کشف ناهنجاری با استفاده از کد کننده خودکار مبتنی بر بلوک‌های LSTM

محمود معلم[۱،*]، علی‌اکبر پویان[۲]

اطلاعات مقاله

دریافت مقاله: ۱۳۹۶/۰۸/۱۸

پذیرش مقاله: ۱۳۹۷/۰۲/۰۵

واژگان کلیدی:

کشف ناهنجاری،

کد کننده خودکار،

LSTM،

یادگیری عمیق.

چکیده

کشف ناهنجاری به معنای یافتن نمونه‌هایی است که با اکثریت هنجار و عادی داده‌ها تفاوت دارند. یکی از اساسی‌ترین چالش‌هایی که در سر راه انجام این کار مهم وجود دارد این است که نمونه‌های برچسب خورده، به‌ویژه برای کلاس ناهنجار کمیاب و گاه نایاب هستند. ما در این مقاله روشی را پیشنهاد می‌کنیم که برای کشف ناهنجاری تنها از داده‌های هنجار استفاده می‌کند. این روش بر مبنای شبکه‌های عصبی تأسیس شده که کد کننده خودکار نام دارند و در مطالعات یادگیری عمیق مورد توجه هستند. یک کد کننده خودکار ورودی خود را در خروجی بازتولید کرده و خطای بازسازی را به عنوان رتبه ناهنجاری مورد استفاده قرار می‌دهد. ما برای ساخت کد کننده، به جای نورون‌های معمولی از بلوک‌های LSTM استفاده کرده‌ایم. این بلوک‌ها در واقع نوعی از شبکه‌های عصبی بازگشتی هستند که در کشف و استخراج وابستگی‌های زمانی و مجاورتی مهارت دارند. نتیجه به‌کارگیری کد کننده خودکار مبتنی بر بلوک‌های LSTM برای کشف ناهنجاری نقطه‌ای در ده نمونه از دادگان‌های رایج نشان می‌دهد که این روش در استخراج مدل درونی داده‌های هنجار و تشخیص داده‌های ناساز موفق بوده است. معیار AUC مدل مذکور، تقریباً در تمامی موارد از AUC یک کد کننده خودکار معمولی و روش مشهور ماشین بردار پشتیبان تک کلاسه یا OC-SVM بهتر است.

۱- مقدمه

در علم داده‌کاوی، کشف ناهنجاری یا کشف داده‌های ناساز[۳] به معنای تلاش برای یافتن اقلام، رویدادها و یا مشاهداتی است که با هیچ یک از الگوهای متداول یک مجموعه داده مطابقت نکرده و با قواعد حاکم بر تولید و رفتار سایر اقلام این مجموعه داده سر ناسازگاری دارند.

یکی از نخستین تعاریف مفهوم ناهنجاری داده‌های آماری در سال ۱۹۶۹ و به وسیله گرابز ارائه شده است [۱]:

"یک مشاهده ناساز یا ناهنجاری، نمونه‌ای است که به شکل قابل ملاحظه‌ای با سایر اعضای مجموعه داده‌ای خود تفاوت داشته باشد."

البته رکورد بیشترین تعداد نقل و استناد، متعلق به تعریف هاوکینز است [۲]:

"یک ناهنجاری، مشاهده‌ای است که با سایر مشاهدات تفاوت دارد. شدت این تفاوت به حدی است که مشاهده‌گر گمان می‌کند این داده، با مکانیسمی متفاوت از داده‌های دیگر تولیدشده است."

گرچه این تعاریف قدیمی هنوز هم معتبر و قابل استفاده هستند، اما اهداف و شیوه‌های کشف ناهنجاری در خلال سال‌های اخیر دستخوش تغییرات و تحولات بزرگی شده است. در گذشته، اصلی‌ترین دلیل جستجو برای یافتن ناهنجاری‌ها، اخراج آن‌ها از مجموعه داده‌ها بود. تا الگوریتم‌های کشف الگو[۴] که غالباً نسبت به وجود اعوجاج[۵] و ناسازی حساس بودند، نتایج دقیق‌تری را تولید کنند. این فرایند، پالایش داده‌ها[۶] یا حذف اعوجاج[۷] نامیده می‌شد. با وضع و ابداع الگوریتم‌های خوش‌بنیه‌تر[۸] علاقه اصحاب

*. پست الکترونیک نویسنده مسئول: moallem@shahroodut.ac.ir

۱. دانشجوی دکتری، دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات، دانشگاه صنعتی شاهرود

۲. استادیار، دانشکده مهندسی کامپیوتر و فن‌آوری اطلاعات، دانشگاه صنعتی شاهرود

[3] Outlier
[4] Pattern recognition
[5] Noise
[6] Data Cleansing
[7] Noise Removal
[8] Robust

داده‌کاوی نسبت به ناهنجاری‌ها کاهش یافت؛ اما این وضع در حوالی سال ۲۰۰۰ میلادی تغییر کرد، این بار محققان خود ناهنجاری‌ها را نه به نیت حذف، بلکه با قصد شناسایی و کشف مورد بررسی قرار دادند. آنها دریافتند که هر ناهنجاری داده‌ای، در عالم واقع از یک رویداد قابل توجه غالباً منفی - مثل حمله به یک شبکه - و ندرتاً مثبت - مثل کشف یک رگه معدنی غنی - حکایت می‌کند. از آن زمان، شاخه جدیدی به نام کشف ناهنجاری در علم داده‌کاوی رو به گسترش نهاد. شاخه‌ای که در آن ارزش و اهمیت داده‌های ناساز از داده‌های هنجار و طبیعی بیشتر است و کاربردهایی مثل تشخیص نفوذ به شبکه، کشف جعل و تقلب در تراکنش‌های مالی، تحلیل تصاویر و مستندات پزشکی به قصد یافتن بیماری و کشف افعال غیرمعمول[1] در دایره شمول آن قرار می‌گیرند.

کشف ناهنجاری مثل طبقه‌بندی[2] به سه شیوه با ناظر، نیمه با ناظر و بدون ناظر انجام می‌پذیرد. از آنجا که طبیعت داده‌های ناهنجار همیشه برای ما آشنا نیست (مثلاً ویروس‌های جدید یا روش‌های ناشناخته تقلب) یا اگر هست، امکان تولید و تکرار آنها وجود ندارد (مثلاً خرابی شاتل فضایی) روش‌های با ناظر در کشف ناهنجاری عملاً کاربرد چندانی ندارند. از سوی دیگر، ارزیابی میزان دقت[3]، صحت[4] ، جامعیت[5] و قابلیت اعتماد روش‌های بدون ناظر موضوع چالش‌برانگیزی است که به‌ویژه در یکی دو سال گذشته، محل مطالعات و مناقشات فراوانی بوده است. در این میان روش‌های نیمه با ناظر، محسنات دو روش دیگر را در خود جمع کرده و معقول‌تر و مقبول‌تر هستند. در این روش‌ها، مدل تنها با استفاده از داده‌های طبیعی و هنجار آموزش داده می‌شود تا ویژگی‌ها و سازوکار درونی این داده‌ها استخراج گردد. خوشبختانه معمولاً تولید و جمع‌آوری داده‌های هنجار کار دشواری نیست. پس از این، هر داده جدیدی که با مندرجات مدل مذکور مطابقت نداشته باشد، به عنوان ناسازی و ناهنجاری در نظر گرفته خواهد شد.

ما در مقاله حاضر یک شبکه کد کننده خودکار یا AE[6] را برای کشف نقاط ناهنجار در مجموعه متنوعی از دادگان[7] استاندارد به کار می‌گیریم. کد کننده‌های خودکار، در واقع یکی از انواع شبکه‌های عصبی هستند که برخلاف شبکه‌های عصبی رایج به شکل بدون ناظر آموزش داده می‌شوند. این شبکه‌ها ورودی خود را در لایه اول دریافت می‌کنند، آن را از پالایه لایه یا لایه‌های میانی گذر می‌دهند و تلاش می‌کنند تا این ورودی را عیناً در خروجی خود تکرار و بازتولید نمایند. بدین ترتیب، در پایان فرایند آموزش، وزن‌های به دست آمده برای اتصالات شبکه، معرف و مبین صفات اصلی و ویژگی‌های ذاتی داده‌های هنجار خواهند بود. اکنون هر واحد داده جدیدی که وارد کد کننده آموزش دیده شده و در لایه خروجی به خوبی بازسازی گردد، هنجار است و داده‌ای که فاصله بین مقادیر اصلی آن در لایه ورودی و مقادیر بازسازی شده آن در لایه خروجی زیاد باشد به عنوان داده ناساز تلقی خواهد شد. در واقع قدر مطلق خطای بازسازی[8] به عنوان رتبه ناهنجاری[9] مورداستفاده قرار می‌گیرد.

به‌کارگیری کد کننده‌های خودکار برای کشف ناهنجاری بی‌سابقه نیست [۳ و ۴]؛ اما تفاوت کار حاضر باکارهای قبلی این است که ما برای ساخت کد کننده به جای نورون‌های عادی از بلوک‌های حافظه طولانی کوتاه‌مدت یا LSTM[10] استفاده کرده‌ایم. بلوک‌های LSTM انواع مدرنی از شبکه‌های عصبی بازگشتی هستند که در سال‌های اخیر و پس از رواج یادگیری عمیق[11] در زمینه‌های زیادی ازجمله پردازش رشته‌ها [۵]، ترجمه ماشینی خودکار [۶] و تحلیل سری‌های زمانی [۷] به کار گرفته شده‌اند. بر اساس مطالعات ما این نخستین باری است که یک کد کننده خودکار مبتنی بر LSTM به عنوان یک روش نیمه با ناظر برای کشف ناهنجاری نقطه‌ای در داده‌های جدولی با ابعاد متفاوت و گاه بسیار بالا (۴۰۰ بعد) مورد استفاده قرار گرفته است.

۲- کارهای قبلی

منابع [۸ و ۹] روش‌های کشف داده‌های ناسازگار را به دو گروه عمده الف) روش‌های آماری و ب) روش‌های مبتنی بر شبکه‌های عصبی تقسیم می‌کنند. روش‌های گروه اول که

[1] Abnormal Activity
[2] Classification
[3] Precision
[4] Accuracy
[5] Recall
[6] AutoEncoder
[7] Dataset
[8] Reconstruction Error
[9] Anomaly Score
[10] Long Short Term Memory
[11] Deep Learning

مشخصاً بر مبنای نظریه‌های آمار و احتمال بنا شده‌اند، داده‌های مجموعه آموزشی[1] را بر اساس ویژگی‌های آماری مدل‌سازی می‌کنند. سپس با عرضه هر یک از اقلام مجموعه آزمایشی[2] به این مدل، میزان هنجار یا ناهنجار بودن آن را اندازه‌گیری می‌نمایند. این روش‌ها تاکنون در حوزه‌های کاربردی از قبیل تحلیل و واکاوی داده‌های جریانی [۱۰]، کشف ناهنجاری و داده‌های ناساز در اهداف زیردریایی [۱۱]، تشخیص بیماری‌ها ازجمله سرطان [۱۲]، بررسی و کشف معایب قطعات مکانیکی [۱۳]، پردازش صوت [۱۴] و بسیاری از زمینه‌های دیگر مورد استفاده قرار گرفته‌اند.

شبکه‌های عصبی از مدت‌ها قبل در زمینه‌های متنوعی از مهندسی مثل تحلیل‌های مرتبط با راه [۱۵]، ساختمان [۱۶]، قدرت [۱۷] و زلزله [۱۸] مورد استفاده قرار گرفته‌اند. به‌کارگیری این شبکه‌ها برای تشخیص اقلام ناساز نیز از دیرباز معمول بوده است. در سال‌های دور، این شبکه‌ها به‌ویژه در زمینه‌ی امنیت شبکه و سامانه‌های تشخیص نفوذ به کار گرفته شده‌اند [۱۹]؛ اما در یکی دو دهه گذشته، به موازات رشد فزاینده علاقه و توجه محققین حوزه‌های متفاوت دانش به یادگیری عمیق، این شبکه‌ها در زمینه‌های دیگری از مقوله کشف ناسازی و ناهنجاری نیز مورداستفاده قرار گرفته‌اند. در میان انواع گوناگون شبکه‌های عصبی، کد کننده‌های خودکار یا AE که شبکه‌های مقلد[3] نیز نامیده می‌شوند، بیشتر از سایر انواع، حوزه کشف ناهنجاری را تحت تأثیر قرار داده‌اند. شدت و قدرت این تأثیر تا حدی است که منبع [۲۰] که یکی از منابع معتبر و پر استناد مرور شیوه‌های کشف ناهنجاری به شمار میاید، یک بخش مستقل و مفصل را به شرح و بحث کد کننده‌های خودکار اختصاص داده است.

روش مورد استفاده در کد کننده‌های خودکار که منبع [۲۰] آن را روش مبتنی بر خطای بازسازی نامیده، برای نخستین بار در سال ۲۰۰۲ میلادی معرفی شده است. بر اساس مرجع یاد شده، در گام اول این روش با به‌کارگیری یک شبکه عصبی حداقل سه لایه، یک مدل رگرسیون از داده‌های نرمال ایجاد می‌گردد. سپس داده‌های جدید از طریق این مدل بازآفرینی می‌شوند و خطای بازسازی، یعنی فاصله بین مقدار بازسازی شده و مقدار واقعی، به عنوان رتبه ناهنجاری در نظر گرفته می‌شود. با ارائه و انتشار روش مذکور، محققان متعددی از آن برای کشف ناهنجاری در داده‌های مختلف استفاده کرده‌اند. از جمله منبع [۲۱] با این روش وجود ناهنجاری در استفاده پردازنده از داده‌های حافظه را برآورد نموده است، منبع [۲۲] وجود صحنه‌های ناساز و نامرتبط را در یک مجموعه تصویری ارزیابی کرده است، و منابع [۲۳] و [۲۴] داده‌های توربین‌های گازی و اطلاعات زمین‌شناختی را برای یافتن نقاط ناهنجار مورد جستجو قرار داده‌اند.

علیرغم قدمت شبکه‌های کد کننده خودکار، به‌کارگیری شبکه‌های عصبی بازگشتی به عنوان واحدهای سازنده یک کد کننده سابقه چندانی ندارد. بنا به اطلاع نگارندگان ترکیب شبکه‌های بازگشتی (به‌ویژه LSTM) و کد کننده‌های خودکار، تنها از سال ۲۰۱۶ میلادی و با منبع [۲۵] آغاز شده است. لازم به ذکر است که این منبع و منابع بعدی مثل [۲۶ و ۲۷] که از LSTM-AD استفاده کرده‌اند، دو ویژگی مشترک دارند. مایه مشترک همه این منابع یافتن ناهنجاری در سری‌های زمانی است که یا تک‌بعدی هستند و یا ابعاد آنها از عدد ۳ فراتر نمی‌رود. بدین ترتیب این منابع بیشتر بر روی بخش LSTM و قدرت آن در پردازش سلسله‌ها و سری‌های زمانی متمرکز شده‌اند و از توانایی شگفت‌انگیز کد کننده‌های خودکار در کاهش غیرخطی ابعاد بهره‌ای نبرده‌اند.

با عنایت به نقطه ضعفی که در تحقیقات موجود وجود دارد، ما در مقاله حاضر توانایی LSTM-AE را بر روی مجموعه کاملاً متفاوتی از دادگان مورد بررسی قرار داده‌ایم. هیچ کدام از دادگان مورد استفاده ما که در بخش‌های بعد معرفی می‌شوند، از جنس سلسله و سری زمانی نیستند. بعلاوه ابعاد آنها محدوده متنوعی (از ۹ تا ۴۰۰ ویژگی) را در برمی‌گیرد. بنا به اطلاع نگارندگان، این اولین باری است که توانایی شبکه‌های کد کننده خودکار مبتنی بر شبکه‌های بازگشتی برای کشف ناهنجاری در داده‌های جدولی با این تنوع به کار گرفته می‌شود. خوشبختانه نتایج تجارب عملی، نشان می‌دهد که این شبکه توان بسیار بالایی را در انجام امور محوله از خود به نمایش گذاشته است.

روش ماشین بردار پشتیبان یا SVM[4] یکی از روش‌های مشهور و رایج دیگر درزمینه کشف ناهنجاری است. این روش در سال ۱۹۹۹ میلادی وارد عرصه کشف ناهنجاری

[1] Training Set
[2] Testing Set
[3] Replicator Network
[4] Support Vector Machine

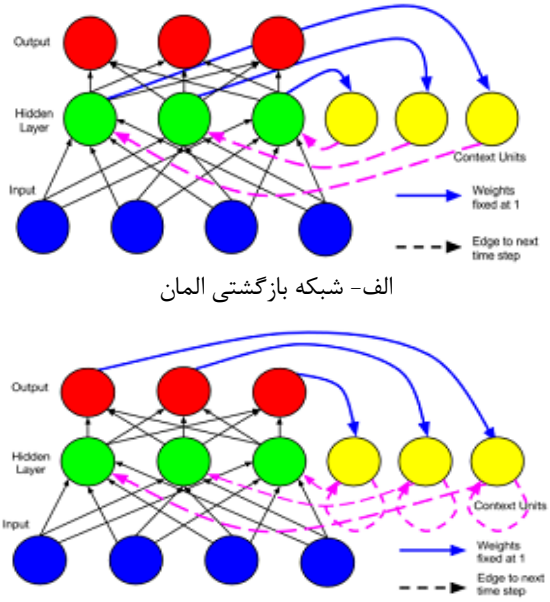
شد [۲۸] و نتایج قابل قبولی را در مقولاتی مثل کشف ناهنجاری در سری‌های زمانی[۲۹]، تحلیل لرزش موتورهای جت[۳۰]، کشف خرابی در موتورهای جت [۳۱]، نظارت بر علائم حیاتی بیماران [۳۲] پردازش تصاویر fMRI [۳۳] و کشف معایب جعبه دنده اتومبیل [۳۴] تولید کرد. از آنجا که این روش نیز غالباً به صورت نیمه با ناظر مورد استفاده قرار می‌گیرد، مقایسه نتایج آن با روش پیشنهادی ما بامعنا خواهد بود.

۳- مبانی

۳-۱- شبکه‌های عصبی بازگشتی و LSTM

شبکه‌های عصبی بازگشتی در واقع سامانه‌های پویایی هستند که وضعیت[۱] درونی خود را در خلال گام‌های زمانی یک فرایند (مثلاً طبقه‌بندی) حفظ می‌کنند. این قابلیت به دلیل ارتباطات حلقوی است که در میان نورون‌های سطوح بالا با نورون‌های لایه‌های پایین‌تر وجود دارد. در برخی از مدل‌ها، این ارتباطات در میان نورون‌های هم لایه یا حتی در ارتباط یک نورون با خودش نیز دیده می‌شود. این ارتباطات بازگشتی به شبکه اجازه می‌دهد که داده‌های مربوط به گام‌های گذشته را تا مراحل بعدی در خود حفظ کند. بدین ترتیب شبکه‌های بازگشتی درواقع از نوعی حافظه برخوردار هستند و در نتیجه مدل محاسباتی آنها از شبکه‌های عصبی پیش‌خور[۲] قدرتمندتر است.

شبکه‌های بازگشتی از انواع کمی متصل تا کاملاً متصل گسترده شده‌اند. دو نوع از مشهورترین شبکه‌های بازگشتی سنتی، به وسیله المان [۳۵] و جوردن [۳۶] معرفی گردیده‌اند. شبکه المان شبیه یک شبکه عصبی معمولی است که از سه لایه تشکیل شده و علاوه بر اتصالات رایج، خروجی لایه پنهان آن در سلول‌های خاصی که سلول‌های زمینه نام دارند ذخیره می‌شود. در هر گام از فرایند آموزش، خروجی‌های سلول‌های زمینه[۳] که از گام قبل باقی مانده‌اند به همراه ورودی‌های گام جاری مجدداً به لایه پنهان تزریق می‌گردند. شبکه‌های جوردن، ساختاری شبیه شبکه‌های المان دارند و تنها تفاوت آنها در این نکته است که محتویات سلول‌های زمینه به جای لایه پنهان، از طریق لایه خروجی تأمین می‌گردد. شکل (۱) نمونه‌هایی از این دو شبکه را نشان می‌دهد.



الف- شبکه بازگشتی المان

ب- شبکه بازگشتی جوردن

شکل ۱: شبکه‌های عصبی بازگشتی الف- المان و ب- جوردن [۳۷]

برای آموزش یک شبکه بازگشتی می‌توان از الگوریتم معمول انتشار پسگرد[۴] استفاده کرد. به این شکل که خروجی سلول‌های زمینه به مجموعه ورودی‌های لایه پنهان اضافه می‌شوند تا ساختار شبکه در بستر زمان گسترش یافته و از حالت بازگشتی خارج شود.

۳-۲- انفجار و اختفای گرادیان

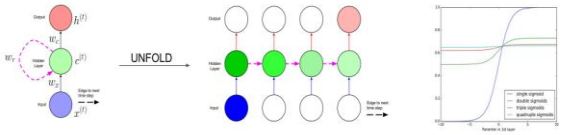
شبکه‌های بازگشتی سنتی از دو مشکل اختفا[۵] و انفجار[۶] گرادیان رنج می‌برند. می‌دانید که در هر مرحله از آموزش یک شبکه عصبی بازگشتی تعداد زیادی ضرب انجام می‌پذیرد. وجود مجموعه بزرگی از وزن‌های بزرگ‌تر از یک، و انجام ضرب‌های یاد شده، شبکه را به سمت اوزان بسیار بزرگ یا همان انفجار گرادیان هدایت می‌کند. در نتیجه این پدیده، حرکت کل شبکه به سمت نقاط مطلوب از کنترل خارج می‌شود و شبکه با گام‌های بسیار بزرگ در اطراف نقاط جواب نوسان می‌کند. شبیه پروانه‌ای که قصد نشستن بر روی یک گل را دارد، اما با هر بال زدن در مضرب بزرگی از اندازه فعلی خود ضرب می‌شود! البته مشکل انفجار گرادیان معمولاً از طریق محدود کردن وزن‌های شبکه قابل حل است.

اما مشکل اختفای گرادیان که مثل انفجار از فراوانی تعداد

[1] State
[2] Feed Forward Network
[3] Context Cell

[4] Backpropagation
[5] Vanishing Gradient
[6] Exploding Gradient

ضرب‌ها در فرایند یادگیری یک شبکه عصبی بزرگ ناشی می‌شود، مشکل بدخیمی است که در نتیجه وجود مجموعه بزرگی از وزن‌های کوچک‌تر از یک ایجاد می‌گردد. در این حالت، گرادیان خطا به تدریج کوچک و کوچک‌تر می‌شود، درنتیجه انتشار پسگرد خطا و اصلاح وزن‌های شبکه در اندازه‌های بسیار کوچک صورت می‌پذیرد، تا جایی که نهایتاً انتشار خطا متوقف شده و شبکه عملاً حساسیت خود را نسبت به وقوع خطا از دست می‌دهد. از این نقطه به بعد، بدون اینکه شبکه به جواب یا حتی اطراف آن همگرا شده باشد، وزن‌ها دیگر هیچ تغییری نمی‌کنند. این مشکل به صورت تصویری در شکل (۲) نشان داده شده است.



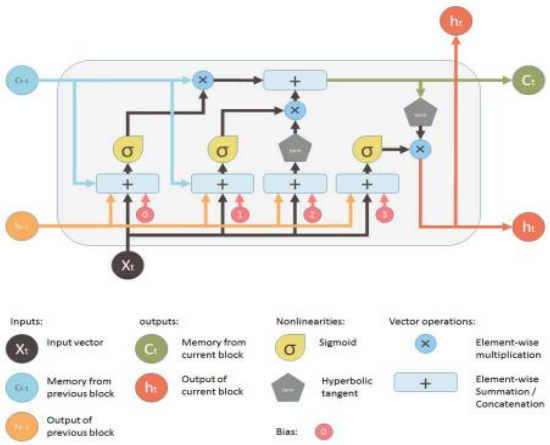
شکل ۲: مشکل اختفای گرادیان در شبکه‌های بازگشتی [۳۷]

مجموعه این مشکلات موجب می‌شد تا شبکه‌های بازگشتی قدیمی امکان پردازش و پویش وابستگی‌های دورودراز را نداشته باشند و مثل مدل‌های مارکوف به یکی دو سه‌گام قبل بسنده کنند.

در سال ۱۹۹۷ سپ هواختر و هوشیدر [۳۸] سعی کردند با ابداع مفهوم درگاه‌های ورودی و خروجی مشکل اختفای گرادیان در شبکه‌های بازگشتی را حل کنند. تلاش ایشان به ارائه نوعی از شبکه‌های بازگشتی به نام حافظه طولانی کوتاه‌مدت یا LSTM منجر گردید. این ساختار که بعدها با اضافه شدن درگاه فراموشی[1] کامل‌تر شد [۳۹] در حل بسیاری از مسائل پیچیده، ازجمله کشف ناهنجاری مورد استفاده قرار گرفته است.

۳-۳- ساختار شبکه‌های LSTM

شبکه‌های LSTM در واقع به‌جای نورون از واحدهای بزرگ‌تر و پیچیده‌تری به نام بلوک حافظه تشکیل شده‌اند. اصلی‌ترین مفهوم در مورد هر بلوک حافظه، وضعیت بلوک است که برآیند پیچیده و غیرخطی از ورودی بلوک، خروجی بلوک‌های مجاور، وضعیت بلوک در گام‌های پیشین، و وضعیت بلوک قبلی است. هر بلوک حافظه از یک سلول حافظه[2] و تعدادی درگاه تشکیل‌شده است. سلول حافظه وضعیت جاری بلوک را نگهداری می‌کند و درگاه‌ها شارش اطلاعات را به سمت داخل و خارج مدیریت می‌نمایند. شکل (۳) ساختار یک بلوک حافظه LSTM را نشان می‌دهد.



شکل ۳: ساختار یک بلوک LSTM

چنانکه از شکل برمی‌آید، هر درگاه درواقع یک لایه شبکه عصبی با تابع فعال سازی سیگمید است که خروجی آن در انتها به یک عملگر ضرب هامادار یا ضرب عنصری[3] ختم می‌شود. در معماری هوشیدر [۳۸] درگاه‌ها، ورودی جاری بلوک (x_t) و خروجی بلوک قبلی (h_{t-1}) را به عنوان ورودی دریافت می‌نمایند و با عبور آنها از پالایه یک لایه سیگمیدی، عددی را در مقام خروجی تولید می‌کنند که در فاصله ۰ تا ۱ قرار دارد و مقدار آن معرف میزان تأثیرگذاری درگاه است. درگاه ورودی تعیین می‌کند که کدام داده‌ها باید در سلول حافظه بلوک ذخیره شوند، درگاه فراموشی تصمیم می‌گیرد که این داده‌ها تا چه زمانی باید در بلوک باقی بمانند و درگاه خروجی مشخص می‌نماید که کدام داده‌ها باید از بلوک خارج بشوند. لازم به ذکر است که نویسندگان مرجع [۳۹] با ملاحظه کاستی‌های موجود در بلوک‌های LSTM زمان خود، وضعیت جاری بلوک را نیز به مجموعه ورودی‌های هر درگاه اضافه کردند و با نام‌گذاری اتصالات مذکور به اتصالات آفتابگردان[4] رایج‌ترین نوع بلوک‌های LSTM، یعنی LSTM آفتابگردانی[5] را پدید آوردند. بلوک‌هایی که ما از آنها استفاده کرده‌ایم از این نوع هستند.

۴-۳- آموزش بلوک‌های LSTM

یک بلوک LSTM را می‌توان به عنوان یک تابع F در نظر

[1] Forget Gate
[2] Memory Cell
[3] Pointwise
[4] Peephole
[5] Peephole LSTM

گرفت که با دریافت ورودی جاری (x_t) وضعیت موجود (c_{t-1}) و خروجی بلوک قبلی (h_{t-1})، خروجی بلوک جاری (h_t) را تولید می‌کند.

$$h_t = F(x_t, h_{t-1}, c_{t-1}) \quad (۱)$$

تابع F از زمان دریافت ورودی تا لحظه تولید خروجی مراحل زیر را طی می‌کند:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} * C_{t-1} + b_i) \quad (۲)$$

$$\hat{C} = \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (۳)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} * C_{t-1} + b_f) \quad (۴)$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (۵)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} * C_t + b_o) \quad (۶)$$

$$h_t = o_t \odot \tanh(C_t) \quad (۷)$$

در عبارات فوق، i_t، f_t و o_t به ترتیب معرف درگاه‌های ورودی، فراموشی، و خروجی هستند. W و b پارامترهای مدل، یعنی وزن‌های اتصالات و بایاس‌ها می‌باشند. Tanh، σ و $\odot$ نیز توابع تانژانت هیپربولیک و سیگمید و عملگر ضرب عنصری را نشان می‌دهند.

برای آشنایی بیشتر با روش کار یک سلول LSTM فرض کنید چنین سلولی در حال پردازش یک متن انگلیسی است و سعی می‌کند با تحلیل متنی که تاکنون دریافت نموده، عبارت بعدی را حدس بزند. در مثال حاضر، جنسیت فاعل جمله جاری، می‌تواند بخشی از وضعیت سلول LSTM را تشکیل بدهد. ابتدا درگاه فراموشی با دریافت توکن بعدی از مجموعه ورودی (x_t) و خروجی سلول قبلی (h_{t-1}) و با ملاحظه وضعیت جاری بلوک (c_{t-1}) تصمیم می‌گیرد که توکن ورودی تا چه حد باید در تغییر وضعیت سلول مورد اعتنا قرار بگیرد.

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} * C_{t-1} + b_f) \quad (۸)$$

به عنوان مثال دریافت یک قید یا صفت، نظر سلول را در مورد جنسیت فاعل تغییر نمی‌دهد، اما مشاهده یک ضمیر به عنوان ورودی، می‌تواند به معنای شروع یک جمله جدید باشد که جنسیت فاعل آن هیچ ربطی به جنسیت فاعل جمله جاری ندارد. درگاه فراموشی در حالت اول صفر، و در حالت دوم یک تولید می‌کند. به عبارت دیگر در حالت اول، بلوک ورودی جاری را کاملاً نادیده خواهد انگاشت، اما در حالت دوم دریافت ضمیر موجب تغییر انقلابی وضعیت بلوک شده و آن را بازنشانی[1] خواهد کرد. بدیهی است که خروجی درگاه فراموشی، یعنی خروجی تابع سیگمید تنها ۰ یا ۱ نیست، بلکه عددی حقیقی در فاصله ۰ و ۱ است که مقدار آن میزان ارتباط ورودی جاری و وضعیت فعلی بلوک را نشان می‌دهد.

در مرحله بعد، بلوک LSTM باید تصمیم بگیرد که کدام اطلاعات را در وضعیت خود تأثیر داده یا ذخیره کند. این کار در دو گام صورت می‌پذیرد، ابتدا یک‌لایه سیگمیدی موسوم به درگاه ورودی، با دریافت همان ورودی‌های درگاه فراموشی، مشخص می‌کند که کدام مقادیر در وضعیت بلوک باید بهنگام شوند:

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} * C_{t-1} + b_i) \quad (۹)$$

سپس یک‌لایه تانژانت هیپربولیک، برداری از مقادیر جدیدی را ایجاد می‌کند که می‌توانند به عنوان نامزدهای احتمالی به وضعیت جاری بلوک اضافه بشوند:

$$\hat{C} = \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (۱۰)$$

سرانجام حاصل‌ضرب عنصری خروجی درگاه فراموشی در وضعیت جاری با حاصل‌ضرب عنصری بردار نامزدهای احتمالی فوق‌الذکر در خروجی درگاه ورودی جمع شده و وضعیت جدید بلوک را ایجاد می‌کنند:

$$C_t = f_t \odot C_{t-1} + i_t \odot \hat{C}_t \quad (۱۱)$$

با عنایت به مثال پردازش متن، اینجا جایی است که اطلاعات مربوط به جنسیت فاعل قبلی از مجموعه وضعیت بلوک LSTM حذف شده و اطلاعات جدید در آن درج خواهد شد.

در آخرین مرحله، بلوک LSTM باید تصمیم بگیرد که چه مقادیری را به عنوان خروجی به بلوک‌های مجاور افقی و / یا عمودی ارسال کند. برای انتخاب این خروجی که درواقع نسخه پالایه شده‌ای از وضعیت بلوک است، درگاه خروجی، ورودی همسان ورودی‌های دو درگاه قبلی می‌گیرد و با عبور آن‌ها از یک‌لایه سیگمیدی، موارد ارسالی را تعیین می‌نماید:

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} * C_t + b_o) \quad (۱۲)$$

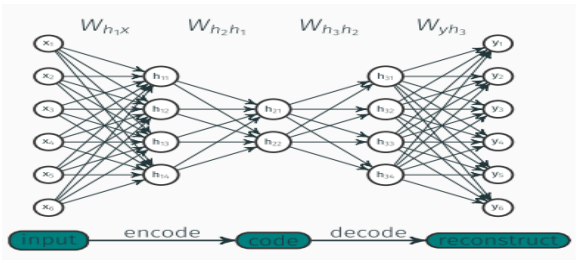
[1] Reset

در آخرین مرحله وضعیت بلوک از یک‌لایه تانژانت هیپربولیک می‌گذرد تا محتویات آن در فاصله ۱- تا ۱ قرار بگیرد. حاصل‌ضرب عنصری این گام در نتیجه درگاه خروجی، تعیین کننده چیزی است که از بلوک جاری به بلوک بعدی انتقال پیدا خواهد کرد:

$$h_t = o_t \odot \tanh(C_t) \quad (۱۳)$$

در مثال پردازش متن، چون در این مرحله یک ضمیر دیده شده، احتمالاً بلوک LSTM مایل است اطلاعاتی را در مورد فعل مربوطه – که در ادامه جمله ذکر شده – به خارج ارسال کند. به عنوان مثال این اطلاعات می‌تواند مبین جمع یا مفرد بودن ضمیر باشد تا بدین ترتیب بلوک‌های بعدی بدانند که در ادامه احتمالاً چه صیغه‌ای از فعل صرف خواهد شد.

آنچه آمد، ساختار و سازوکار یکی از رایج‌ترین انواع بلوک‌های LSTM یعنی LSTM آفتابگردانی است. بلوک‌های LSTM انواع متفاوت و متنوعی دارد که بسته به نیاز کاربردهای گوناگون و به‌وسیله محققین مختلفی پیشنهاد و پیاده‌سازی شده‌اند. LSTM دوطرفه یا BLSTM[1] و واحد بازگشتی درگاهی یا GRU[2] از مهم‌ترین و مشهورترین این انواع هستند. برای آشنایی با سایر گونه‌های LSTM و مقایسه کاربردی آنها می‌توانید به منابع [۳۷ و ۴۰] مراجعه کنید.



شکل ۴: یک کد کننده خودکار عمیق

۳-۵- کد کننده خودکار یا AE (AutoEncoder)

یک کد کننده خودکار، درواقع یک شبکه عصبی سه لایه است که ورودی‌های خود را در لایه اول دریافت می‌کند و با گذر این ورودی‌ها از لایه پنهان، مجدداً آنها را در لایه خروجی خود بازسازی می‌نماید. شکل (۴) یک کد کننده خودکار را نشان می‌دهد.

آموزش یک کد کننده شامل دو مرحله کدگذاری[3] و کدگشایی[4] است. در مرحله اول کد کننده خودکار ورودی خود را در قالب یک بردار d بعدی دریافت می‌کند (x^d) و با عبور دادن این بردار از پالایه لایه پنهان آن را به یک فضای d' بعدی نگاشت می‌نماید ($y^{d'}$):

$$y = s(W * x + b) \quad x \in R^d, y \in R^{d'} \quad (۱۴)$$

تابع s می‌تواند یک تابع خطی مثل ReLU یا یک تابع غیرخطی مثل سیگمید باشد.

در مرحله بعد، کد اخیر با تحویل به لایه خروجی به x تبدیل می‌شود که در واقع بازسازی و بازآفرینی مجدد ورودی x است. این مرحله کدگشایی نام دارد.

$$\hat{x} = s(W' * y + b') \quad y \in R^{d'}, \hat{x} \in R^d \quad (۱۵)$$

آموزش کد کننده خودکار یعنی تعیین مقادیر بردارهای W و W' متغیرهای b و b' که معرف وزن‌ها و بایاس‌های شبکه هستند و با کمینه‌سازی خطای بازسازی (فاصله بردارهای x و $\hat{x}$) محقق می‌گردد. معیار اندازه‌گیری خطای یاد شده به توزیع احتمالاتی داده‌های ورودی بستگی دارد. به عنوان مثال در حالی که این داده‌ها از یک توزیع گاوسی پیروی می‌کنند، بهتر است از تابع مرسوم مربع خطا استفاده شود:

$$L(x, y) = \|x - y\|^2 \quad (۱۶)$$

اما در صورتی که داده‌های ورودی بردارهای بیتی یا بردارهایی از احتمالات وقوع بیت‌ها باشند، تابع آنتروپی متقابل که به فرم زیر تعریف می‌شود نتایج دقیق‌تری را تولید می‌کند:

$$L(x, y) = \sum_{k=1}^{d} [x_k] * \log y_k + (1 - x_k) * \log[(1 - y_k)] \quad (۱۷)$$

کد کننده خودکار با PCA ارتباط نزدیکی دارد. در واقع اگر تنها یک‌لایه پنهان متشکل از k نورون با تابع فعال‌سازی خطی وجود داشته باشد، و برای محاسبه خطا از روش کمینه‌سازی مجموع مربعات یا MSE استفاده کنیم، آنگاه نورون‌های لایه پنهان k مؤلفه اصلی[5] داده‌ها را تولید خواهند کرد. در صورت به‌کارگیری نگاشت‌های غیرخطی مثل تانژانت هیپربولیک (tanh) یا تابع سیگمید، کد کننده خودکار وجوه پیچیده‌تری از داده‌های ورودی را استخراج

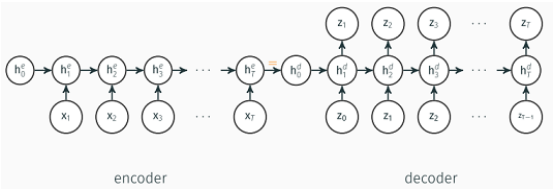
[1] Bidirectional LSTM
[2] Gated Recurrent Unit
[3] Encode
[4] Decode
[5] Cross Entropy

خواهد کرد.

معمولاً برای اینکه کد کننده به دام پیروی از نگاشت همانی در نیفتد، تعداد نورون‌های لایه پنهان باید کمتر از ابعاد داده‌های ورودی باشد. بدین ترتیب کد کننده را می‌توان به عنوان یک فشرده‌ساز دارای خطا در نظر گرفت که هیچ‌وقت ورودی خود را به طور کامل و مطلق بازسازی نمی‌کند. وجود همین خطای بازسازی، کد کننده خودکار را به ابزار مناسبی برای کشف داده‌های ناساز تبدیل کرده است. کد کننده خودکار را می‌توان به عنوان یک روش بدون ناظر یا نیمه با ناظر برای کشف ناهنجاری مورد استفاده قرار داد. برای انجام این کار ابتدا مجموعه آموزشی را در اختیار کد کننده خودکار می‌گذاریم تا با گذر آن از لایه پنهان، ویژگی‌های اساسی و ذاتی داده‌های این مجموعه را استخراج کند. سپس کد کننده خودکار آموزش‌دیده را با داده‌های جدید مجموعه آزمایشی محک می‌زنیم. اگر داده‌های هر دو مجموعه از طریق فرایندهای مشابهی تولیدشده باشند، خطای بازسازی در طول داده‌های آزمایشی زیاد نبوده و دارای نوسانات شدید نیست؛ اما درصورتی‌که داده‌های مجموعه آزمایشی حاوی نقاط ناساز و ناهنجار باشد، قاعدتاً خطای بازسازی نقطه‌های مذکور از نقاط مجاور بزرگ‌تر خواهد بود.

۳-۶- روش پیشنهادی: کد کننده خودکار LSTM

ما با ترکیب ایده‌های AE و LSTM کد کننده خودکاری را می‌سازیم که هر یک از بخش‌های کدگذار و کدگشای آن‌یک لایه LSTM است. شمای کلی این ساختار در شکل (۵) ترسیم شده است.



شکل ۵: یک کد کننده خودکار مبتنی بر بلوک‌های LSTM

چنانکه گفته شد هر بلوک حافظه LSTM محتوی چندلایه سیگمیدی است، به همین دلیل کد کننده خودکاری که با استفاده از این بلوک‌ها ساخته می‌شود، در ساده‌ترین حالت معادل یک کد کننده خودکار عمیق است، یعنی کد کننده‌ای که به جای یک‌لایه پنهان از چندلایه پنهان برای کدگذاری و کدگشایی بهره می‌برد. بعلاوه وجود درگاه‌های ورودی، فراموشی، و خروجی در بلوک‌های LSTM موجب می‌گردد که بلوک‌های یاد شده بتوانند وابستگی‌های زمانی و مکانی را به‌خوبی لحاظ نمایند. با توجه به این موارد انتظار می‌رود که مدل پیشنهادی، در استخراج ساختار ذاتی داده‌های هنجار، و در نتیجه در کشف ناسازی و ناهنجاری از یک کد کننده خودکار عادی و حتی عمیق قوی‌تر عمل کند. نتایج تجربیاتی که در بخش ۴-۴ مقاله حاضر ذکر شده، ثابت می‌کند که این انتظار بیجا نیست.

۴- شرح تجربیات عملی

۴-۱- معیار کار آیی

مقوله کشف ناهنجاری، علیرغم شباهت‌های کلی که با طبقه‌بندی دارد، دارای ویژگی‌های منحصر به فردی است که آن را به یک حوزه مستقل تبدیل می‌کند. درست است که در کشف ناهنجاری داده‌ها به دو طبقه یا کلاس هنجار و ناهنجار تقسیم می‌شوند، اما داده‌های ناهنجار معمولاً نادر و کم تعداد هستند و به همین دلیل اندازه طبقات یاد شده، متوازن نیست. این موضوع باعث می‌شود تا معیارهای رایج در ارزیابی روش‌های طبقه‌بندی، مثل دقت، صحت، و جامعیت کار آیی خود را از دست بدهند. به عنوان مثال دادگانی را در نظر بگیرید که حاوی ۱۰% ناهنجاری باشد. به‌کارگیری یک طبقه بند مبتذل[1] که همیشه همه چیز را به عنوان هنجار طبقه‌بندی می‌کند، بر روی این دادگان با ۹۰% دقت همراه خواهد بود!

مشکل دیگری که روش‌های مبتنی بر خطای بازسازی - مثل AE یا PCA - با آن روبرو هستند، تعیین یک حد آستانه مناسب برای تفکیک نقاط هنجار و ناهنجار است. در صورتی که این عدد را به شکل سخت‌گیرانه‌ای بزرگ در نظر بگیریم، دقت روش افزایش می‌یابد، اما جامعیت آن کم می‌شود؛ یعنی احتمالاً بیشتر نقاط کشف شده ناهنجار هستند، اما بخش بزرگی از نقاط ناهنجار، به عنوان هنجار در نظر گرفته خواهند شد. در طرف مقابل، انتخاب یک حد آستانه پایین، باعث افزایش جامعیت و کاهش دقت می‌گردد؛ یعنی احتمالاً بیشتر نقاط ناهنجار کشف خواهند شد، اما تعدادی از نقاط هنجار نیز به عنوان ناهنجاری برچسب می‌خورند.

مجموعه مطالب مذکور نشان می‌دهد که معیارهای رایج سنجش، مثل دقت، صحت، جامعیت و حتی معیار F1، که

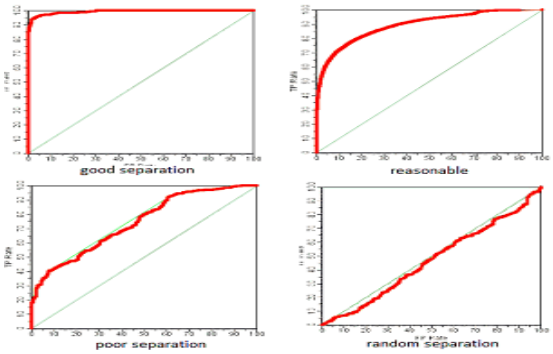
[1] Trivial

معمولاً برای تحلیل روش‌های طبقه‌بندی مورد استفاده قرار می‌گیرند، در حوزه کشف ناهنجاری از انعطاف و کارآیی لازم برخوردار نیستند. به همین دلیل اغلب تحقیقاتی که در حوزه یاد شده صورت پذیرفته است، از معیار دیگری به نام منحنی مشخصه عملکرد سیستم یا ROC استفاده کرده‌اند [۴۱ و ۴۲].

۴-۲- منحنی مشخصه عملکرد سیستم (ROC)

در روش‌های بدون ناظر یا نیمه با ناظر کشف ناهنجاری، برخلاف الگوریتم‌های طبقه‌بندی دو کلاسه، افراز دقیق و صریح داده‌ها به دو مجموعه هنجار و ناهنجار ممکن نیست. این روش‌ها معمولاً به هر نقطه فضای آزمایش یک رتبه ناهنجاری نسبت می‌دهند. این رتبه در فاصله ۰ تا ۱ قرار دارد و مقدار آن مبین میزان اطمینان الگوریتم به ناهنجار بودن نقطه مربوطه است.

برای ترسیم منحنی ROC، نقاطی را در فاصله کوچک‌ترین تا بزرگ‌ترین رتبه (معمولاً ۰ تا ۱) انتخاب می‌کنند و پس از انتصاب هر نقطه به عنوان حد آستانه، نسبت اقلام ناهنجاری که به درستی به عنوان ناهنجاری در نظر گرفته شده‌اند یا TPR[1] و نسبت اقلام هنجاری که به غلط به عنوان ناهنجار طبقه‌بندی گردیده‌اند یا FPR[2] را محاسبه می‌کنند. سپس در یک نمودار دوبعدی که محور عمودی آن TPR و محور افقی آن FPR است، این نقاط را به هم متصل می‌نمایند تا درنهایت نمودار ROC به دست بیاید. شکل (۶) نمونه‌هایی از نمودار ROC را نشان می‌دهد.



شکل ۶: چهار نمونه از نمودارهای ROC

یکی از بزرگ‌ترین امتیازات نمودار ROC این است که نمودار یاد شده از نحوه توزیع رده‌ها مستقل است و با تغییر نسبت نمونه‌های هنجار و ناهنجار تغییر شکل نخواهد داد [۴۳] این ویژگی ROC را به ابزاری مناسب برای تحلیل الگوریتم‌های کشف ناهنجاری تبدیل کرده است.

سطح زیر منحنی نمودار ROC که AUC[3] نامیده می‌شود، مندرجات گرافیکی این نمودار را در یک عدد خلاصه می‌کند. این عدد در فاصله ۰ تا ۱ قرار دارد. مرجع [۴۱] عدد یاد شده را چنین تحلیل می‌کند:

"معنای AUC این است: چقدر احتمال دارد که یک الگوریتم کشف ناهنجاری به یک داده هنجار که تصادفاً انتخاب شده است، رتبه ناهنجاری را اختصاص بدهد که از رتبه‌ای که همین الگوریتم به یک نقطه ناهنجار تصادفی نسبت می‌دهد کوچک‌تر باشد."

با این توضیح، واضح است که مقادیر معقول AUC در فاصله ۰/۵ تا ۱ قرار دارند. در واقع AUC الگوریتمی که نقاط هنجار و ناهنجار را به شکل تصادفی و الله‌بختکی (مثلاً با پرتاب سکه) مشخص می‌کند نزدیک یا برابر ۰/۵ است، و AUC الگوریتمی که تمامی نقاط سازگار و ناساز را به طور دقیق و صریح بازشناسی می‌نماید برابر ۱ است.

نمودار ROC علیرغم محسنات خود، معایبی نیز دارد. از جمله اینکه تنها ترتیب و چیدمان رتبه‌های ناهنجاری برای آن اهمیت دارد و به اندازه این رتبه‌ها یا میزان تفاوت‌های نسبی و عددی آنها توجهی نمی‌کند. با این حال، و با وجود تلاش‌هایی که برای معرفی معیارهای کارآتر صورت پذیرفته، در سال‌های اخیر نمودار ROC و سطح زیر منحنی آن AUC به یک مقیاس استاندارد برای مقایسه الگوریتم‌های کشف ناهنجاری تبدیل‌شده است. ما با الهام از مراجع [۴۴ و ۴۵] و به ویژه مرجع [۴۲] برای اندازه‌گیری و مقایسه قابلیت و کارآیی روش پیشنهادی خود، از ROC و AUC استفاده خواهیم کرد.

۴-۳- دادگان

در یکی دو سال گذشته، روش‌های بدون ناظر و نیمه با ناظر کشف ناهنجاری مورد توجه زیادی قرار گرفته‌اند، و مقالات متعددی در زمینه نقد و تحلیل و مقایسه این روش‌ها به رشته تحریر درآمده است [۴۲، ۴۴ و ۴۵]. یکی از اهداف و جهت‌گیری‌های اصلی مطالعات یاد شده، تلاش برای تهیه، تدوین، و ارائه دادگانی است که از ویژگی‌های لازم برخوردار باشند تا محققین بتوانند روش‌های پیشنهادی خود را بر اساس آنها با یکدیگر مقایسه کنند.

مرجع [۴۲] یک مجموعه ده‌تایی از این دادگان را فراهم

[1] True Positive Rate
[2] False Positive Rate
[3] Area Under Curve

کرده و آنها را از طریق یک وب‌سایت در اختیار عموم قرار داده است. بنا به نوشته مؤلفان، این دادگان در واقع برای کاربردهای طبقه‌بندی مورد استفاده قرار می‌گرفته‌اند؛ اما با به‌کارگیری تکنیک‌های داده‌پردازی، به گونه‌ای پالایش شده‌اند که ملزومات یک کاربرد کشف ناهنجاری را ارضا کنند. به طور کلی این ملزومات شامل دو فرض زیر هستند:

- ناهنجاری‌ها و نقاط ناساز نادرند.
- این نقاط - قطعاً - در برخی از ویژگی‌ها با نقاط طبیعی و هنجار تفاوت دارند.

مشخصات این مجموعه دادگان که از نوع جدولی بوده و تنها حاوی ناهنجاری‌های نقطه‌ای هستند به شرح زیر است:

Breast-cancer: محتویات این دادگان از مجموعه تصاویر پزشکی مربوط به سرطان سینه اخذ شده است [۴۶]. نسخه اصلی که در مخزن دادگان دانشگاه کالیفرنیا در ایروین یا UCI نگهداری می‌شود شامل ۳۵۷ نمونه خوش‌خیم و ۲۱۲ نمونه بدخیم است و معمولاً برای ارزیابی الگوریتم‌های طبقه‌بندی دودویی مورد استفاده قرار می‌گیرد. برای نادر شدن نمونه‌های ناهنجار، تنها ۱۰ مورد نخستین تومورهای بدخیم حفظ گردیده و سایر موارد حذف شده‌اند [۴۷]. این دادگان شامل ۳۶۷ رکورد، ۳۰ ویژگی، و ۱۰ ناهنجاری (معادل ۲/۷۲ درصد) است.

Pen-global: نسخه اصلی این دادگان به مخزن UCI تعلق دارد و حاوی ارقام ۰ تا ۹ است که به شکل دست‌نویس و به وسیله ۴۵ فرد مختلف نوشته شده است. در نسخه global تنها دست‌نویس‌های رقم ۸ به تمامی حفظ گردیده و از هر یک از ارقام باقیمانده ۱۰ نمونه به عنوان ناهنجاری باقی مانده است. بدین ترتیب، دادگان حاضر شامل یک خوشه ۷۱۹ تایی هنجار (رقم ۸) و یک خوشه ۹۰ تایی ناهنجار (۱۰ نمونه از ۹ رقم باقی) است که به شکلی خلوت و سراسری توزیع شده‌اند. این دادگان شامل ۸۰۹ رکورد، ۱۶ ویژگی، و ۹۰ ناهنجاری (معادل ۱۱/۱ درصد) است.

Pen-local: این دادگان و دادگان قبلی از منشأ واحدی اخذشده‌اند. برخلاف pen-global برای ایجاد دادگان اخیر، همه نمونه‌های رقم ۴ به جز ۱۰ نمونه اول حذف گردیده و سایر ارقام دست‌ن خورده باقی مانده‌اند. در نتیجه نسخه local برخلاف نسخه global حاوی ناهنجاری ناچیزی است (۰/۱۵ درصد) که به شکل محلی و متمرکز توزیع شده است [۴۷]. این دادگان شامل ۶۷۲۴ رکورد، ۱۶ ویژگی، و ۱۰ ناهنجاری (معادل ۰/۱۵ درصد) است.

Letter: مرجع [۴۸] ، اصل این دادگان را از مخزن UCI برگزیده و آن را طوری پردازش کرده که برای کشف ناهنجاری بدون ناظر مناسب باشد. در این پردازش ابتدا سه تا از حروف برای تشکیل فضای هنجار انتخاب شده‌اند، فضای ناهنجار نیز با نمونه‌برداری تصادفی از میان سایر حروف ایجاد گردیده است. در مرحله بعد هر یک از نمونه‌های هنجار به طور تصادفی به یک نمونه دیگر چسبانده شده‌اند، که این نمونه ممکن است هنجار یا ناهنجار باشد. بدین ترتیب فضای ۱۶ بعدی دادگان اصلی به یک فضای ۳۲ بعدی تغییر کرده و ناهنجاری سراسری و ساده گام اول، به یک ناهنجاری ترکیبی و پیچیده تبدیل شده است که کشف آن برای الگوریتم‌های بدون ناظر و نیمه با ناظر کار چندان ساده‌ای نیست. این دادگان شامل ۱۶۰۰ رکورد، ۳۲ ویژگی، و ۱۰۰ ناهنجاری (معادل ۶/۲۵ درصد) است.

Speech: این دادگان هم به‌وسیله نویسندگان مرجع [۴۸] مهیا شده است. دادگان speech حاوی ۳۶۸۶ قطعه گفتار انگلیسی است که بیشتر آنها با لهجه آمریکایی ادا شده‌اند. این اکثریت ۳۶۲۵ تایی نمونه‌های هنجار را تشکیل می‌دهند و ۶۱ قطعه باقیمانده که با ۷ لهجه مختلف دیگر بیان شده‌اند به عنوان ناهنجاری تلقی می‌شوند. برای رقومی کردن گفتار از روش i-vector استفاده شده که از جمله روش‌های رایج و نسبتاً روزآمد پردازش صوت به شمار می‌آید. در مجموعه مورد استفاده ما، این دادگان با داشتن ۴۰۰ ویژگی بالاترین تعداد ابعاد را دارد. این دادگان شامل ۳۶۸۶ رکورد، ۴۰۰ ویژگی و ۶۱ ناهنجاری (معادل ۱/۶۵ درصد) است.

Satellite: نسخه اصلی این دادگان در مخزن UCI نگهداری می‌شود. هر رکورد این نسخه حاوی ویژگی‌هایی است که از تصاویر ماهواره‌ای یک منطقه از زمین استخراج شده و با نوع خاک آن منطقه برچسب خورده است. این ویژگی‌های ۳۶ گانه، خاک مناطق زمین را به ۶ گروه تقسیم می‌کنند. برای تطبیق این دادگان با مقوله کشف ناهنجاری، چهار کلاس رایج‌تر در هم ادغام شده و کلاس هنجار را تشکیل داده‌اند، نمونه‌های تصادفی دو کلاس دیگر نیز در کنار هم کلاس کوچک‌تر ناهنجار را ایجاد نموده‌اند. این دادگان شامل ۵۱۰۰ رکورد، ۳۶ ویژگی، و ۷۵ ناهنجاری (معادل ۱/۴۹ درصد) است.

Annthyroid: این دادگان نیز ازجمله دادگان‌هایی است

که نسخه اصلی آن از مخزن UCI اخذ شده است. مرجع [۴۹] نسخه اصلی را طوری پردازش کرده است که برای استفاده در شبکه‌های عصبی مناسب باشد. نمونه‌های سالم نسخه اخیر به عنوان فضای هنجار در نظر گرفته شده‌اند و انتخاب تصادفی نمونه‌هایی از دو کلاس تیروئید پرکار و تیروئید کم‌کار، منجر به ساخت فضای ناهنجار گردیده است. این دادگان شامل ۶۹۱۶ رکورد، ۲۱ ویژگی، و ۲۵۰ ناهنجاری (معادل ۳/۶۱ درصد) است.

Shuttle: نسخه اصلی این دادگان که در مخزن UCI نگهداری می‌شود، از طریق ۹ ویژگی عددی وضعیت رادیاتور کشتی فضایی ناسا – شاتل – را بیان می‌کند. این نسخه که در کشف ناهنجاری با ناظر کاربرد دارد دارای یک کلاس هنجار است که ۸۰% تعداد نمونه‌ها را به خود اختصاص داده است. باقیمانده رکوردها در ۶ کلاس ناهنجار توزیع شده‌اند. برای کاهش فضای ناهنجاری، کلاس ۱ به عنوان کلاس هنجار حفظ شده و از کلاس‌های ۲ و ۳ و ۴ و ۵ و ۶ و ۷ به شکل طبقه‌بندی شده نمونه‌برداری گردیده است. شبیه این کار در مراجع [۵۰ و ۵۱] نیز صورت پذیرفته است. این دادگان شامل ۴۶۴۶۴ رکورد، ۹ ویژگی، و ۸۷۸ ناهنجاری (معادل ۱/۸۹ درصد) است.

Aloi: این مجموعه از دادگان Aloi[1] [۵۲] استخراج شده است. این دادگان حاوی حدود ۱۱۰ عکس از ۱۰۰۰ شیء کوچک است که در شرایط نوری مختلف و زوایای متفاوت گرفته شده‌اند. مرجع [۵۳] این تصاویر را با استفاده از هیستوگرام‌های HSB به بردارهای ۲۷ بعدی تبدیل کرده است. سپس تصاویر برخی از اشیا به عنوان ناهنجاری در نظر گرفته شده‌اند و در نهایت تعداد کل نمونه‌ها تا رسیدن به عدد ۵۰۰۰۰ کاهش یافته است. این دادگان شامل ۵۰۰۰۰ رکورد، ۲۷ ویژگی، و ۱۵۰۸ ناهنجاری (معادل ۳/۰۲ درصد) است.

Kdd99: دادگان kdd99 در ادبیات امنیت شبکه و سامانه‌های تشخیص نفوذ شهرت فراوانی دارد. در دادگان مذکور ترافیک یک شبکه کامپیوتری در سطح لایه IP شبیه‌سازی شده است. در این ترافیک هم بسته‌های عادی و هم بسته‌های ناشی از حمله و نفوذ وجود دارند، به همین دلیل بسیاری از مقالات نسخه کامل یا مجموعه‌های کاهش یافته‌ای از آن را برای ارزیابی روش‌های کشف ناهنجاری مورد استفاده قرار داده‌اند [۵۴]؛ اما این دادگان ساختار همگنی ندارد، مثلاً حملات نوع DDos آن از جنس ناهنجاری نقطه‌ای نیستند، در حالی که انواع دیگر حملات در زمره این نوع از ناهنجاری‌ها قرار می‌گیرند. با عنایت به این مشکلات مجموعه kdd99 به شکل بنیادی مورد پالایش قرار گرفته است [۵۵]. این پردازش‌ها به شرح زیرند:

- بسته‌های همه پروتکل‌ها به غیر از پروتکل HTTP حذف شده است.
- با توجه به فراوانی بسته‌های حملات Dos، برای تبدیل ناهنجاری به یک پدیده نادر، تنها ۵۰۰ بسته از این حملات حفظ شده است.
- به دلیل اینکه فقط بسته‌های HTTP وجود دارند، فیلدهای protocol، port و flags حذف شده است.

نتیجه این تغییرات دادگانی است که ۶۲۰۰۸۹ رکورد، ۳۸ ویژگی، و ۱۰۵۲ ناهنجاری (معادل ۰/۱۷ درصد) دارد.

خلاصه اصلی‌ترین مشخصات این مجموعه در جدول ۱ فهرست شده است. توجه به این جدول نشان می‌دهد که مجموعه مورد استفاده، گستره وسیعی از کاربردها، شامل تشخیص پزشکی، امنیت شبکه و تشخیص نفوذ، پردازش صوت و تصویر، و تجزیه و تحلیل سامانه‌های پیچیده را در برمی‌گیرد. بعلاوه این مجموعه به لحاظ حجم دادگان، تعداد ابعاد و ویژگی‌ها، و درصد وجود ناهنجاری از تنوع خوبی برخوردار است. به همین دلیل به نظر می‌رسد که به‌کارگیری مجموعه مذکور برای سنجش کارآیی شبکه‌های LSTM در کشف ناهنجاری، انتخاب مناسبی باشد. همچنین، چنانکه گفته شد مجموعه دادگانی مورد استفاده، از طریق اینترنت در دسترس عموم محققان قرار دارد، بنابراین امکان تکرار تجربه حاضر یا مقایسه آن با سایر روش‌های دیگر نیز مهیا خواهد بود.

۴-۴- آزمایش‌ها و نتایج

شبکه کد کننده خودکار، از دولایه LSTM تشکیل شده که لایه اول به عنوان کدگذار و لایه دوم در مقام کدگشا عمل می‌کنند. هر یک از این دو لایه دارای ۲۸ بلوک LSTM هستند و - چنانکه گفته شد - این بلوک‌ها از نوع آفتابگردانی می‌باشند. تعداد مناسب برای بلوک‌های

[1] Amsterdam Library of Object Images

LSTM، یعنی عدد ۲۸ پس از سعی و خطای فراوان به دست آمده است. ما مجموعه متنوعی از اعداد را مورد بررسی قرار دادیم و در میان این اعداد، عدد ۲۸ به طور متوسط در مورد مجموعه ده‌گانه دادگان مورد استفاده، بهترین نتایج را تولید کرده است. بنا به اطلاع نگارندگان، انتخاب تعداد لایه‌ها و نورون‌های یک شبکه عصبی، قاعده نظری مطمئنی ندارد و معمولاً به همین روش انجام می‌پذیرد.

هر دادگان به دو بخش ۷۰ و ۳۰ درصدی تقسیم شده، و این دو بخش به ترتیب برای آموزش و آزمایش شبکه کد کننده مورد استفاده قرار گرفته‌اند. برای اینکه کد کننده بتواند با دقت ویژگی‌های درونی و ذاتی داده‌های هنجار را استخراج کند، باید با داده‌هایی آموزش ببیند که شامل هیچ ناهنجاری نباشند. به همین دلیل ما پیش از آنکه داده‌های مجموعه آموزش را در اختیار این شبکه بگذاریم، همه ناهنجاری‌های مجموعه یاد شده را حذف کرده‌ایم. البته بدیهی است که مجموعه آزمایشی حتماً باید حاوی نقاط ناهنجار باشد تا ارزیابی توان شبکه در کشف این نقاط میسر گردد. این شیوه که آن را کشف ناهنجاری به روش نیمه با ناظر می‌نامند، غالباً در مورد ابزاری از قبیل SVM، PCA و AE بسیار رایج است.

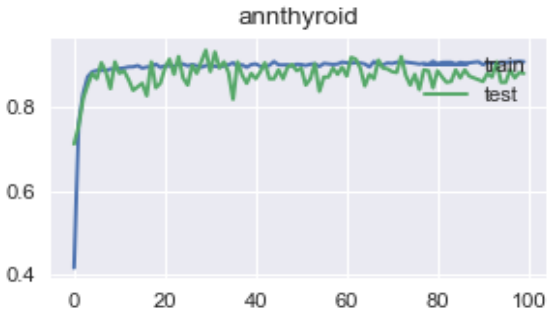
پس از حذف ناهنجاری‌های مجموعه آموزشی، این مجموعه در قالب بسته‌های ۱۰ تایی (batch size = 10) در اختیار کد کننده قرار گرفته است تا وزن‌های اتصالات خود را در جهت کمینه‌سازی خطای بازآفرینی این بسته‌ها، تصحیح و تنظیم نماید. این عمل، که اصطلاحاً آموزش شبکه عصبی نامیده می‌شود، ۱۰۰ بار تکرار شده است (Epochs = 100) تا در نهایت اوزان شبکه، به سمت مقادیر مناسب همگرا گردند. نمودار تغییرات دقت کد کننده خودکار در خلال گام‌های ۱۰۰ گانه مذکور، برای هر یک از دادگان ۱۰گانه در شکل‌های (۷) تا (۱۶) نمایش داده شده است.

جدول ۱: مشخصات مجموعه دادگان مورد استفاده

نام دادگان	تعداد رکورد	تعداد ابعاد	تعداد ناهنجاری	درصد ناهنجاری	شرح
breast-cancer	۳۶۷	۳۰	۱۰	۲/۷۲	تشخیص پزشکی
pen-global	۸۰۹	۱۶	۹۰	۱۱/۱	پردازش تصویر- تشخیص دست خط
Letter	۱۶۰۰	۳۲	۱۰۰	۶/۲۵	پردازش تصویر- تشخیص دست خط
Speech	۳۶۸۶	۴۰۰	۶۱	۱/۶۵	پردازش صوت
Satellite	۵۱۰۰	۳۶	۷۵	۱/۴۹	پردازش تصویر
pen-local	۶۷۲۴	۱۶	۱۰	۰/۱۵	پردازش تصویر
Annthyroid	۶۹۱۶	۲۱	۲۵۰	۳/۶۱	تشخیص پزشکی
Shuttle	۴۶۴۶۴	۹	۸۷۸	۱/۸۹	سامانه‌های مکانیکی
Aloi	۵۰۰۰۰	۲۷	۱۵۰۸	۳/۰۲	پردازش تصویر
Kdd99	۶۲۰۰۹۸	۳۸	۱۰۵۲	۰/۱۷	امنیت شبکه

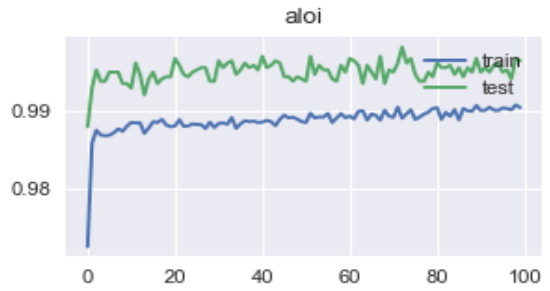
جدول ۲: مقادیر AUC روش‌های مختلف در کشف ناهنجاری دادگان ده‌گانه

	Bcancer	Pglobal	Plocal	Letter	Speech	Satellite	Thyroid	shuttle	Aloi	Kdd99
کدکننده LSTM	۰،۹۹۶۴	۰،۹۹۹۶	۰،۷۹۷۶	۰،۹۲۱۲	۰،۵۹۵۴	۰،۹۸۴۶	۰،۸۵۸۳	۰،۹۹۸۸	۰،۵۳۸۶	۰،۹۹۹۸
کد کننده عمیق	۰،۹۶۴۳	۰،۹۱۶۲	۰،۴۴۸۷	۰،۵۵۷۸	۰،۵۶۳۸	۰،۹۸۵۱	۰،۶۹۴۸	۰،۹۹۴۶	۰،۵۴۹۳	۰،۹۹۹۷
[۴۲] Oc-SVM	۰،۹۷۲۱	۰،۹۵۱۲	۰،۹۵۴۳	۰،۵۱۹۵	۰،۴۶۵	۰،۹۵۴۹	۰،۵۳۱۶	۰،۹۸۶۲	۰،۵۳۱۹	۰،۹۵۱۸
نزدیک‌ترین همسایگی [۴۲]	۰/۹۸۱۶	۰/۹۸۷۲	۰/۹۸۷۷	۰/۹۰۶۸	۰/۵۳۴۷	۰/۹۷۰۱	۰/۶۸۹۳	۰/۹۴۷۴	۰/۷۸۹۹	۰/۹۷۹۶
خوشه‌بندی [۴۲]	۰/۹۴۹۶	۰/۸۷۲۱	۰/۹۷۲۷	۰/۸۹۰۲	۰/۵۰۸۱	۰/۹۶۲۷	۰/۷۸۴۳	۰/۹۷۱۶	۰/۵۸۵۵	۰/۹۹۶۴

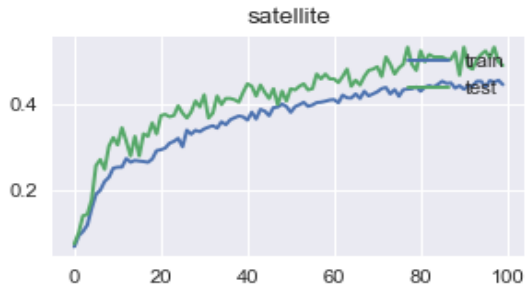


شکل ۷: تغییر دقت در دادگان annthyroid

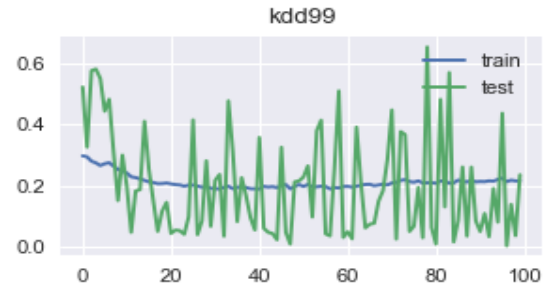
شکل ۱۲: تغییر دقت در دادگان Letter



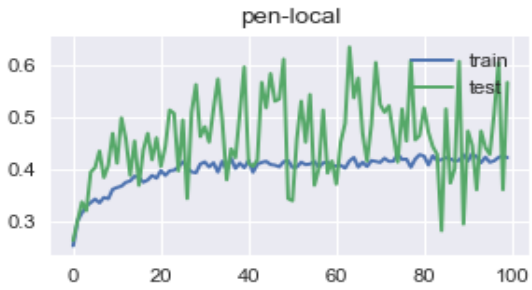
شکل ۸: تغییر دقت در دادگان Aloi



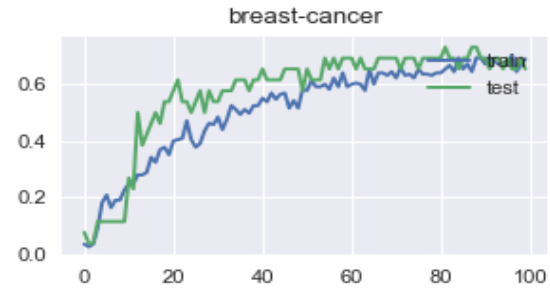
شکل ۱۳: تغییر دقت در دادگان Satellite



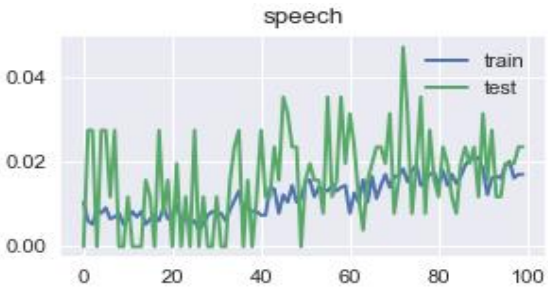
شکل ۹: تغییر دقت در دادگان KDD99



شکل ۱۴: تغییر دقت در دادگان Pen-Local

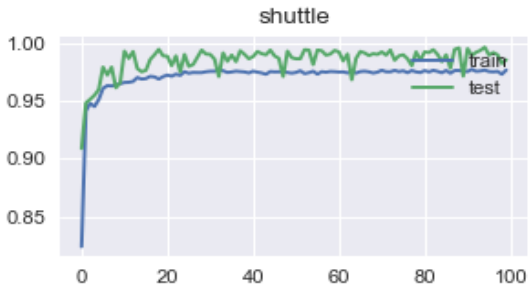


شکل ۱۰: تغییر دقت در دادگان Breast-Cancer



شکل ۱۵: تغییر دقت در دادگان Speech

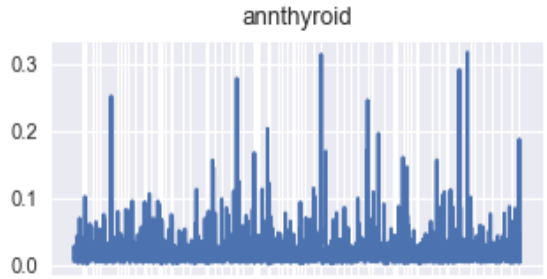
شکل ۱۱: تغییر دقت در دادگان Pen-Global



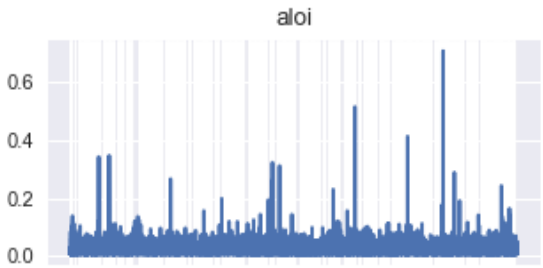
شکل ۱۶: تغییر دقت در دادگان Shuttle

پس از پایان مرحله آموزش، داده‌های مجموعه آزمایشی به مدل نهایی تزریق شده است تا آنها را بازتولید کند و خطای این فرایند، یعنی فاصله مقادیر واقعی و مقادیر تولیدی را به عنوان رتبه ناهنجاری به ما عرضه نماید.

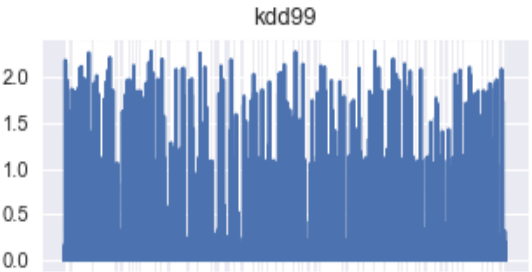
شکل‌های (۱۷) تا (۲۶) نمودار خطای بازسازی را برای هر یک از ۱۰ دادگان به کار گرفته شده، نشان می‌دهند. در این نمودارها، نقاط ناهنجار واقعی، با ترسیم خطوط عمودی مشخص گردیده‌اند.



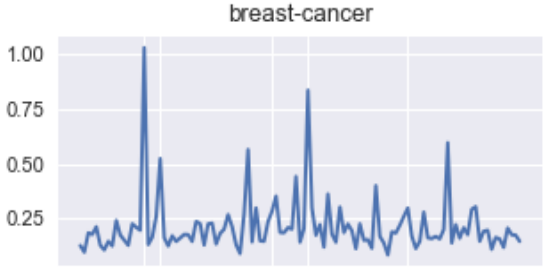
شکل ۱۷: نمودار خطای بازسازی در دادگان Annthyroid



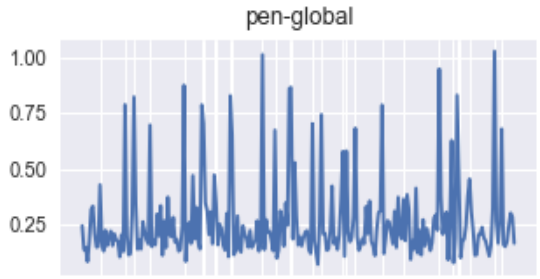
شکل ۱۸: نمودار خطای بازسازی در دادگان Aloi



شکل ۱۹: نمودار خطای بازسازی در دادگان KDD99

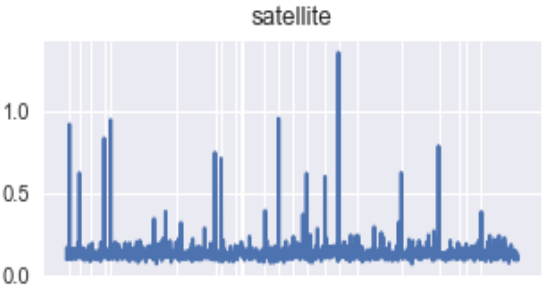


شکل ۲۰: نمودار خطای بازسازی در دادگان Breast-Cancer

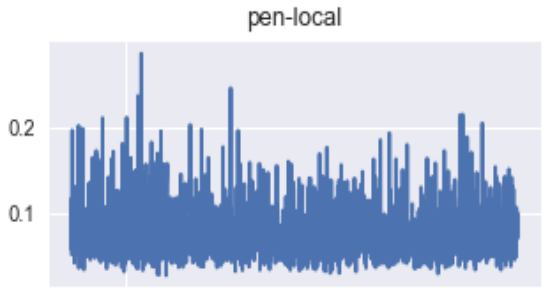


شکل ۲۱: نمودار خطای بازسازی در دادگان Pen-Global

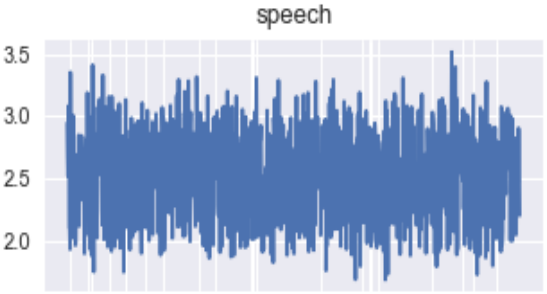
شکل ۲۲: نمودار خطای بازسازی در دادگان Letter



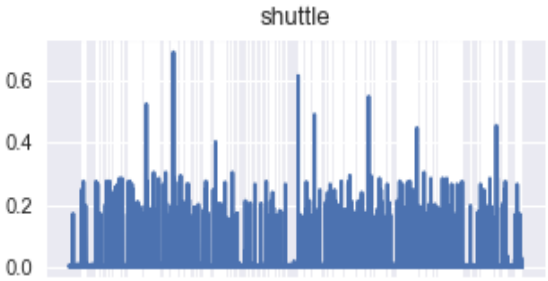
شکل ۲۳: نمودار خطای بازسازی در دادگان Satellite



شکل ۲۴: نمودار خطای بازسازی در دادگان Pen-Local

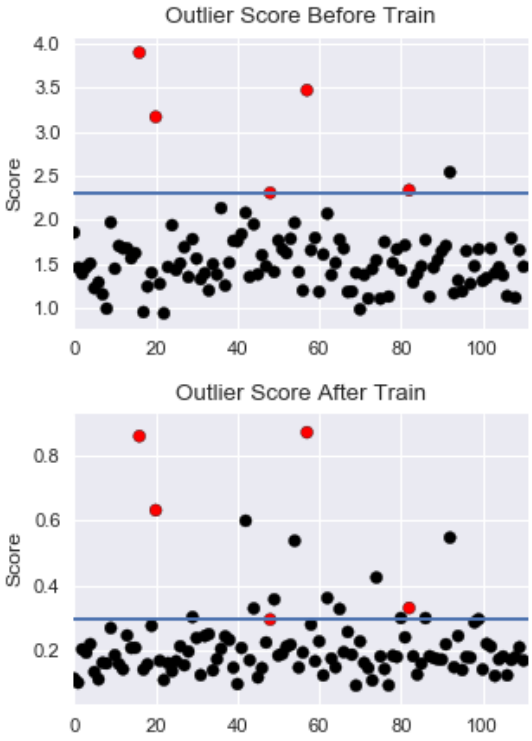


شکل ۲۵: نمودار خطای بازسازی در دادگان Speech



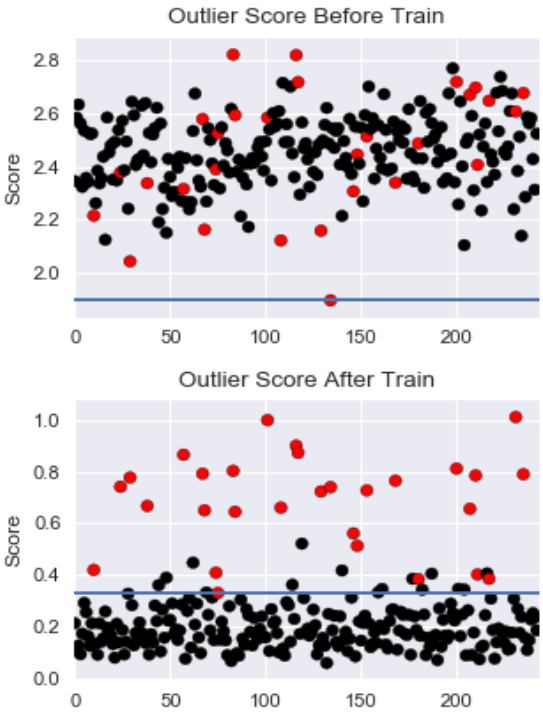
شکل ۲۶: نمودار خطای بازسازی در دادگان Shuttle

از آنجا که همه دادگان مورد استفاده، در دو کلاس هنجار و ناهنجار برچسب خورده‌اند، به سادگی می‌توان میزان دقت و صحت این رتبه‌بندی را مورد بررسی قرار داد. ما با استفاده از این اطلاعات، AUC روش پیشنهادی را برای هرکدام از ده دادگان محاسبه کرده‌ایم. این مقادیر در جدول ۲ فهرست شده‌اند.



شکل ۲۷: توزیع رتبه ناهنجاری در دادگان breast-cancer

بعلاوه ما این مقادیر را با چهار مجموعه دیگر مقایسه کرده‌ایم. مجموعه اول با به‌کارگیری یک کد کننده خودکار عمیق به دست آمده که برای ساخت آن از نورون‌های عادی با تابع فعال‌سازی ReLU استفاده شده است. لایه‌های این کد کننده عمیق، به جز لایه‌های ورودی و خروجی، ۵ تاست که در ساختاری به شکل‌های (۲۸، ۱۴، ۱۰، ۱۴، ۲۸) چیده شده‌اند. تعداد نورون‌های لایه‌های ورودی و خروجی نیز با تعداد ابعاد و ویژگی‌های دادگان مربوطه برابر است. مجموعه‌های دوم، سوم و چهارم هم به منظور مقایسه از مرجع [۴۲] نقل شده است. مجموعه اول حاصل به‌کارگیری روش ماشین بردار پشتیبان یک کلاسه یا oc-SVM[1] برای کشف ناهنجاری در دادگان ده‌گانه مورد استفاده است. مجموعه‌های بعدی نیز به ترتیب بهترین نتایجی است که از روش‌های مبتنی بر نزدیک‌ترین همسایگی (شامل خانواده KNN، LOF، LoOP و LOCI) و خوشه‌بندی (شامل خانواده CBLOF و CMGOS) به دست آمده است.



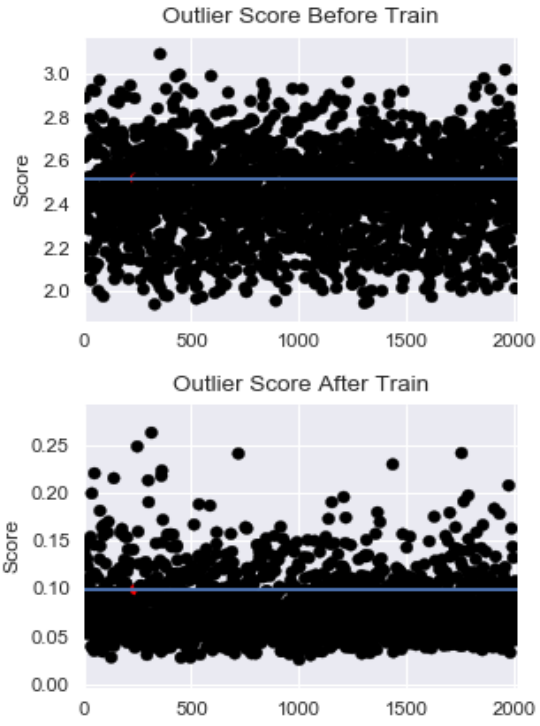
شکل ۲۸: توزیع رتبه ناهنجاری در دادگان pen-global

برای مشاهده تأثیر آموزش در توانایی کد کننده خودکار برای کشف ناهنجاری، به شکل‌های (۲۷) تا (۳۶) توجه کنید. در این شکل‌ها نشان داده شده است که کد کننده خودکار قبل و بعد از آموزش، چه رتبه ناهنجاری را به نقاط مجموعه آزمایشی اختصاص می‌دهد. لازم به ذکر است که کتابخانه مورد استفاده ما، وزن‌های اولیه یک شبکه عصبی آموزش ندیده را بر اساس روش مشهور به Xavier از یک توزیع تصادفی یکنواخت[2] انتخاب می‌کند. شکل‌های مذکور که در آنها نقاط ناهنجار با دایره‌های قرمزرنگ مشخص شده، نشان می‌دهد که روش پیشنهادی ما، غالباً در استخراج ویژگی‌های ذاتی داده‌های هنجار بسیار خوب عمل

[1] One-Class SVM

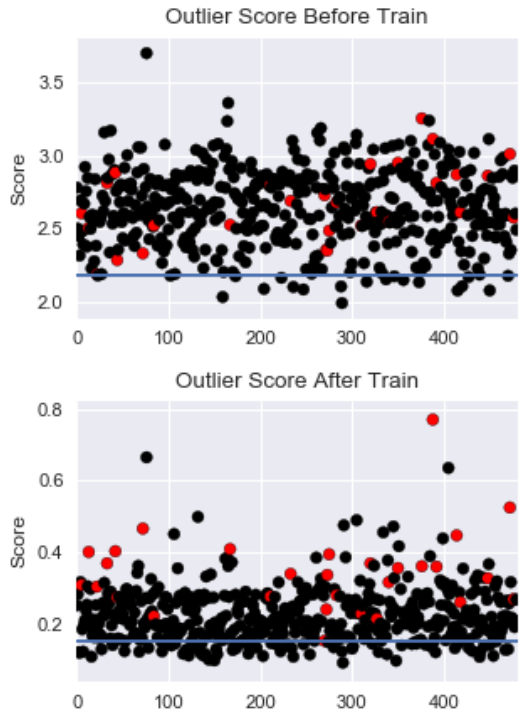
[2] Uniform

کرده است. در واقع تا پیش از انجام فرایند آموزش، نقاط هنجار و ناهنجار به هم آمیخته هستند و رتبه تصادفی که در ابتدای کار به هر یک از نقاط اختصاص داده می‌شود نمی‌تواند به عنوان معیاری برای جداسازی و شناخت ناهنجاری‌ها عمل کند. اما آموزش کد کننده خودکار ضرایب این شبکه عصبی را در جهتی بهینه می‌کند که برآیند کلی آن‌ها نشانگر خصایص اصلی داده‌های مجموعه آموزشی – یعنی داده‌های هنجار – باشد. حالا اگر داده‌های مجموعه آزمایشی را از طریق این شبکه بازسازی کنیم، داده‌هایی که از الگوی غالب مجموعه آموزشی پیروی می‌نمایند با دقت بیشتر و خطای کمتری بازسازی می‌شوند و در نتیجه رتبه ناهنجاری پایین‌تری را به خود اختصاص می‌دهند.

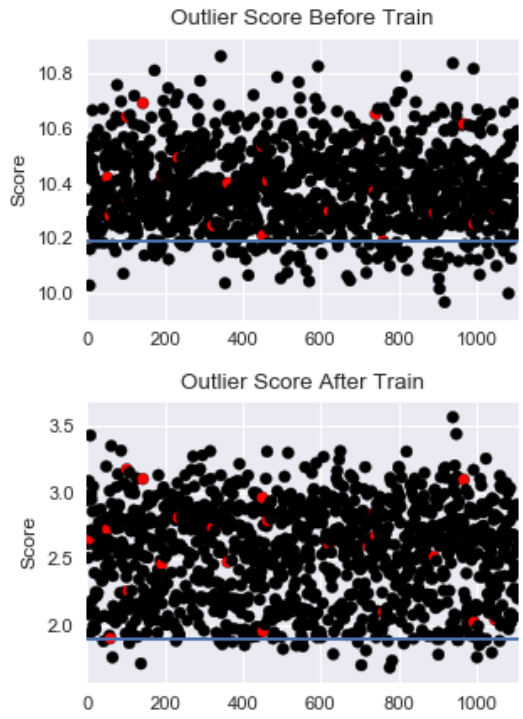


شکل ۲۹: توزیع رتبه ناهنجاری در دادگان pen-local

در مقابل، بازسازی داده‌هایی که از الگوی یاد شده فاصله دارند با خطای بیشتر و رتبه ناهنجاری بزرگ‌تری همراه خواهد بود. بدین ترتیب در شکل‌های (۲۷) تا (۳۶) داده‌های هنجار به شکل نقاط سیاه ته‌نشین شده و نقاط ناهنجار در قالب نقاط قرمز رنگ به سمت بالا حرکت کرده و از داده‌های هنجار جدا شده‌اند. طبیعی است که هرچه معیار کارآیی روش پیشنهادی در مورد یک دادگان بالاتر باشد (جدول ۲) درجه این تفکیک در شکل مربوط به آن دادگان بیشتر است. در هر یک از این شکل‌ها یک خط آبی رنگ عددی را نشان می‌دهد که می‌تواند به عنوان حد آستانه جداسازی داده‌های هنجار و ناهنجار مورد استفاده قرار بگیرد.



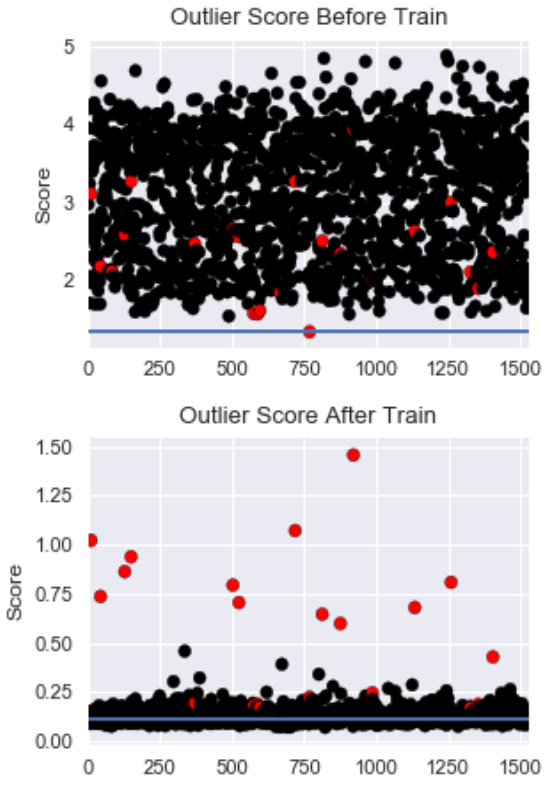
شکل ۳۰: توزیع رتبه ناهنجاری در دادگان letter



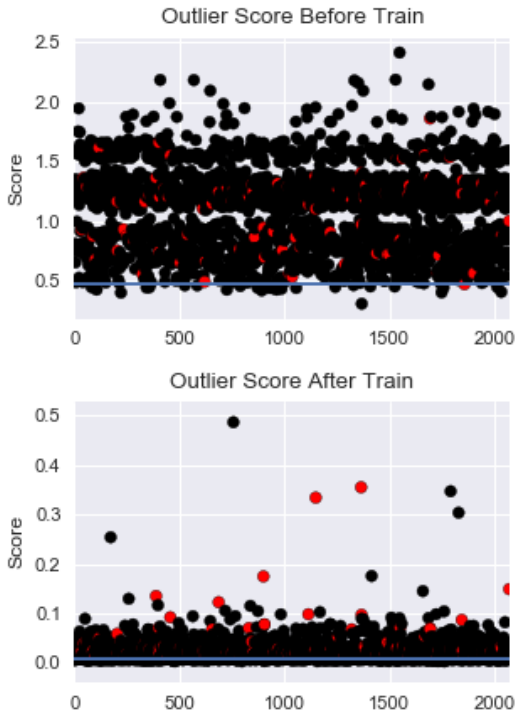
شکل ۳۱: توزیع رتبه ناهنجاری در دادگان speech

رابطه شکل‌های (۱۷) تا (۲۶) و شکل‌های (۱۷) تا (۳۵) هم به نوبه خود درخور توجه است. در واقع این دو مجموعه یکدیگر و اعداد مندرج در جدول ۲ را به خوبی تأئید

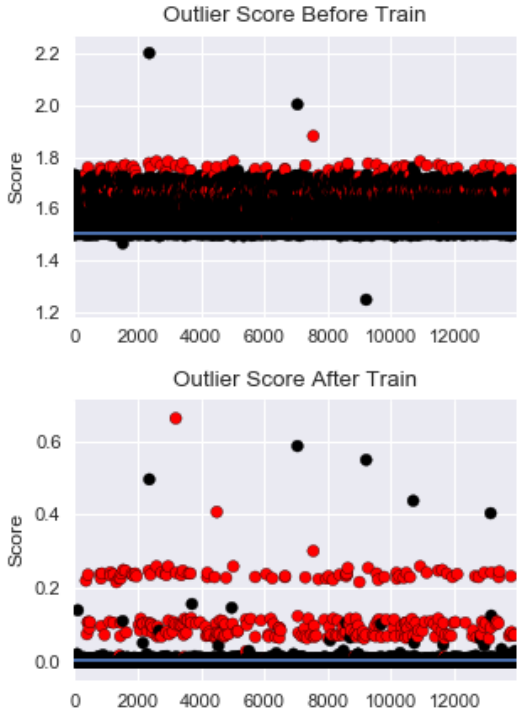
می‌کنند. چنانکه گفته شد شکل‌های (۱۷) تا (۲۶) خطای بازسازی مجموعه آزمایشی هر یک از ده دادگان مورد استفاده را به تصویر می‌کشند. در این نمودارها نقاط ناهنجار واقعی با خطوط سفید رنگ نمایش داده شده‌اند. به همین دلیل نقاطی که خطای بازسازی آنها بزرگ‌تر است، قاعدتاً باید بر روی همین خطوط سفید قرارگرفته باشند. هر چقدر میزان این مطابقت بیشتر باشد یعنی شبکه کد کننده خودکار دریافتن نقاط ناهنجار واقعی درست‌تر عمل کرده است. مثلاً در شکل‌های (۲۱) و (۲۶) که به دو دادگان Pen-global و Shuttle تعلق دارند تطابق مقادیر بزرگ‌تر و خطوط سفیدرنگ کاملاً مشهود است. مقادیر AUC این دو دادگان بر اساس جدول ۲ به ترتیب برابر ۰/۹۹۹۶ و ۰/۹۹۸۸ است. بعلاوه مرزبندی واضحی که در شکل‌های ۲۸ و ۳۴ در میان نقاط سیاه و قرمز وجود دارد نیز ثابت می‌کند که روش پیشنهادی ما در مورد دو دادگان فوق‌الذکر خوب عمل کرده است. در مقابل آشفتگی موجود در شکل (۲۵)، مقدار پایین AUC دادگان Speech (۰/۵۹۵۴) و آمیختگی نقاط سیاه و قرمز شکل (۳۱) را توجیه می‌کند و نشان می‌دهد که چرا روش پیشنهادی ما (و سایر روش‌های جدول ۲) دریافتن ناهنجاری‌های این دادگان ناموفق بوده و شبیه یک روش تصادفی عمل کرده‌اند.



شکل ۳۲: توزیع رتبه ناهنجاری در دادگان satellite



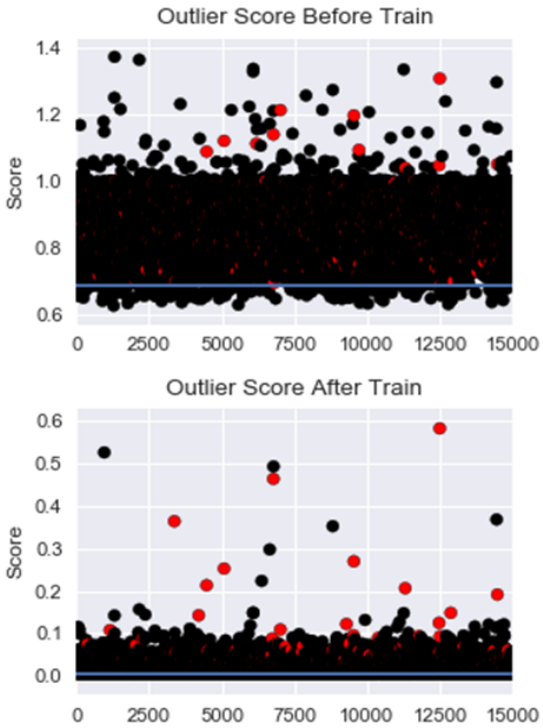
شکل ۳۳: توزیع رتبه ناهنجاری در دادگان annthyroid



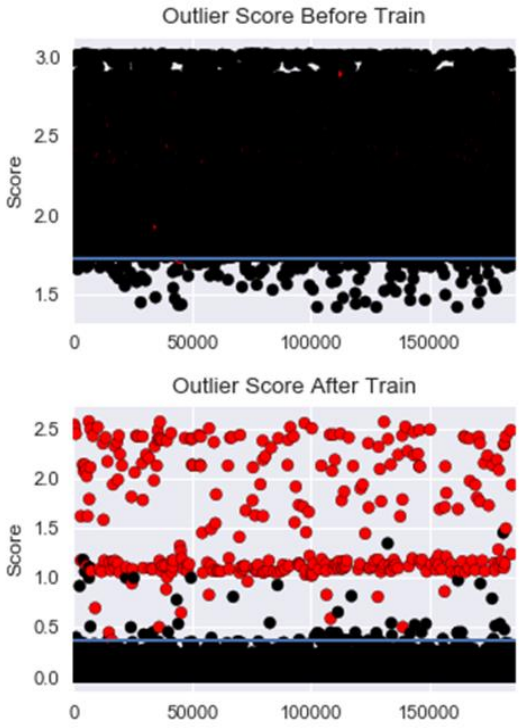
شکل ۳۴: توزیع رتبه ناهنجاری در دادگان shuttle

نهایتاً توجه به نمودارهای مندرج در شکل‌های (۷) تا (۱۶) ثابت می‌کند که گرچه دقت در اغلب حوزه‌های داده‌کاوی معیار مهمی محسوب می‌شود، اما در مقوله کشف ناهنجاری صرف توجه به این معیار مهم می‌تواند بسیار گمراه کننده باشد. مثلاً روش پیشنهادی ما در مورد دو دادگان Aloi و Kdd99 به دقت‌های حدود ۰/۹۹ و ۰/۳ دست پیدا کرده

است لیکن در کشف ناهنجاری‌های دادگان Aloi بسیار ناموفق (AUC برابر ۰/۵۳۸۶) و دریافتن ناهنجارهای دادگان Kdd99 بسیار موفق (AUC حدود ۰/۹۹۹۸) بوده است. ملاحظه دو شکل (۳۵) و (۳۶) تفاوت اخیر را به وضوح نشان می‌دهد.



شکل ۳۵: توزیع رتبه ناهنجاری در دادگان aloi



شکل ۳۶: توزیع رتبه ناهنجاری در دادگان kdd99

لازم به ذکر است که تمام برنامه‌های مورد نیاز به زبان Python و با استفاده از کتابخانه‌های tensorflow و keras نوشته شده‌اند. این برنامه‌ها بر روی سیستم عامل linux Mint اجرا گردیده‌اند و برای استفاده از توان کارت گرافیکی، به امکانات نسخه ۸ کتابخانه CUDA متکی بوده‌اند. کامپیوتر مورداستفاده نیز به یک کارت گرافیکی Nvidia GeForce 920 MX با ۲۵۶ هسته کودا، یک پردازنده چهار هسته‌ای Intel Core i7 7500U 2.7 Hz و ۸ گیگابایت حافظه اصلی مجهز بوده است.

۵- نتیجه‌گیری

در این مقاله ما از یک کد کننده خودکار مبتنی بر بلوک-های LSTM برای کشف ناهنجاری نقطه‌ای یک مجموعه دادگان جدولی استفاده کرده‌ایم. تجربیات ما که به شکل نیمه با ناظر انجام شد نشان داد کد کننده خودکاری که از بلوک‌های LSTM ایجاد شده است، در کشف ناهنجاری از روش‌های مشهور و رایجی مثل SVM بهتر عمل می‌کند. بعلاوه این کد کننده، نتایج یک کد کننده عادی عمیق را نیز پشت سر می‌گذارد.

دلیل توفیق کد کننده‌ها – به ویژه کد کننده خودکار مبتنی بر – LSTMاین است که مثل سایر مدل‌های یادگیری عمیق برای استخراج خصایص ذاتی داده‌ها از لایه‌های غیرخطی استفاده می‌کنند. به همین دلیل از روش‌های رایج در داده کاوی سنتی قوی‌ترند. البته این مدل‌ها، از جمله مدل پیشنهادی برای آموزش به داده کافی نیاز دارند. در مورد کشف ناهنجاری داده کافی می‌تواند به معنی زیاد بودن تعداد اقلام دادگان (مثل دادگان Kdd99)، یا بالا بودن درصد فراوانی ناهنجاری‌ها (مثل دادگان Pen-global) باشد. البته نتایج به دست آمده، در چند مورد به ویژه در مورد ناهنجاری‌های محلی ضعیف است. در واقع به نظر می‌رسد که شبکه پیشنهادی در استخراج مجاورت‌ها و شباهت‌های مکانی درست عمل نمی‌کند. خوشبختانه نوع رایجی از ساختارهای یادگیری عمیق، یعنی شبکه‌های کانولوشنی در این مقوله بسیار توانا هستند؛ بنابراین ترکیب این دو نوع شبکه و ایجاد یک کد کننده خودکار که در ساختار آن از کانولوشن و LSTM به طور توأم استفاده شده، می‌تواند موضوع مطالعات بعدی باشد.

۶- مراجع

[1] F.E. Grubbs, "Procedures for Detecting Outlying Observations in Samples", Technometrics, Vol. 11, No. 1, 1969, pp. 1–21.

[2] W. Rechenberg, "Identification of outliers", Fresenius' Zeitschrift für analytische Chemie, Vol. 311, No. 6, 1982, pp. 590–597.

[3] Y. Ma, P. Zhang, Y. Cao, and L. Guo, "Parallel auto-encoder for efficient outlier detection", 2013 IEEE International Conference on Big Data, Vol. 2, No. 3, 2013, pp. 15–17.

[4] C. Zhou and R.C. Paffenroth, "Anomaly Detection with Robust Deep Autoencoders", Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Min. - KDD '17, 2017, pp. 665–674.

[5] M.S. Aldosari and E. Blaisten-Barojas, "Unsupervised Anomaly Detection in Sequences Using Long Short Term Memory Recurrent Neural Networks", George Mason University, 2016, pp. 1-25.

[6] S.P. Singh, A. Kumar, H. Darbari, L. Singh, A. Rastogi, and S. Jain, "Machine translation using deep learning: An overview", 2017 International Conference on Computer, Communications and Electronics (Comptelix), 2017, pp. 162–167.

[7] S. Chauhan and L. Vig, "Anomaly detection in ECG time signals via deep long short-term memory networks", in Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics, DSAA 2015, 2015, pp. 1–7.

[8] M. Markou and S. Singh, "Novelty detection: a review—part 1: statistical approaches", Signal Processing, Vol. 83, No. 12, 2003, pp. 2481–2497.

[9] M. Markou and S. Singh, "Novelty detection: a review—part 2: neural network based approaches", Signal Processing, Vol. 83, No. 12, 2003, pp. 2499–2521.

[10] E.R. de Faria, I.R. Goncalves, J. ao Gama, and A.C.P. de L.F. Carvalho, "Evaluation of Multiclass Novelty Detection Algorithms for Data Streams", IEEE Trans. Knowl. Data Eng., Vol. 27, No. 11, 2015, pp. 2961–2973.

[11] C. Satheesh Chandran, S. Kamal, A. Mujeeb, and M.H. Supriya, "Novel class detection of underwater targets using Self-Organizing neural networks", in 2015 IEEE Underwater Technology (UT), 2015, pp. 1–5.

[12] L. Tarassenko, "Novelty detection for the identification of masses in mammograms", in 4th International Conference on Artificial Neural Networks, Vol. 1995, 1995, pp. 442–447.

[13] K. WORDEN, G. MANSON, and D. ALLMAN, "Experimental Validation of a Structural Health Monitoring Methodology: Part I. Novelty Detection on a Laboratory Structure", Journal of Sound and Vibration, Vol. 259, No. 2, 2003, pp. 323–343.

[14] J. Foote, "Automatic audio segmentation using a measure of audio novelty", in 2000 IEEE International Conference on Multimedia and Expo. ICME2000. Proceedings. Latest Advances in the Fast Changing World of Multimedia (Cat. No.00TH8532), Vol. 1, 2000, pp. 452–455.

[۱۵] غ. شفابخش، ح. نادر پور، ف. فصیحی، " انتخاب الگوریتم بهینه شبکه عصبی در تحلیل روسازی‌های انعطاف‌پذیر راه‌ها"، مدل‌سازی در مهندسی، دوره ۸، شماره ۲۱، ۱۳۸۹، صفحه ۵۶-۴۵.

[۱۶] ع. مرتضایی، ع. خیرالدین، "مدل‌سازی و تخمین طول مفصل پلاستیک ستون‌های بتن‌آرمه به کمک شبکه‌های عصبی مصنوعی"، مدل‌سازی در مهندسی، دوره ۱۰، شماره ۲۹، ۱۳۹۱، صفحه ۱۷-۱.

[۱۷] ز. مروج، ج. آذرخش، "شبیه‌سازی و طبقه‌بندی وقایع کیفیت توان با استفاده از شبکه عصبی"، مدل‌سازی در مهندسی, دوره ۱۳، شماره ۴۱، ۱۳۹۴، صفحه ۱۴۶-۱۳۷.

[۱۸] س.ع. سلیمانی ایوری، م. فدوی امیری، ح. مروی، "تولید سیگنال مصنوعی زلزله به کمک مدلی جدید در فشرده‌سازی و آموزش شبکه‌های عصبی مصنوعی"، مدل‌سازی در مهندسی, دوره ۱۴، شماره ۴۶, ۱۳۹۵، صفحه ۸۵-۷۵.

[19] E.W. Tavares Ferreira, G. Arantes Carrijo, R. de Oliveira, and N. Virgilio de Souza Araujo, "Intrusion Detection System with Wavelet and Neural Artifical Network Approach for Networks Computers", IEEE Latin America Transactions, Vol. 9, No. 5, 2011, pp. 832–837.

[20] M.A.F. Pimentel, D.A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection", Signal Processing, Vol. 99, 2014, pp. 215–249.

[21] B.B. Thompson, R.J. Marks, J.J. Choi, M.A. El-Sharkawi, and C. Bunje, "Implicit learning in autoencoder novelty assessment", Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290), 2002, pp. 2878–2883.

[22] M. Sabokrou, M. Fathy, M. Hoseini, and R. Klette, "Real-time anomaly detection and localization in crowded scenes", 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW),

2015, pp. 56–62.

[23] W. Yan and L. Yu, "On Accurate and Reliable Anomaly Detection for Gas Turbine Combustors: A Deep Learning Approach", Annual Conference of the Prognostics and Health Management Society, 2015, pp. 1–8.

[24] Y. Xiong and R. Zuo, "Recognition of geochemical anomalies using a deep autoencoder network", Computers & Geosciences, Vol. 86, 2016, pp. 75–82.

[25] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long Short Term Memory Networks for Anomaly Detection in Time Series", in European Symposium on Artificial Neural Networks, No. April, 2015, pp. 22–24.

[26] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based Encoder-Decoder for Multi-sensor Anomaly Detection", in Anomaly Detection Workshop at 33rd International Conference on Machine Learning (ICML 2016), 2016, pp. 25–30.

[27] M. Cheng, Q. Xu, J. Lv, W. Liu, Q. Li, and J. Wang, "MS-LSTM: A multi-scale LSTM model for BGP anomaly detection", Proc. - Int. Conf. Netw. Protoc. ICNP, Vol. 2016–Decem, no. NetworkML, 2016, pp. 1–6.

[28] B. Schölkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt, "Support vector method for novelty detection", Proceedings of the 12th International Conference on Neural Information Processing Systems. MIT Press, 1999, pp. 582–588.

[29] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines", in Proceedings of the International Joint Conference on Neural Networks, 2003. Vol. 3, 2003, pp. 1741–1745.

[30] P. Hayton, B. Schölkopf, L. Tarassenko, and P. Anuzis, "Support vector novelty detection applied to jet engine vibration spectra", Proceedings of the 13th International Conference on Neural Information Processing Systems. MIT Press, 2000, pp. 907–913.

[31] L. Tarassenko, A. Nairac, N. Townsend, and P. Cowley, "Novelty detection in jet engines", in IEE Colloquium on Condition Monitoring: Machinery, External Structures and Health (Ref. No. 1999/034), 1999, pp. 1–5.

[32] L. Clifton, D. A. Clifton, Y. Zhang, P. Watkinson, L. Tarassenko, and H. Yin, "Probabilistic Novelty Detection With Support Vector Machines", IEEE Transactions on Reliability, Vol. 63, No. 2, 2014, pp. 455–467.

[33] D. R. Hardoon and L. M. Manevitz, "One-class machine learning approach for fMRI analysis", in Proceedings of Postgraduate Research Conference in Electronics, Photonics, Communications and Networks, and Computer Science (PREP), Lancaster, UK, 2005b, 2000, pp. 1–2.

[34] M. Davy, F. Desobry, A. Gretton, and C. Doncarli, "An online support vector machine for abnormal events detection", Signal Processing, Vol. 86, No. 8, 2006, pp. 2009–2025.

[35] J. Elman, "Finding structure in time* 1", Cognitive Science, Vol. 14, No. 1, 1990, pp. 179–211.

[36] M. Jordan, "Serial order: A parallel distributed processing approach", Advances in Psychology, Vol. 121, 1997, pp. 471–495.

[37] Z.C. Lipton, J. Berkowitz, and C. Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning", 2015, pp. 1–38.

[38] S. Hochreiter and J. Urgen Schmidhuber, "Long Short-Term Memory", Neural Computation, Vol. 9, No. 8, 1997, pp. 1735–1780.

[39] F.A. Gers and J. Schmidhuber, "Recurrent nets that time and count", in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium, Vol. 3, 2000, pp. 189–194.

[40] K. Greff, R.K. Srivastava, J. Koutník, B.R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey", IEEE Transactions on Neural Networks and Learning Systems, Vol. 28, No. 10. 2017, pp. 2222–2232.

[41] T. Fawcett, "An introduction to ROC analysis", Pattern Recognition Letters, Vol. 27, No. 8, 2006, pp. 861–874.

[42] M. Goldstein, and S. Uchida, "A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data", PLoS One, Vol. 11, No. 4, 2016, pp. 1–31.

[43] R.C. Staudemeyer, "Applying long short-term memory recurrent neural networks to intrusion detection", Sacj, Vol. 56, No. 56, 2015, pp. 136–154.

[44] G.O. Campos, A. Zimek, J. Sander, R.J.G.B. Campello, B. Micenková, E. Schubert, I. Assent, and M.E. Houle, "On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study", Data Mining and Knowledge Discovery, Vol. 30, No. 4, 2016, pp. 891–927.

[45] A. Emmott, S. Das, T. Dietterich, A. Fern, and W.-K. Wong, "A Meta-Analysis of the Anomaly Detection Problem", Oregon State University Libraries & Press, 2015, pp. 12-23.

[46] O.L. Mangasarian, W.N. Street, and W.H. Wolberg, "Breast Cancer Diagnosis and Prognosis Via Linear Programming", Oper. Res., Vol. 43, No. 4, 1995, pp. 570–577.

[47] H.-P. Kriegel, P. Kröger, E. Schubert, and A. Zimek, "LoOP: local outlier probabilities", Proceedings of the 18th ACM Conference on Information and Knowledge Management, 2009, pp. 1649–1652.

[48] B. Micenková, B. McWilliams, and I. Assent, "Learning Outlier Ensembles: The Best of Both Worlds – Supervised and Unsupervised", Proc. ACM SIGKDD Work. Outlier Detect. Descr. ODD., 2014, pp. 1–4.

[49] W. Schi, M. Joost, R. Werner, and D.- Koblenz, "Synthesis and Performance Analysis of Multilayer Neural Network Architectures", Koblenz, 1992, pp. 100-130.

[50] N. Abe, B. Zadrozny, and J. Langford, "Outlier detection by active learning", in Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '06, 2006, pp. 504–509.

[51] M. Reif, M. Goldstein, A. Stahl, and T.M. Breuel, "Anomaly detection by combining decision trees and parametric densities", in 2008 19th International Conference on Pattern Recognition, 2008, pp. 1–4.

[52] J.-M. Geusebroek, G.J. Burghouts, and A.W.M. Smeulders, "The Amsterdam Library of Object Images", International Journal of Computer Vision, Vol. 61, No. 1, 2005, pp. 103–112.

[53] E. Schubert, R. Wojdanowski, A. Zimek, and H.-P. Kriegel, "On Evaluation of Outlier Rankings and Outlier Scores", Proceedings of the 2012 SIAM International Conference on Data Mining, 2012, pp. 1047–1058.

[54] U. Carrasquilla, "Benchmarking Algorithms for Detecting Anomalies in Large Datasets", Rev. Lit. Arts Am., 2010, pp. 1–16.

[55] K. Leung and C. Leckie, "Unsupervised anomaly detection in network intrusion detection using clusters", Proceedings of the Twenty-eighth Australasian Conference on Computer Science, Vol. 38, 2005, pp. 333–342.