

الله الرحمن الرحيم



دانشگاه صنعتی شاهرود

دانشکده مهندسی کامپیوتر

رساله دکتری مهندسی هوش مصنوعی

تشخیص و پیش‌بینی ناهنجاری در خانه‌های هوشمند با استفاده از یادگیری عمیق

نگارنده: محمود معلم

استاد راهنما

دکتر علی‌اکبر پویان

استاد مشاور

دکتر حمید حسن‌پور

شهریور ۱۳۹۸

تقدیم بہ عزیزانم ، بہ پاس مہربانی و ہمہلی و ہمراہی بیارشان

شکر و قدردانی

پس از حمدی حدخداى سرمد و نعت رسول تخی مرتب (ص) و وصف و تحت خاندان پاک نبوت، بر خویشتن فرض و

و غیبه می دانم که از استاد راهمای فرهیخته و کرامتدارم جناب آقای دکتر علی اکبر پویان شکر کنم، که این کار بی هدایت و حمایت

ایشان هرگز به سلمان نمیرسید. از استاد مشاور بزرگوارم جناب آقای دکتر حمید حسن پور بسیار سپاسگزارم، سهم ایشان در این رساله و

حق جنابشان بر ذمه بخارنده از حد مشاوری بی فراتر است.

از داوران محترم دفاع خود، آقایان دکتر بهروز مینایی، دکتر منصور فاتح و دکتر حسین خسروی قدردانی می کنم که از هدایت و

امداد و ارشاد ایشان بهره ای وافر برداشتم.

آقای دکتر عبدالرحیم نوه ابراهیم، آقای دکتر علی تحقیقی اصل، آقای دکتر عبدالعلی بصیری و سرکار خانم دکتر بتول

اخلاقی نیا، هر یک به نوعی در این مقام شایسته شکر هستند.

سرانجام از صمیم قلب ممنون محبت مادر، همسر، خواهر و پسران مانزینم هستم.

تعمیرنامه

اینجانب **محمود معلم** دانشجوی دوره دکتری رشته کامپیوتر-هوش مصنوعی دانشکده مهندسی کامپیوتر دانشگاه صنعتی شاهرود نویسنده پایان نامه تشخیص و پیش‌بینی ناهنجاری در خانه‌های هوشمند با استفاده از یادگیری عمیق تحت راهنمایی دکتر علی‌اکبر پویان متعهد می‌شوم.

- تحقیقات در این پایان‌نامه توسط اینجانب انجام شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورد استفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه صنعتی شاهرود می‌باشد و مقالات مستخرج با نام « دانشگاه صنعتی شاهرود » و یا « **Shahrood University of Technology** » به چاپ خواهد رسید .
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه ، در مواردی که از موجود زنده (یا بافتهای آن‌ها) استفاده شده است ضوابط و اصول اخلاقی رعایت شده است .
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده شده است اصل رازداری ، ضوابط و اصول اخلاق انسانی رعایت شده است .

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج ، کتاب ، برنامه های رایانه ای ، نرم افزار ها و تجهیزات ساخته شده است) متعلق به دانشگاه صنعتی شاهرود می‌باشد . این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود .
استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

چکیده

از مهم‌ترین ویژگی‌هایی که یک‌خانه هوشمند باید از آن برخوردار باشد پیش‌بینی و تشخیص اعمال غیرمعمول و ناهنجار است. پیشرفت چشمگیر فناوری‌های ساخت و به‌کارگیری حسگرها از سویی و افزایش قابل توجه تعداد افراد سالخورده یا بیماری که به‌تنهایی زندگی می‌کنند از سوی دیگر، تحقق این ویژگی را به امری ضروری تبدیل کرده است. مشکل اینجاست که تعدد و تنوع حسگرها و حجم بالای اطلاعات دریافتی از آن‌ها، ابعاد مسئله را به مرزهای مقوله کلان‌داده نزدیک کرده و لزوم همجوشی داده‌ها، کار را دشوارتر می‌نماید. به همین دلیل پردازش این داده‌ها، به‌ویژه استخراج ویژگی‌های مناسب متضمن پیچیدگی محاسباتی قابل توجهی (زمانی و مکانی) است.

در این رساله روشی برای تشخیص و پیش‌بینی اعمال غیرمعمول و ناهنجار در یک‌خانه هوشمند با استفاده از یادگیری عمیق ارائه می‌دهیم. داده‌های این خانه بنا به سنت غالب خانه‌های هوشمند، از طریق حسگرهای بیسیم غیر مزاحم مثل حسگرهای نوری تشخیص و ردیابی و حسگرهای باز و بسته بودن درب و نظیر آن‌ها گردآوری می‌شود. دادگان خروجی این حسگرها مجموعه‌ای از رویدادهای غیرهمزمان و گسسته است که در بستر پیوسته زمان روی می‌دهند. فواصل زمانی این رویدادها یکسان نیست و فاصله زمانی بین دو رویداد می‌تواند بیانگر ویژگی‌های رفتاری مهمی باشد. به همین دلیل ما به‌جای استفاده از روش‌های معمول در تحلیل سری‌های زمانی، از نظریه فرایندهای نقطه‌ای بهره برده‌ایم. این نظریه چارچوب ریاضی قدرتمندی است که زبان آماری مناسب را برای صورت‌بندی و تحلیل رویدادهای غیرهمزمان فراهم می‌کند.

مدل‌های رایج در نظریه مدرسی فرایندهای نقطه‌ای غالباً مفروضات خاص و محدودکننده‌ای را در مورد شیوه تولید رویدادها لحاظ می‌کنند. به همین دلیل ما برای احاطه بر رفتار یک فرایند و استخراج الگوهای هنجار، از یک شبکه عصبی بازگشتی عمیق استفاده کرده‌ایم. این شبکه در قامت یک نگاشت غیرخطی، با دریافت تاریخچه رویدادها نوع و زمان وقوع رویداد بعدی را پیش‌بینی می‌نماید. ما این پیش‌بینی را با استخدام جستجوی پرتو محلی به مجموعه‌ای از رشته رویدادهای محتمل بسط می‌دهیم تا محدوده احتمالی افعال آتی مشخص گردد. فاصله سلسله رویدادهای واقعی با محدوده مذکور، مبین هنجار یا ناهنجار بودن این سلسله خواهد بود. نتایج تجربیات انجام‌شده نشان می‌دهد که این روش امکان کشف افعال ناهنجار را با دقت قابل قبولی فراهم می‌کند.

کلمات کلیدی: تشخیص ناهنجاری، یادگیری عمیق، فرایند نقطه‌ای، محیط هوشمند، پردازش رویداد

لیست مقالات مستخرج از پایان نامه

- ۱- محمود معلم، علی اکبر پویان، «تشخیص افعال روزمره با استفاده از تلفن همراه و یادگیری عمیق»، سومین کنفرانس پردازش سیگنال و سیستم‌های هوشمند ایران، ۱۳۹۶.
- ۲- محمود معلم، علی اکبر پویان - «کشف ناهنجاری با استفاده از کدکننده خودکار مبتنی بر بلوک‌های LSTM» - مدل‌سازی در مهندسی، دوره ۱۷، شماره ۵۶، بهار ۱۳۹۸
- ۳- محمود معلم، علی اکبر پویان، حمید حسن پور، «کشف افعال ناهنجار در خانه‌های هوشمند با استفاده از یادگیری عمیق»، چهارمین کنفرانس پردازش سیگنال سیستم‌های هوشمند، ۱۳۹۷.
- ۴- Moallem, M., Hassanpour, H., and Pouyan, A. A., (2019), “Anomaly Detection in Smart Homes Using Deep Learning,” *Iran. J. Energy Environ.*, vol. 10, no. 2, pp. 126–135.

فهرست مطالب

د	فهرست جداول
ه	فهرست اشکال
ز	فهرست علائم
۱	فصل ۱ مقدمه
۲	۱-۱ درآمد
۳	۲-۱ صورت و مفروضات
۴	۳-۱ جهت گیری
۶	۴-۱ ساختار رساله
۹	فصل ۲: مبانی
۱۰	۱-۲ خانه هوشمند
۱۰	۱-۱-۲ حسگرها
۱۱	۲-۱-۲ افعال
۱۳	۳-۱-۲ پردازش و تحلیل افعال
۱۴	۴-۱-۲ پردازش ناهنجاری
۱۵	۲-۲ تشخیص ناهنجاری
۱۶	۱-۲-۲ چالشها
۲۰	۲-۲-۲ تکنیکهای تشخیص ناهنجاری
۲۳	۳-۲ یادگیری عمیق

۲۷..... ۱-۳-۲ شبکه‌های پیچشی (CNN)

۲۹..... ۲-۳-۲ شبکه‌های عصبی بازگشتی و LSTM

۳۷..... ۳-۳-۲ توجه

۴۲..... ۴-۳-۲ کد کننده خودکار یا AE (AutoEncoder)

فصل ۳: مروری به کارهای گذشته ۴۷

۴۸..... ۱-۳ مقدمه

۴۸..... ۲-۳ شناسایی فعالیت

۴۹..... ۱-۲-۳ شناسایی فعالیت مبتنی بر بینایی ماشین

۵۰..... ۲-۲-۳ شناسایی فعالیت مبتنی بر حسگر

۵۶..... ۳-۳ تشخیص ناهنجاری

فصل ۴: فرایندهای نقطه‌ای ۶۳

۶۴..... ۱-۴ مقدمه

۶۵..... ۴-۲ فرایندهای نقطه‌ای

۶۵..... ۱-۲-۴ تعریف یک فرایند نقطه‌ای

۶۸..... ۲-۲-۴ انواع فرایندهای نقطه‌ای

۷۴..... ۳-۲-۴ استخراج پارامترهای تابع شدت شرطی - شیوه آماری

فصل ۵: راه‌حل پیشنهادی ۷۹

۸۰..... ۱-۵ مقدمه

۸۱..... ۲-۵ پیش‌پردازش

۸۳..... ۳-۵ شرح روش پیشنهادی

۸۹..... ۴-۵ دادگان

۹۲..... ۱-۴-۵ شیوه ایجاد و درج ناهنجاری مصنوعی

۵-۵ معیار کار آبی ۹۵

۵-۶ پیاده‌سازی ۹۹

۵-۶-۱ بحث و بررسی ۱۰۱

۵-۷ استفاده از مکانیزم توجه ۱۰۶

۵-۸ عوامل زمانی ۱۱۲

۱۱۴ فصل ۶ نتیجه‌گیری و کارهای آتی

۶-۱ مقدمه ۱۱۵

۶-۲ نتایج ۱۱۶

۶-۳ کارهای آینده ۱۱۷

۱۲۰ مراجع

فهرست جداول

- جدول ۱-۵: مشخصات تجربه‌ها ۹۹
- جدول ۲-۵: مشخصات شبکه‌های مورد استفاده ۱۰۰
- جدول ۳-۵: مقادیر AUC برای تجربه‌های مختلف ۱۰۰

فهرست اشکال

- شکل ۱-۱: روند افزایش تعداد افرادی که به تنهایی زندگی می‌کنند. ۲.....
- شکل ۱-۲: ناهنجاری‌های نقطه‌ای در فضای دوبعدی..... ۱۷.....
- شکل ۲-۲: ناهنجاری‌های زمینه‌ای در داده‌های دمای ماهانه ۱۸.....
- شکل ۳-۲: ناهنجاری مجموعه‌ای در یک الکترودیگرام انسانی ۱۹.....
- شکل ۴-۲: تشخیص ناهنجاری و رویکردهای با دسترس‌پذیری داده‌های آموزشی و آزمون ۲۰.....
- شکل ۵-۲: تکنیک‌های تشخیص ناهنجاری مبتنی بر رده‌بندی ۲۱.....
- شکل ۶-۲: افزایش انتزاع در لایه‌های یک ساختار سلسله‌مراتبی عمیق ۲۴.....
- شکل ۷-۲: مقایسه شیوه استخراج ویژگی‌ها در مکاتب متفاوت یادگیری ماشین ۲۶.....
- شکل ۸-۲: یک CNN نمونه ۲۹.....
- شکل ۹-۲: شبکه‌های عصبی بازگشتی (الف) المان و (ب) جوردن ۲۹.....
- شکل ۱۰-۲: مشکل اختفای گرادیان در شبکه‌های بازگشتی ۳۲.....
- شکل ۱۱-۲: ساختار یک بلوک LSTM ۳۳.....
- شکل ۱۲-۲: تغییر دقت شبکه بازگشتی نسبت به طول رشته ۳۸.....
- شکل ۱۳-۲: مکانیسم توجه در بینایی انسان ۳۹.....
- شکل ۱۴-۲: جزئیات لایه توجه ۴۱.....
- شکل ۱۵-۲: یک کد کننده خودکار عمیق ۴۲.....
- شکل ۱۶-۲: یک کد کننده خودکار مبتنی بر بلوک‌های LSTM ۴۵.....
- شکل ۱-۴: نمونه‌ای از یک فرایند پوآسن ناهمگن ۶۹.....
- شکل ۲-۴: پراکندگی نقاط و تابع شدت شرطی یک فرایند خود اصلاح‌گر ۷۰.....
- شکل ۳-۴: شبیه‌سازی یک فرایند هاوکس با سه حسگر وابسته ۷۳.....
- شکل ۴-۴: تفکیک رویدادهای درهم به گروه‌های هم‌جنس ۷۴.....
- شکل ۱-۵: شمای روش پیشنهادی ۸۳.....
- شکل ۲-۵: نمایش روزهای یک فصل با شماره روزها و تبدیل قطبی ۸۴.....
- شکل ۳-۵: شیوه عملکرد جستجوی پرتو محلی ۸۷.....
- شکل ۴-۵: بخشی از دادگان aruba ۹۱.....
- شکل ۵-۵: الگوی فعال شدن حسگر درب سرویس بهداشتی در دادگان هنجار ۹۳.....
- شکل ۶-۵: الگوی فعال شدن حسگر درب سرویس بهداشتی در دادگان ناهنجار ۹۳.....
- شکل ۷-۵: الگوی زمان فعال شدن و مدت فعال ماندن حسگر تختخواب - دادگان هنجار ۹۴.....
- شکل ۸-۵: الگوی زمان فعال شدن و مدت فعال ماندن حسگر تختخواب - دادگان ناهنجار ۹۵.....

- شکل ۹-۵: چهار نمونه از نمودارهای ROC ۹۸
- شکل ۱۰-۵: نمودار ROC برای تجربیات متفاوت ۱۰۱
- شکل ۱۱-۵: گراف مدل N1 در Tensorboard ۱۰۳
- شکل ۱۲-۵: گراف مدل N1 ۱۰۴
- شکل ۱۳-۵: روند افزایش دقت مدل N1 در پیش‌بینی رویداد حسگر بعدی ۱۰۵
- شکل ۱۴-۵: روند کاهش خطای مدل N1 در تخمین زمان رویداد بعدی ۱۰۵
- شکل ۱۵-۵: شیوه محاسبه توجه در مدل بهبودیافته ۱۰۷
- شکل ۱۶-۵: روند افزایش دقت مدل N1 در پیش‌بینی رویداد حسگر بعدی - پس از توجه ۱۱۰
- شکل ۱۷-۵: روند کاهش خطای مدل N1 در تخمین زمان رویداد بعدی - پس از توجه ۱۱۰
- شکل ۱۸-۵: گراف مدل N1 پس از اضافه شدن لایه توجه ۱۱۱
- شکل ۱۹-۵: تاثیر مکانیزم توجه بر دقت مدل N1 ۱۱۲

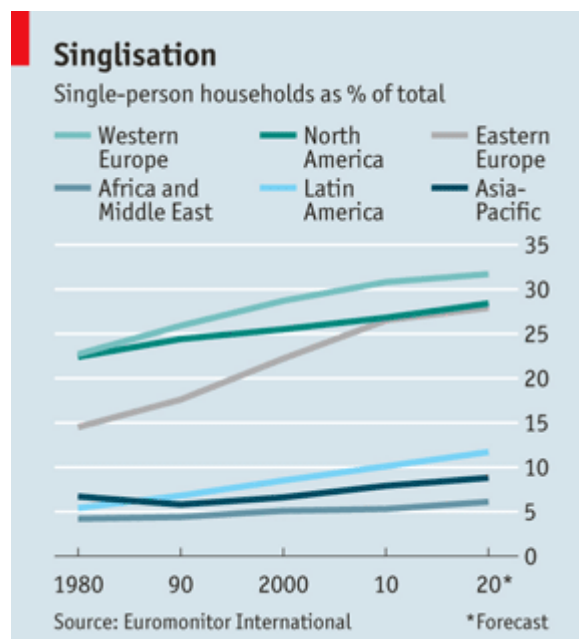
فہرست علامت

AAL	Ambient Assisted Living
ADL	Activities of Daily Living
AE	AutoEncoder
ANN	Artificial Neural Network
ARIMA	AutoRegressive Integrated Moving Average
AUC	Area Under Curve
BLSTM	Bidirectional Long-Short Term Memory
CASAS	Center for Advanced Studies in Adaptive Systems
CNN	Convolutional Neural Network
COV	Covariance
CRF	Conditional Random Field
CUDA	Compute Unified Device Architecture
DTW	Discrete Time Warping
FPR	False Positive Rate
GRU	Gated Recurrent Unit
HMM	Hidden Markov Model
LCS	Longest Common Subsequence
LSTM	Long-Short Term Memory
MLE	Maximum Likelihood Estimation
MLP	Multi Layer Perceptron
MSE	Mean Squared Error
OCSVM	One-Class Support Vector Machine
PCA	Principal Component analysis
Relu	Rectified Linear Units
RFID	Radio-Frequency Identification
RNN	Recurrent Neural Network
ROC	Receiver Operator Curve
SOM	Self Organized Map
SVM	Support Vector Machine
TPR	True Positive Rate

فصل ۱ مقدمه

۱-۱ درآمد

بنا به مواردی مثل گسترش شهرها، شیوع آپارتمان‌نشینی و ترویج خانواده‌های کوچک و کم تعداد که غالباً از تبعات اجتناب‌ناپذیر سبک زندگی معاصر هستند، هرساله بر تعداد افرادی که تنها زندگی می‌کنند افزوده می‌شود. همانطور که در شکل ۱-۱ نیز نشان داده شده است، این مسئله گرچه بیشتر مناطق پیشرفته دنیا خصوصاً اروپا را درگیر کرده، اما منطقه خاورمیانه و کشور ما نیز از دستبرد آن در امان نمانده‌اند [1].



شکل ۱-۱: روند افزایش تعداد افرادی که به تنهایی زندگی می‌کنند [1].

درصد قابل توجهی از این افراد، سالخورده یا بیمار هستند و برای انجام فعالیت‌های روزمره خود به کمک نیاز دارند یا باید تحت توجه و مراقبت باشند تا در صورت بروز مشکل، کسی به کمک آن‌ها بشتابد. متأسفانه در بسیاری از موارد، امکان اینکه کسی از بستگان در کنار این افراد زندگی کند وجود ندارد. استخدام پرستار ماهر و دلسوز نیز همیشه مقدور نیست، چراکه چنین پرستاری (در صورت وجود) معمولاً دستمزد قابل توجهی را طلب می‌کند. البته همه این‌ها به شرطی است که مشکل

عدم پذیرش شخص آشنا یا بیگانه از طرف فرد سالمند یا بیمار را حل شده یا نادیده بینگاریم و فرض کنیم که او با حضور دائم ندیم و ملازم خود مشکلی ندارد، از مراقبت و دلسوزی گاه‌وبیگاه وی افسرده و بی‌حوصله نمی‌شود و درصدد آزار و حتی اخراج او بر نمی‌آید.

بررسی مطالعات متعدد انجام‌شده در دهه‌های اخیر، بیانگر این مطلب است که افراد تنها در خانه، به‌ویژه سالخوردگان، تمایل ندارند تنهایی خود را ترک کنند و در یک زندگی جمعی با افرادی که شبیه خودشان هستند دمساز شوند (مثل زندگی در سرای سالمندان یا اقامت در بیمارستان). همچنین از مراقبت دائمی یک پرستار، یا حتی فردی از بستگان خود نیز استقبال نمی‌کنند [2]؛ بنابراین به نظر می‌رسد بهترین راه کاهش هزینه‌های عاطفی و اقتصادی تمشیت این افراد، آن است که در خانه خود و به شکلی نامحسوس تحت مراقبت باشند. خانه‌ای هوشمند که ظاهراً هیچ تفاوتی با خانه قبلی فرد ندارد اما با بهره‌گیری از حسگرهای نامحسوسی که در گوشه و کنار نصب شده، دائماً اوضاع جسمی و حتی روحی ساکنان خود را به افراد ذی‌ربط گزارش می‌کند.

۱-۲ صورت و مفروضات

بستر عملیاتی موردنظر رساله حاضر، خانه‌ای است که به دلیل وجود حسگرهای متفاوت امکان جمع‌آوری اطلاعات رفتاری ساکنان آن وجود دارد. حسگرهای اصلی خانه یادشده، حسگرهای تشخیص حضور و حسگرهای تشخیص باز و بسته بودن درب‌ها هستند که خروجی آن‌ها به شکل دودویی و ON و OFF است. اما در صورت لزوم می‌توان از حسگرهای دودویی دیگر، مثل حسگرهای تشخیص وضعیت شیر آب، حسگرهای تشخیص وجود فشار و امثال این‌ها نیز استفاده نمود. گرچه در برخی از پروژه‌های خانه هوشمند از حسگرهای عددی مثل فشارسنج، دماسنج و رطوبت‌سنج و برخی ادوات پایش مثل دوربین‌های نظارتی و حرارتی بهره می‌برند، اما چنانکه گفته شد تمرکز رساله حاضر بر روی حسگرهای دودویی، به‌ویژه حسگرهای تشخیص حضور و تشخیص باز و بسته بودن

درب‌هاست. این روش برای کشف افعال و رویدادهای غیرطبیعی از خروجی خام حسگرها استفاده می‌کند و به اعلام و برچسب‌گذاری افعال کاربر نیازی ندارد. چنانکه گفته شد این حسگرها همه دودویی هستند و در فواصل نامتساوی و بر اساس افعال کاربر، فعال یا غیرفعال شده و مقادیر روشن/خاموش، حاضر/غایب و کلاً ۰ و ۱ را به ایستگاه مرکزی مخابره می‌کنند.

بعلاوه روش پیشنهادی با فرض وجود تنها یک ساکن تدوین شده است، بنابراین در صورتی که ساکنین متعددی در یک‌خانه وجود داشته باشند باید به طریقی از هم تفکیک بشوند. این کار معمولاً با استفاده از تگ‌های RFID صورت می‌پذیرد، یعنی با تهیه تگ‌های RFID متمایز برای هر یک از ساکنین و الصاق این تگ‌ها به بدن، لباس یا وسایل شخصی (مثل ساعت) ایشان، امکان بازشناسی و تفکیک این ساکنان برای سیستم مهیا می‌شود. به‌هر حال چنانکه گفته شد ورودی سیستم پیشنهادی، داده‌هایی است که از تعقیب و ردیابی افعال تنها یک ساکن ناشی شده است.

سه دیگر آنکه، عنوان و هدف این تحقیق، تشخیص و پیش‌بینی ناهنجاری و افعال غیرمعمول در یک‌خانه هوشمند است، بنابراین علی‌القاعده، نوع افعال و نسبت آن‌ها با یکدیگر (متداخل، موازی یا مستقل) موضوعیت چندانی ندارد و تنها عدول و خروج این افعال از محدوده افعال معمول مورد کنکاش و تأمل قرار خواهند گرفت. در واقع در روش پیشنهادی، سیستم از طریق خوشه‌بندی داده‌های دریافتی، به تدریج رشته‌های مکرر و معمول را در اعمال کاربر استخراج می‌کند تا امکان تشخیص و پیش‌بینی رشته‌های غیرمعمول و بی‌سابقه فراهم گردد. تا پیش از تکمیل این مرحله، برای ساده‌تر کردن مسئله، فرض می‌کنیم که افعال کاربر ساده و پیاپی هستند.

۱-۳ جهت‌گیری

کارهایی که تاکنون با موضوع کشف ناهنجاری در خانه‌های هوشمند انجام شده است، غالباً بر بازشناسی نوعی مشخص (مثلاً زمین خوردن) یا حداکثر کلاسی معین (مثلاً غیرفعال بودن

طولانی مدت) از افعال غیرمعمول متمرکز بوده‌اند. بعلاوه در این کارها زمینه زمانی و مکانی افعال کمتر مورد توجه قرار گرفته است. مثلاً دیر برخاستن از خواب در یک روز تعطیل یا اقامت ثابت و بدون حرکت در اتاق نشیمن، به ترتیب در زمینه‌های زمانی و مکانی خود غیرمعمول نیستند، اما بدون توجه به این زمینه‌ها می‌توان آن‌ها را به‌عنوان دو فعل نابهنجار دیر بیدار شدن و غیرفعال بودن طولانی در نظر گرفت.

به‌طور کلی ناهنجاری‌های موجود در افعال ساکنان یک‌خانه هوشمند را می‌توان در قالب انواع چهارگانه زیر تقسیم‌بندی کرد:

- ناهنجاری در مکان وقوع: اجرای یک عمل معمول در یک مکان غریب، مثل خوابیدن در آشپزخانه
- ناهنجاری در مدت وقوع: کوتاهی یا درازای غیرمعمول انجام یک فعل
- ناهنجاری در زمان وقوع: آغاز زود یا دیرهنگام یک فعل معمول مثل خوردن دارو
- ناهنجاری در تعداد یا ترتیب مراحل انجام یک فعل

توجه داشته باشید که سه نوع اول، از گونه ناهنجاری‌های نقطه‌ای هستند و نوع چهارم در زمره ناهنجاری‌های زمینه‌ای قرار می‌گیرد. در واقع وقوع این نوع ناهنجاری مستلزم وجود نوعی رابطه ترتیبی (زمانی-مکانی) یا معنایی (گراف) در میان داده‌های مربوطه است. مثلاً روشن شدن تلویزیون، روشن شدن چراغ اتاق خواب و خوابیدن بر روی تختخواب هیچ‌کدام غیرطبیعی نیست؛ اما وقوع متوالی این افعال احتمالاً به این معناست که ساکنان خانه بدون خاموش کردن تلویزیون به رختخواب رفته‌اند. ما در مسیر انجام تحقیق حاضر کار را از انواع سه‌گانه اول آغاز کردیم. این سه نوع علیرغم تفاوت‌های مشهود، در واقع از یک سنخ هستند: در همه آن‌ها ما با سلسله‌ای از اعداد سروکار داریم که در حالت هنجار از الگوهای تکرارشونده برخوردارند؛ و در همه آن‌ها وقوع ناهنجاری به معنای وجود نقطه یا نقاط پرتی است که با هیچ‌یک از الگوهای یادشده تطبیق نمی‌کنند. این اعداد می‌توانند کد مکان، زمان شروع و طول مدت انجام یک فعل باشند؛ بنابراین کشف سه نوع اول از انواع چهارگانه فوق را می‌توان به کشف ناهنجاری در سری‌های زمانی تعبیر و تحویل کرد.

برای کشف ناهنجاری در سری‌های زمانی روش‌های متفاوتی وجود دارد که از مدل‌های آماری تا شبکه‌های عصبی گسترده شده‌اند. یکی از این ابزارها کدکننده خودکار است که رایج‌ترین مدل یادگیری عمیق برای کشف ناهنجاری به شمار می‌آید. تلاش برای آشنایی و کار با این ابزار قدرتمند به نگارش و انتشار مقاله‌ای منجر شد که مشخصات آن در ابتدای رساله حاضر ذکر شده است.

اما جهت‌گیری اصلی ما در نگارش این رساله متوجه کشف ناهنجاری‌های نوع چهارم بوده است. این نوع ناهنجاری از سایر انواع پیچیده‌تر است و به دلیل اشتغال بر مقولات زمان، مکان، مدت و ترتیب وقوع رویدادها، همه این انواع را دربر می‌گیرد. چنانکه گفته شد دادگان مورد استفاده، خروجی حسگرهایی است که در فواصل نامتساوی و بنا به افعال کاربر فعال می‌شوند. این داده‌ها دارای ترتیب و برچسب زمان هستند، اما با رشته‌ها و سری‌های زمانی تفاوتی اساسی دارند. در مجموعه رویدادهای خروجی حسگرها، زمان نقشی فراتر از اندیس صرف را به عهده گرفته و خود، متغیری تصادفی به شمار می‌آید. به همین دلیل هر نوع فعالیتی که - مثل کشف ناهنجاری - مبتنی بر پیش‌بینی باشد مستلزم تحلیل هم‌زمان و توأمان برچسب و زمان وقوع رویداد است.

ما برای انجام این تحلیل از نظریه فرایندهای نقطه‌ای استفاده کرده‌ایم. این نظریه ابزار قدرتمندی است که برای مدل‌سازی سلسله رویدادها بسیار مناسب است و تاکنون در زمینه‌های متنوع و متفاوتی مثل شبکه‌های اجتماعی، تجارت و تعمیر و نگهداری به کار گرفته شده است. بنا به اطلاع نگارنده این نخستین باری است که نظریه فرایندهای نقطه‌ای در حوزه خانه‌های هوشمند و کشف ناهنجاری سلسله‌ای از رویدادها به کمک یادگیری عمیق مورد استفاده قرار می‌گیرد.

۱-۴ ساختار رساله

فصل ۱ حاوی مقدمه است. در این فصل مفروضات و جهت‌گیری عمده در نگارش رساله، به رشته تحریر کشیده شده است. فصل ۲ شامل مبانی است که اطلاع از آن‌ها برای مطالعه رساله لازم

می‌نماید. این مقولات مشتمل بر مفاهیم خانه‌های هوشمند، روش‌های کشف ناهنجاری، و آشنایی با یادگیری عمیق می‌باشد. فصل ۳ برجسته‌ترین تحقیقاتی که تاکنون در زمینه شناسایی افعال و کشف ناهنجاری در خانه‌های هوشمند صورت پذیرفته است را مرور می‌کند. مبانی نظری و چارچوب ریاضی فرایندهای نقطه‌ای در فصل ۴ بیان شده است. فصل ۵ رساله حاوی راه‌حل پیشنهادی و شرح تجربیات انجام شده می‌باشد. سرانجام فصل ۶ شامل نتیجه‌گیری و پیشنهادهایی برای آینده است.

فصل ۲ : مبانی

۱-۲ خانه هوشمند

محور این رساله، کشف و پیش‌بینی افعال غیرمعمول^۱ در یک‌خانه هوشمند است. در اینجا مفهوم هوشمندی دال بر این است که چنین خانه‌ای دائماً افعال ساکنان خود را تحت مراقبت قرار می‌دهد تا انجام اعمال روزانه^۲ (ADL) مثل استحمام، خوابیدن و ... را برای آن‌ها ساده‌تر کند و در صورت تشخیص ناهنجاری^۳ و بروز اتفاقات غیرمترقبه به یاری ساکنان خود شتافته یا بلافاصله هشدارهای لازم را برای افراد امدادگر صادر نماید [3].

۱-۱-۲ حسگرها

در ساختار یک‌خانه هوشمند حسگرهایی تعبیه‌شده است تا مراقبت فوق‌الذکر از طریق آن‌ها صورت پذیرد. این حسگرها بر سه نوع کلی هستند:

- حسگرهای مبتنی بر بینایی ماشین^۴
- حسگرهای پوشیدنی^۵
- حسگرهای غیرتهاجمی^۶ (نامحسوس): حسگرهای نوری، حسگرهای مغناطیسی، ...

^۱ Abnormal Activities

^۲ Activities of Daily Living

^۳ Anomaly

^۴ Machine Vision based Sensors

^۵ Wearable Sensors

^۶ Non-intrusive Sensors

طبیعی است که در هر خانه‌ای (هوشمند یا غیرهوشمند) راحتی و آسایش ساکنان در درجه اول اهمیت قرار دارد؛ بنابراین در پروژه‌های واقعی و صنعتی خانه هوشمند، معمولاً از به‌کارگیری حسگرهای نوع اول و دوم، به دلیل ایجاد اختلال در حریم خصوصی و آسایش ساکنان منزل، اجتناب می‌شود. بیشتر پروژه‌ها و تحقیقات موجود از جمله پژوهش حاضر، بر مبنای داده‌های حاصل از حسگرهای نامحسوس و غیرتهاجمی بنا شده‌اند.

البته این داده‌ها به دلایل زیر خوش‌فرم نیستند و فرآوری آن‌ها از داده‌های خام به اطلاعات قابل‌استفاده برای مراحل بعدی، می‌تواند بسیار چالش‌برانگیز باشد:

- احتمال همراه بودن با نویز به دلیل کیفیت حسگرهای مربوطه
 - نیاز به همجوشی^۱ و اجتماع^۲ داده‌ها به دلیل تولید شدن آن‌ها از منابع متعدد
 - دارا نبودن بسیاری از خواص سری‌های زمانی مرتب (رخ دادن وقایع در زمان‌ها و مکان‌های متفاوت)
 - حجم عظیم داده‌ها (بی‌پایان بودن جریان ثبت وقایع در یک‌خانه هوشمند)
 - پیچیده‌تر شدن ماهیت داده‌ها با تعدد افراد و اعمال موازی یا تودرتو
- استفاده از تکنیک‌های یادگیری عمیق^۳ در فاز پیش‌پردازش این داده‌ها، خصوصاً در مرحله کاهش ابعاد^۴ و استخراج ویژگی‌ها^۵ می‌تواند بسیار مفید و تأثیرگذار باشد.

۲-۱-۲ افعال

هر حسگر به‌طور ادواری داده‌های محسوس خود را در قالب سه‌تایی رابطه (۱-۲) که «رویداد حسگر»^۱ نامیده می‌شود، برای واحد جمع‌آوری ارسال می‌کند:

^۱ Fusion

^۲ Association

^۳ Deep Learning

^۴ Dimensionality Reduction

^۵ Feature Extraction

$$e = \langle t, s, m \rangle \quad (1-2)$$

در این سه تایی، t به معنای زمان، s به معنای شناسه حسگر و m به معنای پیام ارسالی است. با این تعریف، هر فعل یا کنش را می‌توان مطابق رابطه (۲-۲) به‌عنوان یک توالی متناهی n تایی از رویدادهای مرتبط با حسگرهای متناظر در نظر گرفت:

$$a = \langle e_1, e_2, \dots, e_n \rangle \quad (2-2)$$

در سلسله‌مراتب انتزاع، حرکت داده‌هایی که از حسگرها حاصل شده‌اند به تدریج باعث شکل‌گیری و پیدایش افعال از میان آن‌ها خواهد شد. افعال موردبررسی ممکن است ساده (باز کردن شیر آب، ...) یا پیچیده (درست کردن غذا، ...) باشند. هر فعل پیچیده از چند فعل ساده تشکیل شده است. از منظری دیگر، افعال می‌توانند در ارتباط باهم در یکی از این سه گروه قرار بگیرند:

- افعال پیاپی^۲

افعالی هستند که یکی پس از دیگری و مستقل از هم انجام می‌شوند مانند:
پاسخگویی به تلفن، درست کردن غذا، ...

- افعال متداخل^۳

فرد، دو یا چند فعل را در میان یکدیگر انجام می‌دهد و با قطع و وصل متداخل این افعال، در میان آن‌ها رفت‌وآمد می‌کند. مثلاً برای پاسخگویی به تلفن درست کردن غذا را متوقف می‌کند و پس از پاسخگویی به پختن غذا بازمی‌گردد.

- افعال هم‌زمان^۴

فرد، دو یا چند فعل را به‌صورت هم‌زمان و در طول یکدیگر انجام می‌دهد. مثلاً در حال طبخ غذا به تلفن نیز پاسخ می‌دهد

^۱ Sensor Event

^۲ Sequential Activities

^۳ Interleaved Activities

^۴ Concurrent Activities

۲-۱-۳ پردازش و تحلیل افعال

پس از اینکه داده‌های خام در قالب بسته‌های درشت‌دانه و مجردتری به نام افعال بسته‌بندی شدند، کار تحلیل بسته‌های مذکور آغاز می‌گردد. این تحلیل شامل تشخیص یا کشف یا پیش‌بینی افعال رخ داده در وقایع است.

تشخیص افعال^۱ در گروه روش‌های بانظارت^۲ قرار می‌گیرد. مسئله اصلی در تشخیص، نگاشت بسته‌های افعال (به دست‌آمده از بسته‌بندی داده‌های حسگرها) به گروهی از برچسب‌های از پیش تعریف شده می‌باشد. با توجه به اینکه افعال انسانی بسیار متنوع هستند و امکان برچسب‌گذاری همه آن‌ها برای انواع حسگرها وجود ندارد، این نوع از تحلیل معمولاً زمانی به کار گرفته می‌شود که تنها شناسایی انواع خاص و محدودی از افعال مدنظر باشد. همچنین تشخیص ناهنجاری در یک سیستم شناسایی افعال، به این معناست که فرد، یکی از اعمال برچسب خورده و شناخته‌شده را به شیوه متفاوتی انجام می‌دهد. در صورتی که یک‌رشته از دریافت‌های حسگرها با هیچ‌یک از افعال یادشده مطابقت نکند، ممکن است با یک ناهنجاری یا یک فعل هنجار بدون برچسب^۳ روبرو باشیم. لازم به ذکر است که بسیاری از ساختارهای عمیق می‌توانند در کنار رده‌بندهای رایج نظیر ماشین بردار پشتیبان یا SVM^۴ قرار گرفته و نتایج درخشانی را تولید کنند.

کشف افعال^۵ (برخلاف تشخیص افعال) یک مکانیسم بی‌ناظر است. هیچ برچسب و کلاس و فعل از پیش تعریف‌شده‌ای وجود ندارد و رشته‌های مقادیر دریافتی از حسگرها بر اساس معیارهایی مثل تواتر، هم‌زمانی، مجاورت، ... خوشه‌بندی می‌شوند. در خلال این فرایند هر رشته بر اساس فاصله یا شباهتی که با گروه‌های موجود دارد به عضویت یکی از آن‌ها درمی‌آید، یا با تشکیل یک گروه جدید، خود در مرکز آن قرار می‌گیرد. از آنجاکه این روش به هیچ برچسب یا فعل از پیش تعریف‌شده‌ای

^۱ Activity Recognition (AR)

^۲ Supervised

^۳ Unlabeled Activity

^۴ Support Vector Machine

^۵ Activity Discovery (AD)

محدود نمی‌شود، به کاربردهای عالم واقع نزدیک‌تر است. در یک سیستم کشف افعال، رشته‌ای را به‌منزله ناهنجاری تلقی می‌کنیم که رؤیت آن، پیش‌ازاین، بسیار نادر بوده یا اصلاً دیده نشده باشد. یک نقطه قوت ساختارهای عمیق، توانایی آن‌ها در بهره‌گیری از حجم انبوه داده‌های بدون برچسب در قالب یک فرایند غیر نظارتی است که همین موضوع این ساختارها را به ابزار مناسبی برای کشف افعال تبدیل کرده است.

در بحث پیش‌بینی افعال^۱ با نگاه به آینده درصدد پاسخگویی به یکی از دو پرسش زیر یا هر دوی آن‌ها هستیم (البته با فرض اینکه از جریان اتفاقات و افعال تا لحظه حاضر اطلاع داریم):

۱. آیا می‌توانیم کنش بعدی فرد موردنظر را حدس بزنیم؟

۲. آیا می‌توانیم زمان وقوع مجدد یک فعل خاص را پیش‌بینی کنیم؟

برای پاسخگویی به سؤال اول معمولاً از روش‌های رایج در رشته کاوی^۲ مثل روش‌های فشرده‌سازی تخمینی استفاده می‌کنند و جهت جواب دادن به سؤال دوم غالباً از شیوه‌های موجود در پیش‌بینی سری‌های زمانی مثل ARIMA بهره می‌جویند. پیش‌بینی افعال، خصوصاً پیش‌بینی افعال غیرمعمول، به‌ویژه در سامانه‌های مراقبت بیماران و سالمندان اهمیت فراوانی دارد. چراکه پیش‌بینی وقوع یک ناهنجاری مثل فراموش کردن نوبت دارو یا وقوع حمله قلبی از تشخیص آن بسیار مفیدتر است.

۲-۱-۴ پردازش ناهنجاری

تشخیص رفتارهای ناهنجارانه (تشخیص امور نامتعارف^۳) در بین فعالیت‌های روزانه یک عامل کاری پرچالش است. این مقوله در زمینه‌هایی نظیر امنیت شبکه، تشخیص تقلب‌های پولی و بانکی، تصویربرداری پزشکی و تشخیص خطا در تولیدات صنعتی به مرزهای پختگی نزدیک شده، اما در

^۱ Activity Prediction (AP)

^۲ Sequence Mining

^۳ Outlier Detection

حوزه منازل هوشمند و تحلیل افعال و رفتار افراد هنوز بسیار جوان محسوب می‌شود. شاید یکی از دلایل این جوانی آن باشد که رفتارها و کنش‌های انسانی بسیار زیاد است و افراد، بسته به معیارهای مثل سن و سواد و فرهنگ فعالیت‌های ساده و روزانه خود را نیز به شیوه‌های متفاوتی انجام می‌دهند. به همین دلیل، تعریف رفتار نرمال و توالی رشته‌های مفردی که حاصل و سرجمع آن‌ها یک کنش معمول^۱ به شمار بیاید کار بسیار دشواری است. عمده کارهایی که تاکنون در زمینه^۲ تشخیص رفتار ناهنجار انجام شده، تمرکز خود را بر کشف افعالی مثل افتادن، از خواب برنخاستن، اقامت یا غیبت طولانی در یک مکان یا زمان مشخص، یا تکرار مراجعات به یک اتاق قرار داده‌اند.

در مرجع [4] که مجموعه داده ارزشمندی را نیز در حوزه تشخیص افعال ناهنجار ایجاد کرده، زمینه‌های ناهنجاری در حیطه افعال انسانی به صورت‌های زیر تقسیم‌بندی شده است:

- Abnormal Behaviour
 - Temporal
 - Duration
 - Time of Occurrence
 - Spatial
 - Pattern of Action
- Environment Changes
- Context Switching

طبیعی است که تشخیص توأم همه انواع یا حتی برخی از آن‌ها کار بسیار دشواری است.

۲-۲ تشخیص ناهنجاری

ناهنجاری‌ها نقاط داده‌ای نادری هستند که با اکثریت داده‌ها فرق دارند. مرجع مشهور [5] رسماً ناهنجاری را به صورت ذیل تعریف نمود:

«خروجی حاصل از یک مشاهده که با سایر مشاهدات در تناقض باشد و موجب بروز این سوءظن

می‌شود که توسط یک مکانیسم متفاوتی تولید شده باشد.»

^۱ Normal Activity

واژه ناهنجاری تنها عبارت استفاده‌شده در ادبیات خودش نیست. نقاط دورافتاده^۱، ناهنجاری‌ها^۲ و انحرافات^۳ اصطلاحات دیگری هستند که در این ادبیات برای بیان این پدیده بکار می‌روند. تکنیک‌های تشخیص ناهنجاری زمانی استفاده می‌شوند که ناهنجاری‌ها علاقه خاص کاربر باشند. برای مثال در یک حساب بانکی، الگوی تراکنش‌ها معمولاً قابل پیش‌بینی است و زمانی که یک تراکنش غیرعادی رخ می‌دهد، احتمالاً بیانگر یک استفاده غیرمجاز از حساب بانکی است. کاربردهای بسیاری برای تشخیص ناهنجاری متصور است، از جمله می‌توان موارد ذیل را برشمرد: جعل کارت اعتباری، تشخیص نفوذ، تشخیص بیماری، علوم زمین، وضع قوانین و حملات شبکه‌های حسگر [6].

۲-۲-۱ چالش‌ها

مسئله تشخیص ناهنجاری را می‌توان به‌طور اجمالی برای شناسایی الگوها یا نقاط داده‌ای که در ناحیه طبیعی خود جای نگرفته‌اند، بررسی نمود؛ بنابراین، با شناسایی یک ناحیه طبیعی می‌توان مسئله را حل کرد و هر نقطه خارج از این ناحیه طبیعی به‌عنوان ناهنجاری علامت‌گذاری می‌شود. البته این مسئله خود در عمل بسیار چالش‌برانگیز است [7]. برای بیان بهتر موضوع در ذیل چندین مسئله که مربوط به سختی‌ها و پیچیدگی‌های تشخیص ناهنجاری می‌باشند ارائه می‌شوند:

- تعریف اینکه اساساً به چه چیزی هنجار اطلاق شود کار سختی است و معمولاً تمیز دادن حالات طبیعی و غیرطبیعی از یکدیگر اصلاً واضح نیست،
- رفتار طبیعی همیشه در حال تغییر و تکامل است و تعریف آن از دامنه‌ای به دامنه دیگر کاملاً عوض می‌شود،
- ناهنجاری‌ها اشکال و انواع مختلفی بخصوص در دامنه کاربردی که در آن مقیم می‌شوند دارند،

^۱ Outliers

^۲ Abnormalities

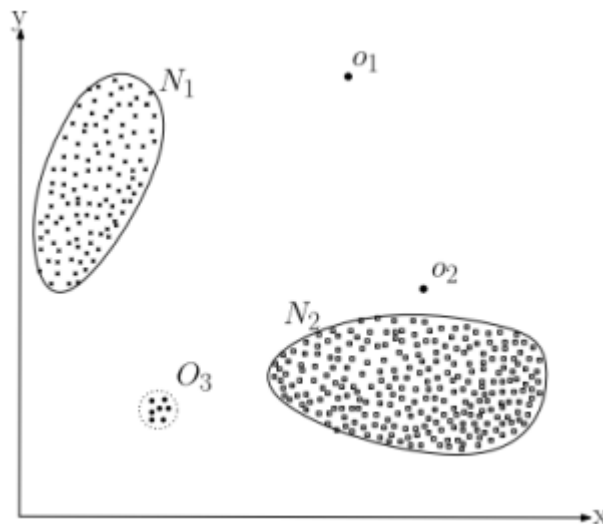
^۳ Deviants

- در دسترس بودن مجموعه داده‌های نمونه با یا بدون برچسب،
- تفکیک درست ناهنجاری‌ها از بین داده‌های دارای اعوجاج^۱.

بنابراین، عمل تشخیص ناهنجاری خیلی واضح و بدیهی نیست و از یک دامنه به دامنه دیگر متفاوت است. نویسندگان مرجع [7] آورده‌اند که بسیاری از تکنیک‌های تشخیص ناهنجاری موجود برای حل مشکلات یک دامنه بخصوص صورت‌بندی می‌شوند و عموماً در سایر دامنه‌ها قابل‌استفاده و بهره‌گیری نیستند.

ماهیت ناهنجاری‌ها جنبه مهمی در تشخیص ناهنجاری دارند. ناهنجاری‌ها به سه دسته تقسیم‌بندی می‌شوند:

- ناهنجاری‌های نقطه‌ای^۲: جایی که نقاط داده‌ها مغایر با سایر داده‌ها در نظر گرفته می‌شوند. در شکل ۱-۲ نواحی N_1 و N_2 را نرمال در نظر می‌گیریم زیرا بخش زیادی از داده‌ها در این دو ناحیه قرار دارند. از طرف دیگر O_1 ، O_2 و O_3 از نواحی نرمال فاصله‌دارند و به‌عنوان ناهنجاری در نظر گرفته می‌شوند.

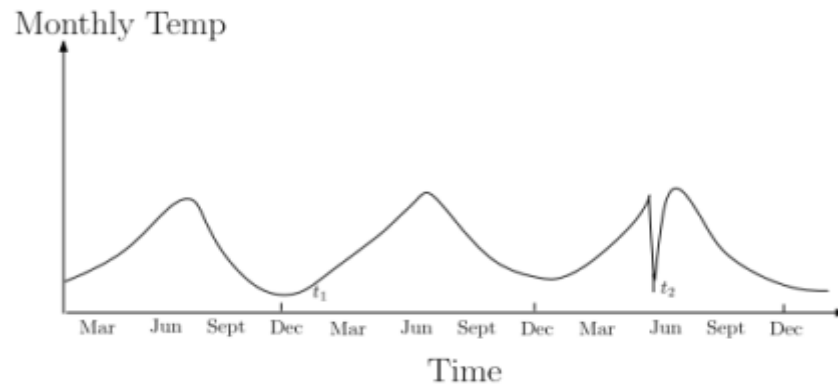


شکل ۱-۲: ناهنجاری‌های نقطه‌ای در فضای دوبعدی [7]

^۱ Noise

^۲ Point Anomaly

- ناهنجاری‌های محدوده‌ای یا زمینه‌ای^۱: که در آن محدوده نقاط داده‌ای ظاهر شده غیرعادی است نه خود نقاط داده. چنانچه در شکل ۲-۲ دیده می‌شود، نقطه داده t_1 متناظر با نقطه داده t_2 است ولی نقطه t_2 به‌عنوان ناهنجاری در نظر گرفته می‌شود زیرا این نقطه در یک محدوده غیرعادی ظاهر شده است.

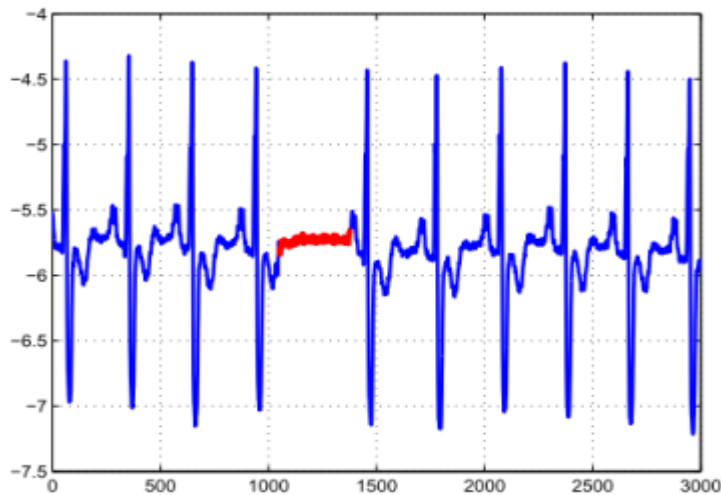


شکل ۲-۲: ناهنجاری‌های زمینه‌ای در داده‌های دمای ماهانه [7]

- ناهنجاری‌های تجمیعی^۲: که مجموعه‌ای از داده‌ها به‌عنوان داده‌های ناهنجار و غیرعادی در نظر گرفته می‌شوند نه به این خاطر که هر نقطه داده ناهنجار هست بلکه به این دلیل که حضور مجموعه این نقاط داده‌ها باهم غیرعادی هستند. در شکل ۳-۲ نقاط داده‌ای در یک الکترودیگرام به‌عنوان ناهنجاری در نظر گرفته شده‌اند که این به دلیل حضورشان به‌عنوان یک مجموعه در این داده‌ها است نه به دلیل خود نقاط داده‌ای.

^۱ Contextual Anomaly

^۲ Collective Anomaly



شکل ۲-۳: ناهنجاری مجموعه‌ای در یک الکترودیگرام انسانی [7]

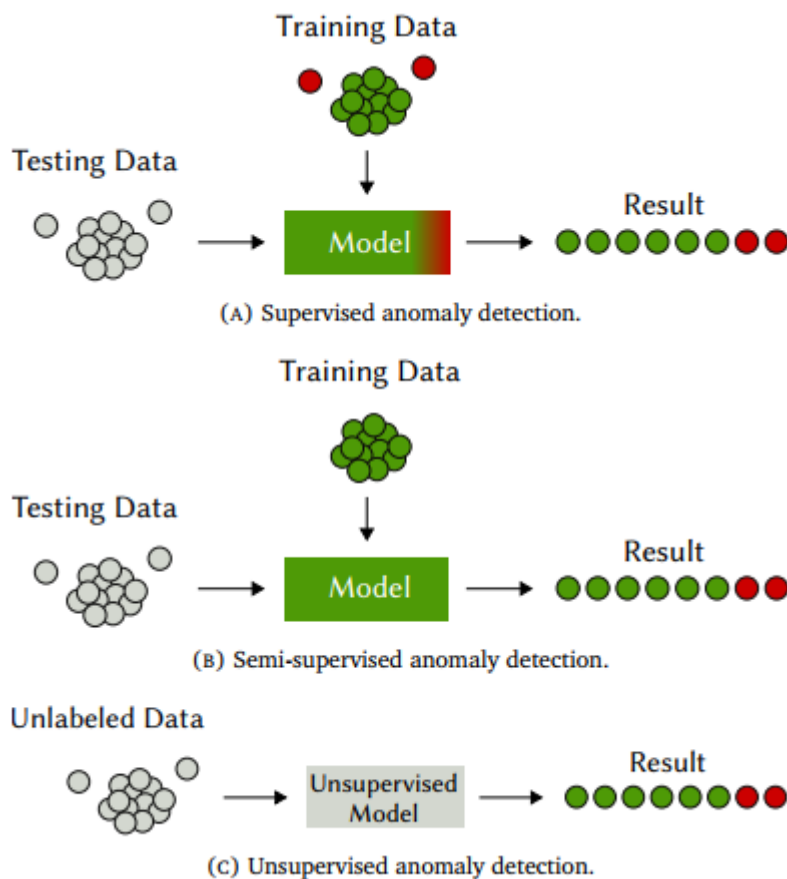
جنبه مهم دیگر در تشخیص ناهنجاری دسترس‌پذیری برجسب‌های داده‌ای است. تکنیک‌های تشخیص ناهنجاری از منظر برجسب زدن نقاط داده‌ای را در سه دسته می‌توان تقسیم‌بندی نمود (شکل ۲-۴):

- باناظر^۱: فرض این است که مجموعه داده آموزشی با برجسب‌هایی برای شناسایی نقاط داده‌ای نرمال از غیر نرمال وجود دارند.
- نیمه نظارتی^۲: فرض این است که مجموعه داده آموزشی در دسترس تماماً نرمال هستند و انحراف از این نقاط داده‌ای نرمال به‌عنوان ناهنجاری در نظر گرفته می‌شود.
- بی‌ناظر^۳: فرض بر این است که داده‌ها برجسب‌گذاری نشده‌اند و به مجموعه داده‌های آموزشی نیز نیازی نیست. تکنیک‌های مورد استفاده در این دسته، فرض را بر این قرار می‌دهند که اکثریت نقاط داده‌ها نرمال هستند و بنابراین، نقاط داده را به دسته‌هایی خوشه‌بندی نموده و هر نقطه داده ایزوله شده به‌عنوان ناهنجاری در نظر گرفته خواهند شد.

^۱ Supervised

^۲ Semi-Supervised

^۳ Unsupervised



شکل ۲-۴: تشخیص ناهنجاری و رویکردهای با دسترس پذیری داده‌های آموزشی و آزمون [8]

۲-۲-۲ تکنیک‌های تشخیص ناهنجاری

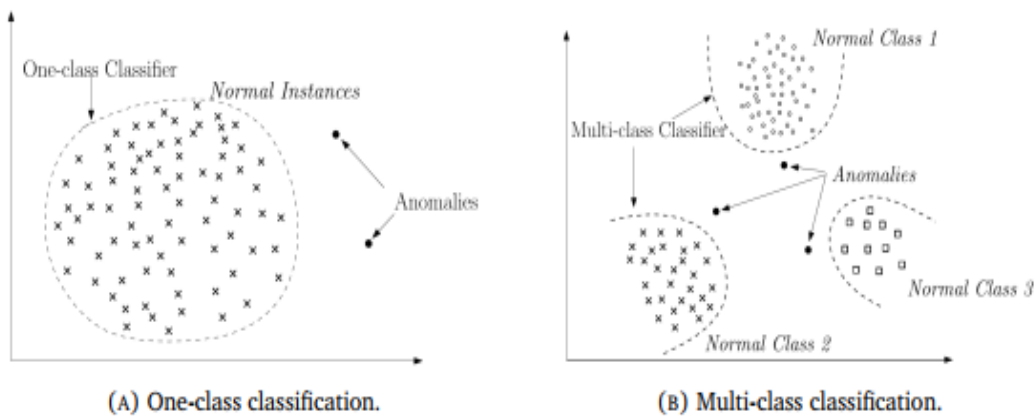
در این بخش، چندین رویکرد و فن شناسایی ناهنجاری‌ها ارائه می‌شوند. تکنیک بنیادی که هر رویکرد مبتنی بر آن است به‌عنوان فاکتور طبقه‌بندی استفاده خواهد شد.

۱-۲-۲-۲ رده‌بندی

در این تکنیک‌ها داده‌های آموزشی شناسایی شده برای یادگیری مدل برچسب‌گذاری می‌شوند. ایده اصلی، آموزش یک رده‌بند^۱ روی نقاط داده نرمال هست و پس‌از آن دقت مدل را بر اساس این مجموعه داده آموزشی روی داده‌هایی که تاکنون هرگز دیده نشده است، ارزیابی می‌کنند، به این داده‌ها، داده-

^۱ Classifier

های آزمون گفته می‌شود. از این جنبه که چه تعداد رده (کلاس) آموزش داده می‌شود، این تکنیک‌ها را می‌توان به دودسته تقسیم‌بندی نمود. تکنیک‌های تشخیص ناهنجاری تک-کلاسه و چند-کلاسه. در تکنیک تک-کلاسه فرض می‌شود که داده‌های آموزشی همگی نرمال هستند؛ بنابراین، مدل ویژگی-های این نقاط داده را یاد گرفته و آن‌ها را به‌عنوان نقاط داده‌ای نرمال رده‌بندی می‌کند. هر نقطه داده‌ای که در این مجموعه قرار نگیرد، توسط مدل به‌عنوان ناهنجاری رده‌بندی خواهد شد. چنانچه در شکل ۲-۵-A نشان داده‌شده، مدل تک-کلاسه تمام داده‌های نرمال را در یک مجموعه بزرگ رده‌بندی نموده و هر نقطه خارج از این رده به‌عنوان ناهنجاری برچسب می‌خورد. برای یادگیری ناحیه نرمال، الگوریتم‌های مختلفی استفاده می‌شوند. یکی از الگوریتم‌های رایج ماشین بردار پشتیبان است [9]. نویسندگان منبع [10] از مدل کرنل فیشر برای یادگیری کلاس داده‌های نرمال استفاده نموده‌اند. رده چند-کلاسه چنانچه در شکل ۲-۵-B نشان داده‌شده است، مشابه همان تک-کلاسه است به‌استثنا این‌که به‌جای یادگیری یک ناحیه نرمال، چند ناحیه نرمال آموزش داده می‌شود. پدیدآورندگان مراجع [11] و [12] در پژوهش خود از تکنیک‌های رده‌بندی چند-کلاسه برای تشخیص ناهنجاری استفاده نموده‌اند.



شکل ۲-۵: تکنیک‌های تشخیص ناهنجاری مبتنی بر رده‌بندی [7]

تکنیک‌های رده‌بندی بر اساس الگوریتمی که در مدل استفاده می‌شوند را می‌توان به چندین دسته از جمله ماشین بردار پشتیبان، شبکه‌های عصبی مصنوعی، شبکه‌های بیزی و قوانین دسته‌بندی نمود. ماشین بردار پشتیبان [13] نواحی نرمال را بر اساس نمونه داده‌های آموزشی یاد می‌گیرد و علاوه بر آن با تکنیک کرنل^۱ [14] می‌تواند نواحی غیرخطی در ابر صفحات^۲ را یاد بگیرد. توسعه این الگوریتم با کرنل‌هایی شبیه RBF^۳ به مدل قابلیت امکان یادگیری نواحی پیچیده‌تری را نیز ارائه می‌دهد. اساساً هر موقع صحبت از ناهنجاری باشد پای SVM و افزونه‌های آن در میان است. کینگ و همکاران [15] فنی را برای تشخیص تازگی^۴ در نیروگاه‌های برق پیشنهاد داده‌اند. دیوی و گودزیل [16] الگوریتم SVM را برای شناسایی ناهنجاری‌ها در سیگنال‌های رادیویی بسط دادند. الگوریتم SVM تکنیک رایج و مرسوم بسیاری از محققان در تشخیص نفوذ به میزبان‌های شبکه‌های کامپیوتری است. از جمله این پژوهش‌ها می‌توان به کارهای انجام‌شده توسط هلر و همکاران [9] و لازارویک و همکاران [17] اشاره نمود. ما و پرکینز [18] SVM ها را در توالی داده‌های موقتی بکار بستند. سونگ و همکاران (۲۰۰۲) نوعی از SVM موسوم به ماشین بردار پشتیبان قوی^۵ یا RSVM پیشنهاد دادند که در مرحله آموزش در تعیین ناهنجاری‌ها قوی‌تر عمل می‌کند. هو و همکاران [19] RSVM را برای محاسبه تشخیص نفوذ سامانه‌های میزبان بکار بسته‌اند.

شبکه‌های عصبی مصنوعی^۶ را می‌توان در مسائل تک و چند کلاسه بکار برد. این تکنیک با بخشی از مجموعه داده‌های مدل شبکه عصبی را آموزش می‌دهد. سپس مجموعه داده‌های آزمون به شبکه تزریق می‌شوند و هر موقع که شبکه نقطه داده‌ای را رد^۷ کند به‌عنوان نقطه داده ناهنجار برچسب می‌خورد [12] و [20]. بسیاری از انواع مختلف شبکه‌های عصبی در این ادبیات استفاده می‌-

^۱ Kernel Trick

^۲ Hyperplanes

^۳ Radial Basis Function

^۴ Novelty

^۵ Robust SVM

^۶ Artificial Neural Networks

^۷ Reject

شوند. آگوستین و فولکرت [21] از پرسپترون چندلایه^۱ برای تشخیص ناهنجاری‌ها استفاده نموده‌اند. تامسون و همکاران [22]؛ و دیاز و هولمن [23] از انواع مختلف شبکه‌های خودکار-باهم‌آیی^۲ برای تشخیص ناهنجاری‌ها استفاده نموده‌اند. گروک و هایر [24] و گروک [25]؛ و مورای [26]، از شبکه‌های هاپفیلد^۳ برای کشف ناهنجاری بهره برده‌اند. سرانجام ویلیامز [27] و هاوکینز [28] از شبکه‌های عصبی مقلد^۴ برای تشخیص ناهنجاری تک کلاسه استفاده نموده‌اند.

۳-۲ یادگیری عمیق

یادگیری عمیق^۵ متعلق به مجموعه‌ای کلی از الگوریتم‌های یادگیری است که با الهام از شیوه یادگیری و ویژگی‌ها ساختاری ذهن انسان، تلاش می‌کنند تا یک نگرش یکسان و یکپارچه برای یادگیری ارائه دهند. نماینده معروف و شناخته‌شده این روش‌ها، روش شبکه‌های عصبی مصنوعی است که از قضا ارتباط بسیار نزدیک و تنگاتنگی با یادگیری عمیق دارد تا جایی که یادگیری عمیق را رنسانس یا رستاخیز شبکه‌های عصبی نیز نامیده‌اند.

ایده اصلی یادگیری عمیق این است که ذهن انسان برای یادگیری و درک مفاهیم جدید از یک ساختار سلسله مراتبی استفاده می‌کند. سطوح پایینی این ساختار حاوی مفاهیم سطح پایین هستند و با حرکت به سمت بالا، مفاهیم موجود در هر سطح از جهت شناختی فربه‌تر می‌شوند تا جایی که در بالاترین سطوح مفاهیم سطح بالا قرار می‌گیرند. بعلاوه هر سطح خوراک و ورودی سطوح بالاتر از خود را فراهم می‌کند. مثال سنتی این مفهوم، ادراک بصری و بازشناسی اشیاء موجود در یک تصویر است. شکل ۲-۶ نشان می‌دهد که چگونه می‌توان با به‌کارگیری یک ساختار سلسله مراتبی از مفاهیم، اشیاء موجود در یک تصویر را از هم تفکیک کرد.

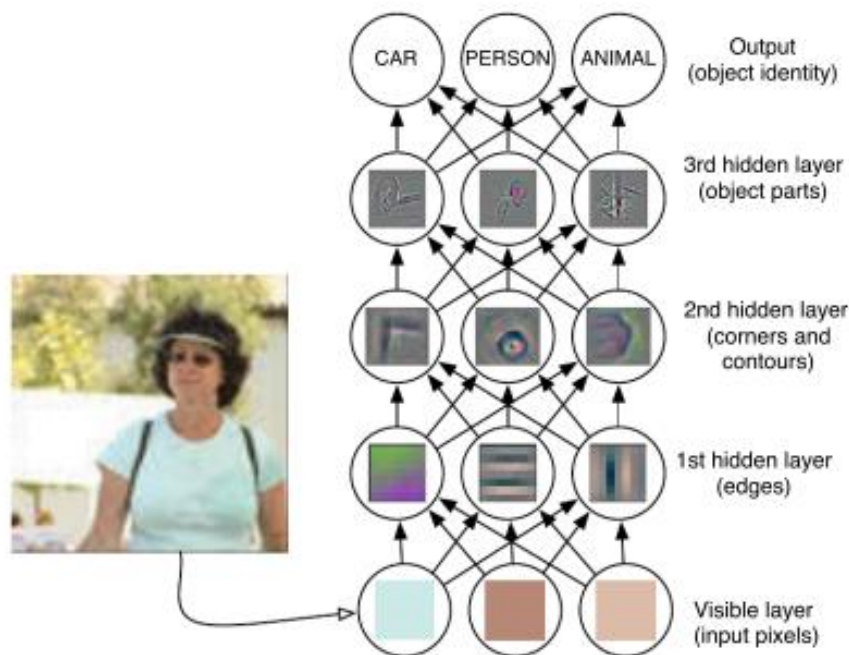
^۱ Multi-Layered Perceptron

^۲ Auto-Associative

^۳ Hopfield Neural Networks

^۴ Replicated Neural Networks

^۵ Deep Learning



شکل ۲-۶: افزایش انتزاع در لایه‌های یک ساختار سلسله مراتبی عمیق [29]

تحقیق در شکل ۲-۶ و مرور شیوه‌هایی که در تماشای یک صحنه برای تفکیک اشیا بکار گرفته می‌شود، درستی ایده یادگیری عمیق را ثابت می‌کند. بعید است که ذهن ما در زمان تحلیل یک صحنه که تصویر آن را از طریق دستگاه بینایی دریافت کرده، از یک روش پیکسل به پیکسل یا حتی ناحیه به ناحیه استفاده کند. چراکه در این صورت، علیرغم تعداد زیاد نورون‌ها و به‌کارگیری پردازش موازی، تحلیل یک صحنه پیچیده باید بسیار طولانی‌تر از آنچه اکنون هست می‌شد. منطقی است بپذیریم که ذهن با تشخیص مفاهیم سطح پایین‌تری مثل لبه‌ها کار خود را آغاز می‌کند و به کمک توازی که از اجتماع انبوه نورون‌ها حاصل شده است، به سرعت سطح مفهومی کشفیات و ادراکات خود را بالاتر برده تا نهایتاً به بالاترین سطح، یعنی کشف و تفکیک اشیا برسد.

از سوی دیگر، یادگیری عمیق با مفهوم بازنمایی^۱ ارتباط نزدیکی دارد [30]. آنچه در یادگیری ماشین سنتی مرسوم است این است که ما برای تدوین یک روش یادگیری، مجموعه‌ای از ویژگی‌ها را

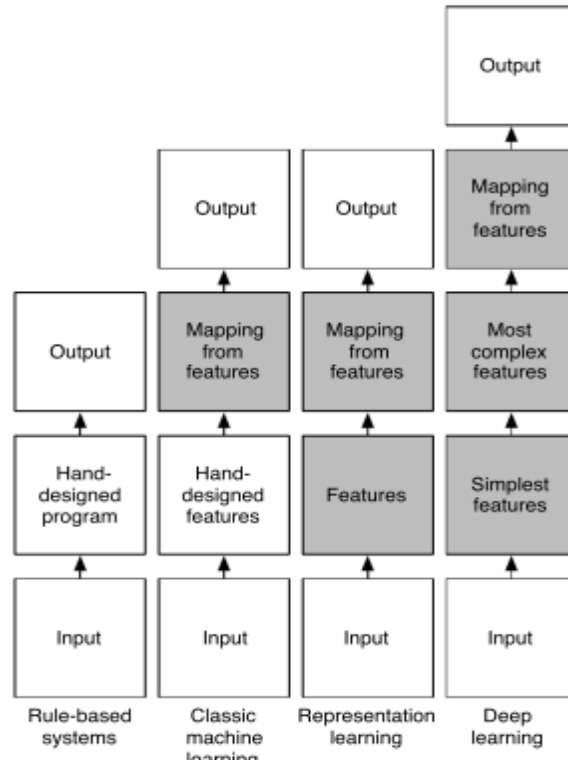
^۱ Representation

به صورت دستی استخراج می‌نماییم و سعی می‌کنیم با استفاده از نگاشت‌های خطی یا غیرخطی، مسیری را برای رسیدن از این ویژگی‌ها به خروجی‌های صحیح و مورد نظر بیابیم. استخراج و انتخاب ویژگی‌ها مناسب و کارآمد (به صورت دستی) وابستگی زیادی به مهارت افراد داشته و می‌تواند به کاری دشوار و زمان‌بر تبدیل شود. بعلاوه ویژگی‌هایی که در حل یک مسئله خاص بسیار خوب عمل می‌کنند، الزاماً برای حل مسائل دیگر، مفید و کارا نخواهند بود.

بنابراین انتخاب شیوه مناسب برای بازنمایی داده‌های ورودی (استخراج و انتخاب ویژگی‌های مناسب از میان انبوه ورودی‌ها) می‌تواند در کیفیت و کارایی روش یادگیری تأثیر فراوانی داشته باشد. در شیوه‌های یادگیری مبتنی بر بازنمایی^۱ از جمله یادگیری عمیق، این کار به وسیله ماشین و به صورت خودکار انجام می‌شود. استخراج ویژگی‌های مناسب یک کار ماهیتاً مفهومی است و چگونگی خودکارسازی آن می‌تواند محل بحث و مناقشه فراوانی باشد، اما آنچه یادگیری عمیق را از سایر شعبات یادگیری مبتنی بر بازنمایی متمایز می‌کند، همان سلسله مراتبی بودن آن است. به عبارت دیگر در یادگیری عمیق ابتدا اولین سطح میانی (بعد از سطح ورودی) ساده‌ترین ویژگی‌ها را استخراج می‌کند. این ویژگی‌ها بسیار ساده بوده و رنگ و بوی مفهومی^۲ کمتری دارند. با حرکت به سمت بالا، هر سطح ویژگی‌های مأخوذ از سطوح تحتانی را دریافت کرده و با اعمال پالایش و تبدیل (معمولاً غیرخطی) آن‌ها را به ویژگی‌های سطح بالاتری نگاشت می‌دهد. الگوریتم رسیدن به نگاشت یادشده می‌تواند باناظر، بی‌ناظر، یا ترکیبی باشد. این مفاهیم در شکل ۲-۷ نشان داده شده است.

^۱ Representation Learning

^۲ Cognitive



شکل ۲-۷: مقایسه شیوه استخراج ویژگی‌ها در مکاتب متفاوت یادگیری ماشین [29]

در یک جمع‌بندی، یادگیری عمیق روشی در یادگیری ماشین است که در آن یک ساختار سلسله‌مراتبی که عمدتاً از تعداد زیادی لایه تشکیل شده است، کار استخراج ویژگی‌ها و انجام فرایند یادگیری را بر عهده دارد. معمولاً همه سطوح، درگیر تبدیلات و نگاشت‌های غیرخطی بوده و هر سطح نسبت به سطوح بالایی از انتزاع و مفهوم‌گرایی کمتری برخوردار است. عمیق بودن این ساختار که برخلاف ساختارهای رایج یادگیری ماشین سنتی (مثل SVM) از سطوح متعددی تشکیل شده، باعث شده است تا این روش را یادگیری عمیق بنامند.

در حال حاضر یادگیری عمیق در روش‌های متعددی نظیر «شبکه‌های عصبی عمیق»^۱، «شبکه‌های عصبی عمیق پیچشی»^۲، «شبکه‌های باور عمیق»^۳، «پشته کد کننده‌های خودکار»^۴ و

^۱ Deep Neural Networks (DNN)

^۲ Convolution deep Neural Networks (CNN)

^۳ Deep Belief Networks (DBN)

^۴ Stack Auto-Encoders (SAE)

«شبکه‌های عصبی بازگشتی»^۱ بکار گرفته می‌شود. این روش‌ها، تغییرات بنیادین و مهمی را در ارکان هوش مصنوعی به‌ویژه بینایی ماشین، تشخیص خودکار گفتار و پردازش زبان طبیعی موجب گردیده‌اند. در این ساختارها بجای آنکه ویژگی‌ها را مستقیماً از داده‌های خام بیرون بکشند، این داده‌ها را از مسیر لایه‌های متعدد می‌گذرانند تا در هر لایه، یک تابع نگاشت غیرخطی (مثل sigmoid یا tanh) مانند یک پالایه جلوه جدیدی از داده‌ها را استخراج کرده و این جلوه را به‌عنوان ورودی در اختیار لایه بعدی بگذارد تا در نتیجه سطح انتزاع^۲ افزایش پیدا کند. نتیجه کار، ساختاری سلسله‌مراتبی است که با دریافت یک مجموعه داده چندبعدی^۳ ابعاد آن را به شکل غیرخطی کاهش داده و ویژگی‌های مناسب را به‌طور خودکار می‌آموزد. این ساختار به‌ویژه در سال‌های اخیر بسیار کارآمد و مفید واقع شده است، چراکه در این سال‌ها مهندسان و متخصصان علوم داده با انبوهی از داده‌های چندبعدی روبرو بوده و هستند. پردازش درست داده‌های مذکور و اخذ نتایج دقیق و قابل‌اعتماد، مستلزم استخراج ویژگی‌هایی است که به شیوه‌ای مجمل و کامل اصلی‌ترین وجوه مشخصه این داده‌ها را در اختیار بگذرانند.

۲-۳-۱ شبکه‌های پیچشی (CNN)

ساختار شبکه‌های CNN کاملاً متصل^۴ نیست، یعنی هر نورون لایه L - برخلاف شبکه‌های عصبی سنتی - به تمام نورون‌های لایه $L + 1$ وصل نمی‌شود. به مجموعه‌ای از نورون‌های لایه L که به یک نورون مشترک و واحد در لایه بعدی متصل باشند، ناحیه ادراکی^۵ می‌گویند. ساختار این شبکه از ساختار ادراکی و به‌ویژه بینایی انسان اخذ شده است که در آن، تعدادی از سلول‌های بینایی ورودی خود را برای یک سلول مغزی ارسال می‌کنند تا هم‌جواری این ورودی‌ها در پردازش تصویر لحاظ شود.

^۱ Recurrent Neural Networks (RNN)

^۲ Abstraction

^۳ High Dimensional

^۴ Fully Connected

^۵ Receptive Field

طبیعی است که این نواحی باهم همپوشانی دارند، البته وجود این همپوشانی در تعریف CNN الزامی نیست.

ویژگی دیگر شبکه‌های CNN این است که لایه‌های آن برخلاف شبکه‌های عصبی رایج، هم‌کارکرد نیستند. در CNN دست‌کم دو نوع لایه وجود دارد که یکی از آن‌ها بر مبنای عملگر کانولوشن و دیگری (لایه متحد‌سازی^۱) بر مبنای عملگرهای دیگر مانند بیشینه‌سازی یا متوسط‌گیری کار می‌کند. این شبکه‌ها با گذر داده‌های ورودی از لایه‌های متعدد، به تدریج حجم و ابعاد این داده‌ها را کاهش می‌دهند، درحالی‌که هم‌زمان، ویژگی‌های ثابت و تغییرناپذیر آن‌ها (مجاورت‌های مکانی، مجاورت‌های زمانی، ...) را حفظ می‌کنند. CNN در هر لایه عملگر کانولوشن را روی داده‌های خود اعمال می‌کند. این عملگر برای داده‌های یک و دوبعدی در روابط (۳-۲) و (۴-۲) تعریف شده است:

$$c(n) = (f * g)(n) = \sum_{u=-\infty}^{\infty} f(u) * g(n-u) = \sum_{u=-\infty}^{\infty} f(n-u) * g(u) \quad (3-2)$$

$$c(m,n) = (f * g)(m,n) = \sum_{u=-\infty}^{\infty} \sum_{v=-\infty}^{\infty} f(u,v) * g(m-u,n-v) \quad (4-2)$$

سپس با اضافه کردن یک بایاس و استفاده از یک تابع غیرخطی مثل \tanh ، نواحی خاصی را از ورودی استخراج می‌کند که نقشه نگاشت^۲ نامیده می‌شوند. این نگاشت از رابطه (۵-۲) به دست خواهد آمد که در آن، h^k بیانگر k امین نقشه نگاشت در یک لایه دلخواه است:

$$h_{i,j}^k = \tanh((W^k * x)_{ij} + b_k) \quad (5-2)$$

در لایه‌های متحدسازی حجم و تفکیک‌پذیری داده‌های پیچش‌خورده کاهش زیادی می‌یابد تا صفات تغییرناپذیر آن‌ها با قدرت بیشتری جلوه کنند، جلوه‌ای که ممکن است به دلیل وجود نویز در ورودی کم‌رنگ شده باشد. چنانکه گفته شد این لایه‌ها ممکن است بر اساس توابع مجموع، بیشینه‌سازی یا

^۱ Pooling Layer

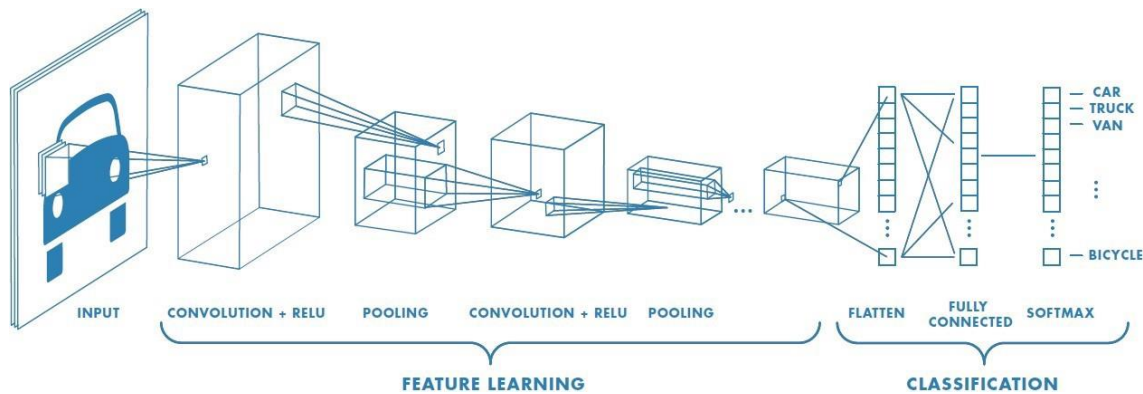
^۲ Feature Map

توابع دیگری عمل کنند. دو نمونه بیشینه‌سازی و میانگین‌گیری به ترتیب در روابط (۶-۲) و (۷-۲) تعریف شده‌اند:

$$v_{i,j}^h = \max_{1 \leq q \leq Q_i} (v_{(i-1),j}^{h+q}) \quad \forall h = 1, 2, \dots, N \quad (6-2)$$

$$v_{i,j}^h = \frac{1}{Q_i} \sum_{1 \leq q \leq Q_i} (v_{(i-1),j}^{h+q}) \quad \forall h = 1, 2, \dots, N \quad (7-2)$$

در این روابط، N تعداد لایه‌ها و Q اندازه ناحیه‌ای است که عمل متحدسازی روی آن انجام می‌پذیرد. شکل ۸-۲ یک CNN نمونه را به نمایش می‌گذارد.

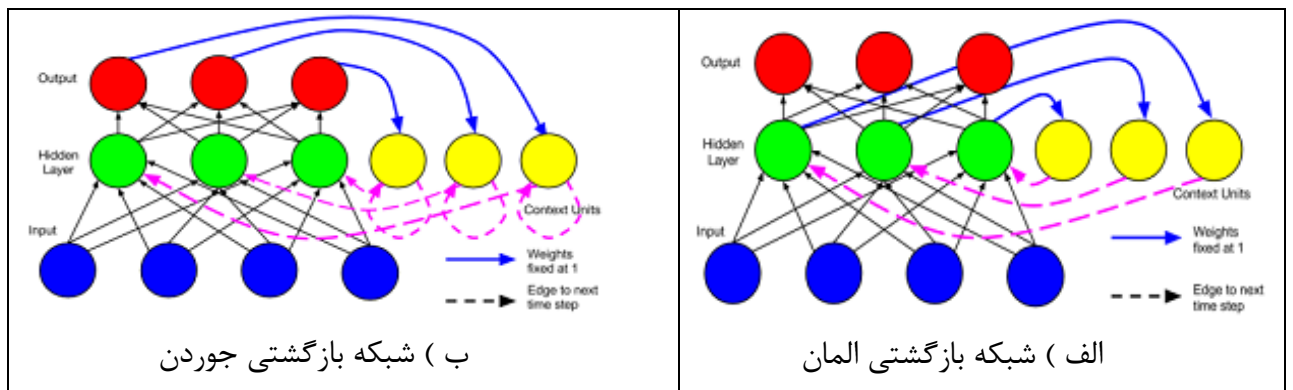


شکل ۸-۲: یک CNN نمونه [31]

۲-۳-۲ 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 رده‌بندی) 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

^۱ State



در خود حفظ کند. بدین ترتیب شبکه‌های بازگشتی در واقع از نوعی حافظه برخوردار هستند و در نتیجه مدل محاسباتی آن‌ها از شبکه‌های عصبی پیش‌خور^۱ قدرتمندتر است.

شکل ۲-۹: شبکه‌های عصبی بازگشتی الف) المان و ب) جوردن [32]

شبکه‌های بازگشتی از انواع کمی متصل تا کاملاً متصل گسترده شده‌اند. دو نوع از مشهورترین شبکه‌های بازگشتی سنتی، به‌وسیله المان [33] و جوردن [34] 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^۲ نام دارند ذخیره می‌شود. در هر گام از فرایند آموزش، خروجی‌های سلول‌های زمینه که از گام قبل باقی‌مانده‌اند 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 به‌جای لایه پنهان، از طریق لایه خروجی تأمین می‌گردد.

نمونه‌هایی از این دو شبکه را نشان می‌دهد.

^۱ Feed Forward Network

^۲ Context Cell

برای آموزش یک شبکه بازگشتی می توان از الگوریتم معمول انتشار پسگرد^۱ استفاده کرد. به این شکل که خروجی سلول های زمینه به مجموعه ورودی های لایه پنهان اضافه می شوند تا ساختار شبکه در بستر زمان گسترش یافته و از حالت بازگشتی خارج شود.

۲-۳-۱ انفجار و اختفای گرادیان

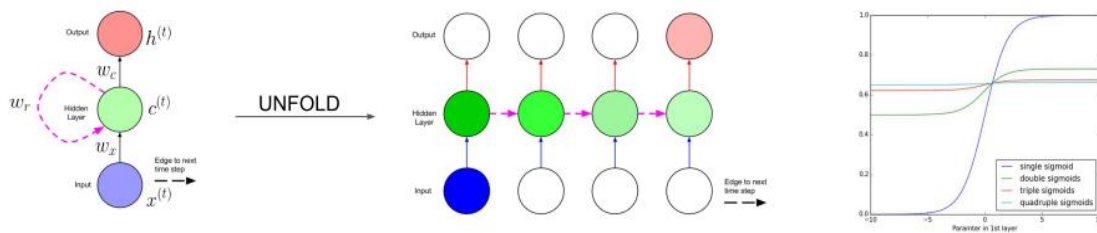
شبکه های بازگشتی سنتی از دو مشکل اختفا^۲ و انفجار^۳ گرادیان رنج می برند. میدانید که در هر مرحله از آموزش یک شبکه عصبی بازگشتی تعداد زیادی ضرب انجام می پذیرد. وجود مجموعه بزرگی از وزنه های بزرگ تر از یک و انجام ضرب های یادشده، شبکه را به سمت اوزان بسیار بزرگ یا همان انفجار گرادیان هدایت می کند. در نتیجه این پدیده، حرکت کل شبکه به سمت نقاط مطلوب از کنترل خارج می شود و شبکه با گام های بسیار بزرگ در اطراف نقاط جواب نوسان می کند. شبیه پروانه ای که قصد نشستن بر روی یک گل را دارد، اما با هر بال زدن در مضر بزرگی از اندازه فعلی خود ضرب می شود! البته مشکل انفجار گرادیان معمولاً از طریق محدود کردن وزن های شبکه قابل حل است.

اما مشکل اختفای گرادیان که مثل انفجار از فراوانی تعداد ضرب ها در فرایند یادگیری یک شبکه عصبی بزرگ ناشی می شود، مشکل بدخیمی است که در نتیجه وجود مجموعه بزرگی از وزنه های کوچک تر از یک ایجاد می گردد. در این حالت، گرادیان خطا به تدریج کوچک و کوچک تر می شود، در نتیجه انتشار پسگرد خطا و اصلاح وزنه های شبکه در اندازه های بسیار کوچک صورت می پذیرد تا جایی که نهایتاً انتشار خطا متوقف شده و شبکه عملاً حساسیت خود را نسبت به وقوع خطا از دست می دهد. از این نقطه به بعد، بدون اینکه شبکه به جواب یا حتی اطراف آن همگرا شده باشد، وزن ها دیگر هیچ تغییری نمی کنند. این مشکل به صورت تصویری در شکل ۲-۱۰ نشان داده شده است.

^۱ Backpropagation

^۲ Vanishing Gradient

^۳ Exploding Gradient



شکل ۲-۱۰: مشکل اختفای گرادیان در شبکه‌های بازگشتی [32]

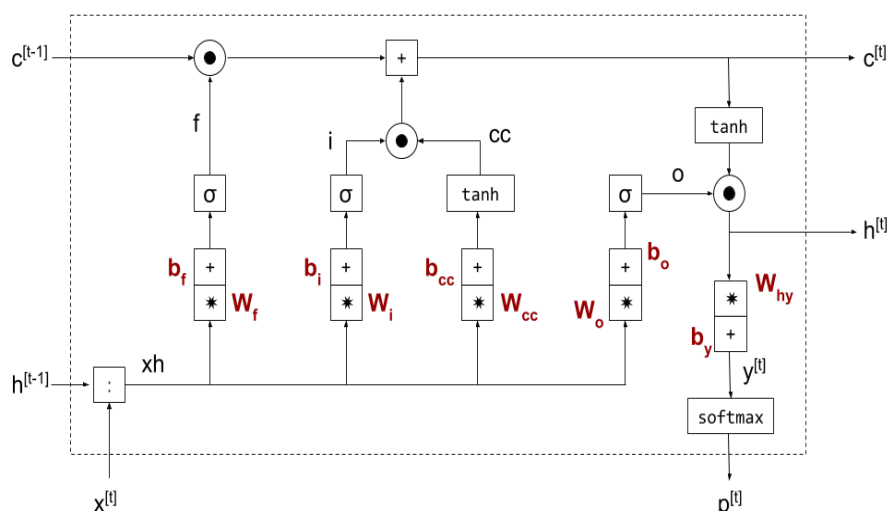
مجموعه این مشکلات موجب می‌شد تا شبکه‌های بازگشتی قدیمی امکان پردازش و پویا و وابستگی‌های دورودراز را نداشته باشند و مثل مدل‌های مارکوف به یکی دو سه گام قبل بسنده کنند. در سال ۱۹۹۷ سپ هواختر و هوشیدر [35] سعی کردند با ابداع مفهوم درگاه‌های ورودی و خروجی مشکل اختفای گرادیان در شبکه‌های بازگشتی را حل کنند. 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^۱ کامل‌تر شد [36] در حل بسیاری از مسائل پیچیده، از جمله کشف ناهنجاری مورد استفاده قرار گرفته است.

۲-۳-۲ ساختار شبکه‌های LSTM

شبکه‌های LSTM در واقع به جای نوروں 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 می‌باشد. هر بلوک حافظه از یک سلول حافظه^۲ و تعدادی درگاه تشکیل شده است. سلول حافظه وضعیت جاری بلوک را نگهداری می‌کند و درگاه‌ها شارش اطلاعات را به سمت داخل و خارج مدیریت می‌نمایند. شکل ۲-۱۱ ساختار یک بلوک حافظه LSTM را نشان می‌دهد.

^۱ Forget Gate

^۲ Memory Cell



شکل ۲-۱۱: ساختار یک بلوک LSTM [37]

چنانکه از شکل برمیآید، هر درگاه درواقع یک‌لایه شبکه عصبی با تابع فعال‌سازی سیگمید است که خروجی آن در انتها به یک عملگر ضرب هامادار یا ضرب عنصری^۱ ختم می‌شود. در معماری هوشیدر [35] درگاه‌ها، ورودی جاری بلوک (x_t) و خروجی بلوک قبلی (h_{t-1}) را به‌عنوان ورودی دریافت می‌نمایند و با عبور آن‌ها از پالایه یک‌لایه 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 [36] با‌ملاحظه 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 روزنه‌ای^۲ رایج‌ترین نوع بلوک‌های LSTM، یعنی LSTM روزنه‌ای^۳ را پدید آوردند.

^۱ Pointwise

^۲ Peephole

^۳ Peephole LSTM

۳-۲-۳-۲ آموزش بلوک‌های LSTM

یک بلوک LSTM روزنه‌ای را می‌توان مطابق رابطه (۸-۲) به‌عنوان یک تابع F در نظر 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:

$$h_t = F(x_t, h_{t-1}, c_{t-1}) \quad (۸-۲)$$

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

$$i_t = \sigma(W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} * c_{t-1} + b_i) \quad (۹-۲)$$

$$f_t = \sigma(W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} * c_{t-1} + b_f) \quad (۱۰-۲)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (۱۱-۲)$$

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} * c_t + b_o) \quad (۱۲-۲)$$

$$h_t = o_t \odot \tanh(c_t) \quad (۱۳-۲)$$

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 وزنه‌های اتصالات و بایاس‌ها می‌باشند. σ ، \tanh 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. این مراحل در رابطه (۱۰-۲) خلاصه شده است:

$$f_t = \sigma (W_{xf} * x_t + W_{hf} * h_{t-1} + W_{cf} * C_{t-1} + b_f) \quad (10-2)$$

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^۱ 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 یک لایه سیگمیدی موسوم به درگاه ورودی، با دریافت همان ورودی‌های درگاه فراموشی، مشخص می‌کند که کدام مقادیر در وضعیت بلوک باید بهنگام شوند. این شرح در رابطه (۹-۲) در قالب یک فرمول ریاضی بیان شده است:

$$i_t = \sigma (W_{xi} * x_t + W_{hi} * h_{t-1} + W_{ci} * C_{t-1} + b_i) \quad (9-2)$$

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

$$\hat{C} = \tanh (W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (14-2)$$

رابطه (۱۴-۲) در واقع بخشی از فرایند محاسبه وضعیت جدید است که در رابطه (۱۱-۲) بیان شده است. سرانجام حاصل ضرب عنصری خروجی درگاه فراموشی در وضعیت جاری با حاصل ضرب عنصری

^۱ Reset

بردار نامزدهای احتمالی فوق‌الذکر در خروجی درگاه ورودی جمع شده و وضعیت جدید بلوک را مطابق رابطه (۱۱-۲) ایجاد می‌کنند:

$$C_t = f_t \odot C_{t-1} + i_t \odot \tanh(W_{xc} * x_t + W_{hc} * h_{t-1} + b_c) \quad (11-2)$$

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

در آخرین مرحله، بلوک LSTM باید تصمیم بگیرد که چه مقادیری را به‌عنوان خروجی به بلوک‌های مجاور افقی و / یا عمودی ارسال کند. برای انتخاب این خروجی که در واقع نسخه پالایه شده‌ای از وضعیت بلوک است، درگاه خروجی، ورودی همسان ورودی‌های دو درگاه قبلی می‌گیرد و با عبور آن‌ها از یک‌لایه سیگمیدی، مطابق رابطه (۱۲-۲) موارد ارسالی را تعیین می‌نماید:

$$o_t = \sigma(W_{xo} * x_t + W_{ho} * h_{t-1} + W_{co} * C_t + b_o) \quad (12-2)$$

در آخرین مرحله وضعیت بلوک از یک‌لایه تانژانت هیپربولیک می‌گذرد تا محتویات آن در فاصله ۱- تا ۱ قرار بگیرد. حاصل ضرب عنصری این گام در نتیجه درگاه خروجی، تعیین‌کننده چیزی است که از بلوک جاری به بلوک بعدی انتقال پیدا خواهد کرد. این مرحله در رابطه (۱۳-۲) خلاصه‌شده است:

$$h_t = o_t \odot \tanh(C_t) \quad (13-2)$$

در مثال پردازش متن، چون در این مرحله یک ضمیر دیده‌شده، احتمالاً بلوک LSTM مایل است اطلاعاتی را در مورد فعل مربوطه - که در ادامه جمله ذکر شده - به خارج ارسال کند. به‌عنوان 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 روزنه‌ای 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^۱ و واحد بازگشتی درگاهی یا GRU^۲ از مهم‌ترین و مشهورترین این انواع هستند. برای آشنایی با سایر گونه‌های LSTM و مقایسه کاربردی آن‌ها می‌توانید به منابع [38] و [39] مراجعه کنید.

ساختار GRU که در سال ۲۰۱۴ معرفی شده [40]، از معماری درگاهی LSTM اخذ گردیده است. البته GRU دست‌کم به دو دلیل از LSTM ساده‌تر است. GRU برخلاف LSTM سلول حافظه ندارد، به همین دلیل وضعیت جاری، یعنی Ct را نگهداری نمی‌کند و آن را بلافاصله پس از تولید به واحد بعدی انتقال می‌دهد. همچنین در طرح GRU، درگاه‌های ورودی و فراموشی در قالب یک درگاه جدید به نام درگاه بهنگام سازی در هم ادغام شده‌اند. به همین دلیل GRU برخلاف LSTM به جای سه درگاه از دو درگاه بهره می‌برد. به دلیل همین تفاوت‌ها، معمولاً آموزش شبکه‌های GRU به زمان و داده‌های کمتری نیاز دارد. در مقابل، حداقل به لحاظ نظری، شبکه‌های LSTM به دلیل وجود سلول حافظه، باید بتوانند وابستگی‌های دورودراز را بهتر از شبکه‌های GRU یاد بگیرند. با این‌همه هیچ قاعده محکمی برای انتخاب یکی از این شبکه‌ها وجود ندارد و معمولاً انتخاب یکی از آن‌ها با توجه به مسئله و داده‌های موردنظر و پس از پیاده‌سازی و سعی و خطا صورت می‌پذیرد [41].

۳-۳-۲ توجه

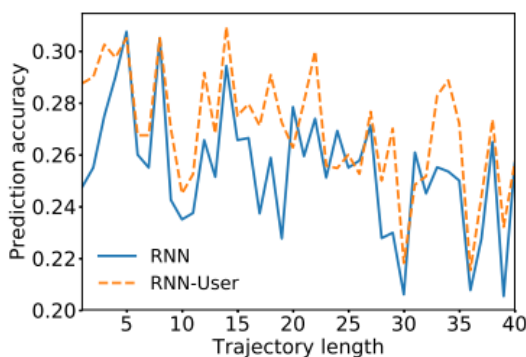
شبکه‌های عصبی بازگشتی به‌عنوان ابزاری قدرتمند برای مدل‌سازی می‌توانند وابستگی‌های دورودراز^۳ را در میان داده‌های رشته‌ای و متوالی لحاظ کنند؛ اما تجربه نشان داده است که اگر طول یک رشته بسیار طولانی باشد - مثلاً یک جمله ۲۰ کلمه‌ای - کارایی این شبکه‌ها به سرعت و به شدت رو به کاهش می‌گذارد [42]. اگر تعداد حسگرهای یک‌خانه هوشمند، به دلیل بزرگی خانه یا تنوع حسگرها زیاد باشد، ثبت بسیاری از اعمال روزمره به ایجاد رشته‌ای بین ۲۰ تا ۱۰۰ رویداد خواهد انجامید. این

^۱ Bidirectional LSTM

^۲ Gated Recurrent Unit

^۳ Long Dependency

موضوع به شکل قابل توجهی دقت و توان شبکه‌های عصبی بازگشتی در تحلیل داده‌های خانه هوشمند را به چالش می‌کشد. شکل ۲-۱۲ نمودار تغییرات دقت یک شبکه بازگشتی را در رده‌بندی افعال یک‌خانه هوشمند نشان می‌دهد. چنان‌که از شکل برمی‌آید با بزرگ‌تر شدن طول رشته رویدادهای معرف افعال، دقت مدل کاهش پیدا می‌کند؛ بنابراین به نظر می‌رسد که شبکه‌های عصبی بازگشتی به‌تنهایی برای تحلیل و کشف ناهنجاری در افعال طولانی مناسب نباشند. مکانیزم توجه یکی از بهترین و رایج‌ترین راه‌حل‌هایی است که برای حل مشکل اخیر پیشنهاد شده است.



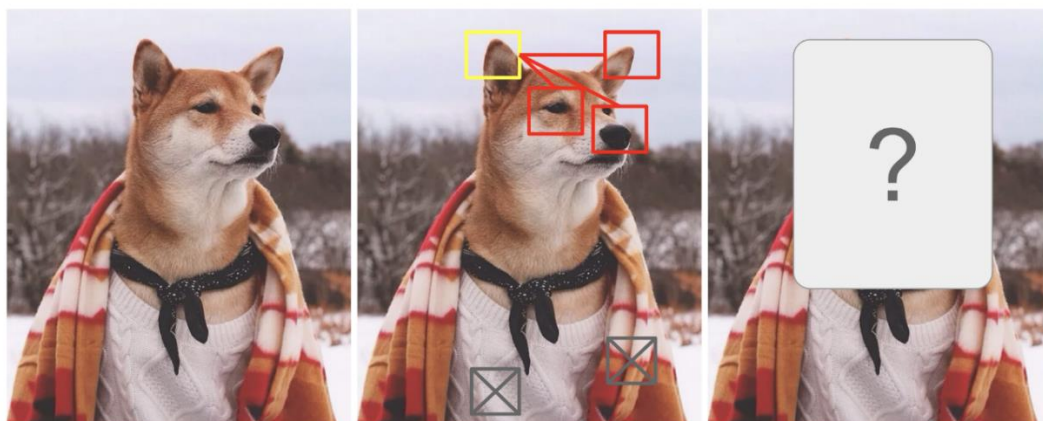
شکل ۲-۱۲: تغییر دقت شبکه بازگشتی نسبت به طول رشته

ایده اصلی مکانیسم توجه^۱ از سیستم استدراک انسانی اخذ شده و از مدت‌ها قبل در روانشناسی مورد بررسی و توجه بوده است. اصل این ایده حکایت از آن دارد که ذهن ما در زمان مواجهه با یک مقوله ادراکی، وجوه اصلی و ممیزه آن را انتزاع کرده و جنبه‌های فرعی‌تر را نادیده می‌گیرد. این کار موجب می‌شود که ما بتوانیم با اختصاص زمان، قوای ذهنی و حافظه کمتر، شناخت بیشتر و بهتری را به دست بیاوریم.

به‌عنوان مثال شیوه توجه سامانه بینایی به شکل ۲-۱۳ را در نظر بگیرید. این سامانه به ما اجازه می‌دهد که روی برخی از نواحی تصویر با توجه بیشتری تمرکز کنیم (مثلاً به گوشی که با مستطیلی زرد مشخص شده است نگاه کنید)، درحالی‌که سایر مناطق را به‌عنوان پس‌زمینه و با دقت کمتری

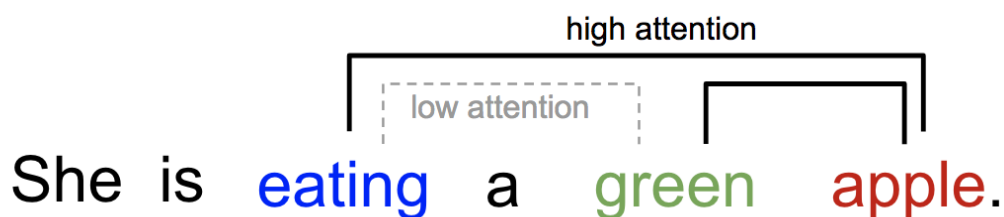
^۱ Attention

می‌بینیم (مثلاً پتو و زمینه برفی تصویر). سیستم ادراکی ما از تصویر شکل ۲-۱۳ تنها نقاط نوع اول را اخذ و ذخیره می‌کند. این نقاط که نقاط کانونی^۱ یا نقاط توجه نامیده می‌شوند، پایه‌های درک ما از تصویر هستند و به ما می‌گویند که در سایر مناطق تصویر توقع دیدن چه چیزهایی را داشته باشیم. مثلاً ما انتظار داریم که در مستطیل زرد یک گوش ببینیم؛ زیرا نقاط کانونی که با مستطیل قرمز نشان داده شده‌اند چنین توقعی را ایجاد می‌کنند. حذف این نقاط در تصویر سوم ثابت می‌کند که بقیه تصویر از جمله لباس، پتو و زمینه برفی، در ایجاد تأثیر یادشده و شکل‌گیری استدلال منتج به آن تأثیری نداشته‌اند.



شکل ۲-۱۳: مکانیسم توجه در بینایی انسان [43]

نظیر این رابطه در میان کلمات یک جمله یا رویدادهای یک فعل نیز وجود دارد. مثلاً عبارت زیر را در نظر بگیرید:



^۱ Focal Points

به‌طور طبیعی، ما انتظار داریم که بعد از فعل خوردن (eating) به‌زودی با نام یک خوراکی یا غذا روبرو شویم. گرچه صفت رنگ (green) هم می‌تواند ذهن ما را به غذا رهنمون سازد، اما به‌احتمال زیاد تأثیر ملاحظه فعل خوردن بسیار بیشتر است.

مکانیسم توجه یادگیری عمیق در سال ۲۰۱۴ و در حوزه بینایی ماشین معرفی شد [44]. هدف اصلی این مکانیسم در بینایی ماشین، تمرکز بر قسمت‌هایی از تصویر و ویدیو است که اهمیت بیشتری دارند. به‌عنوان مثال یک تصویر بزرگ ۵۰ مگاپیکسلی را در نظر بگیرید که تنها بخش کوچکی از آن حاوی اطلاعات موردنیاز ماست. در صورتی که بخواهیم پالایه‌های شبکه پیچشی را روی این تصویر اعمال کنیم، باید میلیون‌ها بار فیلتر را از طول آن عبور دهیم. از آنجا که تنها بخش کوچکی از تصویر برای ما اهمیت دارد، این عملیات پرهزینه و زمان‌گیر خواهد بود. مکانیسم توجه به ما کمک می‌کند با تشخیص نقاط کانونی و محدود کردن پردازش به این نقاط، مشکل را حل نماییم. این ایده درخشان پس از مدتی کوتاه به عرصه ترجمه ماشینی راه پیدا کرد [45] و از آنجا به عرصه سری‌های زمانی و پردازش داده‌های رشته‌ای و جریانی نیز سرایت نمود.

جزئیات محاسبه توجه در شکل ۲-۱۴ نمایش داده شده است. از دیدگاه فنی، عنصر اصلی مکانیسم توجه لایه توجه است^۱ که معمولاً به‌سادگی توجه نامیده می‌شود. ورودی‌های این لایه را پرس‌وجو^۲ می‌نامند. لایه توجه با دریافت یک پرس‌وجو، حافظه خود را مورد جستجو قرار داده و پاسخ این پرس‌وجو را باز می‌گرداند. حافظه لایه توجه حاوی مجموعه‌ای از زوج‌های مرتب (کلید، مقدار) است و مکانیسم توجه بر اساس میزان شباهت یا فاصله پرس‌وجو با کلیدها، پاسخ مربوطه را از ترکیب مقادیر ایجاد می‌کند. برای بیان دقیق‌تر فرض کنید $q \in \mathbb{R}^{d_q}$ بوده و حافظه لایه توجه از n زوج مرتب $(k_1, v_1), \dots, (k_n, v_n)$ تشکیل شده است. یعنی :

Query : $q \in \mathbb{R}^{d_q}$

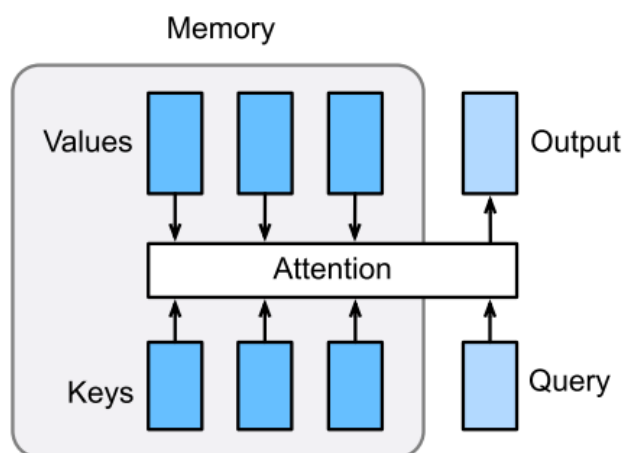
^۱ Attention Layer

^۲ Query

Attention Layer Memory : $\{(k_i, v_i) \mid k_i \in \mathbb{R}^{d_k}, v_i \in \mathbb{R}^{d_v}, i \in \{1, 2, \dots, n\}\}$

با این تعاریف پاسخ لایه توجه به پرس و جو q که ما آن را o می‌نامیم از شکل و نوع مقادیر v_i است،

یعنی $o \in \mathbb{R}^{d_v}$



شکل ۲-۱۴: جزئیات لایه توجه [46]

برای محاسبه خروجی فرض می‌کنیم تابع امتیازدهی به نام α وجود دارد که میزان شباهت یا فاصله پرس و جو و یک کلید را اندازه‌گیری می‌کند. اکنون لایه توجه امتیاز هر یک از کلیدهای حافظه خود را با توجه به پرس و جو محاسبه می‌نماید:

$$a_i = \alpha(q, k_i) \quad (۱۵-۲)$$

سپس با استفاده از تابع سافت مکس ضرایب توجه را به دست می‌آورد:

$$b_1, b_2, \dots, b_n = \mathbf{Softmax}(a_1, a_2, \dots, a_n) \quad (۱۶-۲)$$

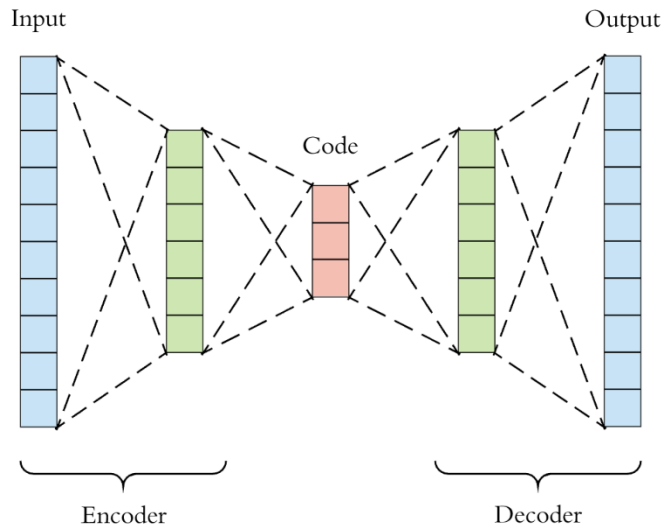
حاصل ضرب این ضرایب یا وزن‌ها در مقادیر حافظه لایه توجه، خروجی این لایه را تشکیل خواهد داد:

$$o = \sum_{i=1}^n b_i v_i \quad (۱۷-۲)$$

به کارگیری توابع امتیازدهی متفاوت، انواع گوناگونی از لایه‌های توجه را ایجاد می‌کند [47].

۴-۳-۲ کد کننده خودکار یا AE (AutoEncoder)

یک کد کننده خودکار، در واقع 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 ۱۵-۲ یک کد کننده خودکار را نشان می‌دهد.



شکل ۱۵-۲: یک کد کننده خودکار عمیق [48]

آموزش یک AE شامل دو مرحله کدگذاری^۱ و کدگشایی^۲ 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 ($y^{d'}$):

$$y = s(W * x + b) \quad x \in R^d, y \in R^{d'} \quad (۱۸-۲)$$

در رابطه (۱۸-۲) تابع s می‌تواند یک تابع خطی، یک تابع نیمه‌خطی مثل ReLU یا یک تابع غیرخطی مثل سیگمید باشد.

در مرحله بعد، کد اخیر با تحویل به لایه خروجی به x^{\wedge} تبدیل می‌شود که در واقع بازسازی و بازآفرینی مجدد ورودی x است. این مرحله که در رابطه (۱۹-۲) بیان شده است کدگشایی نام دارد.

^۱ Encoding

^۲ Decoding

$$\hat{x} = s(W' * y + b') \quad y \in R^{d'} , \hat{x} \in R^d \quad (19-2)$$

آموزش AE یعنی تعیین مقادیر بردارهای W و W' و متغیرهای b و b' که معرف وزن‌ها و بایاس‌های شبکه هستند و با کمینه‌سازی خطای بازسازی (فاصله بردارهای X و X^{\wedge}) محقق می‌گردد. معیار اندازه‌گیری خطای یادشده به توزیع احتمالاتی داده‌های ورودی بستگی دارد. به‌عنوان مثال درحالی‌که این داده‌ها از یک توزیع گاوسی پیروی می‌کنند، بهتر است از تابع مرسوم مربع خطا استفاده شود. این تابع که معمولاً در رگرسیون بکار گرفته می‌شود در رابطه (20-2) نمایش داده‌شده است:

$$L(x, y) = \|x - y\|^2 \quad (20-2)$$

اما در صورتی که داده‌های ورودی بردارهای بیتی یا بردارهایی از احتمالات وقوع بیت‌ها باشند، تابع آنتروپی متقابل^۱ که غالباً در رده‌بندی مورد استفاده قرار می‌گیرد نتایج دقیق‌تری را تولید می‌کند. این تابع در رابطه (21-2) نمایش داده‌شده است:

$$L(x, y) = \sum_{k=1}^d (x_k * \log y_k + (1 - x_k) * \log(1 - y_k)) \quad (21-2)$$

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 متوسط مجموع مربعات یا MSE استفاده کنیم، آنگاه نورون‌های لایه پنهان k مؤلفه اصلی^۲ 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 خواهد کرد.

معمولاً برای اینکه AE به دام پیروی از نگاشت همانی در نیفتد، تعداد نورون‌های لایه پنهان باید کمتر از ابعاد داده‌های ورودی باشد. بدین ترتیب AE را می‌توان به‌عنوان یک فشرده‌ساز دارای خطا در نظر گرفت که هیچ‌وقت ورودی خود را به‌طور کامل و مطلق بازسازی نمی‌کند. وجود همین خطای

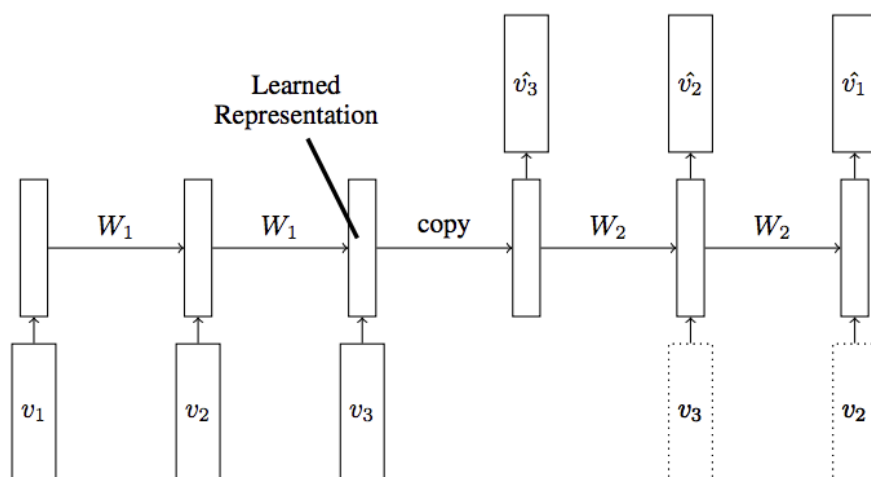
^۱ Cross Entropy

^۲ Principal Component

بازسازی، کد کننده خودکار را به ابزار مناسبی برای کشف داده‌های ناساز تبدیل کرده است. کد کننده خودکار را می‌توان به‌عنوان یک روش بی‌ناظر یا نیمه نظارتی برای کشف ناهنجاری مورد استفاده قرارداد. برای انجام این کار ابتدا مجموعه آموزشی را در اختیار AE 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 تولید شده باشند، خطای بازسازی در طول داده‌های آزمایشی زیاد نبوده و دارای نوسانات شدید نیست؛ اما در صورتی که داده‌های مجموعه آزمایشی حاوی نقاط ناساز و ناهنجار باشد، قاعدتاً خطای بازسازی نقطه‌های مذکور از نقاط مجاور بزرگ‌تر خواهد بود.

۲-۳-۴-۱ کد کننده خودکار LSTM

با ترکیب ایده‌های AE و LSTM کد کننده خودکاری ایجاد می‌شود که هر یک از بخش‌های کدگذار و کدگشای آن یک لایه LSTM است. شمای کلی این ساختار در شکل ۱۶-۲ ترسیم شده است.



شکل ۱۶-۲: یک کد کننده خودکار مبتنی بر بلوک‌های LSTM [49]

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 یادشده بتوانند وابستگی‌های زمانی و مکانی را به خوبی 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.

فصل ۳ : مروری به کارهای گذشته

۳-۱ مقدمه

در این بخش، پیشینه تحقیق را در دو حیطه پژوهشی یعنی شناسایی فعالیت و تشخیص رفتار ناهنجارانه عامل‌ها بیان و بررسی می‌کنیم. شناسایی فعالیت خود متشکل از دو رهیافت شناسایی فعالیت با بینایی ماشین و شناسایی فعالیت مبتنی بر حسگر است.

۳-۲ شناسایی فعالیت

شناسایی یا تشخیص فعالیت فرایندی است که به‌موجب آن رفتار عامل و شرایط محیطی که در آن واقع است تحت نظر قرار گرفته، تجزیه و تحلیل می‌شود تا بدین طریق به فعالیت‌های عامل پی برده شود [50]. در خلال دهه‌های گذشته محققین در حوزه‌های کاربردی مختلف و با توسعه روش‌های متنوعی، شناسایی فعالیت را مورد بررسی و پژوهش قرار داده‌اند. شناسایی فعالیت را به‌طور کلی می‌توان بر اساس امکانات نظارتی که در سامانه‌های تشخیص فعالیت به‌منظور پی بردن به فعالیت‌های عامل به کار می‌رود، دسته‌بندی نمود. در حال حاضر دو رهیافت اصلی برای شناسایی فعالیت‌ها وجود دارند: نوع اول شناسایی فعالیت مبتنی بر بینایی ماشین و نوع دوم، شناسایی فعالیت مبتنی بر حسگر است. مهم‌ترین چالش، یافتن الگوریتم‌هایی است که بتوانند به نحو مؤثری رفتار فعالیت‌های انسانی را بر اساس داده‌های حسگری سطح پایین تشخیص داده و مدل نمایند.

۳-۲-۱ شناسایی فعالیت مبتنی بر بینایی ماشین

ردیابی و درک رفتار عامل‌ها از طریق ویدئو به سبب نقش مهمی که این روش در حوزه‌های مختلف تعامل انسان با ماشین دارد، سال‌ها به‌عنوان کانون پژوهش برای بسیاری از پژوهشگران مطرح بوده و هست. در شناسایی فعالیت‌ها با بینایی ماشین، محققان روی شیوه‌های متنوعی همچون، «جریان نور»^۱، فیلتر کالمن، «مدل‌های پنهان مارکوف»^۲، «فیلدهای تصادفی شرطی»^۳، در حالات مختلفی مثل تک‌دوربین، استریو و مادون‌قرمز کار کرده‌اند. بعلاوه پژوهشگران جنبه‌های متنوعی را نیز در این زمینه بررسی کرده‌اند که از آن جمله می‌توان ردیابی تک‌عامله، ردیابی چندعاملی، شناسایی فعالیت، شناسایی فعالیت ترکیبی و نهایتاً شناسایی تعاملات چندعاملی را نام برد.

فرایند شناسایی فعالیت عموماً از چهار مرحله تشکیل می‌شود که عبارت‌اند از:

- شناسایی و تشخیص عامل
- ردیابی^۴ عامل
- شناسایی فعالیت
- ارزیابی فعالیت سطح بالا^۵

چنگ و همکاران در [51] این‌گونه جمع‌بندی نموده‌اند که رهیافت‌های مبتنی بر شناسایی فعالیت با بینایی ماشین با مشکلات اساسی همچون گسترش‌پذیری و قابلیت استفاده مجدد که به سبب وجود شرایط پیچیده در دنیای واقعی (تنوع بی‌شمار فعالیت‌ها و محیط) حادث می‌شوند، مواجه هستند. بعلاوه، دوربین‌ها در برخی جوامع به‌عنوان یک عامل مزاحم تصور می‌شوند که ممکن است خود مانع از آن شود که این روش در برخی کاربردها نظیر محیط‌های خانگی گسترش یابد.

^۱ Optical Flow

^۲ Hidden Markov Models (HMM)

^۳ Conditional Random Fields (CRF)

^۴ Tracking

^۵ High-level Activity Evaluation

۳-۲-۲ شناسایی فعالیت مبتنی بر حسگر

در تشخیص و شناسایی فعالیت مبتنی بر حسگر از طیف گسترده‌ای از حسگرها نظیر شتاب سنج‌ها، برچسب‌های RFID، آشکارسازهای صوتی و حرکتی و... استفاده می‌شود تا رفتار عامل و محیطی که در آن قرار گرفته است نظارت و پایش شود. این حسگرها در هدف، زیرساخت فنی، سیگنال‌های خروجی و اصول زیربنایی تئوری با یکدیگر متفاوت هستند.

تمام حسگرها را می‌توان از نقطه نظر استقرار در محیط تحت نظارت خود به دودسته اصلی شامل حسگرهای پوشیدنی و حسگرهای تعبیه شده^۱ تقسیم نمود.

۳-۲-۲-۱ حسگرهای پوشیدنی

حسگرهای پوشیدنی به‌طور مستقیم یا غیرمستقیم در بدن عامل قرار می‌گیرند تا در حین فعالیت عامل، حسگر سیگنال‌هایی را تولید نماید. در مواقعی که عامل تحت نظر انسان باشد، حسگر پوشیدنی درون لباس‌ها، عینک، کمربند، کفش‌ها، تجهیزات همراه و ... تعبیه شده یا مستقیماً درون بدن وی جاسازی می‌شود. از حسگرها برای جمع‌آوری اطلاعاتی نظیر موقعیت، سرعت، شتاب بخش‌های مختلف بدن، نبض و دمای پوست می‌توان استفاده نمود [52]. در ادامه، مشخصات چند حسگر اینرسیایی (مثل شتاب سنج‌ها، گردش سنج‌ها، مغناطیس‌سنج‌ها)، حسگرهای تشخیص علائم حیاتی فرد (ضربان قلب، دمای بدن) و نشانگرهای بصری به‌طور خلاصه بیان می‌شود.

حسگرهای شتاب سنج، با توجه به قیمت پایین و اثرپذیری، شاید پرکاربردترین نوع در دسته حسگرهای پوشیدنی باشند. روش‌های نسل اول بر اساس شتاب سنج سه محوره با الگوریتم‌های آستانه‌ای بود. بورکه و همکاران [53] یک الگوریتم آستانه‌ای را برای تمایز بین فعالیت‌های طبیعی (نظیر نشست و برخاست، دراز کشیدن و ایستادن، نشستن روی یا بلند شدن از روی صندلی خودرو، پیاده‌روی) و افتادن به‌عنوان گونه‌ای ناهنجاری حرکتی معرفی کردند. این توانایی با استفاده از یک

^۱ Embedded Sensors

گردش سنج دومحوری نصب شده روی بالاتنه با اندازه گیری سرعت های زاویه ای حاصل می شود. آن ها یک الگوریتم آستانه را برای اندازه گیری حداکثر مقدار سیگنال سرعت زاویه ای، شتاب زاویه ای و تغییر زاویه بالاتنه بکار گرفتند. نسل دوم روش ها، قادر است تا فعالیت ها را با روش هایی نظیر درخت های تصمیم، ماشین بردار پشتیبان، نزدیک ترین همسایگان و رده بند بیزی ساده رده بندی نماید.

روش جالبی در [54] توسط هوین و همکاران برای تشخیص فعالیت های روزانه ارائه شده است. مطابق این روش، حرکت با سه شتاب سنج بدن پوش حس می شود، در حالی که تشخیص ۱۵ فعالیت سطح پایین و ۳ فعالیت سطح بالا با استفاده از چهار راهکار شامل خوشه بندی کی-میانگین^۱، ماشین بردار پشتیبان، رده بندی نزدیک ترین همسایه، و مدل های پنهان مارکوف انجام می گیرد. آن ها در محیط آزمایشی عملکرد سیستم به دقت ۶۹ تا ۸۰ درصد در فعالیت های سطح پایین (مثل نشستن، ایستادن، خوردن، راه رفتن) و دقت ۸۳ تا ۹۲ درصد در فعالیت های سطح بالا (مثل آماده شدن برای کار، خرید، کار منزل) رسیدند.

در منبع [55] تاپیا و همکاران، یک الگوریتم بلادرنگ^۲ ارائه دادند که فعالیت های فیزیکی را به صورت خودکار تشخیص می داد و در برخی حالات، شدت آن ها را نیز با استفاده از پنج شتاب سنج بی سیم و یک ناظر ضربان قلب بی سیم مانیتور می نمود. شتاب سنج ها روی شانه ها، مچ دست، مفصل ران، قسمت بالایی ران و مچ ها نصب می شدند. ویژگی هایی نظیر واریانس، انرژی، ضرایب همبستگی، از زمان و دامنه های فرکانس با استفاده از یک اندازه پنجره از پیش تعریف شده روی سیگنال استخراج می شوند. رده بندی فعالیت با رده بندهای C4.5 و رده بند بیزی ساده در سه دسته جای می گیرد: که شامل حالات (مثل ایستادن، نشستن)، فعالیت ها (مثل پیاده روی، دوچرخه سواری) و سایر فعالیت ها (مثل دویدن، استفاده از پله ها) می شدند. آن ها برای این سه کلاس به دقت تشخیصی برابر با ۹۴/۶

^۱ K-means

^۲ Real Time

درصد با استفاده از آموزش وابسته به موضوع^۱ و ۵۶/۳ درصد با استفاده از آموزش مستقل از موضوع^۲ رسیدند.

کواپیسز و همکاران [56] از یک شتاب سنج نصب شده روی ران افراد استفاده کرده و نتایج روش‌های مختلف رده‌بندی را برای فعالیت‌های پویایی نظیر راه رفتن، دویدن، نرمش (دویدن آرام) مقایسه نموده‌اند. یک رده‌بندی وزنی-سلسله‌مراتبی توسط بانوس و همکاران در مرجع [57] ارائه شده است که در این روش، هر حسگر در سطح اول بر اساس رده‌بندهای دودویی^۳ تصمیمی در مورد تشخیص فعالیت اخذ می‌نماید و در سطح بعد، شمای رأی اکثریت بارأی اخذ شده ادغام می‌شود تا تصمیم نهایی اتخاذ گردد.

در منبع [58] سامانه‌ای توسط ساکتانکار و سیکارا ارائه شده که در آن، با بازسازی حالت کاربران رفتارهای از پیش تعیین شده آن‌ها را تشخیص می‌دهد. داده‌ها به کمک ۴۳ نشانگر^۴ و ۱۲ دوربین با نرخ الگوبرداری ۱۲۰ هرتز جمع‌آوری شده‌اند. آن‌ها یک مدل انسانی را از مختصات خام نشانگرها و ویژگی‌های اخذ شده از مدل، مثل زوایای بین بخش‌های مختلف بدن، طول اندام‌ها، بازه حرکات، ... بازسازی نمودند. یادگیری با استفاده از روش ماشین بردار پشتیبان انجام شد که این روش به دقتی بالغ بر ۷۶٫۹٪ در تشخیص حرکاتی نظیر راه رفتن، دویدن، حرکت دزدانه^۵، جستجو، سینه‌خیز رفتن و برخاستن رسید. رفتار به صورت توالی از فعالیت‌های ابتدایی تعریف شد و با استفاده از مدل پنهان مارکوف مدل گردید. مؤلفان چند مدل رفتاری را تعریف کرده و توالی جدیدی از فعالیت‌ها که بهترین سنخیت با مدل را دارا باشد رده‌بندی کردند.

در کل شناسایی و تشخیص فعالیت‌ها با استفاده از حسگرهای پوشیدنی با محدودیت‌هایی نیز همراه است، بدین معنی که بسیاری از حسگرها باید دائماً و بدون دخالت دست در حال کار باشند.

^۱ Subject Dependent Training

^۲ Subject Independent Training

^۳ Binary Classifiers

^۴ Body Marker

^۵ Sneaking

جدای از این، مسائل عملیاتی مثل پذیرش کاربر و توانایی پوشیدن حسگر تا مسائل فنی مثل اندازه، طول عمر باتری و کاربری ساده از دیگر محدودیت‌های حسگرها هستند [50]. ضمن اینکه حسگرهای پوششی شاید برای نظارت فعالیت‌هایی که شامل تعامل کاربر با محیط است مناسب نباشند. در نتیجه بهتر است که حسگرهای پوششی را با حسگرهای تعبیه‌شده ترکیب نمود.

۲-۲-۲-۳ حسگرهای تعبیه‌شده

حسگرهای تعبیه‌شده که اغلب از آن‌ها به‌عنوان حسگرهای متراکم^۱ نیز یاد می‌شود، به اشیاء متصل شده و فعالیت‌ها با تشخیص تعاملات عامل-شیء نظارت می‌شوند. حسگری متراکم با استفاده از تعداد زیادی از حسگرهای مینیاتوری و معمولاً کم‌هزینه انجام می‌شود که روی انواع اشیاء و مکان‌های درون محیط نصب‌شده‌اند. این روش بر مبنای این فرض عمل می‌کند که فعالیت‌ها با دست‌کاری اشیاء در مدت‌زمان کارکرد آن‌ها رده‌بندی می‌شوند، یعنی می‌توان فعالیت‌ها را از داده‌های حسگری که تعاملات کاربران را با اشیاء در محیط، نظارت و مونیتور می‌نماید تمیز داد.

تشخیص فعالیت با استفاده از حسگرهای تعبیه‌شده، به‌طور گسترده‌ای در مقوله «زندگی به کمک محیط»^۲ و با الگوی خانه هوشمند و به‌منظور نظارت بر حرکات ساکنین و رویدادهای محیطی استفاده می‌شود. به‌عنوان مثال، استورف و همکاران در منبع [59] روی شناسایی اعمال روزانه از حسگرهای تعبیه‌شده در محیط مطالعه نموده‌اند. آن‌ها یک رهیافت چندعاملی معرفی کرده‌اند که با استفاده از یک‌زبان تشخیص فعالیت رویداد گرا، فعالیت‌های اتمیک را به فعالیت‌های سطح بالا تبدیل می‌کند. نویسندگان در مقاله خود دقتی بالاتر از ۸۰٪ را گزارش نموده‌اند. در شرایطی مشابه، کوک و هولدر در [60] مدل‌های پنهان مارکوف را برای تشخیص و شناسایی اعمال روزمره با استفاده از انواع

^۱ Dense Sensors

^۲ Ambient Assisted Living (AAL)

حسگر که برای شناسایی استفاده می‌شدند، بکار بستند. دقت حاصل بین ۸۰ تا ۹۰٪ متغیر بود و زمانی که تعداد قابل توجهی از حسگرها حذف می‌شدند، دقت حاصل به زیر ۷۵٪ می‌رسید.

تنوع مختلفی از حسگرها و روش‌ها برای شناسایی فعالیت‌های مختلف استفاده می‌شوند و غیرممکن است که ادعا کنیم عملکرد صرفاً یک نوع حسگر بهتر است. تناسب و کارایی به شدت به نوع فعالیت مورد ارزیابی و ویژگی‌های کاربردی که قرار است حسگرها در آن بکار رود وابسته است.

علاوه بر این بسیاری از پژوهشگران، شناسایی رفتار و فعالیت را با استفاده از تکنیک‌های آماری انجام داده‌اند. برای مثال، در سال ۲۰۰۴، رده‌بند بیزی ساده توسط تاپیا و همکاران، برای دسته‌بندی و شناسایی فعالیت‌های مربوط به حسگرهای چسبان مورد استفاده قرار گرفته است [61]. در این مقاله، مؤلفین از دو نوع رده‌بند تشخیص فعالیت استفاده کردند: رده‌بند بیزی چند کلاسه و چند رده‌بند بیزی دودویی که اولین رده‌بند، فعالیت‌هایی را که برای شناسایی و تشخیص مورد نیاز هستند بیان می‌کند، در حالی که دومین رده‌بند فعالیت‌هایی که شناسایی می‌شود را بیان می‌نماید. مشکل اصلی شبکه‌های باور بیزی، عدم انعطاف‌پذیری استنتاج احتمالی دقیق است [62].

مدل پنهان مارکوف تکنیک آماری دیگری است که برای مدل‌سازی رفتار انسان استفاده می‌شود. این روش به‌طور گسترده‌ای برای شناسایی فعالیت‌های یک کاربر از داده‌های حسگر استفاده می‌شود [63]. جدی‌ترین مشکل در استفاده از HMM پایه، مشکلات موجود در پردازش مجموعه‌های بزرگ داده‌های حسی (برای مثال، داده‌های زمانی به دست آمده از مقیاس‌های زمانی مختلف) است. همچنین، برای هر فعالیت منفرد، یک توالی از رویداد حسگر را با استفاده از این مدل‌ها نمی‌توان تفکیک نمود [64].

به‌عنوان روشی جایگزین برای روش‌های آماری، از تکنیک‌های هوش محاسباتی به‌طور گسترده‌ای برای تشخیص اعمال روزمره استفاده می‌شود. برای مثال، در منبع [65] از پرسپترون چندلایه^۱ به‌منظور شناسایی اطلاعات حرکتی جمع‌آوری شده از یک شبکه حسگر بی‌سیم استفاده می‌شود. در

^۱ Multi Layer Perseptron (MLP)

این مقاله با به کارگیری الگوریتم‌های مختلف و استفاده از الگوی حرکتی مثل راه رفتن افراد، اطلاعات دیگری را نظیر سن و وضعیت سلامت ایشان استخراج می‌نمایند. این الگوریتم‌ها عبارت‌اند از پرسپترون چندلایه، درخت تصمیم، ماشین بردار پشتیبان، رده‌بند بیزی ساده و شبکه بیزی که البته با توجه به اینکه اندازه مجموعه داده‌های آموزشی و داده‌های آزمون کوچک هستند، پرسپترون چندلایه بالاترین دقت را در رده‌بندی گروه‌ها ارائه می‌کند. ژنگ و همکاران در مرجع [66] نوعی از نگاشت خودسازمان‌ده^۱ را برای خوشه‌بندی اعمال روزمره پیشنهاد دادند. با استفاده از الگوریتم مذکور الگوهای فعالیت اصلی داده‌ها به انضمام داده‌های غیرمعمول و ناهنجاری‌های رفتاری، آشکار و شناسایی می‌شوند. یکی از مشکلات این روش این است که لازم است پارامترهای یادگیری بهینه پیش از مقداردهی نرخ یادگیری و اندازه همسایگی مشخص باشند.

سیستم فازی نیز کاندیدای بسیار مناسبی برای مدل‌سازی داده‌های غیرقطعی و مشکوک در شبکه‌های حسگر است. برای مثال، یک سیستم تشخیص افعال روزانه در توسط مجاهد و همکاران با استفاده از منطق فازی پیشنهاد شده است تا با بررسی اعمال روزانه یک فرد، شرایط یک‌خانه را برای رسیدن به محیطی ایمن، راحت و مناسب تنظیم کند [67]. البته در این تحقیق، منطق فازی تنها روی اطلاعات شبیه‌سازی شده بکار گرفته شد. همچنین اخلاقی‌نیا در رساله دکتری خود [68]، یک مدل پیش‌بینی کننده فازی ایجاد کرده و نتایج حاصله از این مدل را با مدل‌های پیش‌بینی سری‌های زمانی سنتی مثل ARMA، «سیستم استنتاج فازی مبتنی بر شبکه انطباقی»^۲ و مدل استنتاج عصبی-فازی با نرمال‌سازی وزنی مقایسه کرده است. یکی از محدودیت‌ها در این است که تکنیک‌های پیشنهاد شده روی داده‌های پیچیده توأم با نویز، در یک فاصله زمانی بلندمدت بکار گرفته نشده‌اند.

^۱ Self Organized Map (SOM)

^۲ Adaptive Network-based Fuzzy Inference Systems (ANFIS)

نویسندگان مرجع [69] یک رهیافت یادگیری و انطباق فازی را برای عامل‌ها موسوم به سیستم استنتاج فازی برخط انطباقی برای محیط‌های محاسباتی فراگیر پیشنهاد کرده‌اند. این روش متشکل از پنج مرحله است:

۱. نظارت و مانیتورینگ رفتار کاربران

۲. استخراج داده‌های واقعی مربوط به فعالیت کاربران

۳. محاسبه توابع عضویت فازی از داده‌های ورودی/خروجی

۴. تولید قوانین فازی از داده‌ها

۵. کنترل فرایند یادگیری و انطباق عامل

پس‌ازاین مراحل، عامل هوشمند نیازهای کاربر را یاد گرفته، پیش‌بینی کرده و با آن‌ها منطبق می‌شود. لازم به ذکر است که آزمایش‌ها به مدت پنج روز در محیط واقعی یک خوابگاه هوشمند انجام شد و نتایج نشان داد که سیستم استنتاج فازی برخط تطبیقی، خطاهای پیش‌بینی کمتری را نسبت به سیستم استنتاج عصبی-فازی تطبیقی و برنامه‌نویسی ژنتیک تولید کرده است.

۳-۳ تشخیص ناهنجاری

تکنیک‌های آماری مختلفی برای تشخیص ناهنجاری پیشنهاد شده‌اند که از آن جمله می‌توان به میانگین، انحراف معیار و z-score (امتیاز استاندارد) اشاره نمود. البته استفاده از معیار امتیاز استاندارد مرسوم‌تر است زیرا در اکثر موارد، داده‌های تولیدشده توزیع نرمال (یا شبیه به نرمال) ندارند، بنابراین میانگین و انحراف معیار، معیارهای خیلی مطلوبی برای تشخیص ناهنجاری‌ها نمی‌باشند.

استفاده از نمودار جعبه‌ای^۱ نیز به‌عنوان روشی برای بیان گرافیکی بررسی مجموعه داده‌ها بکار می‌رود. در منبع [70]، نمودار جعبه جهت شناسایی ناهنجاری‌هایی که به‌طور غیرمنتظره خارج از چارچوب داده‌ها یافت می‌شدند بکار گرفته شد. یک مزیت مهم نمودار جعبه در این است که می‌تواند

^۱ Box Plot

اختلافات بین جمعیت‌هایی بدون وجود فرضی درباره توزیع آماری زیربنایی و فاصله بین بخش‌هایی که جعبه درجه پخش^۱ و عدم‌توازن^۲ آن‌ها را مشخص می‌کند نشان دهد.

در حوزه کشف ناهنجاری در خانه‌های هوشمند، دو مقوله کنش‌های معمول و مدل‌سازی رفتار مفاهیمی پایه و کلیدی به شمار می‌آیند. بیشتر محققان تلاش کرده‌اند تا با تعریف و صورت‌بندی این مفاهیم بر اساس داده‌های حسگرها، ناهنجاری را به‌عنوان انحراف از آن‌ها معرفی کنند. به بیان فنی‌تر، داده‌های خام حسگرها با استفاده از روش‌های بدون ناظر خوشه‌بندی می‌شوند. این خوشه‌ها در واقع معرف افعال و کنش‌های معمول هستند و هرگونه انحراف و فاصله معنادار از آن‌ها، به‌منزله ناهنجاری تلقی خواهد شد. محققان مذکور برای انجام این کار، یک یا چند گروه از روش‌های زیر را مورد استفاده قرار داده‌اند :

روش‌های منطقی، روش‌های آماری، روش‌های مبتنی بر قاعده و روش‌های یادگیری ماشین.

ماهیت تصادفی و غیر یقینی افعال انسانی، مدل‌سازی این افعال را از طریق منطق رایج دشوار کرده است. به همین دلیل برخی از محققین تلاش کرده‌اند تا با توسل به منطق و استنتاج بیزی، این عدم قطعیت را در کشف و بازشناسی افعال ناهنجار لحاظ کنند. از جمله مرجع [71] سعی کرده با در نظر گرفتن احتمال فعال شدن یک حسگر منفرد، احتمال فعال شدن رشته‌ای از حسگرها، و احتمال فعال ماندن یک حسگر، به‌عنوان سه متغیر تصادفی رفتار ساکنان خانه هوشمند را مدل نموده و ناهنجاری‌های این رفتارها را شناسایی کند. نویسندگان مرجع [72] نیز تلاش کرده‌اند با به‌کارگیری یک شبکه منطقی مارکوف^۳، قدرت منطق درجه اول را با توان احتمال تجمیع کرده و با ابزار منطق احتمالاتی در فضای غیرقطعی داده‌های یک‌خانه هوشمند به جستجوی افعال و حالات ناهنجار بپردازند.

^۱ Spread

^۲ Skewness

^۳ Markov Logic Network

مدل پنهان مارکوف را نیز می‌توان یک روش متداول برای کشف الگوی رفتار ناهنجارانه به‌وسیله شبکه‌های حسگر در شرایط داخلی و خارجی دانست. فی‌المثل در مرجع [73]، خان و همکاران از سه فرم برای کاربرد HMM استفاده کرده‌اند. اولین فرم برای فعالیت‌های طبیعی بکار می‌رود. فرم دوم همه فعالیت‌های طبیعی را درون یک کلاس جای‌داده و فعالیت‌های ناهنجاری بر مبنای یک حد آستانه کشف می‌شوند. در فرم سوم، یک HMM روی فعالیت‌های طبیعی آموزش داده می‌شود درحالی‌که فعالیت‌های ناهنجارانه با استفاده از هم‌وردایی^۱ نقاط داده‌ای فعالیت‌های طبیعی تقریب زده می‌شوند. در اینجا، تمام نقاط داده‌ای رویدادهای طبیعی برای یادگیری و نقاط داده‌ای رویدادهای ناهنجار برای آزمون استفاده می‌شوند.

منبع [74] برای کشف افعال ناهنجار، سلسله رویدادهای خروجی حسگرها را با استفاده از مدل پنهان مارکوف مدل‌سازی کرده و مشاهدات نادر را به‌عنوان ناهنجاری در نظر گرفته است. نویسندگان این مرجع در تدوین مدل خود هیچ عنصر زمینه‌ای مثل زمان یا مکان را در نظر نگرفته‌اند؛ اما روز را به بازه‌های زمانی تقسیم کرده‌اند و با ایجاد یک مدل جداگانه برای هر بازه، دقت روش خود را در کشف افعال ناهنجار افزایش داده‌اند. نویسندگان مرجع [75] نیز برای تشخیص ناهنجاری از مدل پنهان مارکوف استفاده کرده‌اند. در این مرجع تفسیر سطح بالای خروجی‌های حسگرها، یعنی افعال، به‌عنوان ورودی به مدل داده‌شده است. بعلاوه تغییرات رفتاری و وضعیت سلامت فرد به ترتیب با تحلیل الگوهای اعمال روزانه و علائم حیاتی او اندازه‌گیری گردیده است. خروجی این سه مدل در اختیار یک سیستم فازی قرار گرفته تا در آنجا تصمیم‌نهایی در مورد هنجار و ناهنجار بودن وضعیت اتخاذ شود. این روش پیچیده از دونقطه ضعف عمده رنج می‌برد. نخست اینکه دقت آن تا حد زیادی به روشی وابسته است که برای کشف و استخراج افعال از خروجی‌های حسگرها و ارائه تفسیر سطح بالای این رویدادها به کار می‌رود. دوم آنکه علی‌رغم توجه به عامل زمینه‌ای زمان، عوامل زمانی دوار مهمی مثل تعطیلات و وضعیت آب‌وهوا را نادیده گرفته است.

^۱ Covariance

مرجع [76] هم روش بسیار پیچیده‌ای را عرضه کرده است. در این روش علاوه بر حسگرهای دودویی، حسگرهای متنوع دیگری مثل حسگر نور، حرارت، فشارخون، و صدا مورد استفاده قرار گرفته است. خروجی این حسگرها با روش‌های مختلفی دسته‌بندی شده و سلسله افعال را تشکیل داده‌اند. این سلسله با استفاده از منطق فازی و تفکر مورد گرا^۱ برای یافتن افعال ناهنجار مورد تحلیل و بررسی قرار گرفته است. در این تحلیل بسیاری از عوامل زمینه‌ای مثل زمان، مکان، روز هفته، و عوامل محیطی و فیزیولوژیک افراد لحاظ گردیده‌اند؛ اما اتکای این روش به تفکر مورد گرا به این معناست که روش یادشده کاملاً داده محور نبوده و به دانش پیشین، سخت محتاج است. در واقع نویسندگان مرجع هیچ مدلی را برای اعمال هنجار ارائه نکرده‌اند. به همین علت لازم است که در شروع کار سامانه، دست‌کم برخی از موارد به‌وسیله‌ی خبره به هنجار و ناهنجار طبقه‌بندی شده باشد تا پس از آن سابقه جریان افعال از طریق این طبقه‌بندی محک خورده و ضمناً پایگاه دانش سامانه هم به‌نگام شود.

شبکه‌های عصبی مصنوعی نیز به‌منظور تمایز بین الگوهای رفتاری طبیعی و ناهنجاری انسان به کمک حسگرهای سطح پایین استفاده فراوانی دارند. در مرجع [77] به امکان وابستگی ناهنجاری‌ها به ترتیب زمانی رخ دادن الگوهای رفتاری اشاره شده است. یک شبکه عصبی مصنوعی زمانی^۲ بر مبنای عامل تعبیه شده پیشنهاد می‌شود. آزمایش‌ها روی داده‌های زمان واقعی جمع‌آوری شده از حسگرها و فعال‌سازهای سطح پایین انجام شده است تا ناهنجاری‌های موجود در الگوهای رفتاری کاربر پس از ظهور فعالیت جدید در این الگوها، مورد شناسایی قرار بگیرد.

در منبع پر مراجعه [78] یک عامل ناهنجاری محلی (برای یافتن سایر ناهنجاری‌ها در داده‌های چندبعدی) توسط برونینگ و همکاران معرفی می‌شود. این عامل، محلی است چراکه این روش به هر شیء مورد بررسی، درجه‌ای تخصیص می‌دهد تا از دیگر همسایگان خود ایزوله بماند. یکی از محدودیت‌های استفاده از عامل ناهنجاری محلی در توضیح استثنا بودن این‌گونه ناهنجاری‌ها است.

^۱ Case-based Reasoning

^۲ Temporal ANN

این مسئله برای داده‌های با ابعاد زیاد خیلی مهم است زیرا ممکن است که ناهنجاری‌های محلی فقط در بعضی از ابعاد دور ریخته شوند اما در بقیه ابعاد، حذف نشده و حضورداشته باشند.

رفتارهای ناهنجارانه یا غیرطبیعی با استفاده از توابع فاصله و همانندی نیز شناسایی می‌شوند. حجم قابل توجهی نگارش علمی در خصوص معیارهای همانندی^۱ یا فاصله تاکنون منتشر شده است. این مطالعات کاربرد این معیارها را روی طیف گسترده‌ای از کاربردها بررسی می‌کنند. با این حال، تاکنون پژوهش‌های محدودی گزارش شده‌اند که این معیارها را در یک محیط هوشمند و خصوصاً شناسایی الگوی رفتاری انسان به کار بسته باشند.

نویسندگان منبع [79] آزمون T^2 فاصله ماهالانوبیس را برای شناسایی رویدادهای ناهنجار در داده‌های زمانی در یک محیط هوشمند بکار گرفته‌اند. مجموعه داده‌ها با استفاده از یک روش همساز کرنل تولید می‌شوند تا به داده‌هایی برسیم که می‌توانند به‌عنوان الگوی رفتار شخص در بازه زمانی یک هفته‌ای مورد استفاده قرار بگیرند.

منبع [80] از فاصله ویرایش لوینستاین برای یافتن شباهت بین دو الگو استفاده کرده است. فاصله ویرایش بین دو توالی برابر با تعداد ویرایش‌های (درج، حذف، جایگزینی) مورد نیاز برای تبدیل یک توالی به توالی دیگر است. رشیدی و همکاران در این پژوهش، یک تکنیک داده‌کاوی جدید موسوم به روش داده‌کاوی ناپیوسته ترتیب-تنوعی^۲ را برای یافتن دفعات تکرار آن دسته از الگوهایی که ممکن است از نظر ترتیب، نامنظم و غیرمعمول باشند، پیشنهاد داده‌اند. همچنین برای گروه‌بندی الگوها به تعاریف فعالیت، از یک تکنیک خوشه‌بندی استفاده می‌شود، بطوریکه مراکز خوشه، منطبق با فعالیت‌هایی هستند که شناسایی و رده‌بندی شده‌اند.

در منبع [81]، سیستمی به‌منظور تشخیص ناهنجاری پیشنهاد شده است. این سیستم، از یک الگوریتم کی-نزدیکترین همسایگی که در آن از «پیش‌گسسته زمان»^۳ به‌عنوان معیار فاصله

^۱ Similarity Measures

^۲ Discontinues Varied-order Mining

^۳ Discrete Time Warping (DTW)

استفاده شده، بهره می برد. ثابت شده است که این الگوریتم جهت تمایز بین داده های طبیعی و ناهنجاری در محیط های هوشمند عملکرد بسیار خوبی دارد. همچنین ادعا شده است که این روش گسسته زمان نسبت به معیارهای فاصله استاندارد (فاصله اقلیدسی، فاصله منهن، چبیشف، ...) بهتر عمل می کند.

به دلیل تنوع و گستردگی افعال انسانی بسیاری از پژوهش های پیشین بر روی نوع خاصی از افعال متمرکز شده اند. مثلاً محققین در [82] سعی کرده اند با به کارگیری شبکه های عصبی خودسازمان ده، وجود ناهنجاری در افعال روزمره را کشف کنند، مشروط به آنکه زمان این افعال کمتر از ۱۵ دقیقه باشد. نویسندگان مرجع [83] ویژگی های زمانی و مکانی متعددی را مورد توجه قرار داده اند، اما تنها به کشف دوره های بی حرکتی و عدم فعالیت بسنده کرده و این دوره ها را به عنوان علائم وقوع ناهنجاری، به ویژه در مورد سالخوردگان، در نظر گرفته اند. پدیدآورندگان مرجع [84] نیز با استفاده از گراف، مجموعه افعال ۴۰۰ فرد سالخورده را برای یافتن ناهنجاری و علائم بیماری مورد تحلیل و بررسی قرار داده اند؛ اما همه این افراد مجموعه مشخص و از پیش تعریف شده ای از افعال را انجام داده اند. کارهای متعدد دیگری مثل [85] نیز وجود دارند که برای کشف ناهنجاری های افعال یک خانه هوشمند از ابزارهایی نظیر OCSVM استفاده کرده اند؛ اما در بیشتر این کارها، تنها ناهنجاری های نقطه ای و یا تجمیعی مورد توجه قرار گرفته اند و زمینه های مهمی مثل ساعت از شبانه روز، روز هفته، فصل، و... از نظر دور مانده اند.

فصل ۲ : فرایندهای نقطه‌ای

۴-۱ مقدمه

مجموعه رویدادهای حسگرها را می‌توان با استفاده از مفهوم فرایندهای^۱ نقطه‌ای مدل‌سازی کرد. این مفهوم اصولاً از حوزه آمار و احتمال اخذ گردیده است و محققین یادگیری ماشین در سال‌های اخیر آن را مورد توجه فراوانی قرار داده‌اند. باید توجه داشت که فرایندهای نقطه‌ای و سلسله رویدادهای حاصل از آن‌ها با سری‌های زمانی تفاوت دارند. رویدادها در فواصل نامساوی و به شکل غیر هم‌زمان اتفاق می‌افتند، درحالی‌که فاصله بین عناصر یک سری زمانی ثابت و مساوی است.

برای تبدیل یک سلسله رویداد به یک سری زمانی باید زمان را به بازه‌های متساوی تقسیم کنیم و در ابتدای هر یک از بازه‌ها از همه رویدادها نمونه‌برداری نماییم. بدین ترتیب در ابتدای هر بازه یک رشته دودویی خواهیم داشت که هر یک از عناصر آن معرف وقوع (۱) یا عدم وقوع (۰) یکی از رویدادها در زمان مربوطه است. مجموعه این رشته‌ها یک سری زمانی متعارف را تشکیل می‌دهند.

اما این رویکرد از دو مشکل رنج می‌برد. نخست اینکه در بسیاری از کاربردها معیاری برای تعیین اندازه بازه‌های زمانی نداریم. اندازه‌های کوچک باعث افزایش حجم داده‌ها و خلوت شدن جدول رویدادها می‌شود و اندازه‌های بزرگ وقوع برخی از رویدادها را پنهان می‌کند. مثلاً فرض کنید خواهیم از حسگرهای یک‌خانه هوشمند نمونه‌برداری کنیم. در چنین حالتی واحد زمانی ساعت بسیاری از افعال (مثلاً مسواک زدن) را پنهان می‌کند و واحد زمانی ثانیه بعضی از افعال طولانی (مثلاً خوابیدن) را به یک جدول خلوت سراسر صفر تبدیل می‌نماید.

دوم اینکه فاصله بین رویدادها در فرایندهای نقطه‌ای گاه به عنصری اساسی بدل می‌شود که

^۱ Point Process

رفتار پویای سیستم پس‌زمینه تولید رویدادها را گزارش و تبیین می‌کند. تبدیل فرایند نقطه‌ای به سری زمانی، می‌تواند امکان استخراج این مقوله مهم را منتفی یا حداقل بسیار دشوار نماید. در این حالت یک متغیر تصادفی به یک اندیس قطعی تبدیل می‌شود و در نتیجه بهره مهمی از بار معنایی خود را از دست می‌دهد.

۲-۴ فرایندهای نقطه‌ای

فرایندهای نقطه‌ای مجموعه‌ای از نقاط تصادفی هستند که در یک فضای نمونه، مثل زمان یا مکان پراکنده شده‌اند. این فرایندها در واقع زبان آماری مناسب را برای بیان زمان‌بندی و سایر ویژگی‌های رویدادها فراهم می‌کنند [86]. مفهوم رویداد، دامنه گسترده‌ای از کاربردها را در برمی‌گیرد. ما بخش حاضر را با تشریح مبانی فرایندهای نقطه‌ای و معرفی یکی از ساده‌ترین انواع آن‌ها یعنی فرایندهای پواسن^۱ آغاز می‌کنیم. مرور این مبانی کمک می‌کند که کار توصیف و تبیین انواع پیچیده‌تر فرایندهای نقطه‌ای مثل فرآیند هاوکس^۲ و فرایند کاکس^۳ در بخش‌های بعد ممکن گردیده و ساده‌تر بشود.

۱-۲-۴ تعریف یک فرایند نقطه‌ای

اگر امتداد غیر منفی محور افقی را به‌عنوان محور زمان در نظر بگیریم، یک فرایند نقطه‌ای نقطه تصادفی است که در t_i محقق شده است. مجموعه این نقاط در طول محور افقی به سمت بی‌نهایت و در T_1, T_2, \dots پراکنده شده‌اند. وجود i امین نقطه در محل t_i مبین آن است که رویدادی در زمان t_i به وقوع پیوسته است [87]. برای روشن‌تر شدن مطلب می‌توان نوع دیگری از فرایندهای تصادفی به

^۱ Poisson Process

^۲ Hawkes Process

^۳ Cox Process

نام فرایند شمارشگر^۱ را به کمک گرفت. یک فرایند شمارشگر $N(t)$ تابع تصادفی است که برای نقاط $t \geq 0$ تعریف می‌شود و مقادیر خروجی آن، اعداد صحیح مثبت یعنی $1, 2, \dots$ است. مقدار خروجی این تابع در نقطه t نشان‌دهنده تعداد رویدادهای فرایند نقطه‌ای مربوطه تا لحظه t است. به همین دلیل فرایند شمارش گر با یک توالی از متغیرهای تصادفی غیر منفی مشخص می‌شود که در شرایط دوگانه $T_i < T_{i+1}$ و $T_i \leq \infty$ صدق می‌کنند [88]. انتخاب نام شمارشگر برای این نوع فرایند بی‌وجه نیست. در واقع می‌توان گفت که فرایند $N(t)$ تعداد رویدادها را تا لحظه t می‌شمارد:

$$N(t) = \sum_{i \geq 1} I_{\{t \geq T_i\}} \quad (1-4)$$

در رابطه (۱-۴) I تابعی است که مقدار آن به شرط وقوع رویداد ۱ و در غیراینصورت ۰ است. مقدار تابع $N(t)$ در نقطه ۰ برابر صفر است: $N(0) = 0$. بعلاوه مقدار این تابع در یک محدوده محلی ثابت است و با گذر از نقطه رویداد T_i به اندازه ۱ واحد جهش کرده و تا رسیدن به نقطه رویداد بعدی (T_{i+1}) ثابت خواهد ماند. به سادگی می‌توان مشاهده کرد که مجموعه رویدادهای T_1, T_2, \dots و فرایند شمارشگر معادل آن‌ها، می‌توانند ارائه هم‌ارز و یکسانی از فرایند نقطه‌ای باشند.

شاید مهم‌ترین معیاری که درک و صورت‌بندی رفتار پویای یک فرایند نقطه‌ای را ممکن می‌کند، نرخ وقوع رویدادهای آن فرایند باشد [86]. به عنوان مثال فرض کنید یک فرایند نقطه‌ای حاوی سه نوع رویداد است و ما اکنون در لحظه t قرار داریم. اگر ما بتوانیم تعداد رویدادهایی که تا لحظه $t + \Delta t$ اتفاق می‌افتد را با دقت قابل قبولی حدس بزنیم، می‌توانیم بگوییم که بر رفتار و دینامیک این فرایند احاطه داریم. در سایه این احاطه، پیش‌بینی نوع و زمان رویداد بعدی و در نتیجه کشف رفتار هنجار و ناهنجار فرایند نیز برای ما میسر می‌گردد.

در ادبیات فرایندهای نقطه‌ای به تابعی که با در نظر گرفتن گذشته، نرخ وقوع رویدادهای آتی را مشخص می‌کند، تابع شدت شرطی^۲ گویند. این تابع به طور مختصر چنین تعریف می‌شود:

^۱ Counting Process

^۲ Conditional Intensity Function

«تابع شدت شرطی یا $\lambda(t)$ تابعی است که با در نظر گرفتن گذشته تا لحظه t نرخ مورد انتظار وقوع

یک رویداد جدید را در پنجره زمانی کوچک $(t, t+dt)$ محاسبه می‌کند.»

فرمول این تابع به شکل زیر است:

$$\lambda(t) = \lim_{\Delta t \rightarrow 0} \frac{E(N(t + \Delta t) - N(t) | H_t)}{\Delta t} = \frac{E(dN(t) | H_t)}{dt} \quad (2-4)$$

در این فرمول H_t تاریخچه وقوع رویدادها تا لحظه t است:

$$H_t = \{ (t_i, c_i) \mid t_i < t, c_i \in C \} \quad (3-4)$$

همچنین E بیانگر امید ریاضی تعداد رویدادهایی است که با در نظر گرفتن این گذشته، در فاصله زمانی $(t, t+dt)$ اتفاق خواهد افتاد. چنان‌که از رابطه (۲-۴) برمی‌آید، تابع شدت شرطی علاوه بر پارامتر زمان به تاریخچه رویدادهای تا این لحظه نیز بستگی دارد.

تابع شدت شرطی در حوزه فرآیندهای نقطه‌ای نقش اساسی دارد. این نقش در دو محور اصلی

زیر قابل واکاوی است:

۱. با تغییر اجزا و پارامترهای رابطه (۲-۴) چه تغییری در رفتار فرایند نقطه‌ای ایجاد می‌شود.

به عبارت دیگر انواع فرایندهای نقطه‌ای با عنایت به رابطه مذکور کدام است.

۲. چگونه می‌توان تابع شدت شرطی یک دادگان - مثل دادگان حسگرهای یک‌خانه هوشمند -

را با یادگیری از درون داده‌های آن آموخت؟

بدیهی است که پرسش دوم در تحقیق حاضر از اهمیت بیشتری برخوردار است. چراکه ما در این

تحقیق به دنبال راهی هستیم که تحلیل یک توالی از رویدادها و تقسیم آن به هنجار و ناهنجار را

مقدور نماید؛ اما بررسی انواع فرایندهای نقطه‌ای و تحلیل آن‌ها بر اساس ویژگی‌های تابع شدت شرطی

کمک می‌کند که ما این فرآیند را بهتر بشناسیم. این شناخت در زمان صورت‌بندی یک دادگان

به‌عنوان یک فرآیند نقطه‌ای بسیار لازم و کارآمد است. به همین دلیل ما بخش حاضر را با معرفی انواع

مشهور و پرکاربرد فرایندهای نقطه‌ای ادامه می‌دهیم تا قسمت اصلی، یعنی پاسخ پرسش دوم برسیم.

۴-۲-۲ انواع فرایندهای نقطه‌ای

۴-۲-۱ فرآیند پواسن

فرآیند پواسن ساده‌ترین نوع فرایندهای نقطه‌ای به شمار می‌آید. این فرآیند به فرآیند کاملاً تصادفی مشهور^۱ است. فرآیند پواسن در کلی‌ترین حالت به دو نوع همگن و ناهمگن تقسیم می‌شود. در فرآیند پواسن همگن، تابع شدت شرطی از تاریخ و زمان مستقل بوده و برابر یک عدد ثابت است:

$$\lambda(t) = \lambda_0 \quad (4-4)$$

در فرآیند پواسن، فاصله‌های بین رویدادهای متوالی از هم مستقل هستند. این فاصله‌ها متغیرهای تصادفی هستند که به شکل همگن در طول یک توزیع نمایی پراکنده شده‌اند [86]. خاصیت دیگر فرآیند مذکور آن است که تعداد رویدادهای دو بازه زمانی مجزا به هم هیچ وابستگی ندارد. این خاصیت فرآیند پواسن را به یک فرآیند بدون حافظه^۲ تبدیل کرده است. به همین دلیل فرآیند یادشده برای مدل‌سازی رویدادهایی که بر هم تأثیر می‌گذارند - مثل ترافیک شبکه و حسگرهای خانه هوشمند - مناسب نیست. فرآیند پواسن توسعه‌های فراوانی دارد. مشهورترین توسعه این فرآیند، فرآیند پواسن غیر همگن است که در آن تابع شدت شرطی همچنان از تاریخچه مستقل است اما به زمان وابستگی دارد. در واقع این شدت به شکل تابعی از زمان بیان می‌شود:

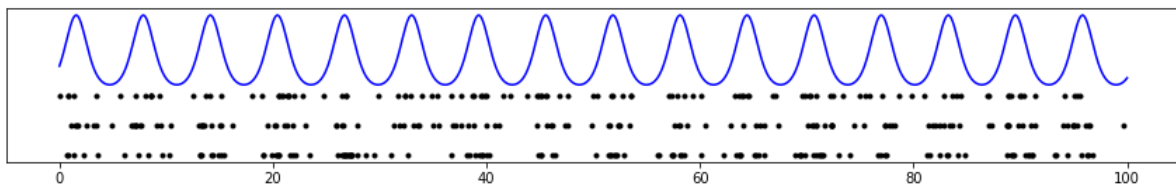
$$\lambda(t) = g(t) \quad (5-4)$$

شکل ۴-۱ نمونه‌ای از یک فرآیند پواسن ناهمگن را نشان می‌دهد که در آن:

$$g(t) = e^{\sin(t)} \quad (6-4)$$

^۱ Completely Random

^۲ Memoryless process



شکل ۴-۱: نمونه‌ای از یک فرایند پواسن ناهمگن

فرایند پواسن تقویتی^۱ و فرایند کاکس دو توسعه مشهور دیگر فرایند پواسن هستند [89]. بسیاری از پدیده‌هایی که در زمره تئوری صف قرار می‌گیرند مثل توالی مراجعه مشتریان و توالی تماس‌های یک مرکز تلفن با فرایند پواسن مدل می‌شوند.

۴-۲-۲-۲ فرایندهای هاوکس

چنان‌که گفته شد فرایند پواسن بدون حافظه است، یعنی بروز یک رویداد در افزایش یا کاهش احتمال وقوع رویدادهای بعدی هیچ تأثیری ندارد. لیکن بسیاری از پدیده‌های تصادفی جهان در این قاعده نمی‌گنجند، به عبارت دیگر در بسیاری از موارد وقوع یک رویداد احتمال پیش آمد رویدادهای آتی را تحت تأثیر قرار می‌دهد. نمونه مدرسی و مشهور پدیده مذکور، زلزله و پس‌لرزه‌های آن هستند [90]. بدیهی است که وقوع یک زلزله احتمال بروز پس‌لرزه‌های بعدی را افزایش می‌دهد. خاصیت اصلی گروه اخیر فرایندهای نقطه‌ای را می‌توان این‌گونه بیان کرد که برخلاف فرایند پواسن، تعداد رویدادهای دو بازه زمانی مجزا ممکن است از هم مستقل نباشد. بیان ریاضی این مطلب به شکل زیر است:

$$Cov(N(s, t), N(t, u)) \neq 0 \quad s < t < u \quad (۷-۴)$$

در رابطه (۷-۴) نماد Cov معرف هم‌وردایی بوده و N فرایند شمارشگر است. به علاوه s و t و u نقاط زمانی روی محور افقی هستند. در صورتی که هم‌وردایی فوق منفی باشد، فرایند از نوع خود اصلاح‌گر^۲

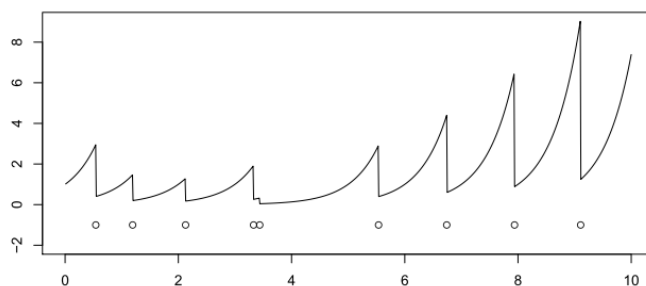
^۱ Reinforced Poisson Process

^۲ Self-correcting Process

خواهد بود [91]. در فرآیندهای خود اصلاح‌گر، وقوع یک رویداد احتمال بروز رویدادهای بعدی را کاهش می‌دهد. این نوع فرایندها معمولاً مدل‌های منظمی را تولید می‌کنند که نقاط آن‌ها در فضا یا زمان به شکلی باقاعده پراکنده شده‌اند. فرآیندهای خود اصلاح‌گر در حوزه‌هایی مثل اقتصاد و جنگلداری به کار می‌آیند و از موارد کاربرد آن‌ها می‌توان به مدل‌سازی کاشت و نگهداری درختان یک جنگل یا زادوولد موجودات یک ناحیه محدود اشاره کرد [92]. رابطه (۸-۴) یک تابع شدت شرطی نمونه را برای یک فرآیند خود اصلاح‌گر نشان می‌دهد:

$$\lambda(t) = \exp(\mu_t - \sum_{t_i < t} \alpha) \quad (8-4)$$

در رابطه (۸-۴) μ و α اعدادی مثبت هستند. تحلیل این رابطه نشان می‌دهد که با گذشت زمان تابع شدت شرطی افزایش پیدا می‌کند، اما درعین حال با وقوع هر رویداد احتمال وقوع رویداد بعدی در یک ضریب کوچک‌تر از یک ضرب شده و در نتیجه کاهش می‌یابد. الگوی پراکندگی نقاط و تابع شدت شرطی یک فرآیند خود اصلاح‌گر در شکل ۲-۴ نمایش داده شده است.



شکل ۲-۴: پراکندگی نقاط و تابع شدت شرطی یک فرآیند خود اصلاح‌گر

گروه دیگر فرآیندهای نقطه‌ای وابسته، فرآیندهایی هستند که در رابطه (۷-۴) صدق می‌کنند اما هم‌وردایی آن‌ها مثبت است. در این گروه از فرآیندها وقوع یک رویداد احتمال بروز رویدادهای بعدی را افزایش می‌دهد. به همین دلیل فرآیندهای مذکور را خود محرک^۱ می‌نامند [93]. فرآیند هاوکس یکی از مشهورترین زیرگونه‌های فرآیندهای خود محرک است. از آنجاکه ما رویدادهای حسگرهای

^۱ Self-exciting Process

خانه‌های هوشمند را بر مبنای این نوع از فرآیندهای نقطه‌ای مدل‌سازی کرده‌ایم، آن را به شکلی تفصیلی مورد بررسی قرار می‌دهیم.

رشته‌ای از رویدادها به شکل زیر را در نظر بگیرید:

$$\{(t_i, u_i)\}_{i=1}^n \quad t_i \in R^+ \quad \text{و} \quad u_i \in U = \{1, 2, \dots\} \quad (9-4)$$

در این رشته t_i و u_i به ترتیب بیانگر زمان و بعد هستند. به عبارت دیگر رویداد i ام در زمان t_i و در بعد u_i رخ داده است. مثلاً u می‌تواند مبین نوع حسگر باشد. در چنین رشته‌ای فرآیند هاوکس با تابع شدت شرطی زیر صورت‌بندی می‌شود [87]:

$$\lambda_u(t, \Theta) = \mu + \sum_{i, t_i < t} h_{uu_i}(t - t_i; \theta_{uu_i}) \quad (10-4)$$

در این فرمول $\Theta = (\mu, \theta)$ پارامترهای مدل هستند و ماتریس H حاوی توابع h_{ij} ماتریس کرنل‌های ماشه^۱ نام دارد:

$$H = [h_{ij}] \quad h_k(t) : R^+ \rightarrow R^+ \quad (11-4)$$

ماتریس H که تابع تحریک^۲ یا کرنل کاهش^۳ نیز نامیده می‌شود بیانگر وابستگی زمانی رویدادهاست و به ازای هر بعد u_i تغییر می‌کند. در واقع فرآیند هاوکس را می‌توان به عنوان یک فرایند پواسن مرکب در نظر گرفت که از دو بخش اصلی تشکیل شده است. در بخش اول یک فرایند پواسن همگن قرار دارد که تابع شدت شرطی آن μ است و به فرآیند پس‌زمینه شهرت دارد. در بخش دوم نیز تعدادی فرآیند پواسن غیر همگن وجود دارند که تابع شدت شرطی i امین فرآیند $h_{uu_i}(t - t_i; \theta)$ است. این دو بخش با عملگر برهم‌نهی^۴ باهم ترکیب شده‌اند و فرآیند جدیدی را ایجاد کردند که خواصی فراتر است از فرایند پواسن دارد [94].

^۱ Triggering Kernels

^۲ Excitation function

^۳ Decay Kernel

^۴ Superposition

میزان و نوع تأثیر رویدادهای گذشته در شانس وقوع رویدادهای آتی از طریق کرنل ماشه مشخص می‌شود. این کرنل را می‌توان مستقیماً مقداردهی کرده و یا آن را از دادگان موجود آموخت. در صورت مقداردهی مستقیم معمولاً کرنل را به گونه‌ای انتخاب می‌کنند که مقدار آن با گذشت زمان کاهش یابد. بدین ترتیب چنان‌که رویدادهای قبلی دورتر می‌شوند میزان تأثیر آن‌ها در رویدادهای فعلی و بعدی نیز کاهش می‌یابد. یکی از بهترین گزینه‌ها برای این مقصود تابع نمایی زیر است:

$$h(t) = \omega e^{-\omega t} \quad (12-4)$$

اضافه کردن یک ضریب تأثیر به رابطه (12-4) اجازه می‌دهد که این تابع را برای هر بعد u_i اختصاصی نماییم:

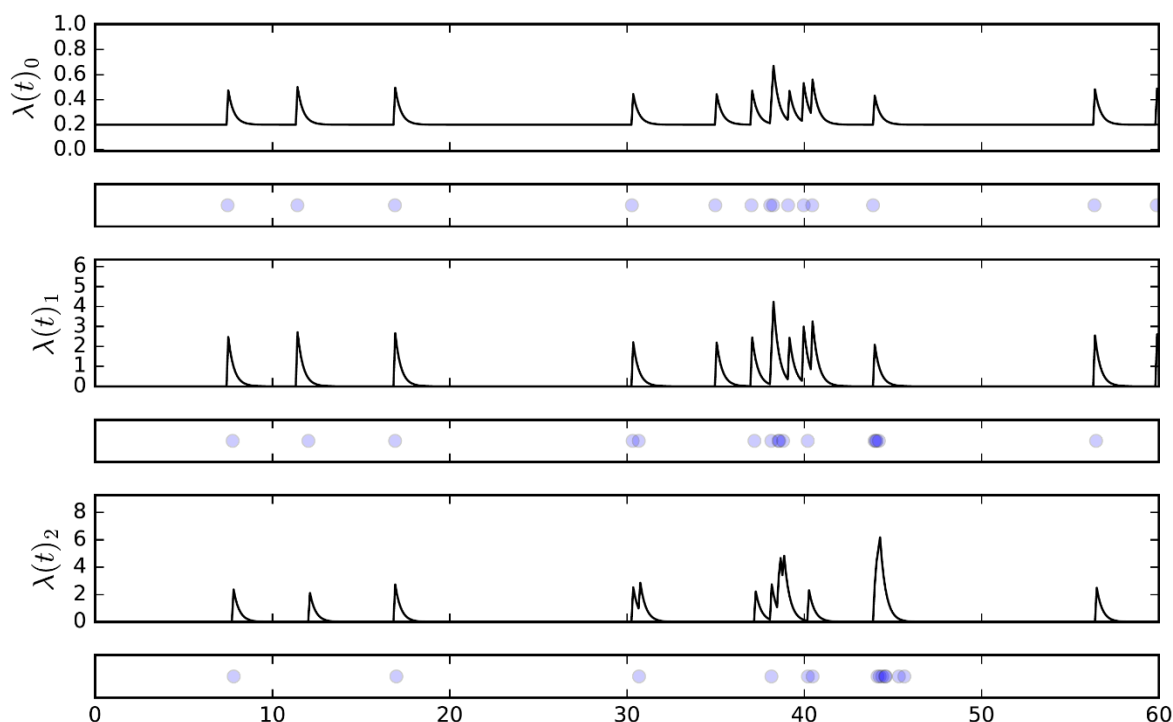
$$h_{uu'}(t) = \alpha_{uu'} \omega_{uu'} e^{-\omega_{uu'} t} \quad (13-4)$$

به‌عنوان مثال فرض کنید سه حسگر به اسامی e_0 ، e_1 و e_2 داریم که به‌صورت متوالی و پشت سر هم قرار گرفته‌اند. در این حالت روشن شدن e_0 بر e_1 تأثیر گذاشته و روشن شدن e_1 نیز احتمال روشن شدن e_2 را افزایش می‌دهد. ما برای شبیه‌سازی این وضعیت پارامترهای μ و α را به شکل زیر مقداردهی می‌کنیم:

$$\mu = [0.2 \quad 0.0 \quad 0.0]$$

$$\alpha = \begin{bmatrix} 0.1 & 0.0 & 0.0 \\ 0.9 & 0.0 & 0.0 \\ 0.0 & 0.9 & 0.0 \end{bmatrix}$$

شکل 3-4 شبیه‌سازی یک فرآیند هاوکس را بر اساس روابط (10-4) و (13-4) و اعداد فوق به نمایش می‌گذارد. از آنجاکه توالی رویدادها می‌بایست به شکلی تصادفی آغاز شود، فرایند پس‌زمینه در مورد e_0 خاموش نشده است. از داده‌های ماتریس تحریک نیز چنین برمی‌آید که e_0 بر e_1 و e_1 بر e_2 تأثیر می‌گذارد. بدین ترتیب رویدادهای e_1 و e_2 دیگر تصادفی نیستند. این رویدادها نتیجه رویدادهای قبلی خود بوده و با دقت خوبی قابل پیش‌بینی هستند.

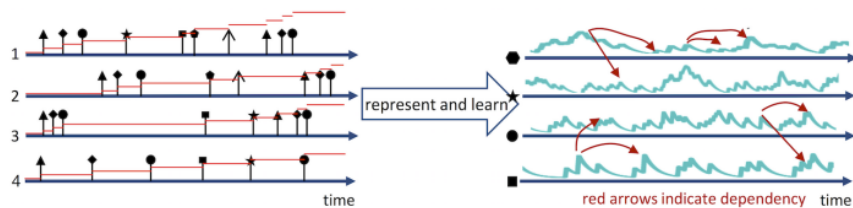


شکل ۴-۳: شبیه‌سازی یک فرایند هاوکس با سه حسگر وابسته

چنانکه دیده می‌شود در فرایندهای معطوف به تاریخچه، مثل فرایند هاوکس زمان از حالت اندیس صرف خارج می‌شود و وجه علی می‌یابد. به همین دلیل در سال‌های اخیر محققین حوزه‌های مختلف از جمله هوش مصنوعی توجه فراوانی به فرایندهای نقطه‌ای به‌ویژه فرایند هاوکس نشان داده و از آن برای کشف و تحلیل روابط علت و معلولی در میان رشته‌های رویدادی استفاده کرده‌اند.

از آنجاکه اساس کار تحقیق حاضر نیز بر رشته‌ای از رویدادها قرار دارد، به نظر می‌رسد که فرایند نقطه‌ای چارچوب مناسبی برای صورت‌بندی و انجام آن باشد. در هر خانه هوشمند تعدادی حسگر وجود دارد که توالی فعال شدن آن‌ها به همراه برچسب زمان ثبت می‌شود. در صورتی که ما این توده درهم را مطابق شکل ۴-۴ به مجموعه‌ای از توالی‌های همگن (هر حسگر یک توالی) تفکیک کنیم، می‌توانیم برای بیان و ارائه صوری آن از یک فرایند هاوکس بهره بگیریم. به کمک این فرایند، تجزیه و تحلیل رویدادهای ثبت‌شده و پیش‌بینی نوع رویدادهای پیش‌رو و زمان وقوع آن‌ها بسیار ساده‌تر خواهد بود. بعلاوه - و برای ما - مهم‌تر اینکه ما می‌توانیم با استخراج روابط علت و معلولی و

تأثیرات متقابل رویدادها از میان توابع تحریک و ضرایب تأثیر، رشته‌های قانونی و هنجار و رشته‌های بی‌قاعده و ناهنجار را پیدا کنیم؛ اما تحقق این مهم در گرو آن است که جان‌مایه و اصل فرآیند نقطه‌ای یادشده، یعنی تابع شدت شرطی آن مشخص باشد. از آنجاکه ما از داده به‌سوی مدل حرکت می‌کنیم و چیزی جز دادگان رویدادهای حسگر یک‌خانه هوشمند را در اختیار نداریم، برای رسیدن به این تابع باید جواب سؤال دوم بخش قبل را پیدا کنیم:



شکل ۴-۴: تفکیک رویدادهای درهم به گروه‌های هم‌جنس

«چگونه می‌توان تابع شدت شرطی یک دادگان - مثل دادگان حسگرهای یک‌خانه هوشمند - را با یادگیری از درون داده‌های آن آموخت؟»

۴-۲-۳ استخراج پارامترهای تابع شدت شرطی - شیوه آماری

در حوزه فرایندهای نقطه‌ای - مثل بسیاری از زمینه‌های دیگر داده‌کاوی - برای تخمین پارامترهای مدل از روش برآورد درست‌نمایی بیشینه^۱ یا MLE استفاده می‌شود [95]. برای توضیح این روش فرض کنید D توالی به شکل زیر مشاهده شده است:

$$S = \{S_1, S_2, \dots, S_D\} \quad (۱۴-۴)$$

هر یک از عناصر این توالی به شکل زیر تعریف می‌شوند:

$$S_d = \{t_j^d\}_{j=1}^{N_d} \quad (۱۵-۴)$$

^۱ Maximum Likelihood Estimation

ما اولین برچسب زمان را برابر صفر و آخرین برچسب را برابر با طول پنجره مشاهده در نظر می‌گیریم، یعنی $t_0 = 0$ و $t_{Nd} = T_d$. بعلاوه فرض می‌کنیم که رابطه زیر در میان زمان‌ها و اندیس‌ها برقرار است:

$$\text{If } i < j \Rightarrow t_i^d < t_j^d \quad (16-4)$$

با این مفروضات تابع چگالی احتمال^۱ برای نقطه t به صورت رابطه زیر بیان می‌شود [87]:

$$f_d(t_j | t_1, \dots, t_{j-1}) = \lambda_d(t_j) \exp\left(-\int_{t_{j-1}}^{t_j} \lambda_d(t) dt\right) \quad (17-4)$$

بنابراین درست‌نمایی لگاریتمی^۲ رشته کلی یا L_d به شکل زیر خواهد بود:

$$\log \prod_{j=1}^{N_d} f(t_j | t_1, \dots, t_{j-1}) = \sum_{j=1}^{N_d} \log(\lambda_d(t_j)) - \int_0^{T_D} \lambda_d(t) dt \quad (18-4)$$

در صورتی که فرایند موردنظر چند متغیره باشد، یعنی در آن بیش از یک نوع رویداد وجود داشته باشد،

آنگاه هر یک از رویدادها علاوه بر برچسب زمان از یک علامت^۳ یا بعد نیز برخوردار است [96]. در این

حالت هر یک از عناصر رشته S به شکل زیر ارائه می‌شوند:

$$S_d = \{(t_j^d, m_j^d)\}_{j=1}^{N_d} \quad (19-4)$$

اگر تعداد انواع رویداد (مثلاً تعداد حسگرهای یک‌خانه هوشمند) را برابر با M در نظر بگیریم، به M

عدد تابع شدت شرطی نیاز خواهیم داشت که هر یک از آن‌ها رفتار یکی از رویدادها را کنترل می‌کند.

در این حالت درست‌نمایی لگاریتمی باید به شکل زیر بازنویسی شود:

$$\sum_{d=1}^D \sum_{m=1}^M L_d^m = \sum_{d=1}^D \left(\sum_{j=1}^{N_d} \log(\lambda_d^{m_j}(t_j)) - \sum_{m=1}^M \int_0^{T_D} \lambda_d^m(t) dt \right) \quad (20-4)$$

روش معمول برای یافتن پارامترهای تابع شدت شرطی این است که مشتق تابع فوق را محاسبه کرده

و با صفر کردن این مشتق به دنبال نقاط عطف (بیشینه یا کمینه) جستجو کنیم. نقاط یادشده در واقع

پارامترهای تابع شدت شرطی موردنظر هستند.

^۱ Probability Density Function

^۲ Log Likelihood

^۳ Mark

استفاده از روش بیشینه‌سازی درست‌نمایی مشکلاتی دارد که به آن‌ها اشاره می‌کنیم [97]:

- صحت روابط (۴-۱۸) و (۴-۲۰) و در نتیجه درستی نتایج روش برآورد درست‌نمایی بیشینه متضمن آن است که متغیرها مستقل باشند و این امر در بسیاری از موارد اتفاق نمی‌افتد.
- به دست آوردن مشتق و حل تحلیلی روابط (۴-۱۸) و (۴-۲۰) برای بسیاری از توابع ممکن نیست. در این حالت باید از روش‌های عددی استفاده کنیم.
- روش بیشینه کردن درست‌نمایی به شدت مستعد درافتادن به ورطه بیش‌برازش^۱ است.
- می‌توان ثابت کرد که در صورت وجود داده کافی روش درست‌نمایی قطعاً به پاسخ همگرا می‌شود؛ اما حتی به شرط وجود داده کافی، معمولاً سرعت این همگرایی با نزدیک شدن به نقاط عطف به شدت کاهش پیدا می‌کند. انتخاب نقطه آغازین تأثیر بسزایی در پیدایش و حل این مشکل دارد.

برای روبرویی با این مشکلات، روش‌های متعدد و کارآمدی مثل به‌کارگیری شیوه‌های تنظیم^۲، استفاده از روش بیشینه‌سازی توقعات^۳ و بهره‌برداری از چهارچوب‌های بیزی^۴ پیشنهاد شده است [98]. در واقع می‌توان گفت که روش برآورد درست‌نمایی بیشینه علی‌رغم مشکلات ذکر شده رایج‌ترین روش برای تخمین پارامترها در داده‌کاوی و آمار به شمار می‌آید؛ اما آیا استفاده از این روش برای تحقیق حاضر هم مناسب است؟

ما دادگانی داریم که از مجموعه رویدادهای تعدادی حسگر دودویی در یک‌خانه هوشمند تشکیل شده است. هر یک از این حسگرها بسته به افعال ساکنان خانه روشن یا خاموش شده‌اند و این رویدادها همراه برچسب زمان مربوطه در دادگان ثبت گردیده است. ما می‌توانیم مطابق شکل ۴-۴ با تفکیک انواع حسگرها، این دادگان را به یک فرایند نقطه‌ای چند متغیره تبدیل کنیم. سپس با استفاده

^۱ Overfitting

^۲ Regularization

^۳ Expectation Maximization

^۴ Bayesian Frameworks

از روش برآورد درست‌نمایی بیشینه پارامترهای مدل این فرآیند را از مشاهدات موجود استخراج نماییم. درواقع ساختن چنین مدلی تمام مشکلات ما را حل خواهد کرد.

اما یک مانع بزرگ ما را از حصول به راه‌حل فوق‌بازمی‌دارد. نکته اینجاست که ما باید پیش از اقدام برای تخمین پارامترهای مدل، ابتدا مدل موردنظر خود را انتخاب کنیم، یعنی با ملاحظه داده‌ها تصمیم بگیریم که روند رویدادها مطابق کدام یک از انواع فرایندهای نقطه‌ای پیش می‌رود: پواسن، هاوکس، کاکس یا نوعی دیگر. توجه داشته باشید که هرکدام از این انواع، گونه خاصی از وابستگی‌ها را در میان تاریخچه رویدادها لحاظ می‌کنند. مدل پواسن فرض می‌کند که رویدادها از هم مستقل هستند و طول مدت آن‌ها یک سری زمانی ایستا را تشکیل می‌دهد. در منظر مدل هاوکس رویدادهای گذشته به شکل خطی بر افزایش احتمال رویداد جاری تأثیر می‌گذارند و مدل خودتصحیح‌گر این تأثیرات را کاهشی و غیرخطی می‌داند. این مدل‌ها جملگی از دانش ما می‌آیند و پدیده‌های عالم واقع مجبور نیستند مطابق تقسیم‌بندی آن‌ها رفتار کنند. حقیقت این است که ما معمولاً قواعد درونی داده‌های خود را نمی‌شناسیم و برای انتخاب یکی از این مدل‌ها ناگزیر از سعی و خطا هستیم. البته روش‌ها و آزمایش‌هایی وجود دارند که به ما کمک می‌کنند در مورد اینکه رویدادهای یک دادگان به کدام مدل فرآیندهای نقطه‌ای نزدیک‌تر است تصمیم بگیریم؛ اما صحت و قابلیت اعتماد این روش‌ها غالباً مستلزم آن است که ما دادگان خود را کوچک‌تر و ساده‌تر کنیم. کوتاه‌سخن اینکه انتخاب یا تأسیس مدلی که یک دادگان بزرگ و متنوع مثل دادگان حسگرهای خانه هوشمند مطابق آن رفتار کند تقریباً غیرممکن است.

البته باید توجه داشت که هدف اصلی ما از تحلیل رویدادهای گذشته، پیش‌بینی رویدادهایی است که در آینده اتفاق خواهند افتاد. به همین دلیل اصراری به مدل کردن تاریخچه رویدادهای گذشته نداریم و به دنبال نگاشتی هستیم که با دریافت این تاریخچه یا بخشی از آن آینده را در افق دلخواه پیش‌بینی کند. این نگاشت می‌تواند به شکل جعبه سیاه باشد، یعنی ما از محتویات و سازوکار و پارامترهای درونی آن اطلاعی نداشته باشیم؛ اما باید ظرفیت دریافت حجم قابل‌توجهی از رویدادهای

مرتبط با گذشته را داشته باشد، با دقت پیش‌بینی کند و بتواند خود را با تغییرات احتمالی رویدادها تطبیق بدهد. خوشبختانه شبکه‌های عصبی بازگشتی همه این ویژگی‌ها را دارند.

فصل ۵ : راه حل پیشنهادی

۵-۱ مقدمه

بر اساس آنچه در فصل قبل گفته شد، ما مجموعه رویدادهای حسگرهای خانه هوشمند را به عنوان یک فرایند نقطه‌ای زمانی در نظر می‌گیریم. برای درک و پیش‌بینی رفتار این فرایند استخراج تابع شدت شرطی آن ضرورت دارد. به کارگیری روش‌های آماری رایج برای تعیین پارامترهای این تابع مستلزم آن است که ما وابستگی‌های موجود در میان رویدادها را به درستی بشناسیم؛ اما تنوع و گوناگونی رفتار انسان و تغییرات ناگهانی و ادواری این رفتار مانع رسیدن به این شناخت می‌گردد. به همین دلیل ناچاریم از مدلی استفاده کنیم که بتواند با ملاحظه تاریخچه رویدادها، تابع شدت شرطی مربوطه را تخمین بزند. این مدل باید دارای نوعی از حافظه باشد تا بتواند تأثیرات رویدادهای گذشته را در پیش‌بینی رویداد جاری لحاظ کند. شبکه‌های عصبی بازگشتی تمام این ویژگی‌ها را دارند.

درواقع ایده اصلی این است که سلسله رویدادها را به ترتیب به شبکه بازگشتی (یا انواع مدرن آن مثل LSTM یا GRU) بخورانیم؛ تا شبکه با عبور این داده‌ها وضعیت درونی خود را بهنگام کند. این وضعیت درونی که ما آن را h_t می‌نامیم، در هر لحظه نشان‌دهنده تأثیری است که شبکه از رویدادهای گذشته به خاطر سپرده است. به عبارت دیگر وقتی ما z امین رویداد، یعنی (t_j, y_j) را در اختیار شبکه می‌گذاریم، شبکه با ملاحظه این رویداد وضعیت درونی خود را از h_{j-1} به h_j بهنگام می‌کند و بدین ترتیب تاریخچه رویدادها را تا رویداد z ام در فضای وضعیت درونی خود می‌نگارد. اکنون می‌توان گفت که تابع چگالی احتمال برای زمان وقوع رویداد $j+1$ به شکل زیر خواهد بود:

$$f(t_{j+1}) = f(t_{j+1} | \mathcal{H}_j) = f((t_{j+1} | t_1, \dots, t_{j-1})) = f(t_{j+1} | h_j) = f(d_{j+1} | h_j) \quad (۱-۵)$$

در این رابطه d_{j+1} بیانگر فاصله زمانی دو رویداد متوالی z و $j+1$ است: $d_{j+1} = t_{j+1} - t_j$

اکنون ما با داشتن تابع چگالی احتمال رابطه (۵-۱) و به استناد رابطه (۴-۱۷) موفق شده‌ایم تابع شدت شرطی موردنظر را برای رویداد $j+I$ تخمین بزنیم. این تخمین بر اساس تاریخچه موجود انجام پذیرفته است و مقدار آن به هیچ پارامتر ثابتی وابسته نیست. بدیهی است که دقت این تخمین تابع ساختار شبکه بازگشتی مورد استفاده و مقادیر ابر پارامترهای آن مثل تعداد لایه‌ها و تعداد نورون‌ها می‌باشد.

۲-۵ پیش پردازش

بیشتر دادگان خانه هوشمند، رویدادهای حسگرها را به شکل سه‌تایی زیر ثبت می‌کنند:

$$e_i = \langle T, S, A \rangle \quad (۲-۵)$$

در این سه‌تایی T مبین زمان فعال شدن حسگر است که غالباً در حد ثانیه یا اجزاء آن بوده و شامل تاریخ نیز هست. S شناسه حسگر است و می‌تواند حاوی اطلاعاتی دیگر، مثل مکان یا شیء مرتبط با حسگر نیز باشد. نهایتاً A مقدار حسگر را نشان می‌دهد. از آنجاکه همه حسگرهای موردنظر ما دودویی هستند، بدون لطمه زدن به کلیت مطلب، این مقدار را می‌توان برابر یکی از دو رشته ON یا OFF در نظر گرفت. وجود اندیس i در تعریف e_i بیانگر آن است که ترتیب موجود در میان رویدادها مهم است.

اما ما رویداد حسگر e_i را به سه‌تایی زیر تبدیل می‌کنیم:

$$e_i = \langle T, SA, D \rangle \quad (۳-۵)$$

در این دو تایی T همان برچسب زمان است و SA رشته‌ای است که از چسباندن دو مقدار S و A در سه‌تایی قبلی به دست آمده است و ما آن را برچسب رویداد می‌نامیم. عنصر سوم، یعنی D نیز فاصله زمانی با رویداد قبلی را نشان می‌دهد. به عبارت دیگر:

$$D_i = T_i - T_{i-1} \quad (۴-۵)$$

طبیعی است که این مقدار در اولین رویداد برابر با صفر است. به عنوان مثال در روش ما سه تایی $T_5, >$

$\langle MI, ON \rangle$ به سه تایی $\langle T_5, MION, T_5-T_4 \rangle$ تبدیل خواهد شد.

مجموعه‌ای از رویدادهای پیاپی، مثل E روند^۱ نامیده می‌شود:

$$E = \langle e_1, e_2, e_3, \dots, e_n \rangle \quad (5-5)$$

ترتیب مندرجات یک روند بر اساس زمان فعال شدن رویدادهای آن (T) مشخص می‌شود. غالباً

می‌توان یک روند را به عنوان یک فعالیت تفسیر کرد. به عنوان مثال اگر فرد از اتاق خواب، با گذشتن از

یک راهرو به آشپزخانه برود و هر یک از مکان‌های یادشده حسگر حضور خود را به اسامی e_1, e_2 و e_3

داشته باشند روند $\langle e_1, e_2, e_3 \rangle$ رؤیت خواهد شد.

ما رفتار ناهنجار را به عنوان روندی تعریف می‌کنیم که:

- تا به حال اصلاً دیده نشده باشد
- در گذشته به ندرت دیده شده باشد
- به اندازه معناداری با روندهای اخیر فاصله داشته باشد

تعاریف فوق برای ناهنجاری، تنها به خود رویدادها و ترتیب وقوع آن‌ها مربوطاند و ناهنجاری‌های

زمانی در آن‌ها لحاظ نشده است. در واقع نوع دیگری از ناهنجاری نیز وجود دارد که در آن:

- روند ناهنجار قبلاً دیده شده اما مختصات زمانی آن با مشاهدات قبلی تفاوت زیادی دارد

مثلاً وجود فاصله‌ای در حد چند ساعت در میان رویدادهای e_2 و e_3 روند بالا باید غیرطبیعی تلقی

شود، زیرا ممکن است به این معنا باشد که فرد پیش از رسیدن به آشپزخانه دچار حادثه‌ای گردیده

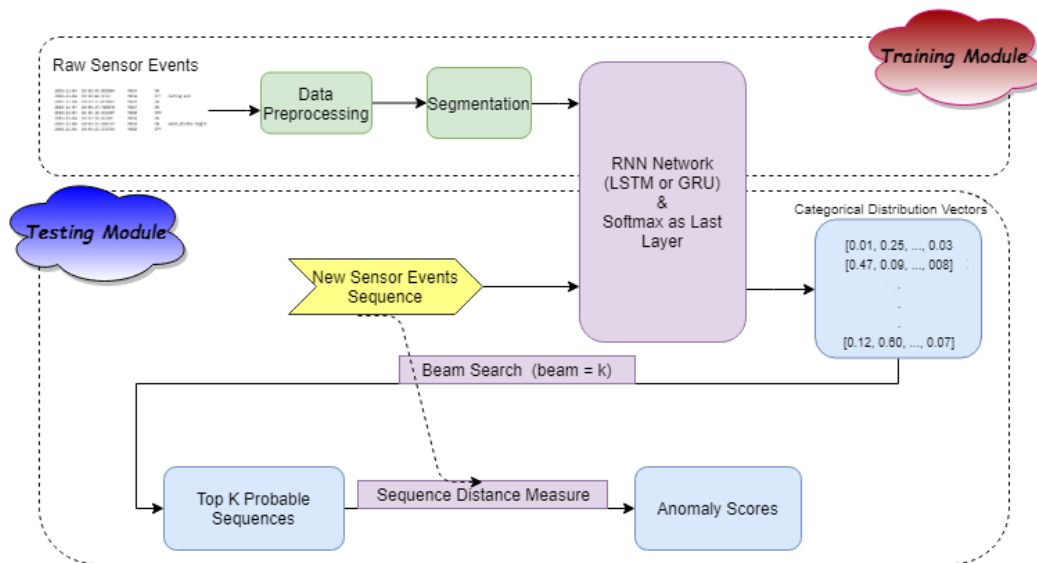
است.

^۱ Episode

۳-۵ شرح روش پیشنهادی

شمای کلی روش پیشنهادی در شکل ۱-۵ ترسیم شده است. ابتدا دادگان موردنظر پیش پردازش می‌شود. این پیش پردازش شامل مراحل زیر است:

- با چسباندن مقادیر S و A مقدار SA حاصل می‌شود.
- رشته کاراکتری SA کدگذاری می‌شود. برای این کدگذاری می‌توان از روش‌های تک‌روشن^۱ یا جاسازی کلمه^۲ استفاده کرد.
- با تفریق برچسب زمان رویداد فعلی از رویداد قبلی، عنصر D محاسبه می‌شود.
- برچسب زمان طوری تبدیل می‌شود که ادواری بودن و چرخش و بازگشت زمان در آن لحاظ شود.



شکل ۱-۵: شمای روش پیشنهادی

در توضیح آخرین پیش پردازش باید به این نکته توجه کرد که به کارگیری روش‌های معمول برای بیان ویژگی‌هایی مثل روزهای هفته یا ماه، می‌تواند رده‌بند را دچار سردرگمی کند. به‌عنوان مثال استفاده از

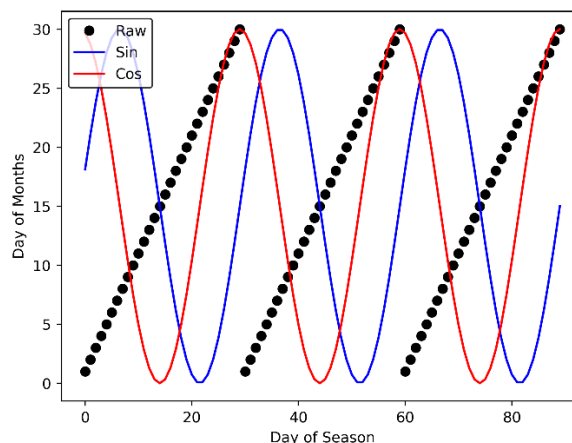
^۱ One-Hot

^۲ Word Embedding

اعداد ۱ تا ۷ برای نمایش روزهای هفته این توهم را برای رده‌بندی ایجاد می‌کند که مثلاً روزهای ۴ و ۵ هفته شباهت بیشتری دارند تا روزهای ۱ و ۵! همین موضوع در مورد روزهای ماه، البته با پیچیدگی بیشتر، صحت دارد، چراکه طول ماه‌ها باهم متفاوت است. برای حل این مشکل، اصولاً بهتر است در بیان متغیرهای گردش و ادواری^۱ از تبدیل قطبی استفاده کنیم. فرمول تبدیل یادشده، مثلاً در مورد ساعات شبانه‌روز به شرح زیر است:

$$a = h \times \left(\frac{2\pi}{24}\right) \quad (۶-۵)$$

این تبدیل ساعت را به سطح یک دایره می‌نگارد. بدین ترتیب فاصله دو ساعت بر اساس زاویه بین دو بردار ناشی از این نگاشت بیان می‌شود، یعنی $(\cos(a), \sin(a))$. شکل ۲-۵ نشان می‌دهد که روزهای یک دوره سه‌ماهه در هر یک از دو روش فوق‌الذکر به چه صورتی نمایش داده خواهند شد.



شکل ۲-۵: نمایش روزهای یک فصل با شماره روزها و تبدیل قطبی

در مرحله بعد داده‌های ورودی در قالب پنجره‌هایی به طول n و با همپوشانی m تقطیع می‌شوند تا مسئله به‌نوعی رده‌بندی باناظر تبدیل گردد. درواقع هر پنجره n تایی از رویدادها به‌عنوان X و رویداد بعدی یعنی e_{n+1} به‌عنوان Y در نظر گرفته خواهد شد. پس‌ازاین، کار ما یافتن نگاشتی خواهد بود که با دریافت یک پنجره n تایی از رویدادها، رویداد بعدی را با دقت بالایی حدس بزند:

$$F(\langle e_i, e_{i+1}, e_{i+2}, \dots, e_{i+n} \rangle) = e_{i+n+1} \quad (۷-۵)$$

^۱ Cyclic Variables

ما برای یافتن نگاشت فوق از شبکه‌های بازگشتی *LSTM* و *GRU* با ساختارهای متفاوت استفاده کرده‌ایم. جزئیات این ساختارها در بخش ۵-۶ بیان خواهد شد. لازم به ذکر است که تمامی شبکه‌های مورد استفاده، دو جزء SA_{i+n+1} و D_{i+n+1} رویداد بعدی را پیش‌بینی می‌کنند و برچسب زمان این رویداد، یعنی T_{i+n+1} از جمع زدن عناصر زمانی رویداد قبلی، یعنی $T_{i+n} + D_{i+n}$ قابل محاسبه است. پس از آموزش مدل و ایجاد نگاشت یادشده از روی داده‌های آموزشی، ما یک مدل پیشگو^۱ در اختیار داریم که با دریافت رشته ورودی $\langle e_1, e_2, \dots, e_t \rangle$ ، رویداد بعدی، یعنی e_{t+1} را پیش‌بینی می‌کند. در کاربردهای سری زمانی معمولاً فاصله این مقدار پیش‌بینی شده با مقدار واقعی به‌عنوان معیاری برای تشخیص ناهنجاری نقطه‌ای به کار می‌رود؛ اما این معیار در مورد مسئله ما قابل استفاده نیست، زیرا ما به دنبال یافتن ناهنجاری‌های تجمعی و زمینه‌دار هستیم. به‌عبارت دیگر، ما به دنبال یافتن سلسله‌ای از رویدادها هستیم که وقوع آن‌ها در یک‌زمان خاص به‌عنوان ناهنجاری تلقی می‌شود. به همین دلیل پس از انجام پیش‌بینی یادشده، e_{t+1} را به ابتدای رشته ورودی ملحق نموده و اولین رویداد این رشته (e_t) را از آن حذف می‌کنیم. اکنون ما یک رشته t تایی جدید به شکل $\langle e_2, e_3, \dots, e_{t+1} \rangle$ داریم که می‌تواند برای پیش‌بینی رویداد دو گام بعد، یعنی e_{t+2} مورد استفاده قرار گیرد. با تکرار این فرایند، ما سلسله l رویداد محتمل بعدی را پیش‌بینی می‌کنیم.

این روش که به نام پیش‌بینی تکراری یا بازگشتی^۲ شهرت دارد، در مطالعات و ادبیات رشته‌ها و سری‌های زمانی شیوه‌ای رایج به شمار می‌آید [99]. نقطه ضعف این روش آن است که علیرغم دقت بالا در گام‌های نخست، با افزایش تعداد قدم‌ها، به دلیل تجمع خطا، دقت آن با شدت رو به کاهش می‌گذارد. به همین دلیل روش یادشده تنها برای پیش‌بینی افق محدودی از رویدادهای پیش‌رو مناسب است [99]. نکته کار ما اینجاست که به‌جای استفاده از روش حریصانه^۳ معمول دریافتن گام

^۱ Predictive Model

^۲ Iterative or Recursive Prediction

^۳ Greedy Search

بعد، از روش جستجوی پرتو محلی^۱ استفاده می‌کنیم. توضیح آنکه، مدل پیشنهادی در هر گام یک عدد واحد را به‌عنوان مدت‌زمان یا D پیش‌بینی می‌کند، اما اوضاع در پیش‌بینی برچسب رویداد یا SA به شکل دیگری است. ما برای پیش‌بینی عنصر مذکور، در آخرین بخش شبکه‌های خود، یک‌لایه $softmax$ را بکار می‌گیریم. خروجی این لایه، برداری است که طول آن برابر با تعداد برچسب‌های موجود است و مقدار منتسب به هر برچسب برابر با احتمال وقوع آن می‌باشد. ما در هر گام m برچسب صاحب بالاترین احتمال را انتخاب می‌کنیم و با الحاق D به این m برچسب و محاسبه T های مربوطه، برای رویداد بعدی m پیشنهاد ایجاد می‌کنیم. سپس در گام بعد هر یک از این m رویداد را به انتهای رشته ورودی می‌چسبانیم و برای این m رشته، m رویداد محتمل بعدی را پیش‌بینی می‌نماییم. با ادامه این فرایند در l مرحله، ما در انتها دارای یک درخت پر و متوازن به عمق l هستیم که هر گره آن صاحب m فرزند است. پیمایش کامل همه مسیرهای این درخت از ریشه تا برگ‌ها، تعداد m^l روند محتمل را با احتمال مربوط به هر گره در اختیار ما می‌گذارد. اکنون ما با استفاده از روش جستجوی پرتو محلی، تعدادی از سلسله رویدادهای محتمل بعدی را از میان این مسیرها استخراج می‌کنیم. تعداد این سلسله‌ها، مقدار دلخواه k است که اندازه پرتو نامیده می‌شود.

جستجوی پرتو محلی روشی مکاشفه‌ای^۲ است که درخت حالات پیش رو را پیمایش می‌کند و تنها رشته‌هایی را باز می‌گرداند که احتمال وقوع آن‌ها بیشتر از بقیه باشد [100]. این روش بر اساس روش جستجوی اول سطح^۳ تدوین شده است، به این معنی که در مسیر جستجو، اول گره‌های همسایه و هم‌سطح را ملاقات می‌کند. تفاوت در اینجاست که روش پرتو محلی در هر مرحله پس از گسترش گره‌های سطح بعد، این گره‌ها را بر اساس مقدار یک تابع هدف f مرتب می‌کند، سپس تعدادی از آن‌ها را برای ملاقات انتخاب کرده و بقیه گره‌ها را حذف می‌نماید. این تعداد که اندازه پرتو^۴ نام دارد

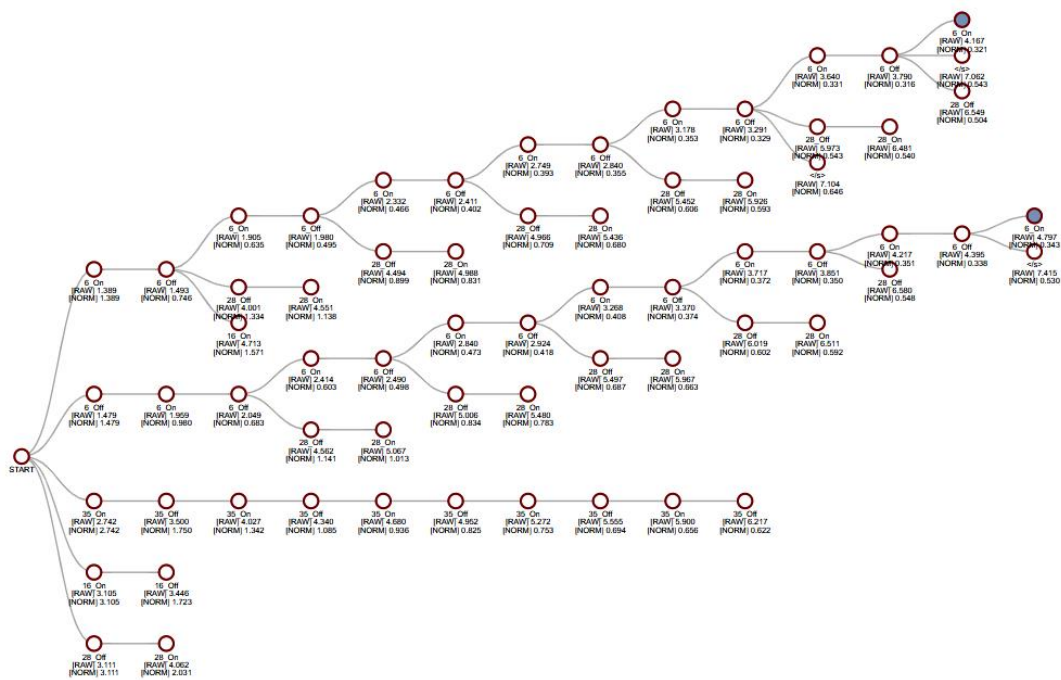
^۱ Local Beam Search

^۲ Heuristic Method

^۳ Breath-First Search

^۴ Beam Size

معمولاً بر اساس محدودیت‌های زمانی و محاسباتی - خصوصاً حافظه - مشخص می‌گردد. طبیعتاً مقادیر بزرگ‌تر زمان و حافظه موردنیاز برای اجرای برنامه را افزایش خواهند داد. با حرکت اندازه پرتو به سمت بی‌نهایت، روش یادشده به الگوریتم اول سطح گرایش پیدا می‌کند. روش جستجوی پرتو محلی در متن کاوی و پردازش زبان طبیعی کاربرد زیادی دارد [101]. شکل ۳-۵ شیوه عملکرد روش جستجوی پرتو محلی را برای یافتن رشته‌ای از بهترین حسگرها نشان می‌دهد. در این شکل اندازه پرتو برابر ۵ می‌باشد.



شکل ۳-۵ شیوه عملکرد جستجوی پرتو محلی

اکنون ما k رشته به طول l داریم که بر اساس داده‌های موجود وقوع هر یک از آن‌ها محتمل است. فاصله رشته l تایی رویدادهای واقعی با این مجموعه می‌تواند به‌عنوان خطای بازسازی تلقی شده و برای ارزیابی رشته l تایی یادشده و انتساب رتبه ناهنجاری به آن مورد استفاده قرار گیرد. این رشته‌ها در واقع حاوی مجموعه‌ای از دوتایی‌های مرتب هستند که یک عنصر آن - یعنی برچسب رویداد - گسسته و عنصر دیگر - یعنی فاصله زمانی تا رویداد بعدی - از نوع پیوسته است. به همین

دلیل هیچیک از روش‌هایی که برای محاسبه فاصله رشته‌های گسسته یا سری‌های زمانی مورد استفاده قرار می‌گیرند، در مورد آن‌ها کارایی ندارند.

برای محاسبه این فاصله ما از روش مرجع [102] استفاده می‌کنیم. این مرجع برای اینکه تمام وجوه ممکن در تفاوت‌ها و شباهت‌های دوتایی‌های یادشده را در نظر بگیرد، ۴ تابع شباهت مختلف را تعریف کرده است. سپس مجموع وزن‌دار این توابع به‌عنوان معیار نهایی شباهت مورد استفاده قرار گرفته است. این توابع در روابط (۸-۵) تا (۱۲-۵) بیان شده‌اند. در این روابط A و B اسامی دو روند مورد مقایسه و n بیانگر طول آن‌هاست. همچنین LCS نام طولانی‌ترین زیررشته مشترک^۱ این دو روند است که طولی برابر k دارد. نمادهای $||$ و $\{\{\}\}$ نیز به ترتیب برای نمایش طول یک ساختار (مثلاً رشته) و چندمجموعه^۲ مورد استفاده قرار گرفته است. مطابق تعریف، چند مجموعه، مجموعه‌ای است که تکرار اعضا در آن مجاز است و چند عضو یکسان به یک عضو کاهش پیدا نمی‌کنند. سرانجام اینکه، معیار شباهت نهایی با S نمایش داده شده است:

$$S(A, B) = \sum_{i=1}^4 w_i S_i \quad (۸-۵)$$

$$S_1 = \frac{|LCS_{A,B}|}{k} \quad (۹-۵)$$

$$S_2 = \frac{|\{\{x \mid x \notin LCS, x \in A \text{ or } x \in B\}\}|}{n} \quad (۱۰-۵)$$

$$S_3 = 1 - \frac{S_t}{t_c} \quad (۱۱-۵)$$

$$S_4 = \frac{\sum_{j=1}^{|LCS|} \frac{\text{Min}(d_j, d'_j)}{\text{Max}(d_j, d'_j)}}{k} \quad (۱۲-۵)$$

در رابطه (۱۱-۵) تابع S_t به شکل زیر تعریف می‌شود:

$$S_t = \frac{\sum_{j=1}^{|LCS|} |t_j - t'_j|}{k} \quad (۱۳-۵)$$

شایان ذکر است که در میان معیارهای فوق، S_3 (رابطه (۱۱-۵)) و S_4 (رابطه (۱۲-۵)) شامل مفهوم زمان هستند. در این روابط، t_j و t'_j و d_j و d'_j به ترتیب بیانگر برچسب زمان و مدت دو رویداد معادل

^۱ Longest Common Subsequence

^۲ MultiSet

و مترادف از روندهای A و B هستند که در طولانی‌ترین زیررشته مشترک یا همان LCS در مقابل هم قرار می‌گیرند. سرانجام، t_C ثابت زمانی برحسب ثانیه است که به طول دوره زمانی موردبررسی بستگی دارد. مثلاً برای مطالعه یک بازه ۲۴ ساعته، این ثابت برابر با ۸۶۴۰۰ قرار می‌گیرد. برای محاسبه تابع شباهت نهایی یا k ، مقدار هر یک از توابع فرعی k_i نرمال‌سازی می‌شود تا در فاصله ۰ تا ۱ قرار بگیرد. سپس حاصل جمع وزن‌دار این مقادیر به‌عنوان معیار نهایی بکار گرفته خواهد شد. با نام‌گذاری این تابع فاصله به F ، تفاوت مجموعه k تایی رشته‌های پیش‌بینی‌شده با رشته واقعی به شکل زیر خواهد بود:

$$d = \frac{1}{k} \sum_{i=1}^k f(E, S_i) \quad (۱۴-۵)$$

در این رابطه، E رشته واقعی و S_i برابر i امین رشته پیش‌بینی‌شده است. با ارزیابی مدل از طریق مجموعه آزمایشی برداری از خطا حاصل می‌شود که به روش فوق‌الذکر به‌دست‌آمده و برای محاسبه نرخ ناهنجاری در هر پنجره ورودی مورداستفاده قرار خواهد گرفت. این بردار بر اساس روش مذکور در مرجع [103] به‌عنوان یک توزیع گاوسی $N(\mu, \Sigma)$ مدل می‌شود. بدین ترتیب احتمال مشاهده خطای p^t در نقطه e^t با محاسبه مقدار N در نقطه e^t به دست می‌آید. برای محاسبه مقادیر μ و Σ نیز از روش برآورد درست‌نمایی بیشینه یا MLE استفاده می‌کنیم. اکنون هر پنجره ورودی که مقدار p^t آن از حد آستانه t بزرگ‌تر باشد به‌عنوان ناهنجاری در نظر گرفته خواهد شد. مقدار این حد آستانه را می‌توان به‌صورت ایستا و بر اساس مشاهدات موجود تعیین کرد یا از طریق بیشینه کردن معیار F_β آن را به‌صورت پویا مشخص نمود.

۴-۵ دادگان

موضوع خانه‌های هوشمند در بسیاری از دانشگاه‌های دنیا به‌عنوان یک موضوع تحقیقاتی اصلی موردتوجه قرار گرفته است. خوشبختانه برخی از این دانشگاه‌ها با تشکیل آزمایشگاه و تأسیس خانه‌های

هوشمند نمونه، دادگان موردنیاز خود را تولید کرده‌اند. البته تعداد زیادی دادگان به صورت رایگان در دسترس محققین و علاقه‌مندان قرار دارد [104]:

- Aware Home at Georgia Tech
<http://awarehome.imtc.gatech.edu/>
- Gator Tech Smart House
<http://www.icta.ufl.edu/gt.htm>
- iSpace/iDorm at University of Essex
<http://cswww.essex.ac.uk/iieg/idorm2/index.htm>
- MARC Smart House at University of Virginia: Medical Automation Research Center
<https://smarthouse.med.virginia.edu/>
- PlaceLab: an MIT and TIAX initiative for studying behavior in a residential condominium
http://architecture.mit.edu/house_n/placelab.html
- Smart Home at Duke University: A live-in laboratory for students
<http://www.smarthome.duke.edu/>
- Tiger Place at University of Missouri: Instrumented assistive living
<http://www.tigerplace.net/>
- The Multicom intelligent flat : an user-oriented smart home platform
<http://domus.imag.fr>

ضمناً تعداد زیادی نرم‌افزار شبیه‌ساز خانه هوشمند وجود دارد که کاربر می‌تواند از طریق آن‌ها خانه موردنظر خود را به شکل دو یا سه‌بعدی طراحی کند، حسگرهای موردنیاز را به نقاط مختلف اضافه کند و سناریوهای مختلف را طراحی و پیاده‌سازی نماید. نتیجه اجرای هر سناریو، جریان مقادیر خروجی حسگرهای مختلفی است که در جریان آن، تحریک و فعال می‌شوند. حسن این نرم‌افزارها این است که در غیاب یک‌خانه هوشمند واقعی می‌توان از آن‌ها به‌عنوان بستر آزمایش^۱ استفاده کرد. SIMACT و OpenSHS دو نمونه از این نرم‌افزارها هستند که به شکل متن‌باز تولید و منتشر شده‌اند [105].

دادگان مورد استفاده در پژوهش حاضر، از مجموعه دادگان‌های «مرکز مطالعات پیشرفته سامانه‌های تطبیقی» (CASAS) انتخاب شده است [106]. این مرکز در دانشگاه ایالتی واشنگتن قرار دارد و یکی از معتبرترین مراکز مطالعات خانه‌های هوشمند به شمار می‌رود. محتویات دادگان

^۱ Testbed

مذکور در یک دوره ۲۲۰ روزه - از ۴ نوامبر ۲۰۱۰ تا ۱۱ ژوئن ۲۰۱۱ - با پایش شبانه‌روزی زندگی یک بانوی کهن‌سال جمع‌آوری شده‌اند. برای گردآوری این دادگان که aruba نام دارد، ۴۰ حسگر حرکت، درب و درجه حرارت در منزل مسکونی این بانو نصب و راه‌اندازی گردیده‌اند. دادگان arub حاوی ۱۱ نوع از افعال روزمره و ۱،۷۱۹،۵۵۳ رویداد حسگر است. چون موضوع کار ما حسگرهای دودویی بوده است، مجموعه رویدادهای حرارت‌سنج را حذف کرده و تعداد این سطرها را به ۱،۶۰۲،۸۱۸ عدد کاهش داده‌ایم. نمونه‌ای از محتویات این دادگان در شکل ۴-۵ نمایش داده شده است.

دادگان aruba اصولاً برای بازشناسی و رده‌بندی افعال روزانه تهیه شده است و هیچ ناهنجاری مشخص و برجسته خورده‌ای در آن وجود ندارد. به همین دلیل ما داده‌های مندرج در آن را به‌عنوان داده‌های کاملاً هنجار در نظر گرفتیم و سپس بر اساس آن‌ها ۱۰۰ نمونه ناهنجاری را به‌صورت دستی تولید نمودیم. این رکوردها به شکل پنجره هستند و دارای یکی از انواع ناهنجاری مذکور در بخش ۵-۱ می‌باشند.

2010-11-04	10:02:46.866804	M014	ON	
2010-11-04	10:02:48.9722	M014	OFF	Eating end
2010-11-04	10:03:17.076067	M020	ON	
2010-11-04	10:03:17.462048	M014	ON	
2010-11-04	10:03:18.631607	M020	OFF	
2010-11-04	10:03:19.85609	M020	ON	
2010-11-04	10:03:21.966597	M018	ON	Wash_Dishes begin
2010-11-04	10:03:22.532394	M020	OFF	

شکل ۴-۵: بخشی از دادگان aruba

لازم به ذکر است که متأسفانه هیچ دادگان عمومی و استاندارد برای کشف ناهنجاری در خانه‌های هوشمند وجود ندارد. به همین دلیل افزودن ناهنجاری‌های دست‌ساز به دادگان استاندارد که در اصل برای شناسایی افعال ایجاد شده، روش رایجی است که بسیاری از مراجع معتبر، از جمله [107]، [71] و [108] آن را به کار گرفته‌اند.

۵-۴-۱ شیوه ایجاد و درج ناهنجاری مصنوعی

ما برای ایجاد افعال و رشته‌های ناهنجار سه گروه از افعال ناهنجار را در نظر گرفته‌ایم :

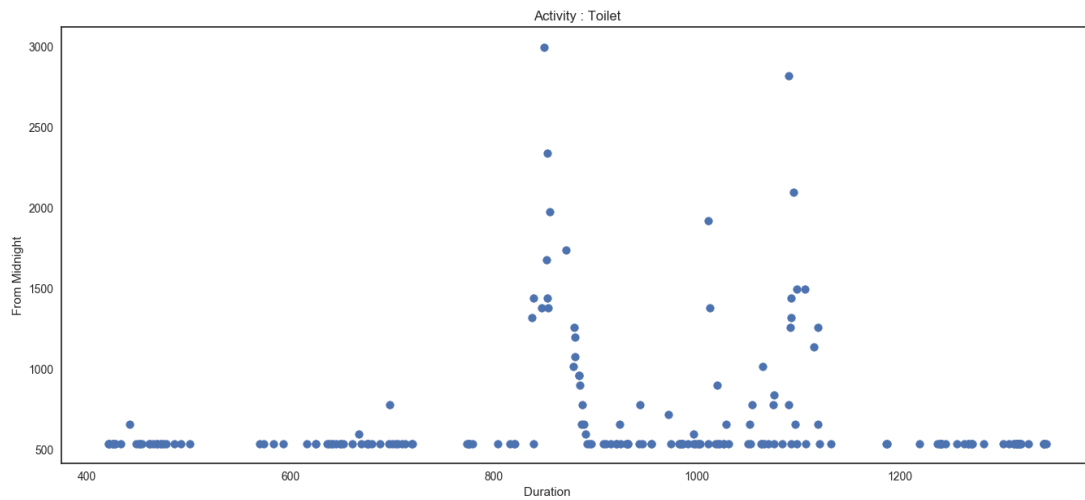
تکرار افعال : تکرار افعال روزمره یکی از انواع رایج ناهنجاری در خانه‌های هوشمند به شمار می‌آید. این ناهنجاری ممکن است به دلیل فراموشی روی دهد. یعنی فرد ساکن در خانه به خاطر نیاورد که یک فعالیت روزمره خاص، مثل مسواک‌زدن یا خوردن دارو را انجام داده است یا خیر و در نتیجه این فعالیت را در طول روز چند بار تکرار کند. این تکرار در مواردی مثل خوردن دارو می‌تواند فرد را دچار مخاطره نماید. بعلاوه افزایش تواتر موارد مذکور باید به‌عنوان اولین علائم بروز برخی از بیماری‌های ذهنی از جمله آلزایمر جدی گرفته شود. افزون بر آنچه گفته شد، تکرار آگاهانه بعضی از افعال، مثل مراجعه مکرر فرد به سرویس بهداشتی ممکن است نشان از بروز یک بیماری جسمی داشته باشد.

ما برای شبیه‌سازی این نوع ناهنجاری مجموعه‌ای از حسگرهای یکسان را در نواحی تصادفی از دادگان نرمال درج کرده‌ایم. در نتیجه رشته‌ای هنجار و طبیعی مثل S به رشته ناهنجار و غیرمعمول S' تبدیل شده است :

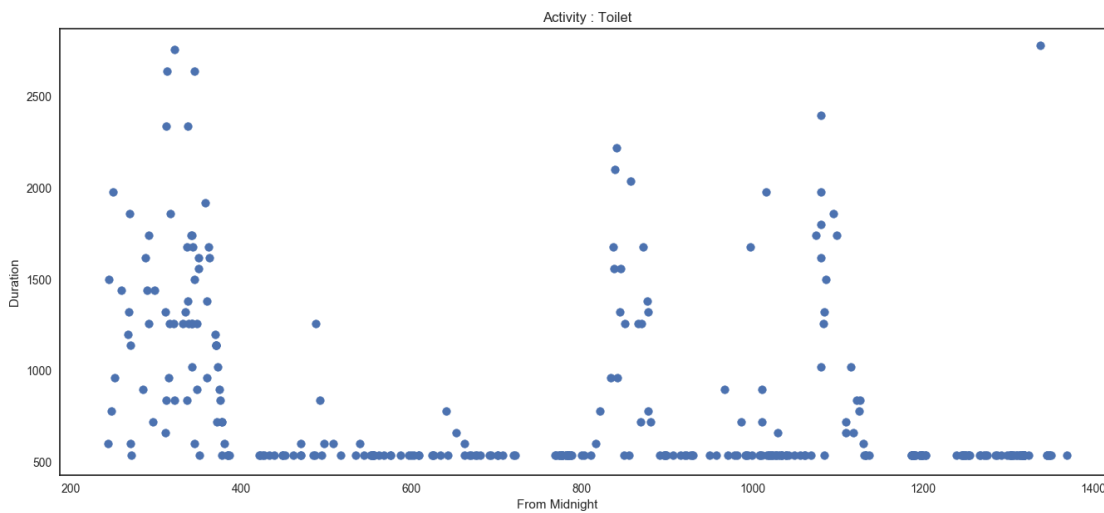
$$S = a1, a2, a3, b1, b2, b3, c1, c2, c3, c4, d1, d2, e1, e2$$

$$S' = a1, a2, a3, d1, d2, b1, b2, b3, c1, c2, c3, d1, d2, e1, e2, d1, d2$$

شکل ۵-۶ نتیجه اعمال این تغییر را بر روی یکی از حسگرها - حسگر درب دستشویی - نشان می‌دهد. چنانکه از شکل ۵-۵ برمی‌آید دفعات فعال شدن این حسگر در دادگان هنجار کمتر بوده است. همچنین الگوی زمان فعال شدن و مدت فعال ماندن این حسگر در دادگان ناهنجار دچار پراکندگی شده است. دلیل این پراکندگی آن است که ما فواصل زمانی بین مجموعه حسگرهای درج‌شده را به شکلی تصادفی مقداردهی کرده‌ایم.



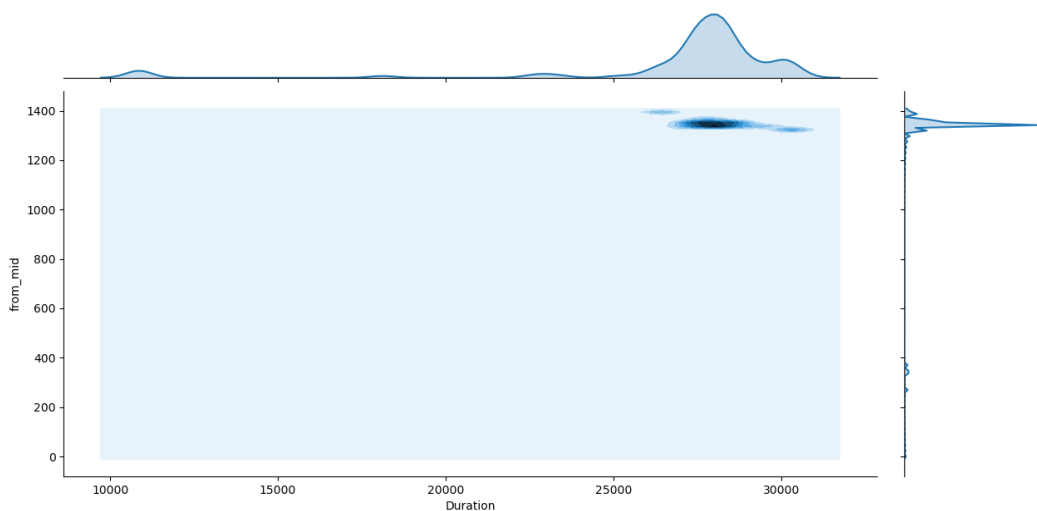
شکل ۵-۵ الگوی فعال شدن حسگر درب سرویس بهداشتی در دادگان هنجار



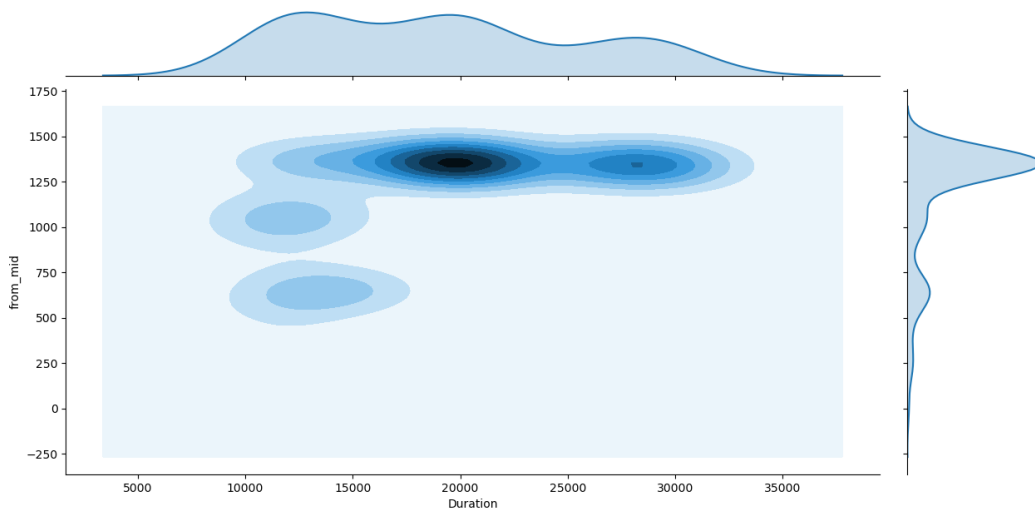
شکل ۶-۵ الگوی فعال شدن حسگر درب سرویس بهداشتی در دادگان ناهنجار

آشفتگی در ساعت شروع و مدت افعال : یک گروه مهم از ناهنجاری‌های موجود در افعال، مربوط به ساعت شروع و مدت آنهاست. مثلاً تماشای تلویزیون در روز و ساعات اولیه شب عادی است، اما همین عمل اگر از ساعت ۳ نیمه‌شب آغاز شود به‌عنوان یک فعل ناهنجار تلقی خواهد شد. مدت افعال هم به‌اندازه زمان شروع آنها اهمیت دارد. به‌عنوان مثال کاهش معنی‌دار ساعات خروج از منزل و افزایش مدت زمان استراحت را می‌توان به‌منزله نشانه‌های آغاز افسردگی در نظر گرفت.

ما برای ایجاد این نوع ناهنجاری زمان شروع و مدت فعال شدن برخی از حسگرها را طوری تغییر داده‌ایم که توزیع و پراکندگی مقادیر آن‌ها با داده‌های هنجار متفاوت باشد. شکل ۷-۵ و شکل ۸-۵ به ترتیب نمودار پراکندگی زمان شروع و مدت فعالیت حسگر تختخواب را در دادگان هنجار و ناهنجار نشان می‌دهند. محتویات شکل ۷-۵ بیانگر وضع خواب یک فرد طبیعی است که معمولاً در ساعات آخر شب آغاز می‌شود و بین ۷ تا ۸ ساعت ادامه پیدا می‌کند. اما در شکل ۸-۵ الگوی فعالیت حسگر دچار پریشانی شده است. به عبارت دیگر فرد گاه‌وبیگاه دچار چرت و خواب‌زدگی می‌شود، درعین حال مدت خواب شبانه‌ای او نیز کوتاه‌تر شده است.



شکل ۷-۵ الگوی زمان فعال شدن و مدت فعال ماندن حسگر تختخواب – دادگان هنجار



شکل ۵-۸ الگوی زمان فعال شدن و مدت فعال ماندن حسگر تختخواب – دادگان ناهنجار

حذف افعال : نوع سوم ناهنجاری حذف برخی از افعال ضروری است. مثلاً نبستن درب ورودی پس از خروج از منزل یا خاموش نکردن تلویزیون پیش از رفتن به رختخواب در زمره این قبیل ناهنجاری‌ها قرار می‌گیرند. ما برای ایجاد نوع اخیر ناهنجاری، به‌طور تصادفی تعدادی از حسگرها را از برخی نواحی دادگان هنجار حذف کرده‌ایم. نتیجه این کار چیزی شبیه شکل ۵-۵ و شکل ۶-۵ را ایجاد می‌کند، با این تفاوت که تعداد دفعات فعال شدن حسگر در نسخه ناهنجار کمتر است.

۵-۵ معیار کار آبی

مقوله کشف ناهنجاری، علیرغم شباهت‌های کلی که با رده‌بندی 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 رده یا کلاس هنجار و ناهنجار تقسیم می‌شوند، اما داده‌های ناهنجار معمولاً نادر و کم تعداد هستند و به همین دلیل اندازه طبقات یادشده، متوازن نیست. این موضوع باعث می‌شود تا

معیارهای رایج در ارزیابی روش‌های رده‌بندی، مثل دقت، صحت و جامعیت کار آبی خود را از دست بدهند. به‌عنوان مثال دادگانی را در نظر بگیرید که حاوی ۱۰٪ ناهنجاری باشد. به‌کارگیری یک رده‌بند مبتذل^۱ که همیشه همه‌چیز را به‌عنوان هنجار رده‌بندی می‌کند، بر روی این دادگان با ۹۰٪ دقت همراه خواهد بود!

مشکل دیگری که روش‌های مبتنی بر خطای بازسازی - مثل کدکننده خودکار 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 کشف‌شده ناهنجار هستند، اما بخش بزرگی از نقاط ناهنجار، به‌عنوان 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 به‌عنوان ناهنجاری برچسب می‌خورند.

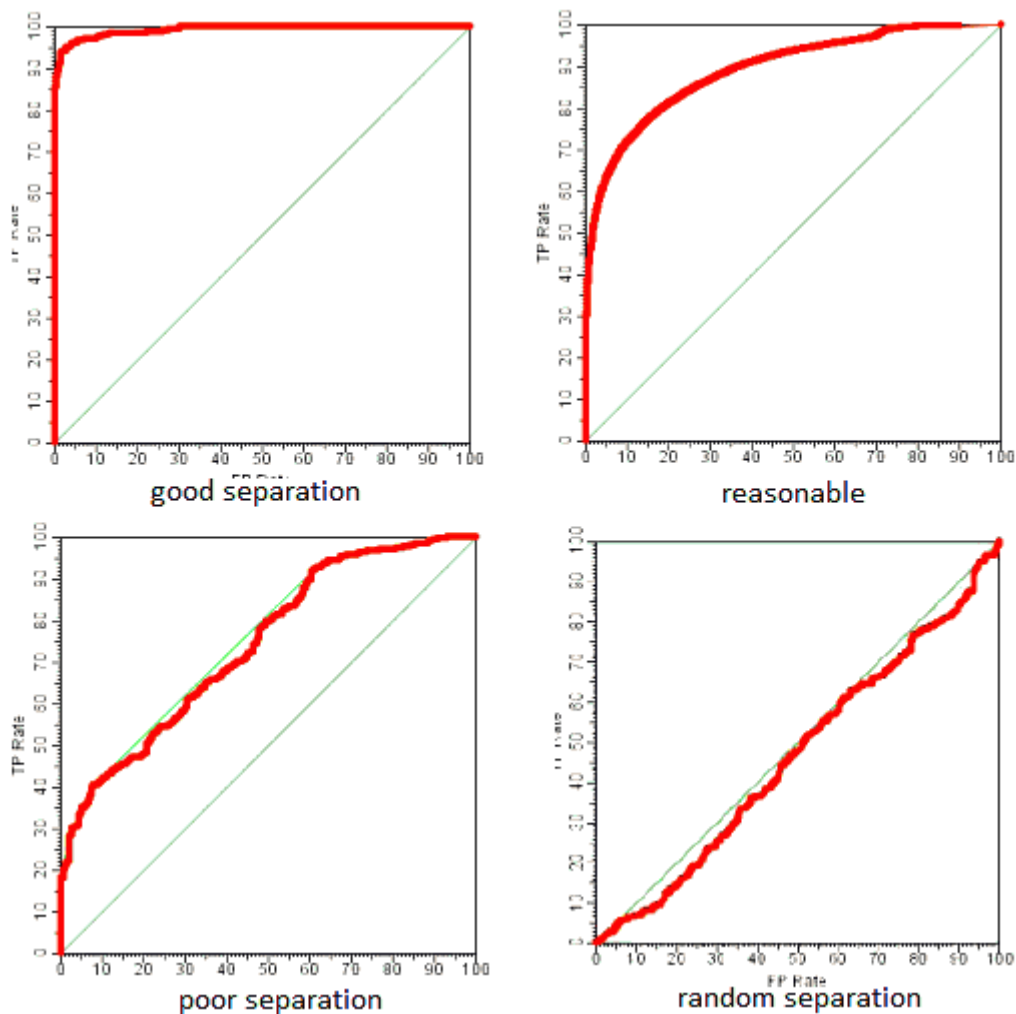
مجموعه مطالب مذکور نشان می‌دهد که معیارهای رایج سنجش، مثل دقت، صحت، جامعیت و حتی معیار $F1$ که معمولاً برای تحلیل روش‌های رده‌بندی مورد استفاده قرار می‌گیرند، در حوزه کشف ناهنجاری از انعطاف و کار آبی لازم برخوردار نیستند. به همین دلیل اغلب تحقیقاتی که در حوزه یادشده صورت پذیرفته است، از معیار دیگری به نام منحنی مشخصه عملکرد سیستم یا منحنی عملیاتی دریافت‌کننده یا ROC ^۲ استفاده کرده‌اند. [8] و [109]

در روش‌های بی‌ناظر یا نیمه‌نظارتی کشف ناهنجاری، برخلاف الگوریتم‌های رده‌بندی 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 می‌باشد.

^۱ Trivial

^۲ Receiver Operator Curve

برای ترسیم منحنی ROC ، نقاطی را در فاصله کوچک‌ترین تا بزرگ‌ترین رتبه (معمولاً ۰ تا ۱) انتخاب می‌کنند و پس از انتصاب هر نقطه به‌عنوان حد آستانه، نسبت اقلام ناهنجاری که به‌درستی به‌عنوان ناهنجاری در نظر گرفته شده‌اند یا TPR ^۱ و نسبت اقلام هنجاری که به‌غلط به‌عنوان ناهنجاری رده‌بندی گردیده‌اند یا FPR ^۲ 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 در نهایت نمودار ROC به دست بیاید. شکل ۵-۹ نمونه‌هایی از نمودار ROC را نشان می‌دهد.



^۱ True Positive Rate

^۲ False Positive Rate

شکل ۵-۹: چهار نمونه از نمودارهای ROC [110]

یکی از بزرگ‌ترین امتیازات نمودار *ROC* این است که نمودار یادشده از نحوه توزیع رده‌ها مستقل است و با تغییر نسبت نمونه‌های هنجار و ناهنجار تغییر شکل نخواهد داد [111]. این ویژگی *ROC* را به ابزاری مناسب برای تحلیل الگوریتم‌های کشف ناهنجاری تبدیل کرده است.

سطح زیر منحنی نمودار *ROC* که *AUC*^۱ نامیده می‌شود، مندرجات گرافیکی این نمودار را در یک عدد خلاصه می‌کند. این عدد در فاصله ۰ تا ۱ قرار دارد. مرجع [109] عدد یادشده 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 شانسی (مثلاً با پرتاب سکه) مشخص می‌کند نزدیک یا برابر ۰,۵ است و *AUC* الگوریتمی که تمامی نقاط سازگار و ناساز را به‌طور دقیق و صریح بازشناسی می‌نماید برابر ۱ می‌باشد.

نمودار *ROC* علیرغم محسنات خود، معایبی نیز دارد. از جمله اینکه تنها ترتیب و چیدمان رتبه‌های ناهنجاری برای آن اهمیت دارد و به‌اندازه این رتبه‌ها یا میزان تفاوت‌های نسبی و عددی آن‌ها توجهی نمی‌کند. باین 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 تبدیل شده است. ما با الهام از مراجع [112] و [113] و به‌ویژه مرجع

^۱ Area Under Curve

[8] برای اندازه‌گیری و مقایسه قابلیت و کار آیی روش پیشنهادی خود، از ROC و AUC استفاده خواهیم کرد.

۵-۶ پیاده‌سازی

ما برای ارزیابی روش پیشنهادی مجموعه‌ای از آزمایش‌ها را انجام داده‌ایم. این آزمایش‌ها، در معیارهایی مثل طول پنجره‌های ورودی، میزان همپوشانی این پنجره‌ها، تعداد بردارهای احتمال و اندازه پرتو (تعداد رشته‌های خروجی جستجوی پرتو محلی) باهم تفاوت دارند. خلاصه مشخصات این آزمایش‌ها در جدول ۵-۱ فهرست شده است.

جدول ۵-۱: مشخصات تجربه‌ها

نام	طول پنجره‌های ورودی	درصد همپوشانی	تعداد بردارهای خروجی	اندازه پرتو
$P1$	۴	۰	۲۰	۶
$P2$	۴	۲۵	۲۰	۶
$P3$	۸	۵۰	۲۰	۸

همچنین شبکه‌های مختلفی را به‌عنوان مدل پیش‌بینی و تشخیص ناهنجاری ایجاد کردیم که در نوع عناصر ($LSTM$ یا GRU)، تعداد عناصر، تعداد لایه‌ها، مختصات لایه‌های کاملاً متصل^۱ و نرخ حذف تصادفی^۲ باهم متفاوت هستند. حذف تصادفی یک روش تنظیم مدل برای کاهش بیش‌برازش در شبکه‌های عصبی است. در این روش، درصدی از وزن‌های یک شبکه عصبی هر بار به‌صورت تصادفی از روند یادگیری حذف می‌شود. خلاصه مشخصات این شبکه‌ها نیز در جدول ۵-۲ فهرست شده است.

^۱ Dense

^۲ Dropout

جدول ۵-۲: مشخصات شبکه‌های مورد استفاده

نام	نوع	لایه‌های بازگشتی	لایه‌های متصل	Dropout
N1	LSTM	۶۴ - ۳۲	۵۱۲-۱۲۸	۰,۴
N2	GRU	۶۴ - ۳۲	۵۱۲-۱۲۸	۰,۴
N3	GRU	۶۴-۳۲-۳۲	۲۵۶-۱۲۸	۰/۲

صرف‌نظر از این تفاوت‌ها، سایر پارامترهای یادگیری و آزمایش در تجربیات یادشده یکسان است. چنانکه گفته شد، در هر اجرا داده‌های ورودی به دو بخش ۷۰ و ۳۰ درصدی تقسیم شده‌اند. بخش اول مجموعه آموزشی را تشکیل داده و بخش دوم، پس از تزریق ناهنجاری‌های دست‌ساز، به‌عنوان مجموعه آزمایشی مورد استفاده قرار گرفته است. داده‌ها در بسته‌های ۱۰۰ تایی تحویل شبکه‌ها شده‌اند ($Batch\ size = 100$) و هر شبکه ۵۰ بار تحت آموزش قرار گرفته است ($Epochs = 50$). نهایتاً در همه موارد نرخ یادگیری برابر ۰/۰۰۲۵ بوده و از تابع آنتروپی متقابل به‌عنوان تابع خطای شبکه استفاده شده است.

علاوه بر ساختارهای یادشده، ما به‌منظور مقایسه روش ماشین بردار پشتیبان تک کلاسه یا $OC-SVM$ را نیز برای کشف ناهنجاری‌ها مورد استفاده قرار داده‌ایم. این روش در منابع متعددی مثل مراجع [9] و [18] برای کشف ناهنجاری در خانه‌های هوشمند به‌کاررفته است.

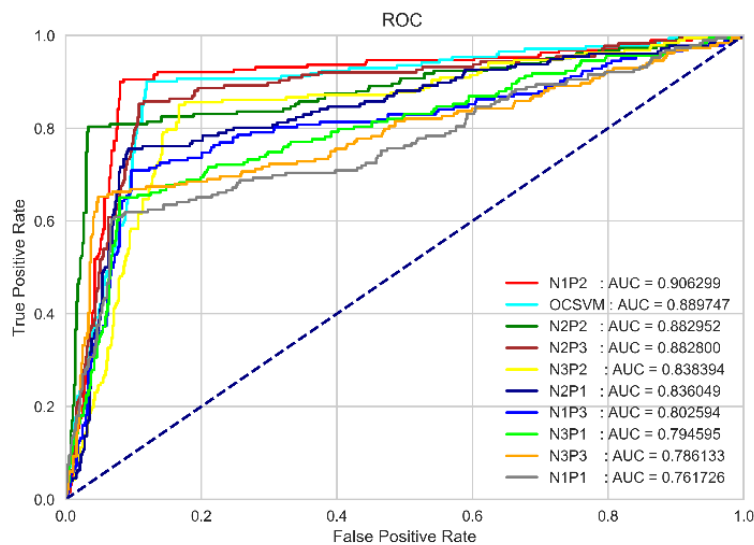
برای مقایسه این روش‌ها از معیارهای منحنی مشخصه عملکرد سیستم (ROC) و سطح زیر نمودار این منحنی (AUC) استفاده شده است. جدول ۵-۳ مقادیر AUC را برای تجربیات متفاوت نشان می‌دهد. شکل ۵-۱۰ نیز نمودار ROC را برای همین تجربیات به نمایش می‌گذارد.

جدول ۵-۳: مقادیر AUC برای تجربه‌های مختلف

مقدار AUC	نام روش
۰/۹۰۶۳	$NIP2$
۰/۸۸۹۷	$OCSVM$
۰/۸۸۳۰	$N2P2$
۰/۸۸۲۸	$N2P3$

.۸۳۸۴	<i>N3P2</i>
.۸۳۶۰	<i>N2P1</i>
/۸۰۲۶	<i>N1P3</i>
.۷۹۴۶	<i>N3P1</i>
.۷۸۶۱	<i>N3P3</i>
.۷۶۱۷	<i>N1P1</i>

تمام برنامه‌های موردنیاز به زبان *Python* و با استفاده از کتابخانه‌های *Tensorflow* و *Keras* نوشته شده‌اند. این برنامه‌ها بر روی سیستم‌عامل *Linux Mint* اجرا گردیده‌اند و برای استفاده از توان کارت گرافیکی، به امکانات نسخه ۸ کتابخانه *CUDA* متکی بوده‌اند. کامپیوتر مورد استفاده نیز به یک کارت گرافیکی *Nvidia GeForce 920 MX* با ۲۵۶ هسته کُودا، یک پردازنده چهار هسته‌ای *Intel Core i7 7500U 2.7 GHz* و ۸ گیگابایت حافظه اصلی مجهز بوده است.



شکل ۵-۱۰: نمودار ROC برای تجربیات متفاوت

۵-۶-۱ بحث و بررسی

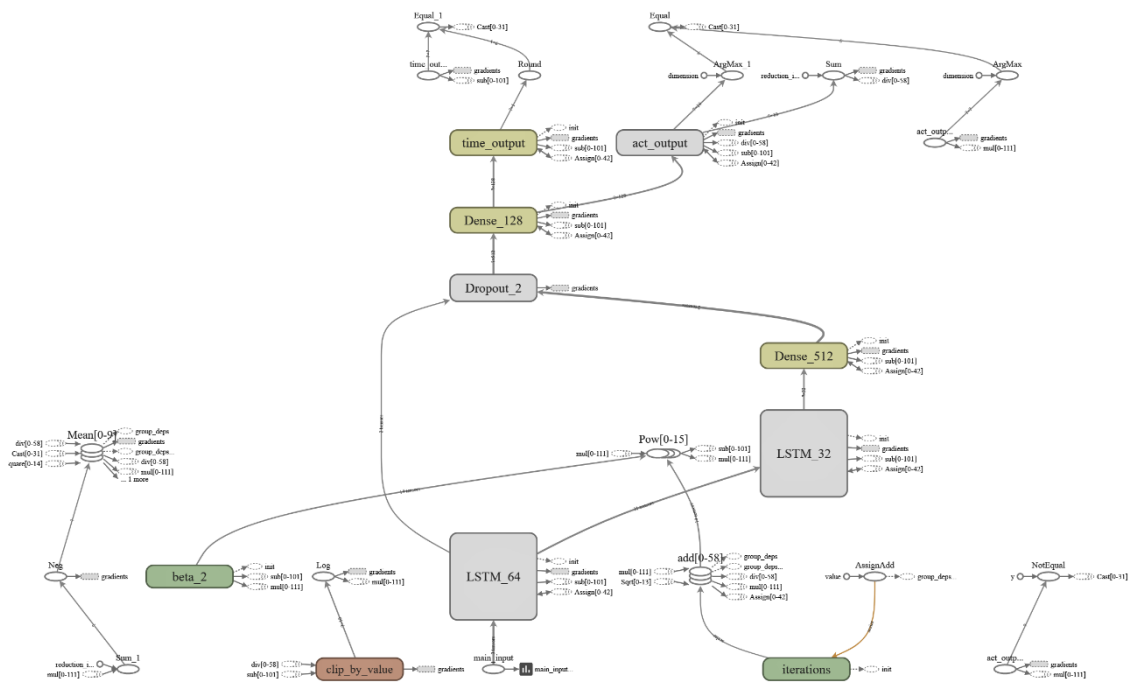
ساختار مدل *NI* شامل نوع و تعداد لایه‌ها بعلاوه ارتباطات آنها در شکل ۵-۱۲ نشان داده شده است. این شکل به وسیله بسته نرم‌افزاری *Keras* ترسیم گردیده است. همین شکل با تفصیل بیشتری در

شکل ۵-۱۱ بازنمایی شده است. گراف موجود در شکل اخیر را نرم‌افزار *Tensorboard* ایجاد و رسم کرده است. این نرم‌افزار شامل مجموعه‌ای از ابزارهای مفید است که برای مصورسازی خصایص، فاکتورها و ویژگی‌های شبکه‌های عصبی در زمان اجرا به کار می‌آیند.

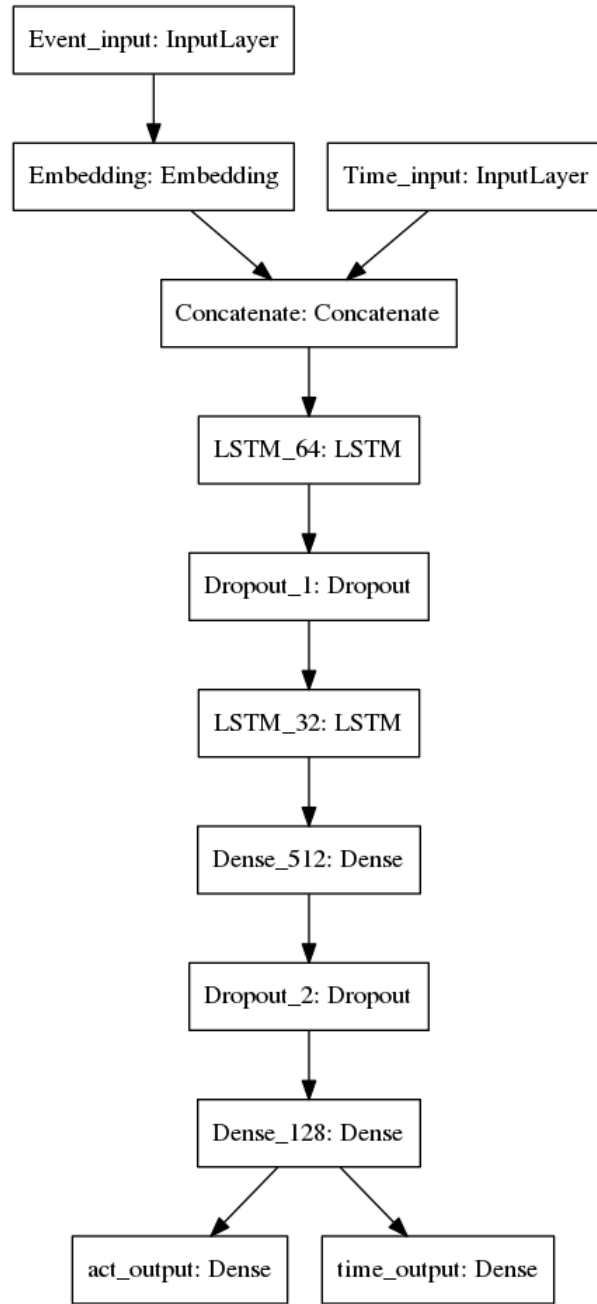
روند آموزش شبکه مدل NI در طول زمان در شکل‌های شکل ۵-۱۳ و شکل ۵-۱۴ نمایش داده شده است. شکل ۵-۱۳ نشان می‌دهد که دقت مدل در پیش‌بینی رویداد حسگر بعدی چگونه افزایش پیدا می‌کند. دقت مدل بیانگر آن است که چند درصد از رویدادهایی که مدل وقوع آن‌ها را در گام بعدی پیش‌بینی می‌کند با واقعیت تطابق دارند. این معیار در دو فاز آموزش و آزمایش اندازه‌گیری و ترسیم شده است. نمودار شکل ۵-۱۴ نیز کاهش خطای مدل در تخمین زمان وقوع رویداد حسگر بعدی را به نمایش می‌گذارد.

چنان‌که از محتویات این دو شکل برمی‌آید، مدل در پیش‌بینی رویداد بعدی بسیار خوب عمل می‌کند؛ اما دقت آن در تخمین زمان این رویداد، در طول فرایند آموزش تغییر چندانی نمی‌یابد. این موضوع بیانگر آن است که مدل پیشنهادی برای بهبود عملکرد خود به توان و ظرفیت بیشتری نیاز دارد. به عبارت دیگر، ما باید با پیش‌پردازش و فرآوری داده‌های ورودی استخراج الگوهای زمانی رویدادها را برای مدل یادشده تسهیل کنیم. همچنین پارامترهای مدل، مثل نرخ یادگیری، تعداد لایه و چیدمان لایه‌ها، تعداد نورون‌های هر لایه و نظایر این‌ها باید تنظیم^۱ شوند.

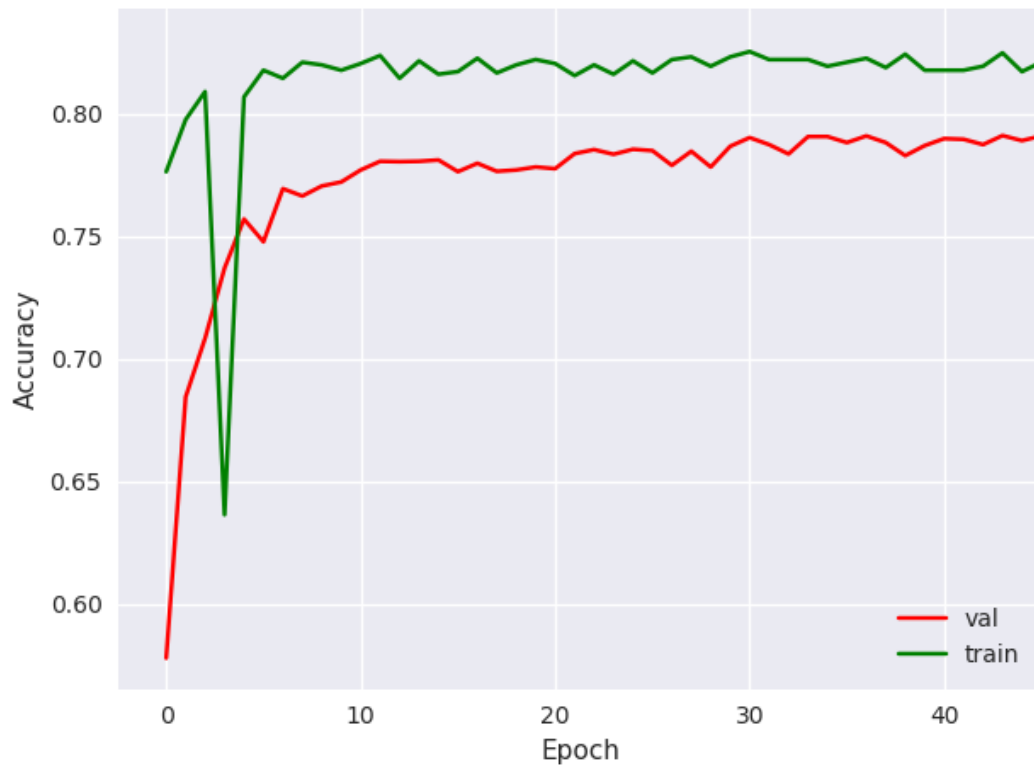
^۱ Tune



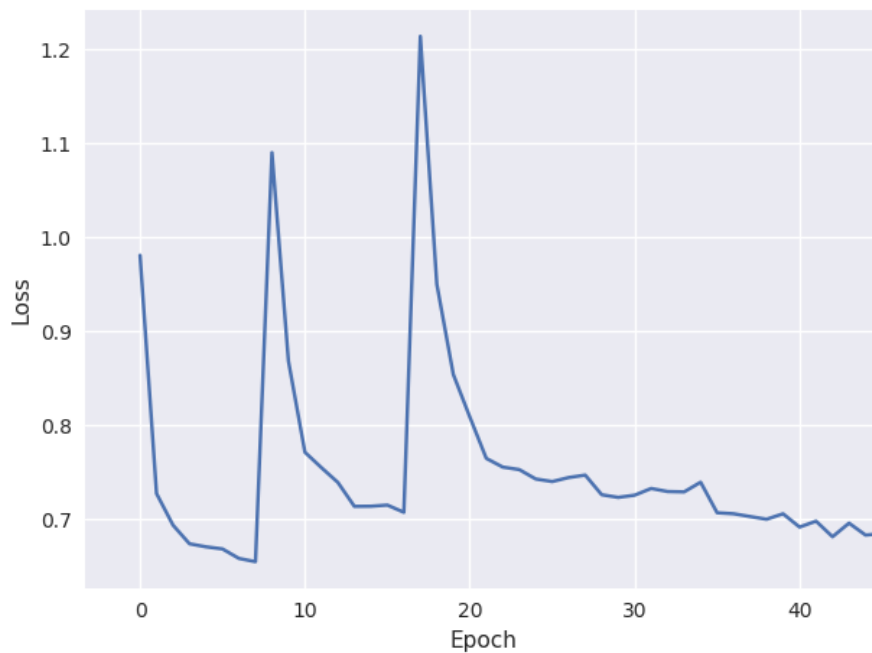
شکل ۱۱-۵ گراف مدل N1 در Tensorboard



شکل ۱۲-۵ گراف مدل N1



شکل ۵-۱۳ روند افزایش دقت مدل N1 در پیش‌بینی رویداد حسگر بعدی



شکل ۵-۱۴ روند کاهش خطای مدل N1 در تخمین زمان رویداد بعدی

۵-۷ استفاده از مکانیزم توجه

ما برای تکمیل مدل پیشنهادی خود از مکانیزم توجه استفاده می‌کنیم. این راه‌حل توانایی مدل را در غلبه بر وابستگی‌های دورودراز بیشتر می‌نماید. بعلاوه نشان می‌دهد که در هر رویداد (t_i, p_i) برای فعال شدن حسگر p_i کدام یک از حسگرهای قبلی مؤثرتر بوده‌اند. بدین ترتیب قابلیت تفسیر مدل هم بیشتر می‌شود و امکان استخراج روابط علت و معلولی از آن فراهم می‌گردد [114].

برای پیاده‌سازی مکانیزم توجه، ما ابتدا روش ارائه تاریخچه H_{t_j} را تغییر می‌دهیم. این تاریخچه در مدل قبلی با آخرین وضعیت شبکه بازگشتی یعنی h_j نمایش داده می‌شد، اما اکنون با مجموعه‌ای از بردارهای وضعیت پنهان به شکل $\{h_i\}_{i=1}^j$ ارائه می‌گردد. هر یک از این بردارها که بردار زمینه^۱ نامیده می‌شوند در واقع خروجی سلول‌های شبکه بازگشتی پس از دریافت ورودی‌های مربوطه $\{z_i, t_i\}_{i=1}^j$ هستند. درایه‌های بردارهای زمینه نشان می‌دهند که میزان تأثیر هر یک از حسگرهای قبلی در فعال شدن حسگر فعلی چقدر بوده است. برآیند این تأثیر در حوزه فرآیندهای نقطه‌ای و به‌ویژه فرآیندهای هاوکس قدرت نفوذ^۲ نام دارد و با $\alpha_{z_i}^z$ نشان داده می‌شود. این مؤلفه اکنون به شکل زیر مدل‌سازی می‌گردد:

$$\alpha_{z_i}^z = f_{att}(h_i, v_z) \quad (۱۵-۵)$$

در این رابطه v_z بردار ویژگی‌های است که شبکه بازگشتی در مورد حسگر z آموخته و f_{att} نقش تابع امتیاز توجه را بازی می‌کند. پس از محاسبه α ما می‌توانیم بردار C_j^z را برای ارائه و ارسال به لایه‌های بعدی ایجاد کنیم:

$$C_j^z = \phi(\{h_i\}_{i=1}^j, \{\alpha_{z_i}^z\}_{i=1}^j) \quad (۱۶-۵)$$

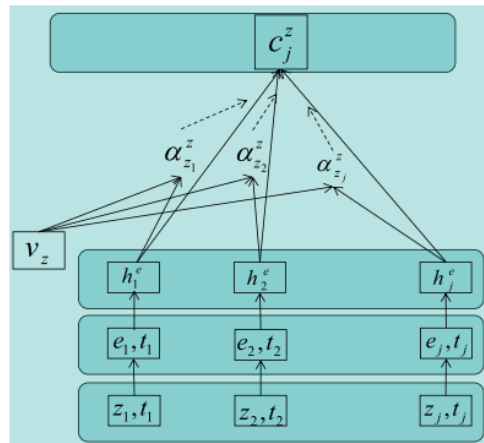
^۱ Context Vector

^۲ Influence Strength

تابع ϕ تابع توجه است که شکل نهایی تاریخچه H_{t_j} بر اساس آن تعیین می‌شود. چون ما از مکانیسم توجه نرم استفاده می‌کنیم تابع ϕ در قالب مجموع حاصلضرب وزن‌های توجه در بردارهای حافظه توجه پیاده‌سازی می‌شود:

$$C_j^z = \sum_{i=1}^s \alpha_{z_i}^z h_i \quad (17-5)$$

شیوه محاسبه توجه در شکل ۵-۱۵ نمایش داده شده است.



شکل ۵-۱۵: شیوه محاسبه توجه در مدل بهبودیافته

چنان‌که از شکل ۵-۱۵ برمی‌آید ما ابتدا رویدادهای ورودی را از یک لایه ویژه می‌گذرانیم و الفبای معرف حسگرها یعنی z_i را به یک فضای برداری حقیقی نگاشت می‌کنیم. این عمل به جاسازی^۱ شهرت دارد و بردارهای حاصل یعنی e_i بردار جاسازی نامیده می‌شود. اکنون بردارهای جاسازی را با برچسب زمانی ترکیب کرده و نتیجه این ترکیب را به‌عنوان ورودی در اختیار لایه بازگشتی قرار می‌دهیم. ساده‌ترین راه برای ترکیب بردار جاسازی و برچسب زمان چسباندن آن‌ها به یکدیگر است. طبیعی است که راه‌های دیگری نیز برای انجام این ترکیب وجود دارد اما ما از روش یادشده استفاده کرده‌ایم. تا این نقطه از فرآیند آموزش را می‌توان به شکل روابط زیر بیان کرد:

$$e_i = w_{em} z_i \quad (18-5)$$

^۱ Embedding

$$h_i^e = LSTM(\{e_i, t_i\}, h_{i-1}^e) \quad (19-5)$$

در این روابط e_i معرف بردار جاسازی ورودی Z_i است و w_{em} بیانگر وزن‌های جاسازی است که در طول فرایند آموزش یاد گرفته شده و مشخص می‌گردند.

برای مجموعه ورودی Z شکل نهایی تاریخچه H_{t_j} پس از گذر از لایه توجه و از طریق C_j^Z مشخص خواهد شد. در مدل ما تابع امتیاز توجه از طریق ضرب داخلی و به روش لانگ [115] پیاده‌سازی شده است یعنی:

$$C_j^Z = f_{att}(h_i^e, v_Z) = \text{DotProduct}(h_i^e * v_Z) \quad (20-5)$$

شکل ۵-۱۸ نشان می‌دهد که مدل NI پس از اضافه شدن لایه توجه به چه شکلی درمی‌آید. این شکل، مثل شکل ۵-۱۲ توسط نرم‌افزار *Keras* ترسیم شده است.

اکنون ما می‌توانیم C_j^Z را برای محاسبه تابع شدت شرطی هر یک از حسگرها مورد استفاده قرار دهیم و سپس نوع حسگر فعال و برچسب زمانی رویداد بعدی را بر اساس آن پیش‌بینی کنیم. برای پیش‌بینی نوع حسگر رویداد بعدی ما تابع سافت‌مکس را بر روی $\{C_j^Z\}_{Z=1}^Z$ اعمال می‌کنیم، درحالی‌که Z برابر تعداد انواع حسگرهاست:

$$u_{j+1} = \text{softMax}(w_u C_j^1, w_u C_j^2, \dots, w_u C_j^Z) \quad (21-5)$$

در این رابطه w_u جزو پارامترهای مدل است که باید در طول فرایند آموزش تنظیم گردد. بر اساس تعریف تابع سافت‌مکس، حسگر فعال رویداد بعدی با انتخاب عنصر بیشینه u_{j+1} مشخص می‌شود:

$$z_{j+1}^* = \text{argmax}(u_{j+1}) \quad (22-5)$$

برای محاسبه برچسب زمان نیز از رابطه زیر استفاده می‌شود:

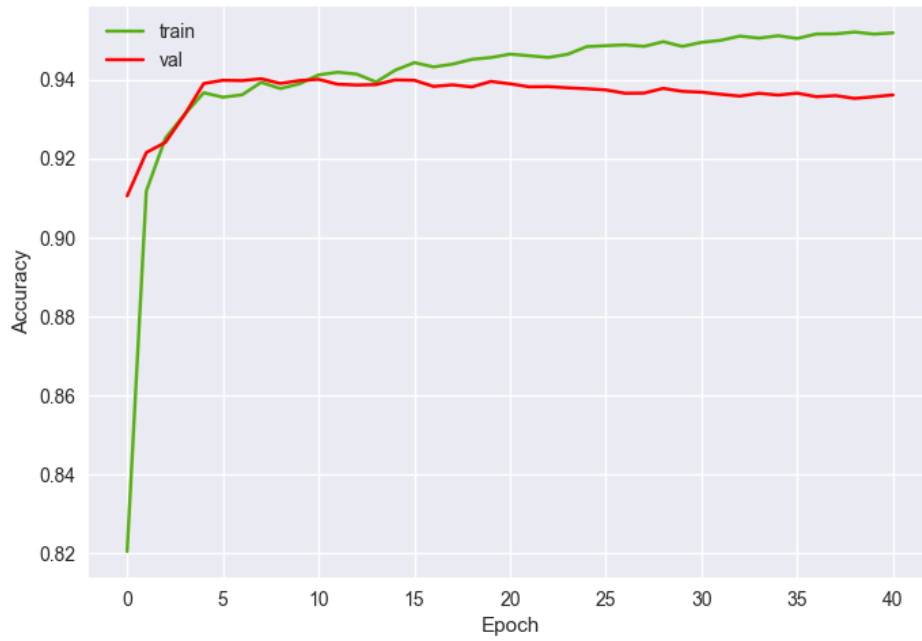
$$t_{j+1}^* = w_c C_j^i + b_c \quad (23-5)$$

در این رابطه w_c جزو پارامترهای مدل است و در زمان یادگیری آموخته خواهد شد.

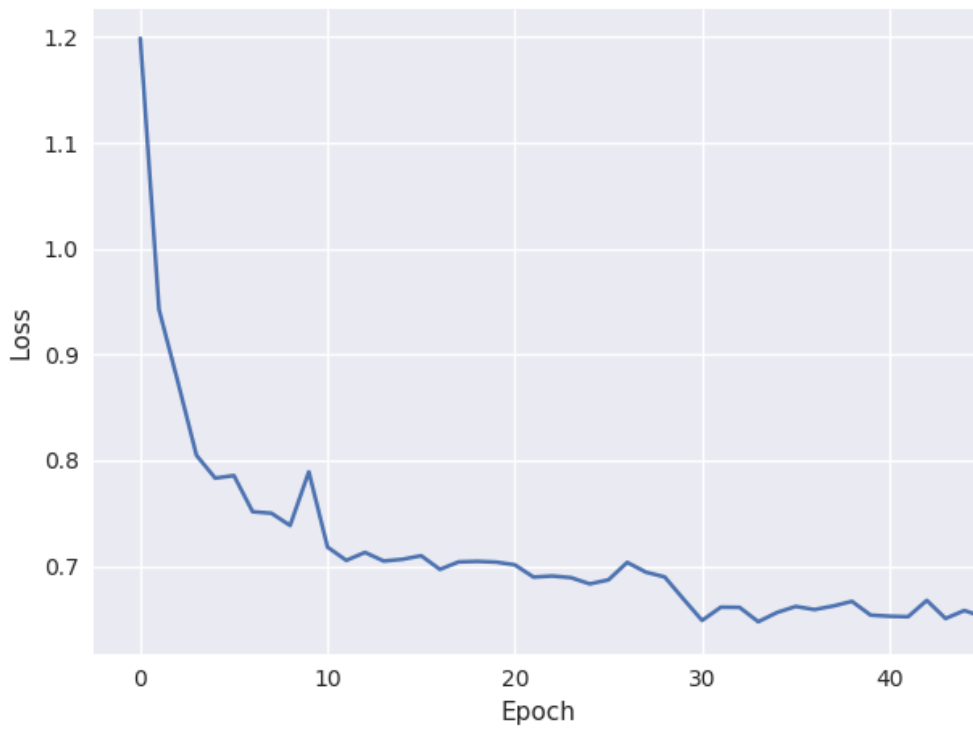
شکل ۵-۱۶ و شکل ۵-۱۷ نشان می‌دهند که دقت مدل پیشنهادی در پیش‌بینی رویداد بعدی و تخمین زمان این رویداد، پس از اعمال مکانیزم توجه چه تغییری می‌کند. بر اساس شکل ۵-۱۶ توجه، دقت مدل را در پیش‌بینی حسگر بعدی به شکل قابل‌توجهی افزایش داده است. اما چنانکه از شکل ۵-۱۷ برمی‌آید، این روش در توانایی مدل برای تخمین زمان رویداد بعد تأثیر چندانی نداشته است. این تفاوت نشان می‌دهد که چیدمان حسگرها در سلسله رویدادهای ثبت‌شده از نظم و قاعده بیشتری برخوردار است. در واقع نظام ترتیبی و علت و معلولی مابین حسگرها به مکانیزم توجه کمک می‌کند که مجموعه حسگرهای مؤثر در فعال شدن یک حسگر خاص را شناسایی کند. اما چنین نظامی در میان نقاط برجسب زمان وقوع وجود ندارد، به همین دلیل به‌کارگیری توجه، دقت مدل را برای تخمین نقطه بعدی بیشتر نکرده است.

شکل ۵-۱۹ تأثیر استفاده از مکانیزم توجه را در فرایند آموزش مدل N1 نشان می‌دهد. در این شکل مدل یادشده سه بار با پارامترهای یکسان، بر روی طول‌های متفاوتی از رشته‌های رویداد حسگرها آموزش داده‌شده است. تفاوت این سه دور در آن است که مدل یک‌بار بدون توجه، یک‌بار با مکانیزم توجه باهداناو [45] و یک‌بار با مکانیزم توجه لانگ آموزش داده‌شده است. این دو روش که به ترتیب به نام Mlp و Dot شهرت دارند در زمره رایج‌ترین شیوه‌های پیاده‌سازی مکانیزم توجه به‌شمار می‌آیند.

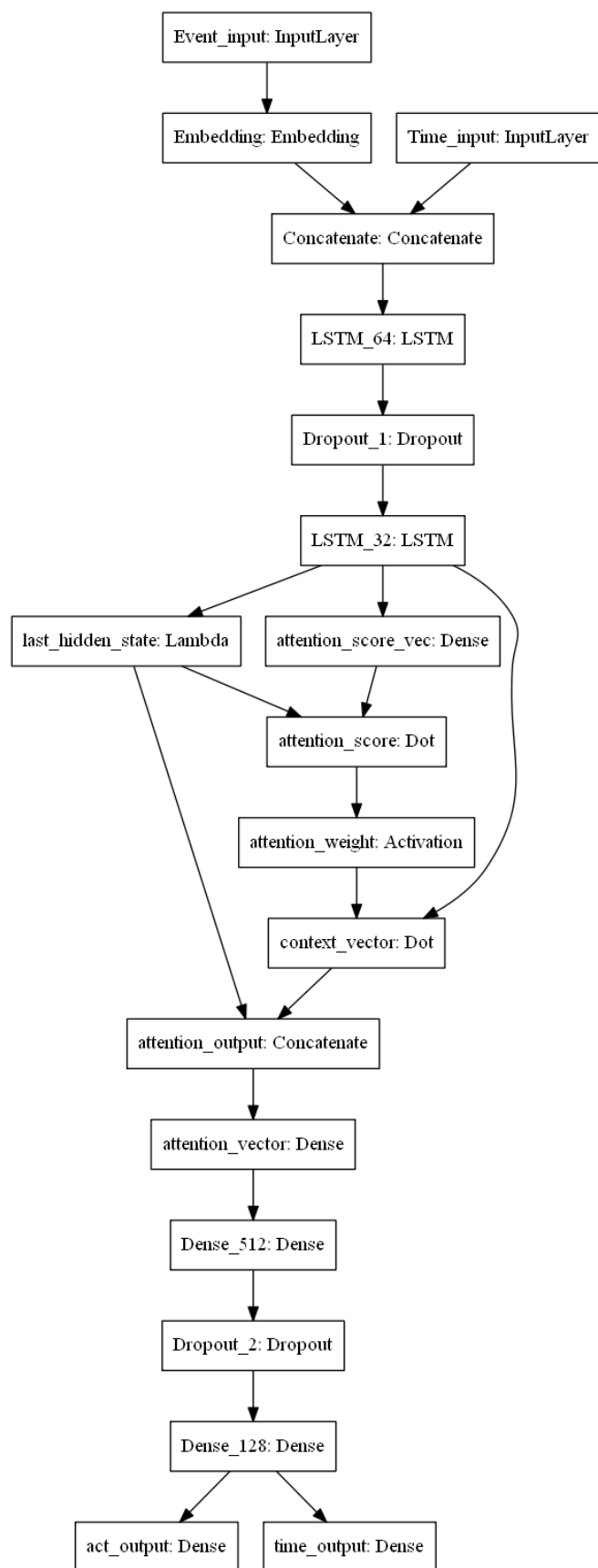
مشاهده نمودار نشان می‌دهد که در هر حال با افزایش طول رشته دقت مدل - دست‌کم در دور اول آموزش ($epoch = 1$) - کاهش پیدا می‌کند. اما به‌کارگیری مکانیزم توجه تقریباً در تمام طول‌ها نتایج بهتری را تولید می‌نماید. ذکر این نکته بیجا نیست که در میان روش‌های متفاوتی که برای پیاده‌سازی تابع امتیاز توجه پیشنهادشده، هیچ‌کدام برتری مطلق ندارند و برای انتخاب شیوه مناسب باید با در نظر گرفتن شرایط مسئله و داده‌های ورودی دست به آزمون و خطا زد.



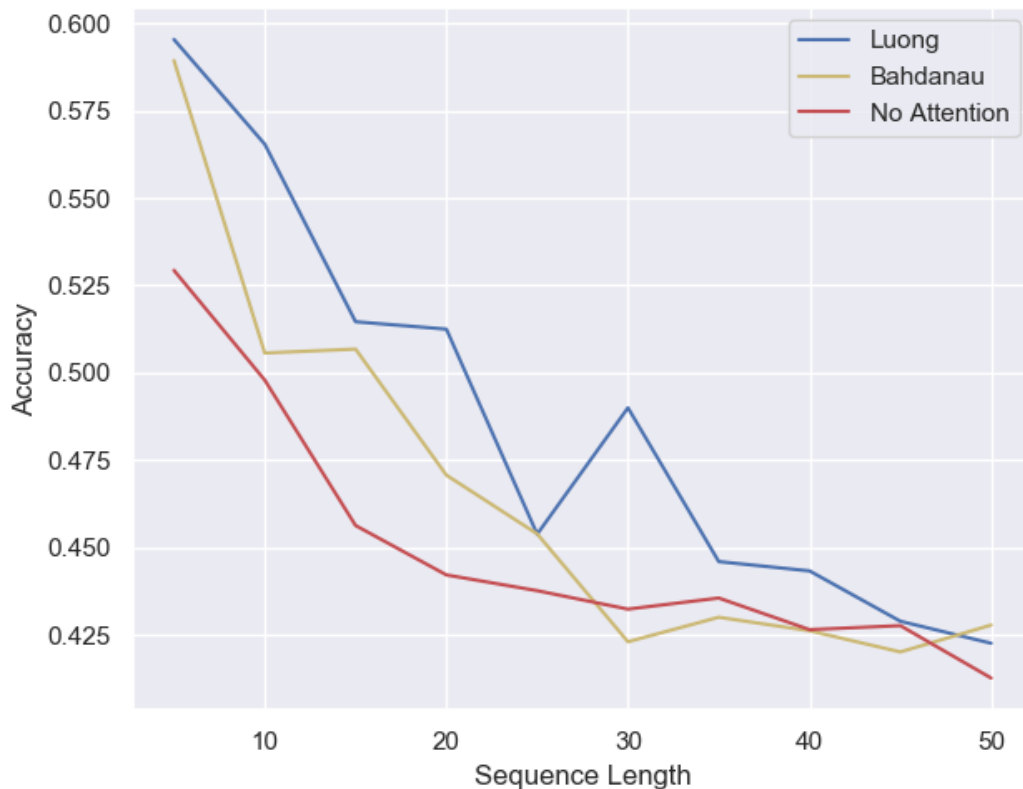
شکل ۵-۱۶ روند افزایش دقت مدل N1 در پیش‌بینی رویداد حسگر بعدی - پس از توجه



شکل ۵-۱۷ روند کاهش خطای مدل N1 در تخمین زمان رویداد بعدی - پس از توجه



شکل ۱۸-۵ گراف مدل N1 پس از اضافه شدن لایه توجه



شکل ۵-۱۹ تاثیر مکانیزم توجه بر دقت مدل N1

۵-۸ عوامل زمانی

افعال انسانی خاصیت دیگری نیز دارد که تحلیل آن‌ها را - حتی در قالب رشته‌های کوتاه - به‌عنوان سری زمانی دشوار می‌کند. این افعال و الگوهای معرف آن‌ها تناوبی و تکراری هستند، اما این تناوب غالباً تک سطحی نیست و از سطوح متعددی برخوردار است. درواقع افعال در دوره‌های تناوب ساعتی، روزانه، هفتگی، ماهانه و ... تکرار می‌شوند. بعلاوه عوامل بیرونی دیگری مثل تعطیلات و تغییرات آب‌وهوا هم وجود دارند که گاه نظم این سیستم تناوبی پیچیده و چند سطحی را به هم می‌ریزند. تأثیرات متقابل دوره‌های یادشده و پیچیدگی عوامل مؤثر مذکور، باعث می‌شود که شبکه‌های عصبی بازگشتی عادی از عهده مدیریت آن‌ها برنیایند [116].

در فصل قبل اشاره کردیم که برای بیان و ارائه یک متغیر گردشی و ادواری مثل زمان بهتر است از تبدیل قطبی استفاده کنیم؛ اما شاید بهتر باشد که در ارائه برچسب زمانی به لایه جاسازی تمهیدات

دیگری را نیز به کار بگیریم. مثلاً الگوی رفتار افراد در روزهای تعطیل و غیر تعطیل تفاوت دارد. ما توقع داریم مدل پس از دریافت داده‌های کافی و رسیدن به نقطه مطلوبی از آموزش، این تفاوت‌ها را استخراج کند تا در معرفی یک فعل به‌عنوان هنجار یا ناهنجار، زمینه مهمی مثل تعطیل یا غیر تعطیل بودن روز مربوطه نیز لحاظ گردد. از آنجاکه خصیصه تعطیل بودن به‌طور مستقیم از ورودی برچسب زمانی برنمی‌آید بهتر است ما به شکل دستی چنین خصیصه‌ای را به ورودی‌های مدل اضافه کنیم. برای انجام این کار کافی است یک فیلد عددی ۲ مقداری (۰ : روز کاری - ۱ : روز تعطیل) را به مجموعه ورودی‌ها بیفزاییم. همین موضوع در مورد ویژگی‌های زمانی دیگری مثل قبل‌ازظهر و بعدازظهر، روز هفته، روز از ماه، ماه از سال و ... نیز صدق می‌کند.

همه این ویژگی‌ها را می‌توان با فیلدهای عددی دو یا چند مقداری به مدل اضافه کرد. علاوه بر این بسته به نمونه داده‌های تحت بررسی می‌توان به خصایص زمانی مهندسی‌شده دیگری نیز فکر کرد. به‌عنوان مثال می‌توانیم ۲۴ ساعت شبانه‌روز را به محدوده‌های مساوی یا نامساوی مثل ساعت کار، ساعت استراحت و ساعت خواب تقسیم کنیم. برای تعیین نوع و تعداد این ویژگی‌های دست‌ساز و بررسی میزان تأثیر آن‌ها ابتدا باید دادگان موردنظر را تحت تحلیل اکتشافی^۱ قرار بدهیم [117].

لازم به ذکر است که تمام ویژگی‌های یادشده به‌طور ضمنی در محتوای برچسب زمانی وجود دارند و به‌رحال از درون این خصیصه قابل‌استخراج و استنباط هستند. ولی استخراج دستی این ویژگی‌ها و اضافه کردن آن‌ها به‌عنوان ورودی سیستم، توان مدل را در کشف الگوهای معنایی موردنظر افزایش داده و کیفیت الگوهای کشف‌شده را بالاتر می‌برد. این خصایص پس از وضع و مقداردهی به شکل تک‌روشن درمی‌آیند، یا با گذر از لایه جاسازی به فضای برداری حقیقی نگاشت می‌شوند.

برخی از دادگان خانه هوشمند در مورد هر رویداد ویژگی‌هایی مثل نوع حسگر و مکان مربوطه را نیز ثبت می‌کنند. در صورتی که نتایج تحلیل اکتشافی دال بر تأثیر این خصایص باشد آن‌ها نیز در قالب توالی‌های تک‌روشن، یا به شکل جاسازی‌شده به مجموعه فوق‌الذکر اضافه خواهند شد.

^۱ Exploratory Analysis

فصل ۶ نتیجه گیری و کارهای آتی

۶-۱ مقدمه

ما در این رساله تلاش کردیم تا با تلفیق نظریه فرآیندهای نقطه‌ای و ابزارهای قدرتمند عرصه یادگیری عمیق، روشی داده محور را برای کشف افعال ناهنجار در خانه‌های هوشمند ارائه دهیم. در حوزه کشف ناهنجاری در خانه‌های هوشمند، مفاهیم «کنش‌های رایج» و «مدل‌سازی رفتار» مفاهیمی پایه و کلیدی به شمار می‌آیند. بیشتر متخصصین حوزه مذکور، سعی کرده‌اند تا با تعریف و صورت‌بندی این مفاهیم بر اساس سطوح انتزاعی متفاوتی از داده‌های حسگرها، ناهنجاری را به‌عنوان انحراف از رفتارهای رایج معرفی کنند. این محققین برای نیل به مقصود خود شیوه‌های متفاوتی را بکار گرفته‌اند که در میان مهم‌ترین آن‌ها می‌توان به روش‌های منطقی، روش‌های آماری، روش‌های مبتنی بر قاعده و روش‌های یادگیری ماشین اشاره کرد.

کاری که در رساله حاضر انجام پذیرفته است، در زمره یادگیری ماشین قرار دارد؛ اما تفاوت آن با کارهای قبلی این است که مفهوم رویداد را به‌عنوان سنگ بنا مورد استفاده قرار داده و این مقوله محوری را با تمسک به یک نظریه آماری صورت‌بندی کرده است. بدین ترتیب علاوه بر توجه به ارتباط زمانی رویدادهای متوالی، وابستگی نوع و زمان وقوع یک رویداد به‌عنوان دو متغیر تصادفی در تدوین مدل لحاظ گردیده است.

نتایج این تحقیق نشان می‌دهد، خانه‌ای که با تعدادی حسگر ساده و غیر مزاحم تجهیز شده باشد می‌تواند اطلاعات ارزشمندی را در مورد اوضاع ساکنان خود ارائه نماید. این تحقیق ثابت می‌کند که پیاده‌سازی و به‌کارگیری سامانه‌های مراقبت مبتنی بر حسگرهای متراکم، برای پایش سالخوردگان، بیماران، کودکان و سایر افراد نیازمند از دیدگاه نظری و فنی، ممکن و موجه است.

۶-۲ نتایج

نتایج اجرای تحقیق حاضر را می‌توان در رئوس زیر خلاصه کرد:

- ضعف غالب بسیاری از پژوهش‌های پیشین در آن است که به‌جای استفاده از داده‌های خام حسگرها، بر مفاهیم درشت‌دانه تر کردار و رفتار تکیه کرده‌اند. این دو مفهوم برای افراد مختلف و در زمان‌های متفاوت بسیار متنوع هستند و ارائه یک تعریف واحد از آن‌ها تقریباً غیرممکن می‌نماید. به همین دلیل ما سعی کرده‌ایم تا به‌جای استفاده از مفاهیم مذکور مستقیماً داده‌های خام حسگرها را برای تشخیص ناهنجاری بکار بگیریم. درعین‌حال توانایی یادگیری عمیق در استخراج خودکار الگوهای تکرارشونده و ویژگی‌های انتزاعی، به ما کمک می‌کند که به‌طور ضمنی از مفاهیم فوق‌الذکر در مدل خود برخوردار شویم.
- استفاده از چارچوب نظری فرآیندهای نقطه‌ای، پشتوانه قابل اتکایی را برای این تحقیق فراهم کرده است. این فرآیندها در علم آمار و احتمال به تفضیل موردبحث و بررسی قرار گرفته‌اند و عملگرهای قدرتمندی مثل نازک‌سازی^۱ و برهم‌نهی در مورد آن‌ها تعریف شده است. همچنین بسته‌های نرم‌افزاری کارآیی مثل tick و PoPPy وجود دارند که کار پردازش و داده‌ورزی را از طریق فرآیندهای نقطه‌ای زمانی و مکانی ممکن و آسان می‌کنند.
- در میان فرآیندهای نقطه‌ای، فرآیند هاوکس روش ساده و قابل‌درک و تفسیری را برای ارائه رویدادهای خروجی حسگرها فراهم می‌کند. این رویدادها از نظر زمانی به هم وابسته هستند و معمولاً فعال شدن یک حسگر باعث فعال شدن سلسله‌ای از حسگرهای بعدی می‌شود. تابع شدت شرطی این فرآیند، بدون نیاز به محاسبات پارامتریک و از طریق یک شبکه عصبی بازگشتی، از درون داده‌ها استخراج شده است.

^۱ Thinning

- رشته‌های رویداد معرف رفتار، بسیار متنوع هستند. این رشته‌ها می‌توانند حتی در مورد یک کار واحد که توسط یک فرد مشخص انجام پذیرفته است، متفاوت باشند. استفاده از روش جستجوی پرتو محلی به‌جای جستجوی حریصانه، تمهیدی است که ما برای غلبه بر این تنوع و گوناگونی اتخاذ کرده‌ایم. بدین ترتیب به‌جای یک‌رشته، چند رشته محتمل را استخراج می‌کنیم تا مدل در فضای واقع‌بینانه‌تری عمل کند.
- به‌کارگیری مکانیسم توجه کمک می‌کند که مدل پیشنهادی بتواند رشته‌های طولانی‌تری از رویدادهای حسگر را پردازش نماید. این روش موجب می‌شود که مدل در هر رشته، تنها حسگرهای اصلی را به‌خاطر بسپارد و حافظه محدود خود را برای ذخیره داده‌هایی که نقش تعیین‌کننده ندارند صرف ننماید.
- ما داده‌های ادواری – مثل ساعت و روز – را با استفاده از تبدیل قطبی به شیوه‌ای مناسب فرآوری کرده‌ایم. بعلاوه مواردی مثل تعطیل/غیرتعطیل یا قبل‌ازظهر/بعدازظهر را از میان برچسب زمان استخراج نموده و آن‌ها را به‌عنوان ورودی در اختیار مدل گذاشته‌ایم. همه این پیش‌پردازش‌ها موجب شده است تا مدل، زمینه‌های زمانی رویدادها را در پیش‌بینی خود صراحتاً لحاظ کند.

۳-۶ کارهای آینده

- برخی از مواردی که می‌توان برای تکمیل و توسعه تحقیق حاضر انجام داد به شرح زیر هستند:
- ما رفتار را به‌عنوان سلسله‌ای از رویدادها تعریف کرده‌ایم. به همین دلیل ابزار یا مدلی که به‌جای پیش‌بینی یک تک‌رویداد، رشته‌ای از رویدادها را پیش‌بینی کند با مسئله ما تناسب بسزایی دارد. مدل رشته به رشته^۱ در یادگیری عمیق، مدلی است که دقیقاً همین کار را می‌کند؛ یعنی با دریافت یک رشته از ورودی، رشته احتمالی بعدی را پیش‌بینی می‌نماید. در سال‌های اخیر این

^۱ Sequence to Sequence

روش با کدگشای جستجوی پرتومحلی^۱ ارائه شده و به همین دلیل توان پیش‌بینی مجموعه‌ای از رشته‌های محتمل آتی را پیدا کرده است. به‌کارگیری مدل رشته به رشته به‌جای استفاده از پیش‌بینی تکراری ممکن است به بهبود کارایی روش پیشنهادی این رساله منجر بشود.

- یکی از بزرگ‌ترین موانعی که ما در خلال انجام این تحقیق با آن روبرو بودیم، فقدان دادگان مناسب و تخصصی برای کشف ناهنجاری در خانه‌های هوشمند بود. تقریباً تمامی دادگان موجود خانه‌های هوشمند با موضوع کشف و شناسایی افعال تدوین‌شده‌اند؛ یعنی از ناهنجاری کنترل‌شده و برچسب خورده تهی هستند. به همین دلیل در صورت استفاده از آن‌ها، امکان ارزیابی دقت و جامعیت مدل فراهم نیست. ایجاد نرم‌افزاری که بتواند یک‌خانه هوشمند را با امکان درج ناهنجاری شبیه‌سازی کند یا دست‌کم ناهنجاری‌های موردنظر کاربر را به یک مجموعه داده موجود بیفزاید، پیشنهاد مفیدی است که کار را برای محققان آینده ساده‌تر خواهد کرد.

- رفتارهای انسانی ماهیت یقینی ندارند، یعنی همیشه و به‌هرحال میزانی از عدم قطعیت در پیش‌بینی رویداد بعدی وجود خواهد داشت. به همین دلیل بسیاری از مطالعاتی که درزمینهٔ بازشناسی و کشف افعال انجام‌شده، بر پایه مدل‌های احتمالاتی -به‌ویژه مدل‌های بیزی- قرار دارند. از آنجاکه روش‌های موجود یادگیری عمیق این عدم قطعیت را لحاظ نمی‌کنند گروهی از محققین دست به تدوین یادگیری عمیق بیزی^۲ زده‌اند. شبکه‌های عصبی بیزی در مقابل بیش‌برازش مصون‌ترند و آموزش آن‌ها از طریق دادگان کوچک و نامتوازن بهتر صورت می‌پذیرد. به این دلایل توقع می‌رود که پیاده‌سازی روش پیشنهادی رساله حاضر با یک شبکه عمیق بیزی نتایج بهتری را به بار آورد.

- چنانکه گفتیم روش پیشنهادی در سطح رویدادهای حسگر عمل می‌کند و امکان برچسب زدن به اعمال معمول و غیرمعمول را ندارد. این موضوع می‌تواند مثل یک تیغ دودم عمل کند. از سویی

^۱ Beam Search Decoder

^۲ Bayesian Deep Learning

نقطه قوت است زیرا سیستم به مجموعه از پیش تعریف شده‌ای از اعمال محدود و مقید نمی‌شود، اما از سوی دیگر می‌تواند با تشخیص یا پیش‌بینی پیاپی افعال جدید به‌عنوان ناهنجاری، حساسیت کاربران خود را کاهش داده و به چوپان دروغ‌گو تبدیل بشود. به همین دلیل کاهش نرخ تشخیص غلط به‌ویژه در مسیر پیاده‌سازی واقعی و تجاری سیستم بسیار مهم بوده و مستلزم تحقیق، آزمایش و ابداع خواهد بود.

مراجع

- [1] Bakar, A. B. U. A. U., Ghayvat, H., Hasanm, F. S., and Mukhopadhyay, S. C., (2016),“Activity and Anomaly Detection in Smart Home: A Survey,” *Smart Sensors, Measurement and Instrumentation*, pp. 191–220.
- [2] Ward, J. A., Lukowicz, P., Tröster, ., and Starner, T. E., (2006),“Activity recognition of assembly tasks using body-worn microphones and accelerometers,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 10, pp. 1553–1566.
- [3] Fan, X., Huang, H., Xie, C., Tang, Z., and Zeng, J., (2014),“Private Smart Space: Cost-Effective ADLs (Activities of Daily Livings) Recognition Based on Superset Transformation,” in *Proceedings - 2014 IEEE International Conference on Ubiquitous Intelligence and Computing, 2014 IEEE International Conference on Autonomic and Trusted Computing, 2014 IEEE International Conference on Scalable Computing and Communications and Associated Sy*, pp. 757–762.
- [4] Tran, A. C., Marsland, S., Dietrich, J., Guesgen, H. W., and Lyons, P., (2010),“Use cases for abnormal behaviour detection in smart homes,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6159 LNCS, pp. 144–151.
- [5] Hawkins, D. M., (1981),*Identification of Outliers.*, vol. 37, no. 4. Dordrecht: Springer Netherlands.
- [6] Aggarwal, C. C., (2013),*Outlier analysis*. Springer New York.
- [7] Chandola, V., Banerjee, A., and Kumar, V., (2009),“Anomaly detection: A Survey,” *ACM Computing Surveys*, vol. 41, no. 3, pp. 1–58.
- [8] Goldstein, M., Goldstein, M., and Uchida, S., (2016),“A Comparative Evaluation of Unsupervised Anomaly Detection Algorithms for Multivariate Data,” *PLoS ONE*, no. April, pp. 1–31.
- [9] Heller, K., Svore, K., Keromytis, A. D., and Stolfo, S., (2003),“One class support vector machines for detecting anomalous windows registry accesses,” in *Workshop on Data Mining for Computer Security (DMSEC), Melbourne, FL, November 19, 2003*.
- [10] Roth, V., (2006),“Kernel fisher discriminants for outlier detection,” *Neural Computation*, vol. 18, no. 4, pp. 942–960.
- [11] Barbará, D., Wu, N., and Jajodia, S., (2001),“Detecting Novel Network

- Intrusions Using Bayes Estimators,” in *Proceedings of the 2001 SIAM International Conference on Data Mining*, pp. 1–17.
- [12] De Stefano, C., Sansone, C., and Vento, M., (2000), “To reject or not to reject: that is the question - an answer in case of neural classifiers,” *IEEE Transactions on Systems, Man and Cybernetics Part C: Applications and Reviews*, vol. 30, no. 1, pp. 84–94.
- [13] Sain, S. R. and Vapnik, V. N., (1996), “The Nature of Statistical Learning Theory,” *Technometrics*, vol. 38, no. 4, p. 409.
- [14] Boser, B. E., Guyon, I. M., and Vapnik, V. N., (1992), “Training algorithm for optimal margin classifiers,” in *Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pp. 144–152.
- [15] King, S. P., King, D. M., Anuzis, P., Astley, K., Tarassenko, L., Hayton, P., and Utete, S., (2002), “The use of novelty detection techniques for monitoring high-integrity plant,” in *IEEE Conference on Control Applications - Proceedings*, vol. 1, pp. 221–226.
- [16] Davy, M. and Godsill, S., (2002), “Detection of abrupt spectral changes using support vector machines an application to audio signal segmentation,” in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, vol. 2, pp. 1313–1316.
- [17] Lazarevic, A., Ertöz, L., Kumar, V., Ozgur, A., and Srivastava, J., (2003), “A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection,” in *Proceedings of the 2003 SIAM International Conference on Data Mining*, pp. 25–36.
- [18] Ma, J. and Perkins, S., “Time-series novelty detection using one-class support vector machines,” in *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, pp. 1741–1745.
- [19] Hu, W., Liao, Y., and Vemuri, V. R., (2003), “Robust anomaly detection using support vector machines,” in *Proceedings of the international conference on machine learning*, pp. 282–289.
- [20] Markou, M. and Singh, S., (2003), “Novelty detection: a review—part 2:: neural network based approaches,” *Signal Processing*, vol. 83, no. 12, pp. 2499–2521.
- [21] Augusteijn, M. F. and Folkert, B. A., (2002), “Neural network classification and novelty detection,” *International Journal of Remote Sensing*, vol. 23, no. 14, pp. 2891–2902.
- [22] Thompson, B. B., Marks, R. J., Choi, J. J., El-Sharkawi, M. A., Huang, M. Y., and Bunje, C., (2002), “Implicit learning in autoencoder novelty assessment,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2878–2883.
- [23] Díaz, I. and Hollmén, J., (2002), “Residual generation and visualization for understanding novel process conditions,” in *Proceedings of the International Joint Conference on Neural Networks*, vol. 3, pp. 2070–2075.
- [24] Crook, P. and Hayes, G., (2001), “A robot implementation of a biologically inspired method for novelty detection,” in *Proceedings of the Towards Intelligent Mobile Robots Conference*.
- [25] Crook, P. A., Marsland, S., Hayes, G., and Nehmzow, U., (2002), “A tale of two filters - on-line novelty detection,” in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 4, pp. 3894–3899.
- [26] Murray, A. F., (2001), “Novelty detection using products of simple experts—a potential architecture for embedded systems,” *Neural networks: the official*

- journal of the International Neural Network Society*, vol. 14, no. 9, pp. 1257–64.
- [27] Williams, G., Baxter, R., Hongxing He, Hawkins, S., and Lifang Gu, “A comparative study of RNN for outlier detection in data mining,” in *2002 IEEE International Conference on Data Mining, 2002. Proceedings.*, pp. 709–712.
- [28] Hawkins, S., He, H., Williams, G., and Baxter, R., (2002), “Outlier detection using replicator neural networks,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 2454 LNCS, pp. 170–180.
- [29] Goodfellow, I., Bengio, Y., and Courville, A., (2016), *Deep Learning*. MIT Press.
- [30] Bengio, Y., Courville, A., and Vincent, P., (2013), “Representation learning: A review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828.
- [31] “A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way.” [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed: 03-Dec-2019].
- [32] Lipton, Z. C., Berkowitz, J., and Elkan, C., (2015), “A Critical Review of Recurrent Neural Networks for Sequence Learning,” *ArXiv*, vol. abs/1506.0, pp. 1–38.
- [33] Elman, J., (1990), “Finding structure in time* 1,” *Cognitive Science*, vol. 14, no. 1, pp. 179–211.
- [34] Jordan, M., (1997), “Serial order: A parallel distributed processing approach,” *Advances in Psychology*, vol. 121, pp. 471–495.
- [35] Hochreiter, S. and Jürgen Schmidhuber, J., (1997), “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780.
- [36] Gers, F. A. and Schmidhuber, J., (2000), “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*, pp. 189–194 vol.3.
- [37] “Minimal character-based LSTM implementation - Eli Bendersky’s website.” [Online]. Available: <https://eli.thegreenplace.net/2018/minimal-character-based-lstm-implementation/>. [Accessed: 05-Dec-2019].
- [38] Lipton, Z. C., Berkowitz, J., and Elkan, C., (2015), “A Critical Review of Recurrent Neural Networks for Sequence Learning.”
- [39] Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J., (2017), “LSTM: A Search Space Odyssey,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.
- [40] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y., (2014), “Learning phrase representations using RNN encoder-decoder for statistical machine translation,” in *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pp. 1724–1734.
- [41] Chung, J., Gulcehre, C., Cho, K., and Bengio, Y., (2014), “Empirical evaluation of gated recurrent neural networks on sequence modeling,” in *NIPS 2014 Workshop on Deep Learning, December 2014*.
- [42] Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., and Jin, D., (2018), “DeepMove: Predicting Human Mobility with Attentional Recurrent Networks,” in *Proceedings of the 2018 World Wide Web Conference on World*

- Wide Web - WWW '18*, pp. 1459–1468.
- [43] “Attention? Attention!” [Online]. Available: <https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>. [Accessed: 05-Dec-2019].
- [44] Mnih, V., Heess, N., Graves, A., and Kavukcuoglu, K., (2014), “Recurrent Models of Visual Attention,” in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, pp. 2204–2212.
- [45] Bahdanau, D., Cho, K., and Bengio, Y., (2015), “Neural Machine Translation by Jointly Learning to Align and Translate,” in *3rd International Conference on Learning Representations, {ICLR} 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- [46] “Dive into Deep Learning — Dive into Deep Learning 0.7.1 documentation.” [Online]. Available: <https://d2l.ai/>. [Accessed: 05-Dec-2019].
- [47] Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., Zemel, R., and Bengio, Y., (2015), “Show, Attend and Tell: Neural Image Caption Generation with Visual Attention,” in *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 2048–2057.
- [48] “Comprehensive Introduction to Autoencoders - Towards Data Science.” [Online]. Available: <https://towardsdatascience.com/generating-images-with-autoencoders-77fd3a8dd368>. [Accessed: 05-Dec-2019].
- [49] Srivastava, N., Mansimov, E., and Salakhutdinov, R., (2015), “Unsupervised learning of video representations using LSTMs,” in *32nd International Conference on Machine Learning, ICML 2015*, vol. 1, pp. 843–852.
- [50] Chen, L., Nugent, C. D., and Wang, H., (2012), “A knowledge-driven approach to activity recognition in smart homes,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974.
- [51] Chen, L. and Khalil, I., (2011), *Activity Recognition: Approaches, Practices and Trends*. Atlantis Press.
- [52] Kunze, K. and Lukowicz, P., (2014), “Sensor placement variations in wearable activity recognition,” *IEEE Pervasive Computing*, vol. 13, no. 4, pp. 32–41.
- [53] Bourke, A. K. and Lyons, G. M., (2008), “A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor,” *Medical Engineering and Physics*, vol. 30, no. 1, pp. 84–90.
- [54] Huynh, T., Blanke, U., and Schiele, B., (2007), “Scalable recognition of daily activities with wearable sensors,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 4718 LNCS, pp. 50–67.
- [55] Tapia, E. M., Intille, S. S., Haskell, W., Larson, K., Wright, J., King, A., and Friedman, R., (2007), “Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor,” in *Proceedings - International Symposium on Wearable Computers, ISWC*, pp. 37–40.
- [56] Kwapisz, J. R., Weiss, G. M., and Moore, S. A., (2011), “Activity recognition using cell phone accelerometers,” *ACM SIGKDD Explorations Newsletter*, vol. 12, no. 2, p. 74.
- [57] Banos, O., Damas, M., Pomares, H., Rojas, F., Delgado-Marquez, B., and Valenzuela, O., (2013), “Human activity recognition based on a sensor weighting hierarchical classifier,” *Soft Computing*, vol. 17, no. 2, pp. 333–343.
- [58] Sukthankar, G. and Sycara, K., (2005), “A cost minimization approach to human

- behavior recognition,” in *Proceedings of the International Conference on Autonomous Agents*, pp. 1189–1196.
- [59] Storf, H., Kleinberger, T., Becker, M., Schmitt, M., Bomarius, F., and Prueckner, S., (2009), “An event-driven approach to activity recognition in ambient assisted living,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5859 LNCS, pp. 123–132.
- [60] Cook, D. J. and Holder, L. B., (2011), “Sensor selection to support practical use of health-monitoring smart environments,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 1, no. 4, pp. 339–351.
- [61] Tapia, E. M., Intille, S. S., and Larson, K., (2004), “Activity recognition in the home using simple and ubiquitous sensors,” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 3001, pp. 158–175.
- [62] Magnanimo, V., Saveriano, M., Rossi, S., and Lee, D., (2014), “A Bayesian approach for task recognition and future human activity prediction,” in *Proceedings - IEEE International Workshop on Robot and Human Interactive Communication*, vol. 2014-October, no. October, pp. 726–731.
- [63] Guo, P., Miao, Z., Zhang, X. P., Shen, Y., and Wang, S., (2012), “Coupled observation decomposed hidden Markov model for multiperson activity recognition,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 9, pp. 1306–1320.
- [64] Singla, G., Cook, D. J., and Schmitter-Edgecombe, M., (2010), “Recognizing independent and joint activities among multiple residents in smart environments,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 1, no. 1, pp. 57–63.
- [65] Tseng, C. C. and Cook, D., (2007), “Mining from time series human movement data,” in *Conference Proceedings - IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 3241–3243.
- [66] Zheng, H., Wang, H., and Black, N., (2008), “Human activity detection in smart home environment with self-adaptive neural networks,” in *Proceedings of 2008 IEEE International Conference on Networking, Sensing and Control, ICNSC*, pp. 1505–1510.
- [67] Medjahed, H., Istrate, D., Boudy, J., and Dorizzi, B., (2009), “Human activities of daily living recognition using fuzzy logic for elderly home monitoring,” in *IEEE International Conference on Fuzzy Systems*, pp. 2001–2006.
- [68] Akhlaghinia, M. J., (2010), “Occupancy Monitoring and Prediction in Ambient Intelligent Environment.”
- [69] Doctor, F., Hagraas, H., and Callaghan, V., (2005), “A Fuzzy embedded agent-based approach for realizing ambient intelligence in intelligent inhabited environments,” *IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans.*, vol. 35, no. 1, pp. 55–65.
- [70] Das, B., Chen, C., Dasgupta, N., Cook, D. J., and Seelye, A. M., (2010), “Automated prompting in a smart home environment,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 1045–1052.
- [71] Ordóñez, F. J., de Toledo, P., and Sanchis, A., (2015), “Sensor-based Bayesian detection of anomalous living patterns in a home setting,” *Personal and Ubiquitous Computing*, vol. 19, no. 2, pp. 259–270.
- [72] Hela, S., Amel, B., and Badran, R., (2018), “Early anomaly detection in smart

- home: A causal association rule-based approach,” *Artificial Intelligence in Medicine*, no. November 2017.
- [73] Khan, S. S., Karg, M. E., Hoey, J., and Kulić, D., (2012), “Towards the detection of unusual temporal events during activities using HMMs,” in *UbiComp’12 - Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pp. 1075–1084.
- [74] Monekosso, D. N. and Remagnino, P., (2010), “Behavior analysis for assisted living,” *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 4, pp. 879–886.
- [75] Forkan, A. R. M., Khalil, I., Tari, Z., Foufou, S., and Bouras, A., (2015), “A context-aware approach for long-term behavioural change detection and abnormality prediction in ambient assisted living,” *Pattern Recognition*, vol. 48, no. 3, pp. 628–641.
- [76] Yuan, B. and Herbert, J., (2014), “Context-aware Hybrid Reasoning Framework for Pervasive Healthcare,” *Personal Ubiquitous Comput.*, vol. 18, no. 4, pp. 865–881.
- [77] Rivera-illingworth, F., Callaghan, V., and Hagra, H., (2006), “Towards the detection of temporal behavioural patterns in intelligent environments,” in *IET Conference Publications*, vol. 2006, no. 518, pp. 119–126.
- [78] Breunig, M. M., Kriegel, H.-P., Ng, R. T., and Sander, J., (2000), “LOF: identifying density-based local outliers,” in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD ’00*, vol. 29, no. 2, pp. 93–104.
- [79] Ye, L., Qin, Z. G., Wang, J., and Jin, J., (2010), “Anomaly event detection in temporal sensor network data of intelligent environments,” in *ICCET 2010 - 2010 International Conference on Computer Engineering and Technology, Proceedings*, vol. 7, pp. V7-414-V7-420.
- [80] Rashidi, P., Cook, D. J., Holder, L. B., and Schmitter-Edgecombe, M., (2011), “Discovering activities to recognize and track in a smart environment,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 4, pp. 527–539.
- [81] Jakkula, V. and Cook, D., (2010), “Outlier detection in smart environment structured power datasets,” in *Proceedings - 2010 6th International Conference on Intelligent Environments, IE 2010*, pp. 29–33.
- [82] Novak, M., Binas, M., and Jakab, F., (2012), “Unobtrusive anomaly detection in presence of elderly in a smart-home environment,” in *Proceedings of 9th International Conference, ELEKTRO 2012*, no. June 2016, pp. 341–344.
- [83] Moshtaghi, M., Zukerman, I., and Andrew Russell, R., (2015), “Statistical models for unobtrusively detecting abnormal periods of inactivity in older adults,” *User Modeling and User-Adapted Interaction*, vol. 25.
- [84] Paudel, R., Eberle, W., and Holder, L. B., (2018), “Anomaly Detection of Elderly Patient Activities in Smart Homes using a Graph-Based Approach Predicting Cognitive Health Problem on Elderly Using Smart Home Data View project Anomaly Detection of Elderly Patient Activities in Smart Homes using a Graph-Bas,” no. July.
- [85] Jakkula, V. and Cook, D. J., (2011), “Detecting Anomalous Sensor Events in Smart Home Data for Enhancing the Living Experience,” in *AAAI Workshop on Artificial Intelligence and Smarter Living The Conquest of Complexity*, no. June 2015, pp. 33–37.

- [86] Daley, D. J. and Vere-Jones, D., (2008), *An Introduction to the Theory of Point Processes*. Springer, New York, NY.
- [87] Rizoiu, M., Lee, Y., and Mishra, S., (2017), "Hawkes processes for events in social media," in *Frontiers of Multimedia Research*, pp. 191–218.
- [88] Cochran, J. J., Cox, L. A., Keskinocak, P., Kharoufeh, J. P., Smith, J. C., and Schoenberg, F. P., (2011), "Introduction to Point Processes," *Wiley Encyclopedia of Operations Research and Management Science*.
- [89] Choi, C. S. and Baccelli, F., (2018), "Poisson cox point processes for vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 10, pp. 10160–10165.
- [90] Molyneux, J., Gordon, J. S., and Schoenberg, F. P., (2018), "Assessing the predictive accuracy of earthquake strike angle estimates using nonparametric Hawkes processes," *Environmetrics*, vol. 29, no. 2, p. e2491.
- [91] Rotondi, R. and Varini, E., (2019), "Failure models driven by a self-correcting point process in earthquake occurrence modeling," *Stochastic Environmental Research and Risk Assessment*, vol. 33, no. 3, pp. 709–724.
- [92] Møller, J., Ghorbani, M., and Rubak, E., (2016), "Mechanistic spatio-temporal point process models for marked point processes, with a view to forest stand data," *Biometrics*, vol. 72, no. 3, pp. 687–696.
- [93] Johnson, N., Hitchman, A., Phan, D., and Smith, L., (2018), "Self-exciting point process models for political conflict forecasting," *European Journal of Applied Mathematics*, vol. 29, no. 4, pp. 685–707.
- [94] Xiao, S., Farajtabar, M., Ye, X., Yan, J., Song, L., and Zha, H., (2017), "Wasserstein learning of deep generative point process models," in *Advances in Neural Information Processing Systems*, vol. 2017-Decem, pp. 3248–3258.
- [95] Kirchner, M. and Bercher, A., (2018), "A nonparametric estimation procedure for the Hawkes process: comparison with maximum likelihood estimation," *Journal of Statistical Computation and Simulation*, vol. 88, no. 6, pp. 1106–1116.
- [96] Islam, K. T., Shelton, C. R., Casse, J. I., and Wetzel, R., (2017), "Marked Point Process for Severity of Illness Assessment," in *Proceedings of the 2nd Machine Learning for Healthcare Conference*, vol. 68, pp. 255–270.
- [97] Yan, J., Liu, X., Shi, L., Li, C., and Zha, H., (2018), "Improving maximum likelihood estimation of temporal point process via discriminative and adversarial learning," in *IJCAI International Joint Conference on Artificial Intelligence*, vol. 2018-July, pp. 2948–2954.
- [98] Thomas, S., Srijith, P. K., and Lukasik, M., (2018), "A Bayesian point process model for user return time prediction in recommendation systems," in *UMAP 2018 - Proceedings of the 26th Conference on User Modeling, Adaptation and Personalization*, no. June, pp. 363–364.
- [99] Kline, D., (2004), "Methods for Multi-Step Time Series Forecasting with Neural Networks," *Neural Networks in Business Forecasting*, pp. 226–250.
- [100] Reddy, D. R. (D. of C. S., (1977), *Speech Understanding Systems: A Summary of Results of the Five-Year Research Effort at CMU*. .
- [101] Freitag, M. and Al-Onaizan, Y., (2017), "Beam Search Strategies for Neural Machine Translation," in *Proceedings of the First Workshop on Neural Machine Translation*, pp. 56–60.
- [102] Park, K., Lin, Y., Metsis, V., Le, Z., and Makedon, F., (2010), "Abnormal human behavioral pattern detection in assisted living environments," *Proceedings of the*

- 3rd International Conference on Pervasive Technologies Related to Assistive Environments - PETRA '10*, no. November 2014, p. 1.
- [103] Malhotra, P., Vig, L., Shroff, G., and Agarwal, P., (2015),“Long Short Term Memory Networks for Anomaly Detection in Time Series,” in *European Symposium on Artificial Neural Networks*, no. April, pp. 22–24.
- [104] Ramasamy Ramamurthy, S. and Roy, N., (2018),“Recent trends in machine learning for human activity recognition-A survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, vol. 8, no. 4, p. e1254.
- [105] Alshammari, N., Alshammari, T., Sedky, M., Champion, J., and Bauer, C., (2017),“OpenSHS: Open Smart Home Simulator,” *Sensors*, vol. 17, no. 5, p. 1003.
- [106] Cook, D. J., Crandall, A. S., Thomas, B. L., and Krishnan, N. C., (2013),“CASAS: A Smart Home in a Box,” *Computer*, vol. 46, no. 7, pp. 62–69.
- [107] Ye, J., Stevenson, G., and Dobson, S., (2016),“Detecting abnormal events on binary sensors in smart home environments,” *Pervasive and Mobile Computing*, vol. 33, pp. 32–49.
- [108] Shin, J. H., Lee, B., and Park, K. S., (2011),“Detection of abnormal living patterns for elderly living alone using support vector data description.,” *IEEE transactions on information technology in biomedicine: a publication of the IEEE Engineering in Medicine and Biology Society*, vol. 15, no. 3, pp. 438–448.
- [109] Fawcett, T., (2006),“An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, pp. 861–874.
- [110] “ROC Analysis - ML Wiki.” [Online]. Available: http://mlwiki.org/index.php/ROC_Analysis. [Accessed: 08-Dec-2019].
- [111] Staudemeyer, R. C., (2015),“Applying long short-term memory recurrent neural networks to intrusion detection,” *Sacj*, vol. 56, no. 56, pp. 136–154.
- [112] Campos, G. O., Zimek, A., Sander, J., Campello, R. J. G. B., Micenková, B., Schubert, E., Assent, I., and Houle, M. E., (2016),“On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study,” *Data Mining and Knowledge Discovery*, vol. 30, no. 4, pp. 891–927.
- [113] Emmott, A., Das, S., Dietterich, T., Fern, A., and Wong, W.-K., (2015),“A Meta-Analysis of the Anomaly Detection Problem,” Oregon State University Libraries & Press.
- [114] Choi, E., Bahadori, M. T., Kulas, J. A., Schuetz, A., Stewart, W. F., and Sun, J., (2016),“RETAIN: An interpretable predictive model for healthcare using reverse time attention mechanism,” in *Advances in Neural Information Processing Systems*, no. Nips, pp. 3512–3520.
- [115] Luong, M. T., Pham, H., and Manning, C. D., (2015),“Effective approaches to attention-based neural machine translation,” in *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421.
- [116] Wu, X., Shi, B., Dong, Y., Huang, C., Faust, L., and Chawla, N. V., (2018),“Restful: Resolution-aware forecasting of behavioral time series data,” in *International Conference on Information and Knowledge Management, Proceedings*, pp. 1073–1082.
- [117] Apte, C. V., Hong, S. J., Natarajan, R., Pednault, E. P. D., Tipu, F. A., and Weiss, S. M., (2003),“Data-intensive analytics for predictive modeling,” *IBM Journal of Research and Development*, vol. 47, no. 1, pp. 17–23.

Abstract

One of the most important features a smart home should have is predicting and detecting unusual and abnormal actions. The rapid development of sensor technologies on the one hand, and the significant increase in the number of elderly or ill living, on the other hand, have made this necessary. The problem is that the multiplicity and variety of sensors and the high volume of information received from them make the dimensions of the problem closer to the big data boundaries and the need for data fusion makes it more difficult. For this reason, the processing of this data, especially the extraction of appropriate features, involves considerable computational (temporal and spatial) complexity.

In this thesis, we present a method for detecting and predicting unusual and abnormal actions in a smart home using deep learning. The data of this home, according to the prevailing tradition of smart homes, is collected through non-intrusive wireless sensors such as PIR and Door sensors. The output data of these sensors is a set of discrete asynchronous events occurring on a continuous time basis. The intervals of these events are not the same, and the interval between the two events may indicate important behavioral characteristics. For this reason, we have used point process theory instead of the usual methods of time series analysis. This theory is a powerful mathematical framework that provides the appropriate statistical language for formulating and analyzing asynchronous events.

Common models in the Classic theory of point processes often take specific and limiting assumptions about the way events are produced. For this reason, we have used a deep neural network to surround the behavior of a process and extract normative patterns. This network, as a nonlinear mapping, predicts the type and timing of the next event, with event logs. We extend this prediction by employing a local beam search to a set of probable events to determine the possible range of future actions. The range of actual events within that range will indicate that the sequence is abnormal. The results of experiments show that this method enables the detection of abnormal activities with acceptable accuracy.

Keywords : Anomaly Detection, Deep Learning, Point Process, Smart Homes, Event Processing



Shahrood University of
Technology

Faculty of Computer Engineering

Ph.D. Thesis in Computer Engineering – Artificial Intelligence

Anomaly Detection and Prediction in Smart Homes using Deep Learning

By: Mahmoud Moallem

Supervisor:
Dr. Ali Akbar Pouyan

Advisor:
Dr. Hamid Hassanpour

September ,2019