Electronic & Cyber Defense
Vol. 13, No. 4, 2026 (Serial No. 52). pp. 23- 39
ISSN: 2322-4347, E-ISSN: 2980-8979

Self-Supervised Anomaly Detection in Multivariate Time Series using Transformer Encoder

Farzaneh Taherizade[1], Vali Derhami [2*]

[1] PhD student, Yazd University, Yazd, Iran. *farzaneh.taherizade@stu.yazd.ac.ir*

[2] Professor, Yazd University, Yazd, Iran (Correspondence: *vderhami@yazd.ac.ir*)

ARTICLE INFO

Article history:

Article Type: Research paper

Receive: 04 October 2024
Revise: 22 November 2025
Accept: 01 December 2025
Available online: 22 December 2025

Keywords:
Anomaly Detection
Anomaly Score
Self-Supervised Learning
Transformer Encoder

ABSTRACT

Anomaly Anomaly detection in multivariate time series data is essential for various applications, including industry, medicine, and urban management. However, quickly and accurately identifying anomalies is challenging due to the absence of labels, complex temporal and spatial relationships, and high-dimensional data. Despite recent advances in deep learning, few methods can effectively address these challenges. This article proposes a transformer encoder that efficiently models temporal and spatial complexity in high-dimensional data using self-supervised learning, thereby eliminating the need for anomaly labels. A novel anomaly score function has also been introduced to reduce false detections. Extensive experiments on a benchmark dataset demonstrate that our method surpasses state-of-the-art baselines in anomaly detection, particularly for anomalous events, achieving a 5% improvement in average composite F-score and a 15% reduction in training time.

Cite this article: Taherizade, F., Derhami, V. Self-Supervised Anomaly Detection in Multivariate Time Series using Transformer Encoder. *Electronic and Cyber Defense*, 2025; 13(4), 23- 38.

DOI: https://doi.org/10.47176/ECDJ.2025.1612

Publisher: Imam Hossein University

«پدافند الکترونیکی و سایبری»
سال سیزدهم، شماره ۴، زمستان ۱۴۰۴، ص ۲۳- ۳۹
شاپا الکترونیکی: ۴۳۴۷ -۲۳۲۲ شاپا چاپی: ۸۹۷۹- ۲۹۸۰

تشخیص ناهنجاری خود نظارت در سری‌های زمانی چند متغیره با استفاده از رمزگذار ترانسفورمر

فرزانه طاهری زاده[1]، ولی درهمی *[2]

[1] دانشجوی دکتری ،دانشگاه یزد، یزد، ایران. farzaneh.taherizade@stu.yazd.ac.ir

[2] استاد، دانشکده‌ی مهندسی کامپیوتر، دانشگاه یزد، یزد، ایران (نویسنده مسئول: *vderhami@yazd.ac.ir*)

چکیده

تشخیص ناهنجاری در داده‌های سری زمانی چند متغیره برای کاربردهای زیادی ازجمله صنعت، پزشکی و مدیریت شهری اهمیت زیادی دارد. برای این‌که یک روش بتواند نقاط و رویدادهای ناهنجار را به‌سرعت و با دقت مشخص کند، با چالش‌های بزرگی مانند عدم وجود برچسب‌های ناهنجاری، پیچیدگی روابط(وابستگی) زمانی و مکانی داده‌ها، ابعاد بالا و سرعت موردنیاز در کاربردهای مدرن روبرو است. علیرغم پیشرفت‌های اخیر روش‌های یادگیری عمیق برای تشخیص ناهنجاری، تنها تعداد کمی از آن‌ها می‌توانند همه این چالش‌ها را برطرف کنند. در این مقاله از رمزگذار ترانسفورمر مبتنی بر توجه برای مدل‌سازی پیچیدگی زمانی و مکانی داده‌های با ابعاد بالا در کمترین زمان ممکن استفاده می‌شود. این مدل با یادگیری خود نظارت آموزش داده می‌شود تا به برچسب ناهنجاری‌ها نیازی نباشد. همچنین تابع امتیاز ناهنجاری مناسبی برای جلوگیری از تشخیص اشتباه ناهنجاری معرفی‌شده است. آزمایش‌های گسترده بر روی 28 سرور مستقل مجموعه داده‌ی *SMD*، تأیید می‌کند که روش ارائه‌شده از اکثر روش‌های پایه پیشرفته کنونی در تشخیص ناهنجاری مخصوصاً رویدادهای ناهنجار عملکرد بهتری دارد، به‌طور خاص میانگین امتیاز *F* ترکیبی 5 درصد بهبودیافته است و زمان آموزش 15 درصد کاهش می‌یابد.

مشخصات مقاله

تاریخچه مقاله:

نوع مقاله: علمی- پژوهشی

دریافت: 1404/07/13

بازنگری: 1404/09/01

پذیرش: 1404/09/10

ارائه آنلاین: 1404/10/01

کلیدواژه‌ها:

تابع امتیاز ناهنجاری

تشخیص ناهنجاری

رمزگذار ترانسفورمر

یادگیری خود نظارت

استناد: طاهری زاده, فرزانه, درهمی, ولی. تشخیص ناهنجاری خود نظارت در سری‌های زمانی چند متغیره با استفاده از رمزگذار ترانسفورمر. پدافند الکترونیکی و سایبری, ۱۴۰۴. ۱۳(۴)، ص ۲۳- ۳۹. https://doi.org/10.47176/ECDJ.2025.1612

ناشر: دانشگاه جامع امام حسین(ع).

1. مقدمه

سامانه‌های فیزیکی سایبری مدرن که در صنعت تولید، هواپیماها، سرور، شبکه و اینترنت [1, 2, 3]، سامانه‌های مالی [4]، پزشکی [5]، اینترنت اشیا [1, 6][1]، روباتیک [7] و مدیریت منابع شهری [5, 6] مورداستفاده قرار می‌گیرند، شامل تجهیزات پیچیده‌ای هستند که می‌توانند اطلاعات قسمت‌های مختلف خود را در بازه‌های زمانی متفاوت ثبت کنند [1]، به‌عنوان نمونه می‌توان به حسگرهای شبکه‌ای توزیع‌شده برای اندازه‌گیری دما و فشار در یک نیروگاه، اطلاعات اجزای متصل به سرورها (به‌عنوان‌مثال، کارکرد واحد پردازنده‌ی مرکزی [2] و ورودی/خروجی دیسک [3]) اشاره کرد.

نظارت مداوم بر این دستگاه‌ها برای اطمینان از عملکرد درست آن‌ها و جلوگیری از خرابی‌هایی که باعث ضررهای هنگفت مالی می‌شوند، امری ضروری است [1]. تشخیص دقیق زمان خرابی آن‌ها برای جلوگیری از ضررهای مالی و تجاری بسیار مهم است، زیرا یک دقیقه توقف در کار یک نیروگاه ممکن است هزاران دلار هزینه داشته باشد.

کنترل و نظارت [4] بر رفتارهای این سامانه‌ها، مقدار قابل‌توجهی از داده‌های سری زمانی چند متغیره تولید می‌کند. سری زمانی چند متغیره به‌نوعی از داده‌ها اشاره دارد که متشکل از چندین متغیر است که در طول زمان با گام‌های زمانی یکسان نمونه‌برداری می‌شوند. وابستگی‌هایی که بین مقدار هر متغیر با مقادیر آن در گام‌های زمانی گذشته وجود دارد را زمینه‌ی زمانی[5] آن متغیر می‌گویند، همچنین وابستگی‌هایی که بین مقادیر متغیرها در هر گام زمانی وجود دارد را زمینه‌ی مکانی آن می‌نامند.

در سامانه‌های نظارتی، داده‌ها از نوع سری زمانی چند متغیره هستند که ازلحاظ عملکرد سیستم در هنگام ثبت آن‌ها به دودسته تقسیم می‌شوند، دسته اول مربوط به زمان‌هایی هستند که سیستم به‌درستی کار می‌کند که به آن‌ها دسته یا کلاس عادی می‌گویند و دسته‌ی دیگر مربوط به زمان‌هایی است که رفتار سیستم با رفتار درست (عادی) خود مطابقت ندارد که به آن‌ها دسته یا کلاس ناهنجاری [6] می‌گویند. تشخیص داده‌های ناهنجاری در این سامانه‌ها، یک‌رشته تحقیقاتی فعال است که تشخیص ناهنجاری در سری‌های زمانی چند متغیره نامیده می‌شود.

با توجه به جنبه‌های زیاد مسائل تشخیص ناهنجاری، دسته‌بندی‌های مختلفی برای این روش‌ها وجود دارد. یکی از آن‌ها، دسته‌بندی بر اساس برچسب‌های در دسترس داده‌های ورودی است. این روش‌ها بر اساس میزان داده‌های بر چسب‌دار در دسترس به سه دسته تقسیم می‌شوند [10]:

الف- یادگیری با نظارت: در تشخیص ناهنجاری با نظارت، فرض می‌شود که مجموعه داده، کاملاً برچسب‌گذاری شده است [11]. هنگامی‌که ناهنجاری‌ها به‌راحتی قابل برچسب‌گذاری باشند، استفاده از روش‌های نظارت‌شده سودمندتر است [12]. در این مرحله، تمایز بین تشخیص ناهنجاری نظارت‌شده و مسئله‌ی طبقه‌بندی باینری ضروری است. ممکن است کسی ادعا کند که اگر داده‌های عادی و غیرعادی در مرحله آموزش در دسترس باشند، می‌توان مسئله را به‌عنوان یک مسئله طبقه‌بندی باینری نظارت‌شده فرمول‌بندی کرد و دیگر یک مسئله‌ی تشخیص ناهنجاری نخواهد بود. بااین‌حال، باید توجه داشت که در تعریف ریاضی، ناهنجاری، نمونه‌ای است که به توزیع داده‌های کلاس عادی (که با P^+ نشان داده می‌شود) تعلق ندارد. کلاس ناهنجاری شامل طیف وسیعی از نقاط داده است که تمام آن‌ها در مرحله‌ی آموزش شناخته‌شده نیستند [10].

ب-یادگیری نیمه نظارت‌شده: در این دسته فرض می‌شود که درصد کمی از مجموعه داده‌ی آموزشی، برچسب‌گذاری شده است و شامل نمونه‌های بر چسب‌دار و بدون برچسب است؛ که برای سناریوهایی که در آن برچسب‌گذاری کل داده‌ها هزینه‌بر است، مناسب هستند. استفاده از این روش‌ها در تشخیص ناهنجاری رایج است زیرا معمولاً داده‌های بر چسب‌دار و بدون برچسب وجود دارند، اما برچسب‌گذاری تمام داده‌ها سخت است و به دانش تخصصی نیاز دارد. ترکیب مجموعه‌ی کوچکی از نمونه‌های ناهنجاری در طول آموزش، می‌تواند به‌طور قابل‌توجهی دقت تشخیص را بهبود بخشد و به‌ویژه در مقایسه با مدل‌های بدون نظارت، استحکام مدل را به حداکثر برساند [13].

ازآنجایی‌که نمونه‌های غیرعادی برچسب‌گذاری شده، اغلب نادر هستند، این روش‌ها مستعد برازش بیش‌ازحد هستند. بنابراین ایجاد فرضیات صحیح در مورد توزیع ناهنجاری‌ها (که با P^- نشان داده می‌شود)، برای ترکیب دقیق ناهنجاری‌های برچسب‌گذاری شده، در فرآیند آموزش بسیار مهم است. توجه به این نکته مهم است که در برخی از مقالات موجود، روش‌هایی که از نمونه‌های آموزشی کلاس عادی و بدون برچسب برای یادگیری استفاده می‌کنند، به‌عنوان روش نیمه نظارتی در نظر گرفته می‌شوند. باید توجه داشت که بر اساس تعاریف بالا، روشی که در آن کل داده‌های آموزشی متعلق به کلاس عادی است، روش بدون نظارت است [10].

ج-یادگیری بدون نظارت: در این دسته فرض می‌شود که فقط داده‌های بدون برچسب برای آموزش مدل در دسترس هستند. در شکل ساده‌شده‌ی یادگیری بدون نظارت، معمولاً فرض

[1] Internet Of Things
[2] Central Processing Unit
[3] Input/ Output
[4] Monitoring
[5] Temporal Context
[6] Anomaly

می‌شود که داده‌های بدون برچسبی که در دسترس هستند، بدون ناهنجاری هستند و توزیع کل داده‌های در دسترس (که با P نشان داده می‌شود) برای آموزش، برابر توزیع داده‌های کلاس عادی است $(P \equiv P^{+})$ [14, 15]. اگر داده‌های پرسروصدا یا ناهنجاری‌های شناسایی نشده در مجموعه داده‌ی آموزشی وجود داشته باشد؛ این مفروضات نقض می‌شوند. بنابراین مدل‌های توسعه‌یافته با این روش‌ها قوی نیستند.

یک رویکرد واقعی‌تر می‌تواند این باشد که فرض شود توزیع داده‌های بدون برچسب در دسترس، ترکیبی از داده‌های عادی و ناهنجاری با درصد ناهنجاری $\eta \in (0,1)$ است یعنی $P = (1-\eta)P^{+} + \eta P^{-}$. در این رویکرد تعیین η و ایجاد یک فرض قبلی در مورد توزیع داده‌های ناهنجاری‌ها (یعنی P^{-}) بسیار مهم است که ممکن است تعمیم روش را کاهش دهد. به‌طورکلی این روش‌ها، علاقه‌ی زیادی دارند که بتوانند بدون نیاز به دسترسی به نمونه‌های آموزشی بر چسب‌دار، الگوهای مشترک موجود در داده‌ها را به دست آورند. باید توجه داشت که روش‌های یادگیری خود نظارتی که در این مقاله مورداستفاده قرار می‌گیرند، زیرگروهی از فن‌های یادگیری بدون نظارت است [10].

داده‌های کلاس عادی فراوان است، اما داده‌های کلاس ناهنجاری خیلی کم و معمولاً نادر هستند و درعین‌حال با نقاط عادی گسترده پنهان می‌شوند. همچنین برچسب‌گذاری این داده‌ها سخت است و نیاز به دانش زیاد در مورد سیستم مربوطه دارد [12] که باعث هزینه‌ی بالای کارشناسی آن می‌شود. درعین‌حال نمی‌توان تضمین کرد که همه‌ی انواع ناهنجاری‌ها دقیقاً برچسب‌گذاری شوند [14]. علاوه براین، در صورت وجود برچسب نیز، عدم تعادل داده‌های عادی و ناهنجاری و دیگر مشکلات اشاره‌شده در قبل، یادگیری به انظارت را دچار مشکل می‌کند. به همین دلیل، در سال‌های اخیر بر روی روش‌های یادگیری بدون نظارت تمرکز شده است.

این روش‌ها نیز در عمل بسیار چالش‌برانگیز هستند. این روش‌ها باید سه بخش مهم را در نظر بگیرند. الف: از مدل‌هایی استفاده کنند که بتوانند بازنمایی‌های آموزنده‌ای از دینامیک زمانی و مکانی پیچیده داده‌های سامانه‌ها در زمان عملکرد عادی را با سرعت مناسب یاد بگیرند. ب: تابع امتیازدهی مناسب برای تبدیل خطای مدل به امتیاز ناهنجاری تعریف کنند. ج: از یک معیار ارزیابی مناسب استفاده کنند که بتواند عملکرد روش را برای داده‌های آینده‌ی سیستم مشخص کند. (در این خصوص در ادامه‌ی مقاله بیشتر بحث شده است.)

جدول(1). مزایا و معایب شبکه‌های عصبی در تشخیص ناهنجاری سری‌های زمانی چند متغیره

شبکه عصبی	معایب	مزایا
RNN [12]	انفجار گرادیان، ناپدید شدن گرادیان، سرعت‌پایین، هزینه‌ی بالای محاسبات، کاهش دقت در داده‌های با زمینه‌ی زمانی طولانی و پیچیده	توانایی مدل‌سازی زمینه‌ی زمانی داده‌های سری زمانی
LSTM [12]	پیچیدگی بیشتر نسبت به *RNN*، سرعت‌پایین، هزینه‌ی بالای محاسبات، کاهش دقت در داده‌های با زمینه‌ی زمانی پیچیده	توانایی مدل‌سازی زمینه‌ی زمانی داده‌های سری زمان ، حفظ اطلاعات در مدت‌زمان طولانی، مناسب برای یادگیری وابستگی‌های بلندمدت
CNN [12]	عدم توانایی در یادگیری وابستگی‌های بلندمدت، نیاز به پیش‌پردازش بیشتر به دلیل محدودیت در فیلترها، کاهش دقت در داده‌های با زمینه‌ی زمانی طولانی و پیچیده	توانایی مدل‌سازی زمینه‌ی زمانی داده‌های سری زمانی، استخراج ویژگی‌های محلی قوی، سریع‌تر در پردازش نسبت به*RNN* و *LSTM*
Transformer [12 ,14 ,16]	نیاز به داده‌های زیاد برای آموزش، پیچیدگی در انتخاب پارامترهای مناسب، نیاز به حافظه زیاد	قابلیت پردازش موازی داده‌ها و افزایش سرعت ، دقت خوب درزمینه‌ی زمانی طولانی و پیچیده

با توجه به فراوانی داده‌های عملکرد عادی سیستم‌ها، شبکه‌های عصبی عمیق، به‌ویژه شبکه‌های عصبی بازگشتی همچون *RNN*[1]، *LSTM*[2] و شبکه‌های پیچشی[3] به‌طور گسترده به‌عنوان مدل در این زمینه استفاده می‌شوند؛ زیرا آن‌ها با توجه به ساختارشان قادر به مدل‌سازی داده‌های سری زمانی تک متغیره هستند. بااین‌حال، مشکلات ناپدید شدن[4]، انفجار[5] گرادیان، سرعت‌پایین، هزینه‌ی بالای محاسبات در شبکه‌های عصبی بازگشتی و محدودیت‌های فیلترهای پیچشی باعث شده است که در مدل‌سازی روابط بلندمدت، مدل‌سازی دینامیک پیچیده‌ی زمینه‌ی زمانی و مکانی و مدل‌سازی وابستگی بین این دو زمینه در داده‌های سری زمانی چند متغیره، عملکرد مناسب را در تشخیص ناهنجاری نداشته باشند [14]. برای این‌که یک روش بتواند در تشخیص ناهنجاری سری‌های زمانی چند متغیره برای سیستم‌های مدرن عملکرد مناسبی داشته باشد؛ باید

[1] Recurrent Neural Network
[2] Long Short-Term Memory
[3] Convolutional Neural Network
[4] Vanishing Gradient
[5] Exploding Gradient

چالش‌های مربوط به پیچیدگی داده‌ها و سرعت استنتاج را حل کند.

شبکه‌های عصبی ترانسفورمر به دلیل ساختاری که دارند قادر به مدل‌سازی پیچیدگی زمینه‌ی زمانی و مکانی داده‌ها هستند. قدرت زیاد آن‌ها در مدل‌سازی پیچیدگی زمینه‌ی زمانی داده‌ها، ابتدا در پردازش زبان طبیعی ثابت شد [16]. مدل‌های زبانی بزرگ مانند *GPT-4*[1] باقدرت فهم قابل‌رقابت با انسان‌ها از این ساختار استفاده می‌کند [17] . به همین دلیل در پژوهش‌های جدید سعی شده است از ترانسفورمرها در داده‌های سری زمانی که مشابه داده‌های زبان طبیعی، نیاز به مدل‌سازی زمینه‌ی زمانی دارند، استفاده شود. در جدول (1) مزایا و معایب این ساختارها آورده شده است.در این مقاله از مدل مبتنی بر رمزگذار ترانسفورمر برای رفع این ضعف‌ها استفاده‌شده است. در این روش برخلاف شبکه‌های بازگشتی و پیچشی، داده‌ها به شیوه‌ی ترتیبی پردازش نمی‌شوند؛ درنتیجه مشکلات مربوط به گرادیان را ندارند؛ در عوض کل توالی داده‌ها پردازش می‌شود ومکانیسمنیزم توجه به خود برای یادگیری وابستگی‌های زمینه‌ی زمانی و مکانی در دنباله استفاده می‌کند که باعث افزایش سرعت استنتاج می‌شود [15]. همچنین ساختار این شبکه باعث می‌شود که برخلاف شبکه‌های قبلی، هیچ سوگیری محلی نداشته باشند؛ به این معنی که داده‌های دوردست، «فرصت برابر» با داده‌های فاصله کوتاه دارند. همچنین این ساختارها پتانسیل مدل‌سازی دینامیک پیچیده داده‌های سری زمانی چند متغیره رادارند. برای نمونه مدل‌های تشخیص ناهنجاری مبتنی بر ترانسفورمری که اخیراً توسط تولی و همکارانش [14] و لینگ و همکارانش [15] پیشنهادشده است، نتایج بهتری از مدل‌های قبلی دارد.

درروش پیشنهادی، علاوه بر مدل، یک تابع امتیازدهی ناهنجاری باهدف کاهش تشخیص اشتباه ناهنجاری نیز تعریف‌شده است که خطای بازسازی مدل را به امتیاز ناهنجاری تبدیل می‌کند. در این تابع، توزیع گوسی مناسب برای خطای بازسازی داده‌های آموزشی، برای هر متغیر به‌صورت جداگانه، به دست می‌آید؛ سپس برای داده‌های تست بر اساس این توزیع‌ها و خطاهای بازسازی، امتیاز ناهنجاری هر متغیر به دست می‌آید. با توجه به این‌که عملکرد غیرعادی سیستم در چند متغیر باعث ناهنجاری در کل سیستم می‌شود؛ درصدی از تعداد کل متغیرهایی که امتیاز ناهنجاری بالایی دارند، در ارزیابی نهایی آن داده، در نظر گرفته می‌شوند.

آزمایش‌ها بر روی مجموعه داده‌ی *SMD*[2] شامل داده‌های 28 سرور بخش‌های مختلف یک شرکت اینترنتی بزرگ انجام‌شده است که در آن‌ها، روش پیشنهادی نسبت به روش‌های پیشرفته اخیر در معیار ارزیابی امتیاز F ترکیبی[3]، بهبود 5 درصدی را نشان می‌دهد. همچنین روش پیشنهادی باعث بهبود سرعت یادگیری می‌شود. به‌طور خلاصه سهم علمی و نوآوری این مقاله به شرح زیر است:

1- فرموله کردن مسئله جهت استفاده از رمزگذار ترانسفورمر که به‌صورت یادگیری خود نظارت آموزش می‌بینند. 2- تعریف یک تابع امتیازدهی ناهنجاری جدید 3- انجام آزمایش بر روی مجموعه داده‌ی *SMD* با 28 سرور.

ساختار این مقاله به شرح زیر است. در بخش دوم، مروری بر روش‌های تشخیص ناهنجاری آورده شده است. در بخش سوم، مفاهیم پایه‌ی مدل و معیار ارزیابی مورداستفاده درروش پیشنهادی توضیح داده‌شده است. در بخش چهارم روش پیشنهادی معرفی‌شده و درنهایت در بخش پنجم نتیجه‌گیری و کارهای آینده ارائه‌شده است.

2. مروری بر روش‌های تشخیص ناهنجاری

تشخیص ناهنجاری در سری‌های زمانی از سال 1995 موردتوجه محققان بوده است. در روش‌های سنتی توزیع سری‌های زمانی را با استفاده از روش‌های مختلف مانند خوشه‌بندی *k-Mean*، ماشین‌های بردار پشتیبان[4] یا مدل‌های رگرسیون مدل‌سازی می‌کنند [18-22]. روش‌هایی که از شبکه‌های عصبی استفاده می‌کنند، از سه نظر قابل‌تحلیل هستند:

2-1. نحوه‌ی مدل‌سازی وابستگی بین متغیرها

بیشتر مدل‌های یادگیری عمیق، بین متغیرهای داده‌های سری زمانی چند متغیره، در هر مرحله‌ی زمانی ارتباط برقرار می‌کنند. این اطلاعات مکانی-زمانی، نه‌تنها زمینه‌ی زمانی بلکه همبستگی بین متغیرها را در نظر می‌گیرد.

یکی از این روش‌های مدل‌سازی همبستگی بین متغیرها، کاهش بعد است. در اکثر موارد، وضعیت یک سیستم در مقیاس بزرگ با استفاده از چند ویژگی مهم آن قابل تفسیر است. بنابراین، می‌توان با استخراج ویژگی‌های اصلی آن میزان محاسبات را کاهش داد. برای این منظور تعدادی از محققان [23-27] از انواع رمزگذارهای خودکار استفاده کرده‌اند. اما کاهش ابعاد باعث می‌شود که تشخیص علت ناهنجاری دشوار شود.

روش دیگر، استفاده از ساختار گراف است. یک گراف می‌تواند یک ساختار توپولوژیکی صریح را تعریف کند و رابطه‌ی علت و معلولی را بین متغیرهای مستقل بیاموزد. در سال‌های اخیر چندین روش [27, 28] که مکانیسم توجه را به شبکه‌های

[1] The latest version of Generative Pre-trained Transformers
[2] Server Machine Dataset
[3] Composite F-score
[4] Support Vector Machines

عصبی گراف اعمال می‌کنند، معرفی‌شده است. درروشی دیگر، دینگ و همکارانش [29] از یک توزیع گوسی چند متغیره برای تعریف همبستگی بین ویژگی‌های داده‌ها استفاده می‌کنند.

2-2. نحوه‌ی مدل‌سازی زمینه زمانی

تاریخچه‌ی یک توالی حاوی اطلاعات زیادی در مورد رفتار آن است که می‌تواند تغییرهای آینده را نشان دهد. استفاده از شبکه‌های عصبی بازگشتی، یکی از رایج‌ترین روش‌های مبتنی بر یادگیری عمیق است که زمینه‌ی زمانی داده‌ها را مدل‌سازی می‌کند. مدل به‌دست‌آمده، برای تشخیص الگوی توالی و پیش‌بینی مقادیر مورد انتظار استفاده می‌شود [29, 30]. تعدادی از محققان از رمزگذار خودکار [1] [22-26] یا رمزگذار خودکار متغیر [2] [31, 32] بر پایه‌ی ساختارهای شبکه‌ی عصبی بازگشتی عمیق برای بازسازی [3] نمونه‌ها استفاده می‌کنند.

اگرچه شبکه‌های عصبی بازگشتی، ساختار اصلی برای مدل‌سازی داده‌های سری زمانی است، گاهی اوقات شبکه‌های عصبی پیچشی عملکرد بهتری را در بعضی کاربردها نشان می‌دهد. این شبکه‌ها با داده‌های کوتاه‌مدت کار می‌کنند [33-35]. آن‌ها از لایه‌های پیچشی که دنبال هم قرارگرفته‌اند و هر لایه سطح بالاتری از ویژگی‌ها را می‌آموزد، تشکیل‌شده‌اند. علاوه بر این، برای این‌که بتوانند توابع غیرخطی را مدل‌سازی کنند و از این طریق ویژگی‌های پیچیده‌ی دنباله‌ها را ثبت کنند، از لایه‌های ادغام [4] بعد از لایه‌های پیچشی استفاده می‌کنند.

این شبکه‌ها، الگوهای زمینه‌ی زمانی را در سری‌های زمانی تقسیم‌بندی شده یاد می‌گیرند. به همین دلیل، مشکل آن‌ها این است که، درک رفتارهایی که در یک دوره طولانی‌مدت اتفاق می‌افتد، برای آن‌ها آسان نیست. برای حل این مشکل، شبکه‌های پیچشی زمانی [5] که نوعی از آن‌ها هستند [15, 36]، پیشنهادشده است. ویژگی متمایزکننده‌ی این شبکه‌ها از شبکه‌های پیچشی عادی این است که می‌توانند مانند شبکه‌های عصبی بازگشتی، دنباله‌ای با هر طول را به‌عنوان ورودی پردازش کنند.

یک سری شبکه‌های ترکیبی از شبکه‌های بازگشتی و پیچشی نیز وجود دارد که مدل‌سازی مکانی و زمانی را به‌طور هم‌زمان انجام می‌دهند. شی و همکاران [37]، یک مدل ترکیبی از شبکه‌ی بازگشتی *LSTM* و پیچشی برای حل مسئله‌ی پیش‌بینی توالی مکانی-زمانی پیشنهاد کردند. آن‌ها ضرب نقطه‌ای سلول *LSTM* را با عملگرهای پیچشی، جایگزین می‌کنند؛ درنتیجه، تمام دروازه‌ها و حالت‌های موجود در سلول به تنسورهای سه‌بعدی تغییر شکل می‌دهند که می‌توانند اطلاعات مکانی-زمانی را ضبط کنند.

خوش نویسان و همکاران [38] معماری دیگری معرفی می‌کنند که ترکیبی از رمزگذار خودکار و شبکه متخاصم مولد است که از ترکیب شبکه‌ی بازگشتی *LSTM* و پیچشی در بخش رمزگذار خود استفاده می‌کند و زمینه‌ی مکانی و زمانی داده‌ها را ضبط می‌کند.

مکانیسم توجه [6] روشی است که در بخش‌هایی از مدل‌های شبکه عصبی استفاده می‌شود. این روش با تمرکز بر اطلاعات مرتبط، به بهبود عملکرد آن‌ها کمک می‌کند و به مدل‌ها اجازه می‌دهد تا به‌طور انتخابی به بخش‌های مختلف سری زمانی ورودی توجه کنند و درجات مختلفی از اهمیت یا وزن را به هر نقطه‌ی زمانی اختصاص دهند. در ساختارهای عصبی جدید مانند ترانسفورمر [7]، مکانیسم توجه، به ابزار اصلی برای مدل‌سازی زمینه‌ی زمانی تبدیل‌شده است. این مدل‌های مبتنی بر توجه، می‌توانند وابستگی بسیار دوربرد و اهمیت نسبی هر نقطه‌ی داده در سری زمانی را بفهمند. دستاوردهای قابل‌توجه این ساختارها در پردازش زبان طبیعی، باعث شده است که در تشخیص ناهنجاری سری‌های زمانی هم نفوذ کنند. در سال‌های اخیر محققان زیادی [27, 39] امکانیسمزم توجه استفاده کرده‌اند.

حافظه‌ی زمانی سلسله مراتبی نیز ساختار عصبی است که برای مدل‌سازی زمانی در تشخیص ناهنجاری استفاده‌شده است [40]. این حافظه‌ها، برای تجسم ساختار و تعامل نورون‌های هرمی در نئوکورتکس طراحی‌شده است. آن‌ها شامل سلول‌های انباشته به شکل درخت هستند و ستون‌های سلول‌ها توسط ورودی و حالت‌های قبلی همسایگان متصل، فعال می‌شوند. این حافظه‌ها می‌توانند زمینه‌های زمانی را ضبط و پیش‌بینی کنند، بنابراین برای تشخیص ناهنجاری در داده‌های سری زمانی مفید هستند. این حافظه‌ها به‌طور مداوم الگوهای زمانی را از جریان داده‌ها، بدون پس انتشار یاد می‌گیرند؛ بنابراین، آن‌ها به حداقل مداخله انسانی نیاز دارند تا بدون نظارت آموزش داده شوند.

3-2. امتیاز ناهنجاری

مدل‌هایی که در بخش‌های قبل به آن‌ها اشاره شد، زمینه‌ی زمانی و وابستگی بین متغیرها در سری زمانی چند متغیره را به شیوه‌ای بدون نظارت یا نیمه نظارت‌شده، یاد می‌گیرند. پس از آموزش مدل‌ها، آن‌ها باید برای تشخیص عملکرد سامانه‌ها مورداستفاده قرار بگیرند. نتایج روش باید به‌صورت عددی بیان شود تا به درک وضعیت کمک کند. به این خروجی عددی، امتیاز ناهنجاری می‌گویند. میزان بزرگی آن، میزان غیرعادی (ناهنجار) بودنش را مشخص می‌کند.

[1] AutoEncoder
[2] Variational AutoEncoder
[3] Reconstruction
[4] Pooling layer
[5] Temporal Convolutional Networks
[6] Attention Mechanism
[7] Transformer

برای استخراج امتیاز ناهنجاری شبکه‌های عصبی رمزگذار خودکار، رمزگذار خودکار متغیر، شبکه‌ی متخاصم مولد [1] و ترانسفورمرها از خطاهای بازسازی استفاده می‌کنند. مدل‌های مبتنی بر رمزگذار خودکار مانند [26]، داده‌های ورودی را با استخراج ویژگی بازسازی می‌کنند. مدل‌های مبتنی بر رمزگذار خودکار متغیر [31, 32]، توزیع داده‌ها را تخمین می‌زنند و نمونه‌هایی را از آن تولید می‌کنند که بسیار شبیه به داده‌های ورودی هستند. مدل‌های مبتنی بر شبکه عصبی مولد [33, 35]، به‌صراحت نمونه‌هایی را تولید می‌کنند که تا حد امکان شبیه به داده‌های ورودی هستند.

در سال‌های اخیر، ساختار رمزگذار خودکار عمیق که در آن رمزگذار و رمزگشا از لایه‌های زیاد و متنوعی تشکیل‌شده است، در تحقیقات زیادی استفاده‌شده است. وانگ و همکاران از لایه‌های توجه در کنار لایه‌های دیگر برای ساخت رمزگذار و رمزگشا استفاده کرده‌اند [27]. به‌طور خاص، ژائو و همکاران [35] هر دو خطای پیش‌بینی و بازسازی را به‌طور مشترک در مدل خود در نظر گرفتند.

مدل‌های گفته‌شده در بالا، از شبکه‌های عصبی، روش‌های آموزش و توابع هدف متفاوت استفاده می‌کنند اما امتیازات ناهنجاری را به‌طور مشابه محاسبه می‌کنند. هدف آن‌ها بازسازی داده‌های ورودی است و خطای بین داده‌های ورودی و خروجی خود را اندازه‌گیری می‌کنند.

برای استخراج امتیازهای ناهنجاری از مدل پیش‌بینی دو راه وجود دارد. یک‌راه برچسب باینری است که بر اساس احتمال طبقه‌بندی نمونه ورودی به‌عنوان یک داده‌ی ناهنجاری، برچسب آن را تشخیص می‌دهد [41, 42]. راه دیگر، پیش‌بینی مقدار مورد انتظار برای زمان بعدی است [40]. در این حالت، خطای پیش‌بینی، باقیمانده‌ی بین مقدار مورد انتظار و مقدار پیش‌بینی‌شده است. روش دوم کاربردی‌تر از اولی است زیرا برچسب‌ها در دنیای واقعی ناکافی هستند.

برای استخراج امتیاز ناهنجاری در مدل‌هایی که توزیع داده‌های عادی را به‌دست‌آمده می‌آورند، معیارهای مبتنی بر عدم تشابه استفاده می‌شود. روش‌های مختلفی برای اندازه‌گیری شباهت ازجمله فاصله‌ی اقلیدسی، فاصله‌ی مینکوفسکی، شباهت کسینوس و فاصله‌ی ماهالانوبیس وجود دارد.

شن و همکارانش [43] با استفاده از شبکه‌ی سلسله مراتبی یک کلاسه‌ی زمانی [2] ، ویژگی‌های سری زمانی را استخراج می‌کنند؛ سپس با بردار ویژگی به‌دست‌آمده خوشه‌بندی انجام می‌دهند. برای هر نمونه‌ی تست، شباهت کسینوسی بین بردار ویژگی آن و خوشه‌ها اندازه‌گیری می‌شود؛ شباهت به‌دست‌آمده از عدد یک‌کم می‌شود و به‌عنوان نمره‌ی ناهنجاری استفاده می‌شود.

درروشی دیگر [44] ویژگی‌های سری زمانی توسط شبکه‌های پیچشی زمانی استخراج و توزیع آن‌ها با استفاده از یک مدل مخلوط گوسی [3] تخمین زده و سپس برای امتیاز ناهنجاری از فاصله‌ی ماهالانوبیس [4] استفاده می‌شود.

درنهایت، در همه‌ی روش‌ها زمانی که امتیاز ناهنجاری به‌دست‌آمده از یک آستانه‌ی خاص فراتر می‌رود، نقطه‌ی زمانی مربوطه به‌عنوان ناهنجاری تعیین می‌شود. درگذشته کارشناس مربوطه، دامنه‌ی این آستانه را به‌صورت تجربی تعیین می‌کرد اما اکنون با توجه به نتیجه‌ی آموزش مدل، تصمیم‌گیری می‌شود. برخی از مدل‌ها [31] از یک آستانه‌ی تطبیقی [5] استفاده می‌کنند که به‌طور مداوم با تغییرات داده‌ها در طول زمان تنظیم می‌شود.

3. مفاهیم پایه

در این بخش توضیح مختصری در خصوص شبکه‌ی عصبی ترانسفورمر و معیارهای ارزیابی روش‌های تشخیص ناهنجاری داده می‌شود.

3-1. ترانسفورمر

مدل ترانسفورمر یک معماری شبکه‌ی عصبی است که در ابتدا برای پردازش زبان طبیعی [6] طراحی‌شده است [16]. این مدل با استفاده از مکانیسم توجه، ارتباط بین کلمات در یک جمله یا پاراگراف ورودی خود را به‌طور مستقیم مدل‌سازی می‌کند. به‌این‌ترتیب بدون نیاز به یک حافظه طولانی‌مدت، ترتیب کلمات ورودی و ارتباط آن‌ها به‌دست‌آمده می‌آید. اولین قدم این است که کلمات به شکل ورودی مناسب برای این مدل تبدیل شوند، بدین منظور کلمات ورودی به بردارهای عددی معنایی با بعد d_{model} تبدیل می‌شوند. این کار را الگوریتم‌های مختلف تعبیه‌ی کلمات [7] انجام می‌دهند.

با توجه به این‌که مدل ما حاوی هیچ بازگشت و پیچشی نیست، برای این‌که مدل از ترتیب کلمات ورودی استفاده کند باید اطلاعاتی در مورد موقعیت نسبی یا مطلق کلمات در دنباله به ورودی اضافه شود. برای این منظور بردار عددی با عنوان، "رمزگذاری موقعیت [8]" با بردار عددی تعبیه‌ی کلمات ورودی جمع می‌شود. بعد این دو بردار باهم برابر است که در ادامه آن را با d_{model} نشان می‌دهیم. رمزگذارهای موقعیت متنوعی در تحقیقات مختلف مورداستفاده قرارگرفته است که دسته‌ای از آن‌ها به‌عنوان پارامترهای مدل یاد گرفته می‌شوند و دسته‌ی دیگر

[1] Generative Adversarial Networks
[2] Temporal Hierarchical One Class
[3] Gaussian mixture model
[4] Mahalanobis
[5] Adaptive Threshold
[6] Natrual Language Proccessing
[7] Word Embedding
[8] Positional encoding

اعداد ثابتی هستند که با توجه به موقعیت کلمه در جمله محاسبه می‌شوند.

2-3. معماری

این مدل از دو بخش رمزگذار[1] و رمزگشا[2] تشکیل‌شده است. هر دو بخش از پشته‌ای از N بلوک یکسان تشکیل‌شده‌اند. هر بلوک دارای دولایه است، اولی لایه‌ی خود توجه چند سر[3] و دومی یک شبکه پیش‌خور ساده‌ی کاملاً متصل[4] است. یک اتصال مستقیم از ورودی هر بلوک به خروجی آن وجود دارد که به آن اتصال باقیمانده[5] می‌گویند. در انتهای هر بلوک نیز از یک نرمال ساز لایه استفاده می‌شود. خروجی هر بلوک از رابطه‌ی زیر محاسبه می‌شود:

$$output = Layer\ Norm(x+SubLayer(x)), \qquad (1)$$

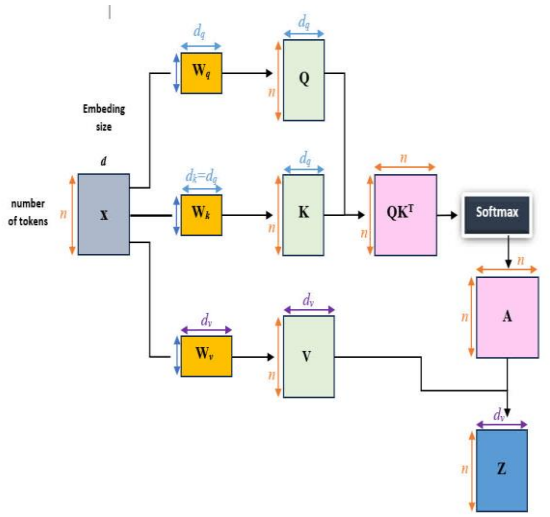
که در آن $SubLayer(x)$ تابعی است که توسط هر لایه روی ورودی آن لایه (x) انجام می‌شود. برای این‌که اتصالات بین لایه‌های هر بلوک و اتصال باقیمانده بتواند به‌راحتی اعمال شود، بعد خروجی تمامی لایه‌ها برابر با بعد ورودی d_{model} است.

1-2-3. لایه‌ی خود توجه چند سر

لایه‌ی توجه[6]: این لایه به مدل اجازه می‌دهد تا روابط بین اجزای مختلف یک دنباله ورودی را درک کند، آن دنباله می‌تواند کلمات در یک جمله، سری زمانی یا هر نوع توالی دیگر باشد. معماری این لایه به شکلی است که به مدل قدرت پردازش کل توالی را در یک مرحله می‌دهد.

این لایه، بردارهای ورودی با عنوان پرس‌وجو[7] (q)، جفت‌های کلید[8] (k) و مقدار[9] (v) را دریافت می‌کند. خروجی آن برابر مجموع وزنی بردارهای v است که وزن اختصاص داده‌شده به هر بردار v برابر میزان شباهت بردار q با بردار k متناظر آن v است. علت استفاده از کلمات پرس‌وجو، کلید و مقدار برای ورودی‌های این لایه این است که مفهوم آن را در دنیای واقعی برساند. برای نمونه، اگر هدف، جستجویی در یوتیوب یا گوگل باشد، متنی که در کادر جستجو تایپ می‌شود پرس‌وجو ، نتایجی که نشان داده می‌شوند، کلید و محتوای داخل آن‌ها مقدار نامیده می‌شود. برای یافتن بهترین مقدارهای منطبق با پرس‌وجوی مدنظر، باید شباهت بین آن پرس‌وجو و کلید متناظر آن مقدار را پیدا کرد.

در نوع خاصی از این لایه که در ترانسفورمر وانیلی استفاده می‌شود، q و k دارای بعد d_k و v دارای بعد d_v هستند. برای اندازه‌گیری میزان شباهت q و k از ضرب نقطه‌ای آن‌ها استفاده می‌شود. برای مقیاس‌پذیری، نتیجه‌ی این ضرب نقطه‌ای بر جذر اندازه‌ی این بردارها یعنی $\sqrt{d_k}$، تقسیم و تابع سیگموئید بر روی آن اعمال می‌شود.



شکل(1). روابط داخل لایه‌ی توجه

در عمل، لایه‌ی توجه روی مجموعه‌ای از پرس‌وجوها به‌طور هم‌زمان اعمال می‌شود. در بلوک اول رمزگذار، همان‌طور که در شکل (1) دیده می‌شود، این مجموعه از بردارهای عددی معادل کلمات یک جمله یا پاراگراف ورودی تشکیل می‌شود که پس از ضرب‌در یک بردار وزن w_q، در یک ماتریس Q بسته‌بندی‌شده‌اند. کلیدها و مقادیر هم، همان بردارهای عددی معادل کلمات ورودی هستند که پس از ضرب‌در بردارهای وزن w_k و w_v، در ماتریس‌های K و V بسته‌بندی می‌شوند. ماتریس خروجی این لایه، همان‌طور که در شکل (1) نشان داده‌شده است، با رابطه‌ی (2) محاسبه می‌شود:

$$Z = Attention(Q,K,V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V. \qquad (2)$$

همان‌طور که گفته شد، هر سه بردار پرس‌وجو، کلید و مقدار از ورودی رمزگذار که کلمات هستند، ساخته می‌شوند، ازاین‌رو یکسان هستند. در بلوک‌های بعدی نیز آن‌ها برابر خروجی بلوک قبلی و باهم برابر هستند. به همین دلیل به این نوع خاص توجه، توجه به خود[10] می‌گویند. توجه چند سر، از چندلایه توجه به خود تشکیل‌شده که به‌طور موازی باهم قرار دارند و ورودی‌های یکسان Q، K و V را طبق رابطه‌ی (3) به خروجی‌های مستقل $head_i$ نگاشت می‌کنند. سپس خروجی‌های آن‌ها، همان‌طور که در قسمت دوم رابطه‌ی (3) دیده می‌شود، به هم متصل[11] می‌شوند. به‌طور شهودی، سرهای توجه متعدد اجازه می‌دهد تا به ورودی با دیده‌ای متفاوتی توجه شود. برای مثال یکسر، وابستگی‌های طولانی‌مدت و سر دیگر وابستگی‌های کوتاه‌مدت را مدل‌سازی می‌کند.

[1] Encoder
[2] Decoder
[3] Multi head self attention
[4] Fully Connected Feed Forward Network
[5] Residual
[6] Attention
[7] Query (q)
[8] Key(k)
[9] Value(v)
[10] Self Attention
[11] Concat

$$MultiHead\,(Q,K,V) = Concat\,(head_1,\ldots,head_h), \quad (3)$$

$$where\ head_i = Attention\,(QW_i^Q, KW_i^K, VW_i^V),$$

که ماتریس‌های W_i^Q و W_i^K و W_i^V در رابطه‌ی (3) پارامترهایی هستند که در هنگام آموزش شبکه تنظیم می‌شوند. ابعاد این ماتریس‌ها در رابطه‌ی (4) آمده است:

$$W_i^Q\ and\ W_i^K \in R^{d_{model}\times d_k}, \quad (4)$$

$$W_i^V\ and\ W_i^O \in R^{d_{model}\times d_v},$$

که d_k و d_v از رابطه‌ی (5) به دست می‌آیند:

$$d_k = d_v = d_{model}/h. \quad (5)$$

با توجه به کاهش ابعاد هر سر، کل هزینه‌ی محاسباتی مشابه هزینه یک‌لایه‌ی توجه با ابعاد کامل است.

3-3. معیارهای ارزیابی روش‌های تشخیص ناهنجاری

روش‌های تشخیص ناهنجاری بعد از محاسبه‌ی امتیاز ناهنجاری، به یک آستانه برای تشخیص نیاز دارند. برخی از روش‌ها، تمام آستانه‌های ممکن را با استفاده از ناحیه‌ی زیر منحنی دقت-یادآوری[1] یا مشخصه‌ی عملکرد گیرنده[2] ارزیابی می‌کنند. اما در کاربردهای عملی تشخیص ناهنجاری، مقدار بالای معیار امتیاز F [3]مهم‌تر است. در کاره‌ای گذشته چند نوع امتیاز F برای تشخیص ناهنجاری تعریف‌شده است:

امتیاز F نقطه‌ای[4]: این امتیاز برای تشخیص ناهنجاری نقطه‌ای مناسب است، اما در عمل، در داده‌های سری زمانی تشخیص رویدادهای ناهنجار اهمیت بیشتری دارد. رویدادهای ناهنجار، مجموعه‌ی پیوسته از نقاط زمانی با عملکرد غیرعادی در سامانه‌ها هستند. تعدادی از محققان [22, 29] برای ارزیابی روش خود از این معیار استفاده می‌کنند که همان امتیاز F معروف است. در هنگام محاسبه‌ی این امتیاز از پیش‌بینی‌های روش موردنظر استفاده می‌شود.

GT	000111111001000111111100111
Pred	100001000000000011000000
PA	100111111000000111111100000

شکل (2). نمونه‌ی پیش‌بینی تنظیم‌شده با نقطه [45]

امتیاز F تنظیم‌شده با نقطه[5]: تنظیم نقطه[6]، پروتکلی است که پیش‌بینی‌ها را قبل از محاسبه‌ی معیارهای عملکرد، تنظیم می‌کند. به این صورت که اگر حداقل یک نقطه از یک رویداد ناهنجار، به‌عنوان ناهنجاری پیش‌بینی شود، همه‌ی پیش‌بینی‌های آن رویداد را به‌عنوان ناهنجاری در نظر می‌گیرد. امتیاز Fی که با این پیش‌بینی‌ها به دست می‌آید را امتیاز F تنظیم‌شده با نقطه می‌گویند [46]. ایده‌ی این پروتکل این است که الگوریتمی که یک هشدار را برای نقطه‌ای از یک رویداد ناهنجار تولید می‌کند، در عمل برای واکنش به‌موقع به آن رویداد کافی است. شکل (2) پروتکل تنظیم نقطه را نشان می‌دهد که برچسب اصلی نمونه‌ای از داده‌های یک سری زمانی در ردیف GT و پیش‌بینی‌های یک مدل برای آن سری زمانی در ردیف $Pred$ آمده است. مقادیر پیش‌بینی‌های مدل پس از پروتکل تنظیم نقطه نیز در PA آمده است. همان‌طور که دیده می‌شود تعداد تشخیص درست ناهنجاری از $Pred$ به PA از 4 به 11 افزایش‌یافته است. امتیاز F نیز، برای $Pred$ برابر 32 درصد است اما در PA به 58 درصد می‌رسد. اختلاف قابل‌توجه بین این دو عدد، نگرانی‌هایی را در مورد کارایی تحقیقاتی که از این معیار ارزیابی استفاده می‌کنند، ایجاد کرده است [45]. به‌عنوان‌مثال، کیم و همکارانش [47] ادعا می‌کنند که اگر از این معیار ارزیابی استفاده شود، تشخیص ناهنجاری تصادفی از تعدادی از آخرین روش‌های تشخیص ناهنجاری بهتر عمل می‌کند. این معیار رویدادها را در نظر می‌گیرد، اما حتی زمانی که تعداد زیادی از رویدادها شناسایی نمی‌شوند، امتیاز بالایی را ارائه می‌کند.

یک روش تشخیص ناهنجاری ایدئال برای داده‌های سری زمانی این است که حداقل یک نقطه زمانی را در هر رویداد تشخیص دهد و هیچ نقطه‌ی با عملکرد عادی را به‌عنوان ناهنجاری (FP[7]) تشخیص ندهد. بنابراین، یک معیار خوب برای تشخیص ناهنجاری باید الف: برای نقاط، دقت[8] بالا و برای رویدادها، یادآوری[9] بالا داشته باشد، ب: یک نشانه مفید از این‌که چقدر باحالت ایدئال فاصله دارد، ارائه دهد. معیار امتیاز F ترکیبی [1]، این موارد را به‌خوبی نشان می‌دهد. این معیار میانگین هارمونیک دقت نقطه‌ای و یادآوری رویداد است که نحوه‌ی محاسبه‌ی آن‌ها در رابطه‌ی (6) آمده است:

$$Precision_t = \frac{TP_t}{TP_t + FP_t}, \quad (6)$$

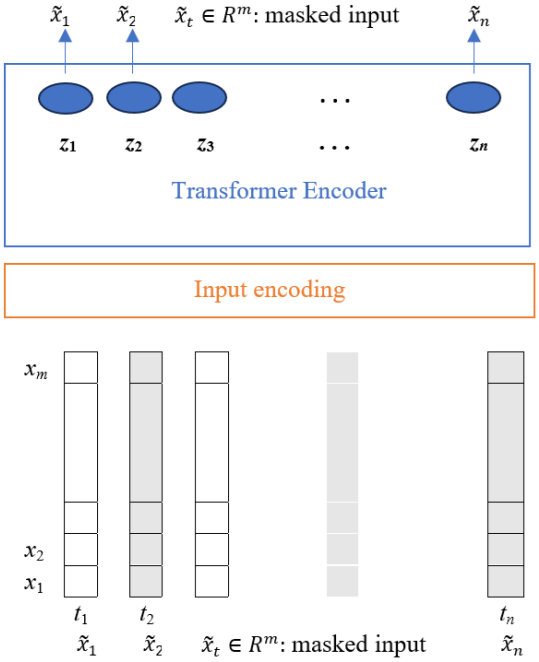
$$Recall_e = \frac{TP_e}{TP_e + FN_e},$$

$$Compositr\ F\text{-}Score = 2 \times \frac{Precision_t \times Recall_e}{Precision_t + Recall_e}.$$

[1] AUPRC (Area under the Precision - Recall graph)
[2] AU-ROC (Area under the receiver operating characteristic)
[3] F-score
[4] Point-wise F-score
[5] Point-Adjusted F-score
[6] Point Adjustment
[7] False Positive
[8] Precision
[9] Recall

4. روش پیشنهادی

همان‌طور که در بخش 2 توضیح داده شد هر روش تشخیص ناهنجاری سری‌های زمانی چند متغیره شامل سه بخش اصلی است؛ مدل‌سازی همبستگی بین متغیرها، مدل‌سازی زمینه‌ی زمانی و استخراج امتیاز ناهنجاری. در این مقاله روش‌های جدیدی برای هر سه بخش ارائه‌شده است.



شکل (3). مدل پیشنهادی

1.4. مدل‌سازی هم‌زمان همبستگی بین متغیرها و زمینه‌ی زمانی

در هسته روش پیشنهادی، از یک مدل مبتنی بر رمزگذار ترانسفورمر ، برای مدل‌سازی هم‌زمان همبستگی بین متغیرهای سری زمانی، زمینه زمانی آن‌ها و ارتباط این دو باهم استفاده‌شده است. یک نمودار شماتیک از این مدل در شکل (3) ارائه‌شده است. در ادامه تغییراتی که در رمزگذار ترانسفورمر معرفی‌شده در بخش مفاهیم، اعمال‌شده تا برای به‌کارگیری در داده‌های سری زمانی چند متغیره مناسب باشد، شرح داده می‌شود. همان‌طور که در شکل (3) دیده می‌شود، هر نمونه آموزشی $X \in R^{m \times n}$ ، یک سری زمانی چند متغیره به طول n و با m متغیر مختلف است که نرمال شده‌اند که یک پنجره زمانی با طول مشخص n را تشکیل می‌دهد. برای ورود این نمونه‌های آموزشی به مدل، استفاده از بردار $x_t \in R^m$ در هر نقطه‌ی زمانی t، به‌جای بردار تعبیه کلمات در کاربرد پردازش زبان طبیعی رمزگذار پیشنهاد می‌شود. درنتیجه، طبق رابطه‌ی پیشنهادی (7) تعداد متغیرهای سری زمانی چند متغیره به‌عنوان بعد مدل مورداستفاده قرار می‌گیرد:

$$d_{model} = m, \qquad (7)$$

سپس اطلاعات مربوط به ترتیب داده‌ها در سری زمانی با رمزگذار موقعیت ثابت با فرمول‌های رابطه (8) به بردار d_{model} تبدیل‌شده و به بردارهای هر نقطه‌ی زمانی اضافه می‌شود [16]. خروجی این حاصل جمع به‌عنوان ورودی به مدل داده می‌شود.

$$PE_{(pos,2i)} = sin\left(\frac{pos}{10000^{2i/d_{model}}}\right), \qquad (8)$$

$$PE_{(pos,2i+1)} = cos\left(\frac{pos}{10000^{2i/d_{model}}}\right),$$

که pos مکان و i بعد بردار خروجی این رمزگذار موقعیت است که باید با بعد d_{model} برابر باشد.

1-1-4. تابع هدف

آموزش مدل ارائه‌شده، به‌صورت یادگیری خود نظارت است، که بعد از انجام آن، مدل، همبستگی بین متغیرهای سری زمانی، زمینه زمانی آن‌ها و ارتباط این دو باهم را یاد می‌گیرد. در تابع هدف این مدل، پیش‌بینی نقاط زمانی پوشانده شده[1]ی داده‌های عملکرد عادی سیستم لحاظ شده است.

برای تولید نمونه‌های آموزشی، همان‌طور که در شکل (3) نشان داده‌شده است از داده‌های با عملکرد عادی سیستم، پنجره‌های زمانی با طول n با هم‌پوشانی l_{ov} در نظر گرفته‌شده، که در هر پنجره mc تا از نقاط زمانی به‌طور تصادفی انتخاب و کل متغیرهای نقاط زمانی انتخاب‌شده با یک بردار پوشانده می‌شوند. این بردار m بعدی ثابت، از اعداد تصادفی بین صفر و یک تشکیل‌شده است. پیش‌بینی مقادیر درست این نقاط زمانی پوشانده شده، طبق رابطه‌ی (9) به‌عنوان هدف مدل ارائه‌شده، لحاظ می‌شود. به‌این‌ترتیب از داده‌های بدون برچسبی که در حجم زیاد وجود دارد، نمونه‌های آموزشی زیادی تولید می‌شود. l_{ov}, n ,mc، فرا پارامترهایی هستند که در آموزش تنظیم می‌شوند. تابع هدف ارائه‌شده برای مدل پیشنهادی در رابطه‌ی (9) آورده شده است:

$$L_{mse} = \frac{1}{mc \times m}\sum_{i=1}^{mc}\sum_{j=1}^{m}\left(\hat{y}(mask_i, j) - x(mask_i, j)\right)^2. \qquad (9)$$

در رابطه‌ی بالا $mask_i$، اندیس iامین نقطه‌ی زمانی پوشانده شده را مشخص می‌کند. ازآنجایی‌که مقادیری که باید پیش‌بینی شود، در بازه‌ی $[0,1]$ قرار دارد، از تابع فعال‌سازی سیگموئید در خروجی هر لایه برای این مدل استفاده‌شده است.

با توجه به ساختار این مدل، در داده‌های داخل پنجره‌ی زمانی هر نمونه‌ی آموزشی و تست، ارتباط هر نقطه‌ی زمانی با هر دو طرفش یعنی هم‌زمان‌های قبل از آن و هم‌زمان‌های بعدازآن،

[1] mask

در نظر گرفته می‌شود. به‌این‌ترتیب نگاه عمیق‌تری به دینامیک زمانی آن انجام می‌شود.

4-2. تابع امتیاز ناهنجاری پیشنهادی

ازآنجایی‌که با توجه به ذات داده‌های سری زمانی چند متغیره، پنجره‌های زمانی با هم‌پوشانی زیاد برای آموزش استفاده‌شده‌اند، مدل، توانایی بازسازی داده‌های با عملکرد عادی سیستم را پیدا می‌کند و می‌توان از خطای بازسازی داده‌های تست برای تشخیص ناهنجاری استفاده کرد. در این مرحله مطابق توضیحات بخش 2-3 باید یک تابع امتیاز ناهنجاری تعریف شود که خطای بازسازی داده‌ی تست را به یک عدد نگاشت کند.

گارگ و همکارانش [1] برای نگاشت خطاهای آموزشی به امتیاز ناهنجاری، یک توزیع گاوسی روی آن‌ها برازش داده و یک امتیاز بر اساس آن به دست آوردند. استفاده مستقیم از توابعی که روی تابع توزیع احتمال [1] تعریف می‌شوند، مانند لگاریتم آن، امتیاز بالایی در هر دو دنباله توزیع به نقاط می‌دهد. این بدان معنی است که حتی نقاطی با خطای بسیار کم به‌عنوان ناهنجاری طبقه‌بندی می‌شوند. آن‌ها برای جلوگیری از این مشکل، این امتیاز را توسط تابعی از تابع توزیع تجمعی [2] مانند $-log(1-cdf)$ به دست آوردند. این تابع با افزایش خطا به‌طور یکنواخت افزایش می‌یابد. سپس برای محاسبه‌ی امتیاز ناهنجاری نهایی در داده‌های سری زمانی چند متغیره، برای هر متغیر، توزیع و امتیاز جداگانه‌ای طبق رابطه‌ی (10) به دست آورده، درنهایت برای به دست آوردن تابع امتیاز a_t از جمع امتیاز متغیرها استفاده کردند [1]. همان‌طور که در بالا توضیح داده شد تابع ϕ در رابطه‌ی (10) همان تابع توزیع تجمعی است:

$$A_t^i = -\log\left(1-\phi\left(\frac{Er_t^i-\hat{\mu}^i}{\hat{\sigma}^i}\right)\right), \qquad (10)$$

$$a_t = \sum_{i=1}^{m} A_t^i.$$

با توجه به‌این‌علت ناهنجاری براثر عملکرد غیرعادی در چند متغیر اتفاق می‌افتد، استفاده از جمع امتیاز متغیرها باعث می‌شود که تعدادی از نقاط با عملکرد عادی، به‌اشتباه به‌عنوان نقاط ناهنجاری تشخیص داده شوند. برای مثال، جمع خطای ده متغیر با خطای 0/01 که برای هر متغیر، خطای کوچکی است و نمی‌تواند به علت ناهنجاری رخ‌داده باشد، برابر 0/1 است که اگر برای یک متغیر رخ دهد، می‌تواند نشانه‌ی ناهنجاری باشد. به همین دلیل درروش پیشنهادی برای محاسبه‌ی امتیاز نهایی a_t، به‌جای جمع امتیاز همه‌ی متغیرها، از میانگین امتیاز k تا از متغیرها که خطای بازسازی ماکزیمم دارند، مطابق رابطه‌ی (11) استفاده‌شده است:

$$AS_t = sort\,(A_t),$$

$$a_t = \sum_{i=1}^{k} AS_t^i. \qquad (11)$$

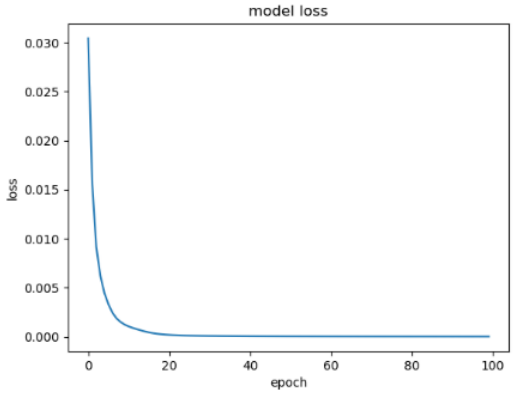
5. نتایج و اعتبارسنجی

برای ارزیابی روش پیشنهادی، از مجموعه‌های داده‌ی معیار *SMD* استفاده‌شده است. این داده‌ها طی 5 هفته از یک شرکت بزرگ اینترنتی جمع‌آوری‌شده است. این شرکت شامل 28 بخش است که داده‌های سرورهای آن‌ها به‌طور منظم در هر دقیقه نمونه‌برداری می‌شوند. این داده‌ها شامل 38 متغیر از ویژگی‌های مختلف سخت‌افزاری و نرم‌افزاری سرورها هستند. در هر سرور نصف داده‌ها که در آن‌ها عملکرد سرورها عادی بوده، برای آموزش مدل استفاده می‌شود و نصف دیگر که شامل عملکرد عادی و ناهنجاری سرورهاست، برای تست تشخیص ناهنجاری استفاده می‌شود. این مجموعه داده برای ارزیابی روش‌های تشخیص ناهنجاری به‌صورت عمومی منتشرشده است.

جدول (2). داده‌های سرور *machine_1_1* و *machine_3_1*

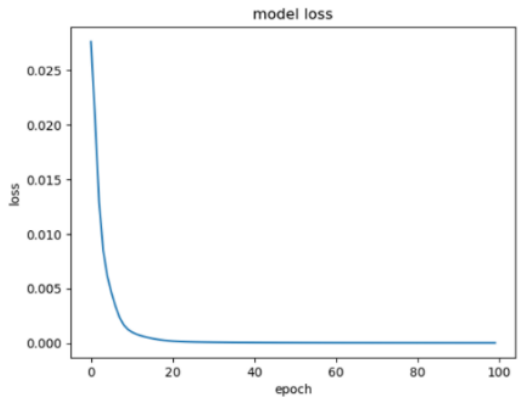
تعداد	***machine_1_1***	***machine_3_1***
داده‌های آموزش	28713	28479
داده‌های تست	28713	28479
تعداد نقاط ناهنجاری در داده‌های تست	303	2694
تعداد رویداد ناهنجار در داده‌های تست	4	8

در ابتدا برای داده‌های هر سرور، نرمال‌سازی کمینه-بیشینه [3] برای هر کانال به‌صورت جداگانه، در داده‌های آموزش انجام‌شده است و داده‌های آموزش در محدوده [0، 1] فشرده‌شده است، همچنین داده‌های تست برای جلوگیری از مقادیر بیش‌ازحد بزرگ یا کوچک در محدوده‌ی $[min-4, max+4]$ از هر کانال خاص محدود می‌شود. این اقدامات می‌تواند به کاهش نویز کمک کند.



شکل (4). خطای آموزش مدل مربوطه به *machine_1_1*

[1] Probability Distribution Function(pdf)
[2] Cumulitive Distribution Function(cdf)
[3] Min-Max



شکل (5). خطای آموزش مدل مربوطه به *machine*_3_1

جدول (3). فرا پارامترهای مدل‌های مربوطه به *machine*_1_1 و *machine*_3_1

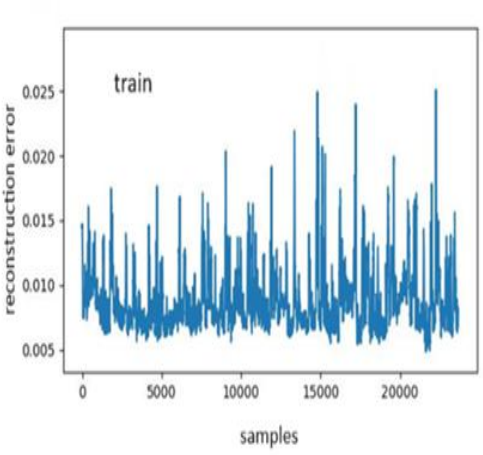
فرا پارامتر	***machine*_1_1**	***machine*_3_1**
اندازه‌ی پنجره‌ی ورودی رمزگذار	32	64
تعداد لایه‌ها در رمزگذار	1	2
تعداد سر[1] در هر لایه رمزگذار	4	8

ازآنجایی‌که در این مجموعه داده، 28 سرور مربوط به بخش‌های مختلف یک شرکت بزرگ هستند که هرکدام مستقل از دیگری است، برای هرکدام یک مدل، آموزش‌دیده شده است که با توجه به تعداد و پیچیدگی داده‌های آن سرور، فرا پارامترهای بهینه‌ی مربوط به ساختار مدل پیشنهادی، متفاوت است. برای مثال، تعداد داده‌های آموزش و تست، نقاط و رویدادهای ناهنجار در داده‌های تست مربوط به سرورهای *machine*_1_1 و *machine*_3_1 در جدول (2) و فرا پارامترهای بهینه‌ی مدل آن‌ها در جدول (3) آورده شده است. همچنین شکل (4) و (5) خطای آموزش این مدل‌ها را نشان می‌دهد. همان‌طور که در شکل‌ها مشخص است در هر دو مدل خطای آموزش با افزایش تعداد گام‌های آموزش کاهش می‌یابد.

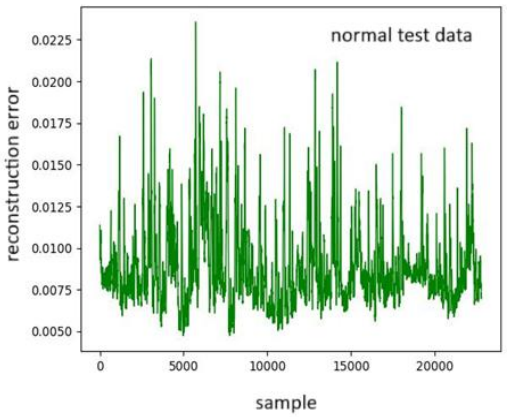
همچنین برای اطمینان از آموزش درست و عدم بیش برازش مدل‌ها، میانگین خطای بازسازی کل پنجره زمانی برای داده‌های آموزش و تست موردبررسی قرار گرفت.

همان‌طور که در شکل (6) مشخص است، میانگین خطای بازسازی داده‌های آموزش سرور *machine*_1_1 که تمام آن‌ها از کلاس عادی هستند، با میانگین خطای بازسازی داده‌های کلاس عادی تست که در شکل (7) آمده است، در یک محدوده است که نشان می‌دهد مدل درست آموزش‌دیده است و برای داده‌های نادیده[2] به‌درستی عمل می‌کند.

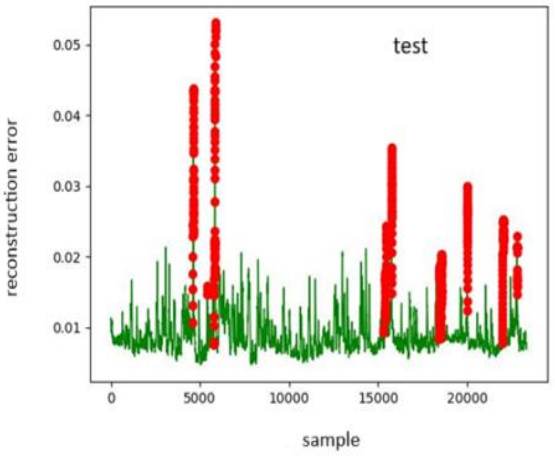
[1] Head
[2] Unseen Data



شکل (6). خطای بازسازی نمونه‌های آموزش *machine*_1_1

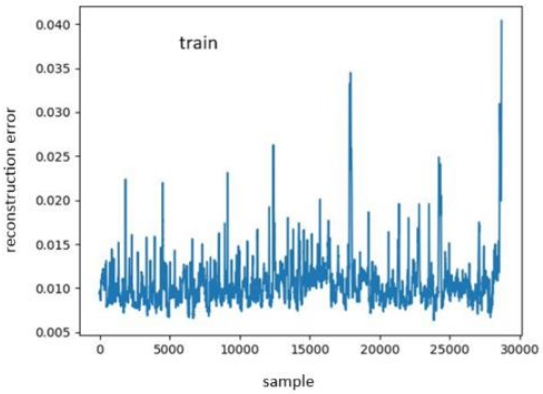


شکل (7). خطای بازسازی نمونه‌های با عملکرد عادی تست *machine*_1_1

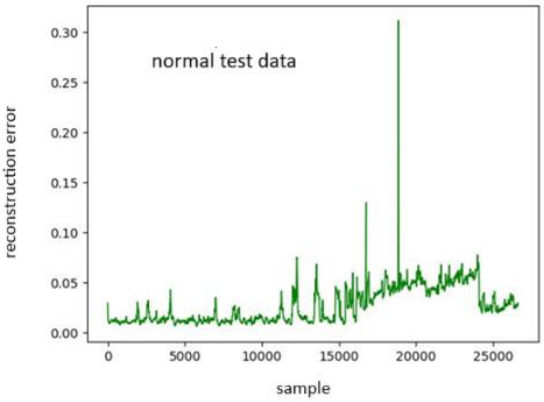


شکل (8). خطای بازسازی کل نمونه‌های تست *machine*_1_1

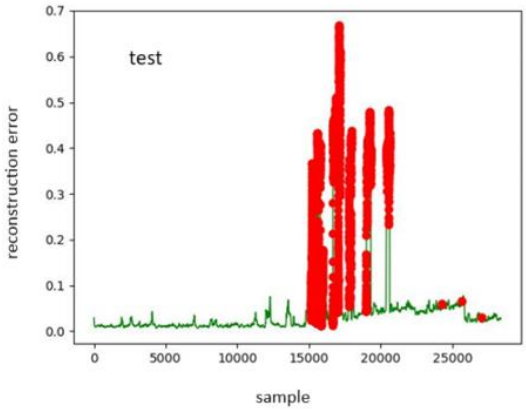
در شکل (8) خطای بازسازی کل داده‌های تست این سرور آورده شده است. نقاط قرمز مربوط به میانگین خطای بازسازی نمونه‌های کلاس ناهنجاری تست و خطوط سبزرنگ مربوط به نمونه‌های کلاس عادی آن هستند. همان‌طور که در شکل دیده می‌شود، با خطای بازسازی می‌توان داده‌های این دو کلاس را به‌خوبی از هم جدا کرد.



شکل (9). خطای بازسازی نمونه‌های آموزش *machine*_3_1



شکل (10). خطای بازسازی نمونه‌های عادی تست *machine*_3_1



شکل (11). خطای بازسازی همه‌ی نمونه‌های تست *machine*_3_1

در شکل‌های (9) تا (11) این خطاهای بازسازی برای سرور دیگر این مجموعه داده معیار بانام *machine*_3_1 آمده است.

همان‌طور که در بخش 2-3 توضیح داده شد، بعد از آموزش مدل باید امتیاز ناهنجاری برای هر نمونه محاسبه شود. درروش پیشنهادی، از تابع پیشنهادی بخش 4-3 برای تبدیل خطای بازسازی به امتیاز ناهنجاری استفاده‌شده است. برای محاسبه‌ی معیار ارزیابی از روش انتخاب حد آستانه‌ی *topk* [1] استفاده‌شده است. همچنین معیار ارزیابی این مجموعه داده، میانگین مقدار آن در 28 سرور موردنظر است. در جدول (4) روش پیشنهادی (*SSTEAD*[1]) با آخرین روش‌های پیشرفته مقایسه شده است. همان‌طور که دیده می‌شود روش پیشنهادی در معیار ارزیابی امتیاز *F* ترکیبی، از روش‌های پیشرفته اخیر بهتر عمل می‌کند. این روش نسبت به بهترین روش پایه، 5 درصد بهبود ایجاد می‌کند.

جدول (4). مقایسه‌ی امتیاز *F* ترکیبی

روش‌ها	*SMD*
UAE [1]	%55
OCAN [48]	%46
LSTM AE [49]	٪52
TCN AE [۳۶]	%58
LSTM VAE [۳۱]	%54
BeatGAN [۳۵]	%54
NASA LSTM [۳۰]	%38
DAGMM [۵۰]	%0.01
OmniAnomaly [۵۱]	%50
OCAN [۴۸]	%46
TranAD [۱۴]	٪53
DTAAD [۱۵]	%56
SSTEAD	**%63**

جدول (5). مقایسه‌ی زمان آموزش مدل‌ها

روش	زمان(ثانیه بر تکرار)
NASA-LSTM	373
DAGMM	204
OmniAnomaly	276
TranAD	43.6
DTAAD	80
SSTEAD	36

در آزمایش‌های انجام‌شده، زمان یادگیری مدل پیشنهادی و آخرین روش‌های پیشرفته بر روی یک سخت‌افزار نیز اندازه‌گیری شد. این زمان، میانگین زمان موردنیاز برای آموزش یک تکرار در 28 سرور است. همان‌طور که در جدول (5) آورده شده است، مدل پیشنهادی به دلیل ساختاری که دارد، سرعت بهتری نیز دارد. این مدل سرعت را نسبت به سریع‌ترین مدل پایه 15 درصد بهبود می‌بخشد.

6. نتیجه‌گیری

در این مقاله، در جهت رفع چالش‌های کمبود داده‌ی بر چسب‌دار و روابط پیچیده‌ی بین نقاط زمانی و متغیرهای داده‌های سری زمانی چند متغیره، یک مدل تشخیص ناهنجاری بر اساس رمزگذار ترانسفورمور طراحی شد و با یادگیری خود نظارت آموزش داده شد. با توجه به این‌که ترانسفورمرها برای پردازش زبان طبیعی طراحی‌شده‌اند، داده‌ی ورودی آن‌ها جملات هستند که نوعی از داده‌های سری زمانی تک متغیره هستند، بنابراین تغییرهای موردنیاز برای ورود این سیگنال‌ها به‌جای کلمات، داده شد و تابع هدف متناسب با این مسئله تعریف گردید. نوآوری

[1] Self Supervised Transformer Encoder for Anomaly Detection

دیگر، ارائه یک تابع امتیاز جدید است. در این تابع، خطاهای بازسازی کوچک که باعث *FP* می‌شود، در نظر گرفته نشد. نتایج نشان داد که مدل ارائه‌شده، امکان تشخیص سریع نقاط و مخصوصاً رویدادهای ناهنجار سری‌های زمانی چند متغیره را به‌طور مناسبی فراهم می‌کند. بر این اساس می‌توان نتیجه گرفت که در مسئله‌ی مذکور روش پیشنهادی باعث افزایش کارایی می‌شود. به‌عنوان نتیجه دیگر، تابع امتیاز مذکور باعث کارایی بهتر روش‌ها می‌شود. روش پیشنهادی در مجموعه داده‌ی *SMD* با 28 سرور مستقل، امتیاز F ترکیبی را 5 درصد بهبود بخشید. در حوزه‌ی زمان هم باعث بهبود 15 درصد زمان آموزش شد. به‌عنوان کار آینده، این روش می‌تواند در تشخیص نفوذ در سامانه‌های نظارتی مورداستفاده قرار بگیرد.

5. مراجع

1) Garg, W. Zhang, J. Samaran, R. Savitha and C. Sheng Foo, "An evaluation of anomaly detection and diagnosis in multivariate time series," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 6, pp. 2508-2517, 2021, https://doi.org/10.1109/TNNLS.2021.3105827 .

2) S. Bejani, M. R. Hasani Ahangar and M. Akhzami, "The Role of Intrusion Detection Systems in Web Services Security," Passive Defense, vol. 6, no. 4, pp. 65-77, 2013, (In Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1397.6.4.2.4

3) H. Tabatabaee and S. Hadavi, "Feature Selection and Intrusion Detection in Wireless Sensor Networks with Unsupervised Extreme Learning Machine (UELM)," Passive Defence, vol. 15, no. 4, pp. 25-40, 2024, DOR: 20.1001.1.20086849.1403.15.4.3.6. (In Persian).

4) K. Koo, M. Park and B. Yoon, "A suspicious financial transaction detection model using autoencoder and risk-based approach," IEEE Access, vol. 12, pp. 68926 - 68939, 2024, https://doi.org/10.1109/ACCESS.2024.3399824.

5) I. Farady, V. Patel, C.-C. Kuo and C. Yang, "ECG anomaly detection with LSTM-autoencoder for heartbeat analysis," in IEEE International Conference on Consumer Electronics (ICCE), Las Vegas, NV, USA, 2024, pp. 1-5, https://doi.org/10.1109/ICCE59016.2024.10444327.

6) S. Siadat, M. Ghafary and M. Rezvanmadani, "A method to detect intrusion into the Internet of Things using the game theory," Electronic and cyber defense, vol. 10, no. 1, pp. 21-31, 2022, DOR: 20.1001.1.23224347.1401.10.1.3.7, (In Persian). https://dor.isc.ac/dor/20.1001.1.23224347.1401.10.1.3.7

7) A. Alaverdov and F. Kanehiro, "Sensor anomaly detection for biped robot using the dynamic equation of a robotic system," in IEEE/SICE International Symposium on System Integration (SII), Ha Long, Vietnam, 2024, pp. 357-362, https://doi.org/10.1109/SII58957.2024.10417448.

8) S. Vila, R. Soto, E. Vega, A. P. Fritz and B. Crawford, "Anomaly detection on bridges using deep learning with partial training," IEEE Access, vol. 12, pp. 116530 - 116545, 2024, https://doi.org/10.1109/ACCESS.2024.3447571.

9) T. A. Siahmarzkooh, "Smart home intrusion detection model based on principal component analysis and random forest classification," Electronic and Cyber Defense, vol. 12, no. 2, pp. 15-25, 2024, DOR: 20.1001.1.23224347.1403.12.2.2.2. (In Persian).https://dor.isc.ac/dor/20.1001.1.23224347.1403.12.2.2.2

10) H. Hojjati, T. K. Khanh Ho and N. Armanfard, "Self-supervised anomaly detection in computer vision and beyond: a survey and outlook," Neural Networks, vol 172, 2024, https://doi.org/10.1016/j.neunet.2024.106106.

11) K. Lee, H. Lee and J. Shin, "A simple unified framework for detecting out-of-distribution samples and adversarial attacks," in Advances in Neural Information Processing Systems 31, Montreal, Canada, 2018.

12) G. PANG, C. SHEN, L. CAO and A. V. D. HENGEL, "deep learning for anomaly detection: A review," ACM Computing Surveys, vol. 54, no. 2, pp. 1-38, 2021, https://doi.org/10.1145/3439950.

13) L. Ruff, R. A. Vandermeulen, N. Görnitz, A. Binder, E. Müller, K. R. Müller and M. Kloft, "deep semi-supervised anomaly detection," Presented at International conference on learning representations, 2020. [Online]. Available: https://iclr.cc/virtual_2020/poster_HkgH0TEYwH.html

14) S. Tuli, G. Casale and N. R. Jennings, "TranAD: deep transformer networks for anomaly detection in multivariate time series data," in Proceedings of the VLDB Endowment, vol. 15, no. 6, pp. 1201-1214, 2022, https://doi.org/10.14778/3514061.3514067.

15) L. r. Yu, Q. h. Lu and Y. X. Yang Xue, "DTAAD: dual tcn-attention networks for anomaly detection in multivariate time series data," Knowledge-Based Systems, vol. 295, 2024, https://doi.org/10.1016/j.knosys.2024.111849.

16) A. Vaswani, N. Shazeer, N. Parmar, J. Uszkorei, L. Jones, N. A. Gomez and Ł. Kaiser, "Attention is all you need," in Advances in Neural Information Processing Systems 30, Long Beach, CA, USA, 2017.

17) J. Achiam, S. Adler, S. Agarwal, L. Ahmad and I. Akkaya, "GPT-4 technical report," arXiv preprint, arXiv:2303.08774, 2024. https://doi.org/10.48550/arXiv.2303.08774.

18) K. Kingsbury and P. Alvaro, "Elle: inferring isolation anomalies from experimental observations," in Proceedings of the VLDB Endowment, 2020. https://doi.org/10.48550/arXiv.2003.10554.

19) P. Boniol, J. Paparrizos, T. Palpanas and M. J. Franklin, "SAND: Streaming Subsequence Anomaly Detection," in Proceedings of the VLDB Endowment, 2021, pp. 1717-1729, https://doi.org/10.14778/3467861.3467863.

20) O. Salem, A. Guerassimov, A. Mehaoua and A. Marcus, "Anomaly detection in medical wireless sensor networks using SVM and linear regression models," International Journal of E-Health and Medical Communications (IJEHMC), vol. 5, pp. 20-45, 2014. https://doi.org/10.4018/ijehmc.2014010102

21) Y. Wang, N. Masoud and A. Khojandi, "Real-time sensor anomaly detection and recovery in connected automated vehicle sensors," IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1411-1421, 2021. https://doi.org/10.1109/TITS.2020.2970295.

22) T. Kieu, B. Yang, C. Guo and C. S. Jensen, "Outlier detection for time series with recurrent autoencoder ensembles," in IJCAI, Macao, China, 2019, pp. 2725-2732.

23) Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen and N. V. Chaw, "A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data," in Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, vol. 33, no. 1, 2019, pp. 1409-1416.

24) J. Audibert, P. Michiardi, F. Guyard and M. A. Zuluaga, "USAD: UnSupervised Anomaly Detection on multivariate time Series," in Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 3395-3404, 2020. https://doi.org/10.1145/3394486.3403392.

25) R. J. Hsieh, J. Chou and C. H. Ho, "Unsupervised online anomaly detection on multivariate sensing time series data for smart manufacturing," in IEEE 12th Conference on Service-Oriented Computing and Applications (SOCA), pp. 90-97, 2019. https://doi.org/10.1109/SOCA.2019.00021.

26) N. Gugulothu, P. Malhotra, L. Vig and G. Shroff, "Sparse neural networks for anomaly detection in high-dimensional time series," presented at AI4IOT Workshop in conjunction with ICML, IJCAI and ECAI, pp. 1551-3203, Stockholm, July. 9-19, 2018.

27) H. Zhao, Y. Wang, J. Duan, C. Hua, D. Cao and Y. Tong, "Multivariate time-series anomaly detection via graph attention network," in IEEE International Conference on Data Mining (ICDM), 2020, pp. 841-850, https://doi.org/10.1109/ICDM50108.2020.00093

28) A. Deng and B. Hooi, "Graph neural network-based anomaly detection in multivariate time series," in The 35th AAAI Conference on Artificial Intelligence, 2021, pp. 4027-4035, https://doi.org/10.1609/aaai.v35i5.16523.

29) N. Ding, H. Ma, H. Gao, Y. Ma and G. Tan, "Real-time anomaly detection based on long short-Term memory and Gaussian Mixture Model," Computers & Electrical Engineering, vol. 79, 2019, https://doi.org/10.1016/j.compeleceng.2019.106458.

30) K. Hundman, V. Constantinou, C. Laporte, I. Colwell and T. Soderstrom, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018, https://doi.org/10.1145/3219819.3219845.

31) Park, Y. Hoshi and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," IEEE Robotics and Automation Letters, vol. 3, no. 3, pp. 1544-1551, 2018, https://doi.org/10.1109/LRA.2018.2801475.

32) Y. Guo, W. Liao, Q. Wang, Y. Lixing, T. Ji and P. Li, "Multidimensional time series anomaly detection: A GRU-based gaussian mixture variational autoencoder approach," in Proceedings of The 10th Asian Conference on Machine Learning, Beijing, China, pp. 97-112, 2018.

33) Y. Choi, H. Lim, H. Choi and J. Kim, "GAN-Based anomaly detection and localization of multivariate time series data for power plant," in IEEE International Conference on Big Data and Smart Computing (BigComp), 2020, pp. 71-74, https://doi.org/10.1109/BigComp48618.2020.00-97.

34) T. Wen and R. Keyes, "Time Series anomaly detection using convolutional neural networks and transfer learning," arXiv preprint, arXiv:1905.13628, 2019, https://doi.org/10.48550/arXiv.1905.13628

35) B. Zhou, S. Liu, B. Hooi, X. Cheng and J. Ye, "BeatGAN: anomalous rhythm detection using adversarially generated time," in Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 4433-4439, https://doi.org/10.24963/ijcai.2019/616.

36) S. Bai, J. Kolter and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," arXiv preprint, arXiv:1803.01271v2, 2018, https://doi.org/10.48550/arXiv.1803.01271.

37) Shi, C. Xingjian, W. Zhourong, Y. Hao, Yan and Dit, "Convolutional LSTM Network: A machine learning approach for precipitation nowcasting," in Proceedings of the 28th International Conference on Neural Information Processing Systems, 2015.

38) Khoshnevisan, F. Farzaneh, C. Zhewen, R and Vitor, "Improving robustness on seasonality-heavy multivariate time Series anomaly," in The 1st Workshop on Artificial Intelligence for Anomalies and Novelties, 2020. https://doi.org/10.1145/1122445.1122456.

39) Z. Chen, D. Chen, X. Zhang, Z. Yuan and C. Xiuzhen, "Learning graph structures with transformer for multivariate time-series anomaly detection in IoT," IEEE Internet of Things Journal, vol. 9, no. 12, pp. 9179-9189, 2022, https://doi.org/10.1109/JIOT.2021.3100509.

40) J. Wu, W. Zeng and F. Yan, "Hierarchical temporal memory method for time-series-based anomaly detection," Neurocomputing, vol. 273, pp. 535-546, 2018, https://doi.org/10.1016/j.neucom.2017.08.026.

41) H. Song, D. Raja, J. J. Thiagarajan and A. Spanias, "Attend and diagnose: clinical time series analysis using attention models," in Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, vol. 32, no. 1, 2018, https://doi.org/10.1609/aaai.v32i1.11635.

42) Yongliang, Y. Xu, H. Zhong and Y. Liu, "HS-TCN: A semi-supervised hierarchical stacking temporal convolutional network for anomaly detection in IoT," in IEEE 38th International Performance Computing and Communications Conference, 2019, pp. 1-7, https://doi.org/10.1109/IPCCC47392.2019.8958755.

43) L. Shen, Z. Li and T. J. Kwok, "Timeseries anomaly detection using temporal hierarchical one-class network," in 33th Conference on Neural Information Processing Systems, vancouver, canada, 2020, pp.13016-13026.

44) J. Liu, H. Zhu, Y. Liu, H. Wu, Y. Lan and X. Zhang, "Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model," Journal of Physics, vol. 1187, no. 4, 2019, https://doi.org/10.1088/1742-6596/1187/4/042111

45) N. Mejri, L. L. Fuentes, K. Roy, P. Chernakov, E. Ghorbel and D. Aouada, "Unsupervised anomaly detection

49) Malhotra, R. Pankaj, A. Anusha, V. Gaurangi, A. Lovekesh, S. Puneet and Gautam, "Lstm-based encoder-decoder for multi-sensor anomaly detection," arXiv preprint, arXiv:1607.00148, 2016, https://doi.org/10.48550/arXiv.1607.00148.

50) Zong, S. Bo, M. Qi, Renqiang, C. Martin, L. Wei, C. Cristian, C. Daeki and Haifeng, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in International conference on learning representations, Vancouver, BC, Canada, 2018.

51) Su, Z. Ya, N. Youjian, L. Chenhao, S. Rong, P. Wei and Dan, "Robust anomaly detection for multivariate time series through stochastic recurrent neural network," in Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828-2837, https://doi.org/10.1145/3292500.3330672.

in time-series: An extensive evaluation and analysis of state-of-the-art methods," Expert Systems With Applications, vol. 256, 2024, https://doi.org/10.1016/j.eswa.2024.124922.

46) H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu and Y. Zhao, "Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications," in Proceedings of the 2018 World Wide Web Conference, 2018, pp. 187-196, https://doi.org/10.1145/3178876.3185996.

47) S. Kim, K. Choi, S. H. Choi, B. Lee and S. Yoon, "Towards a rigorous evaluation of time-series anomaly detection," in The Thirty-Sixth AAAI Conference on Artificial Intelligence, 2022. https://doi.org/10.1609/aaai.v36i7.20680

48) Zheng, Y. Panpan, W. Shuhan, L. Xintao, L. Jun and Aidong, "One-class adversarial nets for fraud detection," in Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, no. 1, 2019, pp. 1286-1293, https://doi.org/10.1609/aaai.v33i01.33011286.