژگی همگرا شده‌اند. در طبقه‌بند بیز، میانگین ویژگی‌های به کار گرفته شده در فرایند انتخاب ویژگی پویا نیز دقیقاً معادل ۷ است. دلیل این مسئله آن است که راهکار پیشنهادی مبتنی بر سیاست min-FNo به محض آن که به مدلی با حداقل تعداد ویژگی‌ها دست یابد، همواره آن مدل را به عنوان مدل توصیه‌شده انتخاب خواهد کرد و هرگز به توصیه سایر مدل‌ها که ممکن است با دارابودن تعداد ویژگی‌های بیشتر دارای دقت مطلوب‌تری باشند، نخواهد پرداخت.

حال اگر این مدل، مبتنی بر کروموزوم‌های تصادفی نسل نخست الگوریتم ژنتیک با هفت ویژگی منتخب ایجاد گردد، عملاً قابلیت انتخاب ویژگی پویا از بین رفته و عملیات تشخیص ناهنجاری، تمام جریان داده‌ها را مبتنی بر همان هفت ویژگی و یا زیرمجموعه هفت‌عنصری دیگری از ویژگی‌ها پردازش خواهد نمود. در نتیجه استفاده از این سیاست به تنهایی به هیچ وجه معیار مناسبی برای انتخاب و توصیه مدل نخواهد بود و در برخی موارد (نظیر طبقه‌بند بیز ساده گاوسی)، اثرات فاحشی بر دقت تشخیص بات‌نت خواهد داشت. در نهایت سیاست min-age نسبت به سایر سیاست‌ها عملکرد میانه‌روتری داشته و با تعداد ویژگی‌هایی که همواره بیشتر از سیاست min-FNo است، دقت نسبتاً مطلوبی (البته پایین‌تر از دقت سیاست max-acc) در تشخیص بات‌نت از خود نشان می‌دهد.

همان گونه که ذکر شد، در این آزمایش‌ها صرفاً از سیاست‌های تک‌متغیره به عنوان سیاست توصیه مدل استفاده شده است. دلیل استفاده از رویکرد تک‌متغیره آن است که این رویکرد، امکان تحلیل مناسب‌تری

در خصوص رفتار و نحوه تأثیرگذاری هر یک از پارامترها در راهکار پیشنهادی فراهم می‌نماید. اما روشن است که بهره‌گیری از سیاست‌های ترکیبی و چندمتغیره، در بسیاری اوقات می‌تواند نتایج مناسب‌تری را به همراه داشته باشد. به عنوان نمونه‌ای از این سیاست‌های ترکیبی می‌توان به الف) سیاست توصیه مدلی با بیشترین دقت و کمترین عمق، ب) سیاست توصیه مدلی با بیشترین دقت و کمترین تعداد ویژگی و ج) سیاست توصیه جدیدترین مدل‌ها با بیشترین دقت و کمترین تعداد ویژگی و مواردی از این دست که همگی مبتنی بر توابع بهینه‌سازی چندهدفه و قابل مدل‌سازی می‌باشند، اشاره نمود.

به عنوان نمونه‌ای از آزمایش‌های صورت‌گرفته در این زمینه می‌توان به مقایسه نتایج حالت DTH۱۱ در هنگام اعمال سیاست‌های max-acc و سیاست ترکیبی ج پرداخت. مطابق آزمایش‌های صورت‌گرفته، گرچه دقت تشخیص راهکار پیشنهادی با اعمال سیاست max-acc از سیاست ترکیبی بیشتر است (دقت ۰/۹۹ در مقابل ۰/۹۸۸)، اما سیاست ترکیبی دارای تعداد ویژگی‌های کمتر (۱۲ ویژگی در مقابل ۱۵ ویژگی در انتهای آزمایش و ۹/۵ ویژگی در مقابل ۷۲/۲ ویژگی به صورت میانگین در طول آزمایش) و نیز عمق درخت کمتر (عمق ۸ و عمق مؤثر ۵ در مقابل عمق ۱۸ و عمق مؤثر ۸/۴) نسبت به سیاست max-acc بوده و در نتیجه از پیچیدگی زمانی کمتری نیز (۱۱۵/۶ میکروثانیه در مقابل ۱۴۰/۵ میکروثانیه) برخوردار است. به عبارت دیگر، سیاست ترکیبی نه فقط بر بهبود دقت که بر بهبود هم‌زمان کلیه پارامترها تمرکز می‌نماید.

۵- نتیجه‌گیری

در این مقاله به ارائه روشی مبتنی بر اعمال انتخاب ویژگی پویا در ترکیب پردازش‌های دسته‌ای و جریانی با هدف تشخیص باتنت‌های حوزه اینترنت اشیا پرداختیم. معماری ارائه‌شده مشتمل بر ۳ واحد پردازش دسته‌ای، پردازش جریانی و واحد ادغام و خدمت‌رسانی می‌باشد که علاوه بر آن که شرایط انجام موازی پردازش‌های دسته‌ای و جریانی و بهره‌گیری از دقت تکنیک‌های پردازش دسته‌ای، هم‌زمان با سرعت و بلادرنگ بودن پردازش‌های جریانی را فراهم می‌آورد، از یک روش انتخاب ویژگی پویا نیز مبتنی بر الگوریتم ژنتیک بهره می‌گیرد که به طور کامل با ماهیت پردازش ترکیبی سازگار بوده و از ظرفیت‌های ذاتی این پردازش‌ها در راستای انتخاب ویژگی‌های مؤثر استفاده می‌نماید. توصیه‌گر مجموعه ویژگی، ویژگی‌های مؤثر در فرایند پردازش را در طول زمان و وابسته به جریان ورودی داده‌ها به صورت پویا تغییر داده و با مجموعه ویژگی‌های مناسب‌تر جایگزین می‌نماید.

واحد پردازش دسته‌ای به طور تکرارشونده به ساخت مدل‌های دسته‌ای دقیق و مبتنی بر مجموعه ویژگی‌های توصیه‌شده توسط واحد توصیه‌گر مجموعه ویژگی می‌پردازد و توصیه‌گر و ارزیاب مدل نیز پس از ارزیابی مدل‌های دسته‌ای ساخته‌شده، مدل مناسب را به تشخیص‌دهنده جریانی باتنت معرفی می‌نماید تا بر پایه آن و به صورت تدریجی به تکامل مدل جریانی خود بپردازد. سرانجام مؤلفه ادغام از طریق ترکیب هوشمندانه نتایج مدل‌های منتخب که مبتنی بر مجموعه ویژگی‌های متفاوتی هستند، عملیات تشخیص باتنت را به صورت بلادرنگ انجام می‌دهد.

به منظور ارزیابی راهکار پیشنهادی، از معیارهای ارزیابی دقت تشخیص ناهنجاری، مساحت زیر نمودار *ROC* پیچیدگی زمانی شامل زمان به‌روزرسانی مدل و زمان برچسب‌گذاری و نیز تعداد ویژگی‌های منتخب به کار گرفته شده توسط راهکار پیشنهادی، مبتنی بر مجموعه دادگان N-BaIoT استفاده نمودیم. همچنین با توجه به مستقل‌بودن راهکار پیشنهادی نسبت به طبقه‌بند، کلیه آزمایش‌ها را برای دو طبقه‌بند درخت تصمیم و بیز ساده گاوسی به صورت مجزا، بسته به آن که از طبقه‌بند دسته‌ای و جریانی به صورت غیر ترکیبی و مجزا استفاده می‌شود یا به صورت ترکیبی و عملیات انتخاب ویژگی و انتخاب هوشمندانه مدل فعال یا غیر فعال هستند، در ۶ حالت B۰۰، S۰۰، H۰۰، H۰۱، H۱۰ و H۱۱ گزارش نمودیم.

آزمایش‌های صورت‌گرفته بیانگر مکمل‌بودن عملیات انتخاب ویژگی و انتخاب مدل در دستیابی به دقت و سرعت بالا در عملیات تشخیص باتنت معماری پیشنهادی هستند. به بیانی دیگر، دقت تشخیص ناهنجاری در حالت‌هایی که عملیات انتخاب مدل، فعال است (حالت‌های H۱۰ و H۱۱)، بیشتر از حالت‌هایی است که این عملیات غیر فعال می‌باشد و در مقابل، پیچیدگی زمانی در حالت‌هایی که عملیات انتخاب ویژگی پویا فعال است (حالت‌های H۰۱ و H۱۱)، اغلب اوقات (در صورتی که پیچیدگی زمانی طبقه‌بند به کار گرفته شده، تابعی از تعداد ویژگی‌های به کار گرفته شده باشد که در بیشتر موارد صادق است) کمتر از حالاتی است که مبتنی بر تمامی ویژگی‌ها عمل می‌کنند.

همچنین در ادامه آزمایش‌ها، تأثیر اتخاذ سیاست‌های توصیه مدل متفاوت در کارایی راهکار پیشنهادی از طریق ۳ سیاست شامل: ۱) توصیه مدلی با بیشترین دقت، ۲) توصیه مدلی با کمترین تعداد ویژگی و ۳) توصیه جدیدترین مدل ایجادشده مورد بررسی و آزمایش قرار گرفت. نتایج کسب‌شده از این آزمایش‌ها گویای آن است که با توجه به تک‌متغیره بودن کلیه سیاست‌های مذکور، کارایی و دقت راهکار پیشنهادی کاملاً متناسب و سازگار با سیاست توصیه مدل اتخاذشده می‌باشد. لذا با توجه به کاربرد و اولویت اهداف مد نظر می‌توان سیاست‌های متفاوتی را با هدف افزایش دقت یا کاهش تعداد ویژگی‌های مؤثر و مانند آن انتخاب نمود.

البته در این زمینه توجه به برخی نکات حایز اهمیت است. نخست آن که با توجه به تک‌بعدی بودن سیاست‌های مذکور، برخی از آنها نظیر سیاست توصیه مدلی با کمترین تعداد ویژگی، به هیچ وجه معیار مناسبی برای انتخاب و توصیه مدل نخواهند بود و به دلیل نادیده‌گرفتن پارامتر دقت تشخیص، در برخی حالات اثرات نامطلوبی بر دقت تشخیص ناهنجاری خواهند داشت. نکته دوم آن که در صورت سازماندهی پارامترهای تک‌متغیره، در قالب یک تابع بهینه‌سازی چندهدفه، آسیب ذکرشده در نکته نخست، قابل رفع بوده و عملاً می‌توان به سیاست‌های جامع‌تری دست یافت که با اهداف و کاربردهای دنیای واقعی سازگاری بیشتری دارد. البته طبیعتاً یافتن پارامترهای مناسب و نیز رویکرد مناسب به منظور ترکیب این پارامترها در قالب توابع هدف بهینه‌سازی، نیازمند انجام بررسی و تحلیل‌های متفاوتی است که می‌توانند در محدوده کارهای آتی این پژوهش قرار گیرند.

از این رو مطابق آزمایش‌های صورت‌گرفته می‌توان گفت که معماری پیشنهادی با تجمیع مناسب و سازگار عملیات انتخاب ویژگی و انتخاب هوشمندانه مدل، برای پردازش‌های ترکیبی دسته‌ای و جریانی، در مجموع از دقت و سرعت مناسب‌تری نسبت به سایر حالات برخوردار است و می‌توان آن را در حذف ویژگی‌های تکراری و نامرتبط پردازش‌های ترکیبی و در عین حال حفظ یا افزایش دقت این پردازش‌ها مؤثر دانست. همچنین این راهکار امکانی را برای کاربران فراهم می‌سازد که با توجه به کاربرد و اولویت اهداف مد نظر می‌توان سیاست‌های متفاوتی را با هدف افزایش دقت یا کاهش تعداد ویژگی‌های مؤثر و مانند آن انتخاب نمود.

همچنین در انتها ذکر چهار نکته حایز اهمیت است:

۱) نخست آن که راهکار پیشنهادی علاوه بر تشخیص باتنت‌های

اینترنت اشیا می‌تواند برای بهبود عملکرد سایر تشخیص‌دهنده‌های ناهنجاری مانند تشخیص بات‌نت‌های غیر اینترنت اشیا، تشخیص بدافزار، تشخیص تراکنش‌های ناهنجار و مانند آن در صورتی که از یک سو نیازمند کشف الگوی رفتارهای ناهنجار مبتنی بر حجم وسیع داده‌های پیشین باشد و از سوی دیگر نیاز به عملیات بلادرنگ داشته باشد، به کار گرفته شود.

۲) نکته دوم آن که الگوریتم‌های به کار گرفته شده در این معماری از جمله الگوریتم ژنتیک، تنها الگوریتم‌هایی نیستند که قابلیت به کارگیری در این معماری را دارا هستند، بلکه هر روش انتخاب ویژگی دیگری، در صورتی که با ویژگی‌های پردازش‌های ترکیبی سازگار باشد و از ظرفیت‌های ذاتی این پردازش‌ها در راستای انتخاب ویژگی استفاده نماید، می‌تواند کاندیدای مناسبی برای توسعه‌های آتی راهکار پیشنهادی محسوب گردد.

۳) نکته سوم آن که ساخت استخرهای متفاوتی که در این مقاله از آنها نام برده شد، در اینجا مبتنی بر آرایه‌ای از چندتایی‌های مرتب شبیه‌سازی شده است. اما این شبیه‌سازی در هنگام عملیاتی‌شدن این معماری مبتنی بر زیرساخت‌های پردازشی کلان‌داده باید مورد تجدید نظر قرار گیرد و شخصی‌سازی‌های لازم متناسب با آن صورت پذیرد. تحلیل دقیق‌تر شخصی‌سازی‌های مذکور می‌تواند از طریق آزمایش‌هایی در همین راستا و مبتنی بر پلتفرم‌های تحلیل کلان‌داده در زمره کارهای آتی این مقاله قرار گیرد.

۴) نهایتاً نکته چهارم آن که روش پیشنهادی به هیچ وجه متناسب با ویژگی‌های مجموعه دادگان N-BaIoT شخصی‌سازی نشده و کوشش گردیده تا کلیه عملیات مربوط به آموزش مدل، آماده‌سازی استخرها، انتخاب پارامترها و مواردی از این دست، به طور کاملاً مستقل و کلی و بدون وابستگی به نوع مجموعه دادگان انجام پذیرد. با این حال ارزیابی عملکرد راهکار پیشنهادی در مقابل سایر مجموعه دادگان موجود در حوزه‌های تشخیص بات‌نت و یا سایر حوزه‌های مربوط به تشخیص ناهنجاری در پژوهش‌های آتی می‌تواند منجر به شناخت وجوه بیشتری از قابلیت‌ها یا کاستی‌های معماری پیشنهادی شود که با رفع آنها به راهکار جامع‌تری می‌رسیم.

مراجع

[1] M. Antonakakis, *et al.*, "Understanding the mirai botnet," in *Proc. 26th USENIX Security Symp.*, pp. 1093-1110, Vancouver, Canada, 16-18 Aug. 2017.

[2] A. Marzano, *et al.*, "The evolution of bashlite and mirai IoT botnets," in *Proc. IEEE Symp. on Computers and Communications, ISCC'18*, pp. 813-818, Natal, Brazil,25-28 Jun. 2018.

[3] S. García, A. Zunino, and M. Campo, "Survey on network-based botnet detection methods," *Security and Communication Networks*, vol. 7, no. 5, pp. 878-903, May. 2014.

[4] R. Alhajri, R. Zagrouba, and F. Al-Haidari, "Survey for anomaly detection of IoT botnets using machine learning auto-encoders," *Int. J. Appl. Eng. Res*, vol. 14, no. 10, pp. 2417-2421, Jul. 2019.

[5] R. Azmi and B. Pishgoo, "STLR: a novel danger theory based structural TLR algorithm," *The ISC International J. of Information Security*, vol. 5, no. 2, pp. 209-225, Mar. 2014.

[6] R. Azmi and B. Pishgoo, "SHADuDT: secure hypervisor-based anomaly detection using danger theory," *Computers & Security*, vol. 39, no. 1, pp. 268-288, Nov. 2013.

[7] L. Yin, L. Qin, Z. Jiang, and X. Xu, "A fast parallel attribute reduction algorithm using Apache Spark," *Knowledge-Based Systems*, vol. 212, Article ID: 106582, Jan. 2021.

[8] Y. Wu and J. Tang, "Research progress of attribute reduction based on rough set in context of big data," *Computer Engineering and Applications*, vol. 55, no. 6, pp. 31-38, May 2019.

[9] I. A. Gheyas and L. S. Smith, "Feature subset selection in large dimensionality domains," *Pattern Recognition*, vol. 43, no. 1, pp. 5-13, Jan. 2010.

[10] D. B. Skillicorn and S. M. McConnell, "Distributed prediction from vertically partitioned data," *J. of Parallel and Distributed Computing*, vol. 68, no. 1, pp. 16-36, Jan. 2008.

[11] M. Riahi-Madvar, A. A. Azirani, B. Nasersharif, and B. Raahemi, "A new density-based subspace selection method using mutual information for high dimensional outlier detection," *Knowledge-Based Systems*, vol. 216, Article ID: 106733, Mar. 2021.

[12] M. Banerjee and S. Chakravarty, "Privacy preserving feature selection for distributed data using virtual dimension," in *Proc. of the 20th ACM Int. Conf. on Information and Knowledge Management*, pp. 2281-2284, Glasgow, Scotland, UK, 24-28 Oct. 2011.

[13] M. Bramer, Principles of Data Mining, vol. 180, pp. 231-238, *London: Springer*, 2007.

[14] J. Qian, P. Lv, X. Yue, C. Liu, and Z. Jing, "Hierarchical attribute reduction algorithms for big data using MapReduce," *Knowledge-Based Systems*, vol. 73, no. 1, pp. 18-31, Jan. 2015.

[15] H. Chen, T. Li, Y. Cai, C. Luo, and H. Fujita, "Parallel attribute reduction in dominance-based neighborhood rough set," *Information Sciences*, vol. 373, pp. 351-368, Dec. 2016.

[16] W. Ding, J. Wang, and J. Wang, "Multigranulation consensus fuzzy-rough based attribute reduction," *Knowledge-Based Systems*, vol. 198, Article IDL. 105945, Jun. 2020.

[17] H. Kalkan and B. Çetisli, "Online feature selection and classification," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, ICASSP'11*, pp. 2124-2127, Prague, Czech Republic, 22-27 May 2011.

[18] D. Levi and S. Ullman, "Learning to classify by ongoing feature selection," *Image Vision Comput*, vol. 28, no. 4, pp. 715-723, Jun. 2010.

[19] N. AlNuaimi, M. M. Masud, M. A. Serhani, and N. Zaki, "Streaming feature selection algorithms for big data: a survey," *Applied Computing and Informatics*, vol. 18, no. 1-2, pp. 113-135, Jul. 2020.

[20] N. Parveen and M. Ananthi, "Data processing for large database using feature selection," in *Proc. 2nd Int. Conf. on Computing and Communications Technologies, ICCCT'17*, pp. 321-326, Chennai, India, 23-24 Feb. 2017.

[21] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: a review," *Applied Sciences*, vol. 8, no. 9, Article ID:. 1521, Sept 2018.

[22] N. Abd-Alsabour, "A review on evolutionary feature selection," in *Proc. European Modelling Symp.*, pp. 20-26, Pisa, Italy, 21-23 Oct. 2014.

[23] N. Heidari, R. Azmi, and B. Pishgoo, "Fabric textile defect detection, by selecting a suitable subset of wavelet coefficients, through genetic algorithm," *International J. of Image Processing*, vol. 5, no. 1, pp. 25-35, Jan. 2011.

[24] R. Azmi, B. Pishgoo, N. Norozi, M. Koohzadi, and F. Baesi, "A hybrid GA and SA algorithms for feature selection in recognition of hand-printed Farsi characters," in *Proc. IEEE Int. Conf. on Intelligent Computing and Intelligent Systems*, vol. 3, pp. 384-387, Xiamen, China ,29-31Oct. 2010.

[25] Y. Xing, H. Shu, H. Zhao, D. Li, and L. Guo, "Survey on botnet detection techniques: classification, methods, and evaluation," *Mathematical Problems in Engineering*, vol. 2021, no. 1, pp. 1-24, Jan. 2021.

[26] S. Almutairi, S. Mahfoudh, S. Almutairi, and J. S. Alowibdi, "Hybrid botnet detection based on host and network analysis," *J. of Computer Networks and Communications*, vol. 2020, no. 1, pp. 1-17, Jan. 2020.

[27] A. Karim, R. B. Salleh, M. Shiraz, *et al.*, "Botnet detection techniques: review, future trends, and issues," *J. of Zhejiang University-Science C*, vol. 15, no. 11, pp. 943-983, Nov. 2014.

[28] K. Sinha, V. Arun, and B. Julian, "Tracking temporal evolution of network activity for botnet detection," https:// arxiv.org/abs/1908.03443, 2013.

[29] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley, "Detecting botnets with tight command and control," in *Proc. 31st IEEE Conf. on Local Computer Networks*, pp. 195-202, Tampa, FL, USA, 14-16 Nov. 2006.

[30] E. Passerini, R. Paleari, L. Martignoni, and D. Bruschi, "Fluxor: detecting and monitoring fast-flux service networks," in *Proc. Int. Conf. on Detection of Intrusions and Malware and Vulnerability Assessment*, pp. 186-206, Paris, France, 10-11 Jul. 2008.

[31] T. F. Yen and M. K. Reiter, "Traffic aggregation for malware detection," in *Proc. Int. Conf. on Detection of Intrusions and Malware, and Vulnerability Assessment*, pp. 207-227, Paris, France, 10-11 Jul. 2008..

[54] M. Di Capua, E. Di Nardo, and A. Petrosino, "An architecture for sentiment analysis in twitter," in *Proc. of Int. Conf. on E-learning, Germany*, 10 pp., Berlin, Germany, 11-12 Sept. 2015.

[55] V. Nair, "Aligning Machine Learning for the Lambda Architecture," 2015.

[56] -, *Madrid. Lambdoop*. Retrieved from www.lambdoop.com, 2014.

[57] -, MemSQL. The Lambda Architecture Simplified, 2016.

[58] V. Astakhov and M. Chayel, Lambda Architecture for Batch and Real-Time Processing on AWS with Spark Streaming and Spark SQL, Amazon Web Services, p. 12, 2015.

[59] S. P. T. Krishnan and J. L. U. Gonzalez, Building Your Next Big Thing with Google Cloud Platform: A Guide for Developers and Enterprise Architects, Apress, 2015.

[60] W. Fan and A. Bifet, "Mining big data: current status, and forecast to the future," *ACM SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 1-5, Apr. 2013.

[61] S. Landset, T. M. Khoshgoftaar, A. N. Richter, and T. Hasanin, "A survey of open source tools for machine learning with big data in the Hadoop ecosystem," *J. of Big Data*, vol. 2, no. 1, pp. 1-36, Dec. 2015.

[62] A. Bifet, "Mining big data in real time," *Informatica*, vol. 37, no. 1, pp. 15-20, Jan. 2013.

[63] A. Mahesh and P. Manimegalai, "An efficient data processing architecture for smart environments using large scale machine learning," *IIOAB J., Special Issue, Emerging Technologies in Networking and Security*, vol. 7, no. 9, pp. 795-803, Aug. 2016.

[64] C. H. Kumar and A. S. Sangari, "An efficient distributed data processing method for smart environment," *Indian J. Sci. Technol.*, vol. 9, no. 31, pp. 380-384, Aug. 2016.

[65] X. Liu and P. S. Nielsen, "Scalable prediction-based online anomaly detection for smart meter data," *Information Systems*, vol. 77, no. 3, pp. 34-47, Sept. 2018.

[66] G. Iuhasz, D. Pop, and I. Dragan, "Architecture of a scalable platform for monitoring multiple big data frameworks," *Scalable Computing: Practice and Experience*, vol. 17, no. 4, pp. 313-321, Oct. 2016.

[67] M. Kiran, *et al*., "Lambda architecture for cost-effective batch and speed big data processing," in *Proc. IEEE Int Conf. on Big Data (Big Data)*, pp. 2785-2792, Santa Clara, CA, USA, 29 Oct.-1 Nov. 2015.

[68] -, *Oryx 1*, Retrieved from https://github.com/certxg/oryx-1, 2013.

[69] -,*Oryx2*, Retrieved from http://oryx.io/, 2014.

[70] R. C. Fernandez, *et al*., "Liquid: unifying nearline and offline big data integration," in *Proc. 7th Biennial Conf. on Innovative Data Systems Research, CIDR'15*, 8 pp., Asilomar, CA, USA, 4-7 Jan. 2105.

[71] D. Namiot, "On big data stream processing," *International J. of Open Information Technologies*, vol. 3, no. 8, pp. 48-51, aUG. 2015.

[72] L. Magnoni, *et al*., "Monitoring WLCG with lambda-architecture: a new scalable data store and analytics platform for monitoring at petabyte scale," *J. of Physics: Conf. Series*, vol. 664, no. 5, Article ID:. 052023, Dec. 2015.

[73] F. Yang, et al., The RADStack: Open Source Lambda Architecture for Interactive Analytics, 2017.

[74] B. Pishgoo, A. A. Azirani, and B. Raahemi, "A hybrid distributed batch-stream processing approach for anomaly detection," *Information Sciences*, vol. 543, pp. 309-327, Jan. 2021.

[75] J. Cai, J. Luo, S. Wang, and S. Yang, "Feature selection in machine learning: a new perspective," *Neurocomputing*, vol. 300, no. 7, pp. 70-79, Jul. 2018.

[76] N. Y. Almusallam, Z. Tari, P. Bertok, and A. Y. Zomaya, "Dimensionality reduction for intrusion detection systems in multi-data streams-a review and proposal of unsupervised feature selection scheme," *Emergent Computation*, vol. 2017, pp. 467-487, Jan. 2017.

[77] J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9-15, Mar. 2017.

[78] R. Xu, *et al*., "Dynamic feature selection algorithm based on Q-learning mechanism," *Applied Intelligence*, vol. 51, no. 10, pp. 7233-7244, Oct. 2021.

[79] V. Bolón-Canedo, N. Sánchez-Maroño, and A. Alonso-Betanzos, "Recent advances and emerging challenges of feature selection in the context of big data," *Knowledge-Based Systems*, vol. 86, no. 9, pp. 33-45, Sept. 2015.

[80] J. Li, K. Cheng, S. Wang, F. Morstatter, R. P. Trevino, J. Tang, and H. Liu, "Feature selection: a data perspective," *ACM Computing Surveys (CSUR)*, vol. 50, no. 6, pp. 1-45, Dec. 2017.

[81] C. Fahy and S. Yang, "Dynamic feature selection for clustering high dimensional data streams," *IEEE Access*, vol. 7, pp. 127128-127140, Jul. 2019.

[32] S. Kondo and N. Sato, "Botnet traffic detection techniques by C&C session classification using SVM," in *Proc. Int. Workshop on Security,* pp. 91-104, Nara, Japan, 29-31 Oct. 2007.

[33] J. François, S. Wang, and T. Engel, "BotTrack: tracking botnets using NetFlow and PageRank," in *Proc. Int. Conf. on Research in Networking*, pp. 1-14, Valencia, Spain, 9-13 May 2011.

[34] G. Gu, R. Perdisci, J. Zhang, and W. Lee, "Botminer: clustering analysis of network traffic for protocol-and structure-independent botnet detection," in *Proc. USENIX Security Symp.,* pp. 139-154, San Jose, CA, USA, 28 Jul.-1 Aug. 2008.

[35] W. Jung, H. Zhao, M. Sun, and G. Zhou, "IoT botnet detection via power consumption modeling," *Smart Health*, vol. 15, Article ID: 100103, Mar. 2020.

[36] Y. Zhang and Z. O. U. Fu-Tai, "Detection method of malicious domain name based on knowledge map," *Communications Technology*, vol. 53, no. 1, pp. 168-173, Jan. 2020.

[37] C. Yin, *Research on Network Anomaly Detection Technology Based on Deep Learning*, University of Information Engineering, Strategic Support Forces, Zhengzhou, China, 2018.

[38] R. Vinayakumar, *et al*., "A visualized botnet detection system based deep learning for the internet of things networks of smart cities," *IEEE Trans. on Industry Applications*, vol. 56, no. 4, pp. 4436-4456, Feb. 2020.

[39] S. I. Popoola, *et al*., "Federated deep learning for zero-day botnet attack detection in IoT edge devices," *IEEE Internet of Things J*., vol. 9, no. 9, pp. 3930-3944, Jul. 2021.

[40] A. Almomani, "Fast-flux hunter: a system for filtering online fast-flux botnet," *Neural Computing and Applications*, vol. 29, no. 7, pp. 483-493, Aug. 2018.

[41] M. Alauthman, N. Aslam, M. Alkasassbeh, S. Khan, A. AL-qerem, and K. K. Raymond Choo, "An efficient reinforcement learning-based botnet detection approach," *J. of Network and Computer Applications*, vol. 52, Article ID: 102479, Jan. 2019.

[42] H. T. Nguyen, Q. D. Ngo, D. H. Nguyen, *et al*., "PSI-rooted subgraph: a novel feature for iot botnet detection using classifier algorithms," *ICT Express*, vol. 6, no. 2, pp. 128-138, Jun. 2020.

[43] D. Zhuang and J. M. Chang, "PeerHunter: detecting peerto-peer botnets through community behavior analysis," in *Proc. of the IEEE Conf. on Dependable and Secure Computing*, pp. 493-500, Taipei, Taiwan, 7-10 Aug.. 2017.

[44] M. Habib, I. Aljarah, H. Faris, and S. Mirjalili, "Multiobjective particle swarm optimization for botnet detection in internet of things," *Evolutionary Machine Learning Techniques,* pp. 203-209, Berlin: Germany, Springer, 2020.

[45] A. Al Shorman, H. Faris, and I. Aljarah, "Unsupervised intelligent system based on one class support vector machine and Grey Wolf optimization for IoT botnet detection," *J. of Ambient Intelligence and Humanized Computing*, vol. 11, no. 7, pp. 2809-2825, Jul. 2020.

[46] S. Y. Huang, C. H. Mao, and H. M. Lee, "Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security*, pp. 101-111, ،Beijing China , 13-16 Apr. 2010.

[47] S. Garg, M. Guizani, S. Guo, and C. Verikoukis, "Guest editorial special section on AI-driven developments in 5G-envisioned industrial automation: big data perspective," *IEEE Trans. on Industrial Informatics*, vol. 16, no. 2, pp. 1291-1295, Nov. 2020.

[48] X. Wang, Q. Yang, and X. Jin, "Periodic communication detection algorithm of botnet based on quantum computing," *J. of Quantum Electronics*, vol. 33, no. 2, pp. 182-187, Mar. 2016.

[49] M. Albanese, S. Jajodia, and S. Venkatesan, "Defending from stealthy botnets using moving target defenses," *IEEE Security & Privacy*, vol. 16, no. 1, pp. 92-97, Feb. 2018.

[50] Z. Zha, A. Wang, Y. Guo, D. Montgomery, and S. Chen, "BotSifter: an SDN-based online bot detection framework in data centers," in *Proc. of the IEEE Conf. on Communications and Network Security, CNS'19*, pp. 142-150, Washington, D.C., USA, 10-12 Jun. 2019.

[51] G. Spathoulas, N. Giachoudis, G. P. Damiris, and G. Theodoridis, "Collaborative blockchain-based detection of distributed denial of service attacks based on internet of things botnets," *Future Internet*, vol. 11, Article ID: 226, Oct. 2019.

[52] J. Warren and N. Marz, Big Data: Principles and Best Practices of Scalable Realtime Data Systems, Simon and Schuster, 2015.

[53] B. Twardowski and D. Ryzko, "Multi-agent architecture for real-time big data processing," in *Proc. IEEE/WIC/ACM Int.Joint Conf. on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)*, vol. 3, pp. 333-337, Warsaw, Poland, 11-13 Aug. 2014.

[82] J. Jesus, A. Canuto, and D. Araújo, "Dynamic feature selection based on pareto front optimization," in *International Joint Conf. on Neural Networks, IJCNN'18*, 8 pp., Rio de Janeiro, Brazi, 8-13 Jul. 2018.
[83] R. D. O. Nunes, C. A. Dantas, A. M. Canuto, and J. C. Xavier-Júnior, "An unsupervised-based dynamic feature selection for classification tasks," in *Proc. Int. Joint Conf. on Neural Networks, IJCNN'16*, pp. 4213-4220, Vancouver, Canada, 24-29 Jul. 2016.
[84] J. P. Barddal, H. M. Gomes, F. Enembreck, and B. Pfahringer, "A survey on feature drift adaptation: definition, benchmark, challenges and future directions," *J. of Systems and Software*, vol. 127, no. 5, pp. 278-294, May 2017.
[85] G. Wei, J. Zhao, Y. Feng, A. He, and J. Yu, "A novel hybrid feature selection method based on dynamic feature importance," *Applied Soft Computing*, vol. 93, no. 8, Article ID:. 106337, Aug. 2020.
[86] S. Perkins, K. Lacker, and J. Theiler, "Grafting: fast, incremental feature selection by gradient descent in function space," *The J. of Machine Learning Research*, vol. 3, no. 3, pp. 1333-1356, Mar. 2003.
[87] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Dynamic feature space and incremental feature selection for the classification of textual data streams," in *Proc. Int. Workshop on Knowledge Discovery from Data Streams, ECML/PKDD'06*, . pp. 107-116, Berlin, Germany, 18-22 Sept. 2006.
[88] J. Zhou, D. Foster, R. Stine, and L. Ungar, "Streaming feature selection using alpha-investing," in *Proc. of the 11th ACM SIGKDD International Conf. on Knowledge Discovery in Data Mining*, pp. 384-393, Chicago, IL, USA, 21-24 Aug. 2005.
[89] X. Wu, K. Yu, H. Wang, and W. Ding, "Online streaming feature selection," in *Proc. 27th Int. Conf. on Machine Learning, ICML'10*, . pp. 1159-1166, 21-24, Jun. 2010.
[90] X. Wu, K. Yu, W. Ding, H. Wang, and X. Zhu, "Online feature selection with streaming features," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1178-1192, Sept. 2012.
[91] C. Zhang, J. Ruan, and Y. Tan, "An incremental feature subset selection algorithm based on boolean matrix in decision system," *Convergence Information Technology*, vol. 6, no. 12, pp. 16-23, Dec. 2011.
[92] M. Masud, J. Gao, L. Khan, J. Han, and B. M. Thuraisingham, "Classification and novel class detection in concept-drifting data streams under time constraints," *IEEE Trans. on Knowledge and Data Engineering*, vol. 23, no. 6, pp. 859-874, Jun. 2010.
[93] H. L. Nguyen, Y. K. Woon, W. K. Ng, and L. Wan, "Heterogeneous ensemble for feature drifts in data streams," in *Proc. Pacific-Asia Conf. on Knowledge Discovery and Data Mining*, 12 pp., Kuala Lumpur, Malaysia, 29 May-1 Jun. 2012.
[94] K. Yu, X. Wu, W. Ding, and J. Pei, "Towards scalable and accurate online feature selection for big data," in *Proc. IEEE Int. Conf. on Data Mining*, pp. 660-669, Shenzhen, China, 14-17 Dec. 2014.
[95] F. Wang, J. Liang, and Y. Qian, "Attribute reduction: a dimension incremental strategy," *Knowledge-Based Systems*, vol. 39, no. 2, pp. 95-108, Feb. 2013.
[96] S. Eskandari and M. M. Javidi, "Online streaming feature selection using rough sets," *International J. of Approximate Reasoning*, vol. 69, no. 2, pp. 35-57, Feb. 2016.
[97] M. M. Javidi and S. Eskandari, "Streamwise feature selection: a rough set method," *International J. of Machine Learning and Cybernetics*, vol. 9, no. 4, pp. 667-676, Apr. 2018.
[98] J. P. Barddal, H. M. Gomes, F. Enembreck, B. Pfahringer, and A. Bifet, "On dynamic feature weighting for feature drifting data streams," in *Proc. Joint European Conf. on Machine Learning and Knowledge Discovery in Databases*, pp. 129-144, Riva del Garda, Italy, 19-23 Sept. 2016.
[99] J. P. Barddal, F. Enembreck, H. M. Gomes, A. Bifet, and B. Pfahringer, "Merit-guided dynamic feature selection filter for data streams," *Expert Systems with Applications*, vol. 116, no. 2, pp. 227-242, Feb. 2019.
[100] J. C. Chamby-Diaz, M. Recamonde-Mendoza, and A. L. Bazzan, "Dynamic correlation-based feature selection for feature drifts in data streams," in *Proc. 8th Brazilian Conf. on Intelligent Systems, BRACIS'19*, pp. 198-203, Salvador, Brazil, 15-18 Oct. 2019.
[101] J. P. Barddal, F. Enembreck, H. M. Gomes, A. Bifet, and B. Pfahringer, "Boosting decision stumps for dynamic feature selection on data streams," *Information Systems*, vol. 83, no. 7, pp. 13-29, Jul. 2019.
[102] S. Sahmoud and H. R. Topcuoglu, "A general framework based on dynamic multi-objective evolutionary algorithms for handling feature drifts on data streams," *Future Generation Computer Systems*, vol. 102, no. 1, pp. 42-52, Jan. 2020.
[103] Y. Meidan, *et al*., "N-baiot-network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-22, Jul.-Sept. 2018.
[104] -, *N-BaIoT Dataset*, Retrieved from https://archive.ics.uci.edu/ml/datasets/detection_of_IoT_botnet_attacks_N_BaIoT#, 2018.
[105] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas, "DDoS in the IoT: mirai and other botnets," *Computer*, vol. 50, no. 7, pp. 80-84, Jul. 2017.
[106] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. the 14th int. joint Conf. on Artificial intelligence, IJCAI'95*, vol. 2, pp. 1137-1145, Montreal, Canada, 20-25 Aug. 1995.
[107] T. G. Dietterich, "Approximate statistical tests for comparing supervised classification learning algorithms," *Neural Computation*, vol. 10, no. 7, pp. 1895-1923, Oct. 1998.

بشری پیشگو تحصیلات خود را در مقطع کارشناسی مهندسی کامپیوتر در سال ۱۳۸۸ در دانشگاه الزهرا (س) و مقطع کارشناسی ارشد مهندسی کامپیوتر گرایش هوش مصنوعی را در سال ۱۳۹۰ در همین دانشگاه به پایان رسانده است. پس از آن، در سال ۱۳۹۲ به دوره دکتری مهندسی کامپیوتر گرایش هوش مصنوعی در دانشگاه علم و صنعت ایران وارد گردید و هم‌اکنون نیز در مراحل پایانی دفاع از رساله می‌باشد. ایشان از سال ۱۳۸۹ تاکنون در دانشگاه الزهرا دروس متفاوتی را تدریس نموده است. زمینه‌های علمی مورد علاقه وی یادگیری جریانی، پردازش ترکیبی، تشخیص ناهنجاری و انتخاب ویژگی می‌باشد.

احمد اکبری ازیرانی دانشیار دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران بوده و دارای ۲۵ سال سابقه تدریس و پژوهش در زمینه‌های مختلف رشته مهندسی کامپیوتر در این دانشکده می‌باشد. ایشان دبیر قطب علمی شبکه‌های ارتباطی و اطلاعاتی نسل جدید هستند و بیش از ۵۰ مقاله در مجلات معتبر بین‌المللی منتشر نموده است. دکتر اکبری مسئولیت‌های علمی و اجرایی مختلفی از جمله ریاست دانشکدهٔ مهندسی کامپیوتر دانشگاه علم و صنعت به مدت ۷ سال و عضویت در هیأت مدیره انجمن کامپیوتر ایران به مدت ۱۶ سال را در کارنامه خوددارند. زمینه‌های پژوهشی مورد علاقه ایشان پردازش داده‌ها، شبکه‌های کامپیوتری و امنیت شبکه می‌باشد.