
بسم الله الرحمن الرحيم

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گروه مهندسی فناوری اطلاعات

پایان‌نامه کارشناسی

رشته مهندسی کامپیوتر گرایش فناوری اطلاعات

عنوان پروژه:

تحلیل زمانی رفتار ترافیکی شبکه با استفاده از الگوریتم‌های تحلیل سری زمانی

استاد راهنما:

دکتر بهروز شاهقلی

پژوهشگران:

رضا پازن

علی هداوند

شهریور ۱۴۰۰

دانشگاه اصفهان

دانشکده مهندسی کامپیوتر

گروه مهندسی فناوری اطلاعات

پروژه کارشناسی رشته‌ی مهندسی کامپیوتر گرایش فناوری اطلاعات

آقایان علی هداوند و رضا پازن

تحت عنوان

تحلیل زمانی رفتار ترافیکی شبکه با استفاده از الگوریتم‌های تحلیل سری زمانی

در تاریخ / / ۱۳ توسط هیأت داوران زیر بررسی و با نمره به تصویب نهایی رسید.

۱- استاد راهنمای پروژه:

دکتر امضا

۲- استاد داور :

دکتر امضا

امضای مدیر گروه

تقدیم به

پدر و مادر عزیزمان، که تلاش بی‌دریغ‌شان روشنی‌بخش آینده‌ی ما است.

چکیده:

در دنیای امروز، شبکه‌های کامپیوتری یکی از فناوری‌های پرکاربرد در تمام زمینه‌های زندگی انسان‌ها است. از تلفن‌های همراه گرفته تا موتورهای جست‌وجو و زیرساخت‌های سازمانی، همگی از شبکه‌های کامپیوتری برای ایجاد ارتباط و انتقال اطلاعات استفاده می‌کنند. با توجه به میزان گستردگی این فناوری، مراقبت و حفظ سلامت شبکه‌های کامپیوتری یکی از دغدغه‌های اصلی متخصصان و ناظران این شبکه‌ها است. یکی از مهم‌ترین عوامل نگهداری و حتی پیش‌بینی شبکه، دانستن الگوی رفتاری آن است. روش‌ها و ابزارهای زیادی برای پایش و بررسی رفتار شبکه وجود دارد ولی هیچکدام به ثبت رفتار شبکه بدون تحلیل‌های انسانی کمک نمی‌کند. این پروژه با استفاده از مفاهیم آماری و سری‌های زمانی در راستای ثبت الگوی رفتاری شبکه‌های سازمانی انجام شده است. سری‌های زمانی داده‌هایی هستند که وابسته به زمان بوده و پیوسته در حال تغییر هستند. داده‌های جاری در شبکه نیز از این نوع بوده و می‌توان آن‌ها را با استفاده از تحلیل‌های سری زمانی و مفاهیم آماری مانند میانگین، انحراف‌معیار و واریانس تحلیل کرد. روش پیاده‌سازی شده در این پروژه، با تعریف دو پنجره‌ی مبدأ و آزمون شروع شده و با محاسبه‌ی میانگین و انحرف معیار برای هر پنجره، پردازش روی داده‌ها آغاز می‌شود. بازه‌هایی برای هر دو پنجره، با مرکزیت میانگین و حد بالا و پایین تعیین شده توسط انحراف داده‌ها از میانگین تعیین شده و هم‌پوشانی این بازه‌ها بررسی می‌شود. در صورت عدم وجود هم‌پوشانی، هر پنجره یک دانه‌ی زمانی شناخته شده و به خروجی اضافه می‌شود. خروجی این پروژه دانه‌های زمانی هستند که هر دانه‌ی زمانی، بازه‌ی مشخصی از زمان است که در آن، شبکه رفتار مشخص و متفاوتی نسبت به سایر بازه‌ها از خود نشان می‌دهد. دانستن نوع رفتار شبکه در هر زمان به ناظران کمک می‌کند تا در صورت بروز ناهنجاری بتوانند به راحتی آن را تشخیص دهند. همچنین از داده‌های به دست آمده در نتیجه‌ی تحلیل‌های سری زمانی می‌توان در راستای پیش‌بینی رفتار شبکه استفاده کرد. نتایج به دست آمده در این پروژه نشان می‌دهند که روش پیاده‌سازی شده می‌تواند روشی مناسب از نظر تحلیل‌های آماری برای تشخیص الگوری رفتاری شبکه‌های سازمانی باشد.

واژگان کلیدی: شبکه‌های کامپیوتری، تحلیل داده‌های جاری، سری زمانی، دانه‌های زمانی، تشخیص ناهنجاری، پیش‌بینی

فهرست مطالب

فهرست شکل‌ها

مخفف‌ها

CN	Computer Networks
TS	Time Series
ACF	Auto Correlation Function
PACF	Partial Auto Correlation Function
AR	Auto Regressive
MA	Moving Average
ARMA	Auto Regressive Moving Average
ARIMA	Autoregressive Integrated Moving Average
ETS	Error Trend Seasonality
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
SNMP	Simple Network Management Protocol
XML	Extensible Markup Language
LSTM	Long Short Term Memory
IoT	Intenet of Things
RMON	Remote Network Monitoring
CMIP	Common Management Information Protocol
FOD	First Order Difference

فصل اول

مقدمه

۱–۱ بیان مسئله

در طول تاریخ یکی از اساسی‌ترین نیازهای انسان برقراری ارتباط بوده است که با گذر زمان، ابزار و روش‌های آن نیز دستخوش تغییر شده‌اند. امروزه، فناوری شبکه‌های کامپیوتری[۱] (CN) یکی از مهم‌ترین ابزارهای برآورده کردن نیازهای تعاملی انسان است. از ابتدای پیداش این فناوری تا کنون، به دلیل گسترش جوامع و به دنبال آن گسترش استفاده از شبکه‌های کامپیوتری، میزان داده‌ی جاری در این شبکه‌ها افزایش یافته و چالش‌هایی نیز پیش روی کاربران و توسعه‌دهندگان این فناوری قرار داده است. مواردی مانند تامین محرمانگی، وجود یا عدم وجود خطا، صحت انتقال اطلاعات، تأخیر و غیره، از جمله این چالش‌ها هستند.

با توجه به گسترده شدن کاربرد شبکه‌های کامپیوتری و حضور این فناری در تمام عرصه‌های زندگی انسان، از سازمان‌های بزرگ تا کاربری خانگی، حفظ سلامت این شبکه‌ها امر بسیار مهمی تلقی می‌شود. یکی از راه‌های کنترل این شبکه‌ها، پایش[۲] آن‌ها است. پایش به معنی جمع‌آوری و تحلیل بسته‌های[۳] انتقال داده شده در شبکه است [۱]. کارشناسان و متخصصان، با استفاده از ابزارها و روش‌های مختلف پایش شبکه مانند پروتکل مدیریت ساده‌ی شبکه[۴] (SNMP) و پایش بلادرنگ[۵] قادر به مشاهده، ذخیره و بررسی داده‌های شبکه هستند که به آن‌ها کمک می‌کند رفتار شبکه‌ی موردنظر را ثبت و در صورت بروز ناهنجاری[۶] یا خطا در جریان داده‌ها، آن را گزارش کنند [۱].

تحلیل داده‌های شبکه و تشخیص الگوی رفتاری آن کار آسانی نیست؛ زیرا جریان داده‌ای شبکه متغیری پیوسته در زمان است و فرایند تحلیل و نتیجه‌گیری در این مورد باید بازده قابل‌قبولی داشته باشد. یکی از اهداف تحلیل داده‌های شبکه را می‌توان ثبت رفتار آن دانست. در این راستا، در این پروژه با استفاده از مفاهیمی به نام سری‌های

---

[1] Computer Networks
[2] Monitoring
[3] Packets
[4] Simple Network Management Protocol
[5] Real–Time Monitoring
[6] Anomaly

زمانی[1] (TS) در کنار علم آمار، روشی برای تحلیل داده‌های شبکه و دسته‌بندی[2] رفتار آن در بازه‌های زمانی مختلف، جهت ثبت یک الگوی ثابت، ارائه شده است.

۱–۲ ارزش پروژه

در زمینه‌ی تحلیل داده‌های شبکه پروژه‌های مشابهی وجود دارد که خروجی آن‌ها با استفاده از ابزارهایی مانند زبان نشانه‌گذاری توسعه‌پذیر[3] (XML) تولید می‌شود. در این پروژه‌ها، بازه‌های زمانی به صورت پیش‌فرض توسط کاربر تعریف می‌شوند و تحلیل جریان داده‌ی شبکه فقط در بازه‌های از پیش تعریف شده و به صورت ایستا انجام می‌شود. پروژه‌ها و مقاله‌های دیگر نیز که از مفاهیم سری زمانی برای تحلیل داده‌های شبکه استفاده کرده‌اند، به هدف تشخیص ناهنجاری و یا دسته‌بندی جریان داده‌ی شبکه بر اساس ماهیت داده‌ها بوده است؛ به طور مثال داده‌های شبکه‌های اجتماعی[4] از داده‌های دیگر سرویس‌ها جداسازی شود.

در این پروژه، داده‌های موجود با استفاده از مفاهیم سری زمانی مورد بررسی قرار گرفته است و خروجی مورد نظر، بازه‌های زمانی هستند که هر بازه‌ی زمانی مشخص کننده‌ی قطعه‌ای از زمان است که میزان داده‌ی جاری در شبکه در آن زمان مقدار مشخصی دارد. این خروجی به سیستم و کاربران قالب مشخصی از رفتار شبکه را ارائه می‌دهد که در تصمیم‌گیری‌های آینده بسیار مؤثر است. همچنین قالب ورودی پروژه به صورت فایل است که این فایل شامل اطلاعات بسته‌های شناسایی شده‌ی جاری در شبکه است. این امر باعث افزایش قابلیت حمل برنامه‌ی پروژه می‌شود.

در نهایت با توجه به نیاز جدی نظارت بر رفتار و جریان داده‌های شبکه‌های کامپیوتری و همچنین پیچیدگی زیاد و زمان‌بر بودن استفاده از ابزارهای موجود برای تحلیل داده‌ها، پیاده‌سازی ابزاری که فرایند تحلیل و نتیجه‌گیری و در نهایت تصمیم‌گیری را به صورت خودکار ارائه می‌دهد و خروجی متفاوت و پرکاربردی تولید می‌کند امری قابل توجه است.

به طور کلی، با استفاده از مفاهیم سری زمانی می‌توان ویژگی‌های رفتاری داده‌های ورودی را مشخص کرد. استفاده از تحلیل سری زمانی، به کاهش خطاهای احتمالی پیش‌آمده در روند تحلیل داده‌ها کمک می‌کند و امکان پیش‌بینی[5] رفتار آینده‌ی شبکه و همچنین پیاده‌سازی سیستم تشخیص ناهنجاری را نیز به کاربر می‌دهد.

۱–۳ هدف پروژه

هدف اصلی از انجام این پروژه، ارائه‌ی روشی جهت تحلیل جریان داده‌ی شبکه‌های کامپیوتری با استفاده از مفاهیم تحلیل سری‌های زمانی است. تحلیل‌ها و پردازش‌های انجام شده بر روی سری زمانی دادهای جاری در شبکه با هدف تولید بازه‌های زمانی است که الگوی رفتاری شبکه موردنظر را نتیجه می‌دهد.

به‌طورکلی این پروژه اهداف زیر را در نظر دارد.

---

[1] Time Series
[2] Classification
[3] Extensible Markup Language
[4] Social Media
[5] Forecasting

- افزایش بهره‌وری از داده‌های جاری در شبکه جهت شناسایی الگوری رفتاری
- ایجاد زمینه برای پیاده‌سازی سیستم‌های تشخیص ناهنجاری در شبکه
- دسته‌بندی رفتار شبکه طبق تغییرات جریان داده‌ها طی گذر زمان
- بهره‌گیری از تحلیل‌های آماری و سری زمانی در توصیف رویدادهای شبکه

۱–۴ رویکرد پیشنهادی

این پروژه با استفاده از داده‌های شبیه‌سازی‌شده انجام شده است. جهت تحلیل و پردازش‌های گسترده‌تر نیاز به داده‌های سالیانه از شبکه‌های فعال بود که دسترسی به این داده‌ها مشکل است.

به طور کلی مراحل انجام پروژه به صورت زیر است:

- **مرحله‌ی اول: پردازش و آماده‌سازی داده‌ها**

در این مرحله فایل داده‌های ورودی در برنامه بارگذاری شده و اطلاعات موردنیاز آن جداسازی می‌شود. همچنین قالب تاریخ و ساعت درج شده در فایل داده‌ها به قالب قابل پردازش در برنامه تبدیل شده و اندیس‌گذاری ستون‌ها بر اساس تاریخ و ساعت به دست آمده مرتب می‌شود.

- **مرحله‌ی دوم: تحلیل داده‌های پردازش شده و استخراج ویژگی‌های اولیه**

در این مرحله با استفاده از توابع آماده‌سازی شده در کتابخانه‌های اضافه شده به برنامه، نمودارهای داده‌های پردازش‌شده را رسم کرده و ویژگی‌های سری زمانی به‌دست‌آمده را بررسی و ثبت می‌کنیم.

- **مرحله‌ی سوم: پردازش نهایی، تعریف پنجره‌های اولیه و دریافت خروجی**

در این مرحله با تعریف پنجره‌های اولیه و محاسبه‌ی واریانس و میانگین داده‌ها، تحلیل نهایی انجام شده و خروجی مورد نظر تولید می‌شود.

۱–۵ ساختار پایان‌نامه

این پایان‌نامه شامل هفت فصل است. در فصل دوم مفاهیم تحلیل داده‌های شبکه‌های کامپیوتری، ابزارها، روش‌ها و در نهایت مفاهیم موردنیاز جهت تحلیل و بررسی نتایج اولیه‌ی سری زمانی در کنار بررسی تحقیقات انجام‌شده بیان شده‌اند. در فصل سوم رویکرد پیشنهادی به طول کامل بررسی شده و مسئله‌ی بیان شده به تفصیل بررسی می‌شود. در فصل چهارم نتایج به‌دست‌آمده ارائه شده و به تحلیل و بررسی آن‌ها به همراه ارائه‌ی کدهای مهم و ابزارهای استفاده شده پرداخته شده است. در فصل پنجم جمع‌بندی و نتیجه‌گیری نهایی بیان شده است. در نهایت بخش ششم پیوست‌ها و بخش هفتم منابع را ارائه می‌دهند.

فصل دوم
مفاهیم و کلیات

۲–۱ مقدمه

با توجه به گسترش علوم و فنون تحلیل داده‌ها، روش‌های زیادی در زمینه‌های تخصصی مختلف ارائه شده است. در این مورد، تحلیل سری‌های زمانی در دهه‌های اخیر بسیاری از محققان را به خود جذب کرده است. این پروژه با استفاده از مفاهیم موجود در علم تحلیل سری‌های زمانی، داده‌های جمع‌آوری شده از شبکه‌های کامپیوتری را مورد بررسی قرار می‌دهد.

روش‌های زیادی در زمینه‌ی تحلیل جریان داده‌ی شبکه و با اهداف مختلف ارائه شده است که در این فصل در کنار مرور این روش‌ها، به بررسی تعاریف و مفاهیم ابتدایی سری‌های زمانی، الگوریتم‌ها و مدل‌های موجود و نکات آن پرداخته شده است. همچنین تحقیقات انجام شده در این زمینه نیز مورد بررسی قرار گرفته و تعاریف بیان‌شده در آن‌ها توضیح داده شده است.

۲–۲ تحلیل جریان داده‌ی شبکه‌های کامپیوتری

امروزه، اهداف مدیران شبکه‌های کامپیوتری از تحلیل داده‌های شبکه به نگهداری وضعیت کنونی شبکه خلاصه نمی‌شود. با پیشرفت علوم هوش مصنوعی[1] (AI) و داده‌پردازی، این امکان برای مدیران شبکه فراهم شده است تا با به‌کارگیری این فناوری‌ها در کنار مفاهیم شبکه‌های کامپیوتری، رفتار آینده‌ی شبکه‌ها را پیش‌بینی کرده، داده‌های جاری در آن‌ها را دسته‌بندی و ناهنجاری‌های رخ داده در شبکه را شناسایی و یا حتی پیش‌بینی کنند. همچنین با استفاده از ابزارهای پایش موجود می‌توان داده‌های شبکه را به صورت بلادرنگ جمع‌آوری و در قالب گزارشات در فرمت‌های مرسوم ضبط بسته[2] (pcap) و یا مقادیر جدا شده با کاما[3] (csv) ذخیره کرد.

از آنجایی‌که امروز یافتن، ثبت و پیش‌بینی الگوی رفتاری شبکه‌های کامپیوتری اهمیت زیادی دارد، تکنیک‌هایی از علوم مختلف در این زمینه استفاده شده است که هرکدام اهداف و ویژگی‌های مشخصی دارد. به طور مثال، حافظه‌های

[1] Artificial Intelligence
[2] Packet Capture
[3] Comma–Seperated Values

طولانی کوتاه‌مدت[1] (LSTM)، که یک معماری در علم یادگیری عمیق[2] (DL) است، در کنار زمینه‌های مختلف شبکه، مانند اینترنت اشیاء[3] (IoT) با اهداف مختلفی مانند پیش‌بینی داده‌های شبکه پیاده‌سازی می‌شود [۱۳، ۱۴]. همچنین تحلیل‌های آماری نیز درمورد داده‌های شبکه کاربرد زیادی دارند. به طور مثال، مفاهیم سری زمانی، بیشتر از جنبه‌های آماری به تحلیل داده‌های شبکه می‌پردازند. استفاده از مدل‌های مختلف این الگوریتم‌ها به طور ترکیبی نیز مرسوم است. استفاده از شبکه‌های LSTM در کنار سری‌های زمانی جهت پیش‌بینی و یا دسته‌بندی کردن داده‌ها نمونه‌ای از این کاربردها است.

اولین مرحله برای شروع فرایند تحلیل داده‌ها، جمع‌آوری آن‌هاست. ابزارهای زیادی برای پایش و جمع‌آوری داده‌های شبکه وجود دارد که به صورت بلادرنگ به جمع‌آوری داده‌های جاری می‌پردازند. اما باید توجه داشت که این نرم‌افزارها صرفا به جمع‌آوری داده‌ها کمک می‌کنند و تحلیل و نتیجه‌گیری را بر عهده‌ی کاربر می‌گذارند. نرم‌افزارهایی مانند وایرشارک و یا اتریل ابزارهای پایش شبکه هستند که امکان جمع‌آوری و ذخیره‌سازی داده‌ها را در فایل‌هایی با فرمت pcap به کاربر می‌دهند. فایل‌های حاصل را می‌توان پایگاه‌داده‌ای از بسته‌های جاری در شبکه دانست که حاوی اطلاعاتی مانند آدرس مبدا، آدرس مقصد، حجم بسته و غیره هستند.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.133.57	68.64.21.62	UDP	1168	53807 → 1853 Len=1126
2	0.000050	172.16.133.57	68.64.21.62	UDP	1168	53807 → 1853 Len=1126
3	0.000050	172.16.133.57	68.64.21.62	ADwin …	94	
4	0.000322	96.43.146.176	172.16.133.82	TCP	60	443 → 61228 [ACK] Seq=1 Ack=1 Win=9659 Len=0
5	0.001160	172.16.133.56	68.64.21.42	UDP	167	49514 → 1853 Len=125
6	0.001306	68.64.21.62	172.16.133.57	UDP	67	1853 → 53807 Len=25
7	0.001307	96.43.146.176	172.16.133.82	TCP	60	443 → 61228 [ACK] Seq=1 Ack=1107 Win=10765 Len=0
8	0.005263	96.43.146.176	172.16.133.82	TCP	60	443 → 60073 [ACK] Seq=1 Ack=1 Win=65535 Len=0
9	0.005988	172.16.133.49	68.64.21.41	UDP	167	58246 → 1853 Len=125

شکل ۲-۱: اطلاعات ذخیره شده در فایل pcap

شکل ۲-۱ داده‌های موجود در یک فایل pcap، که توسط نرم‌افزار wireshark تهیه شده است را نشان می‌دهد. لازم به ذکر است که در این پروژه، فایل‌های pcap به عنوان ورودی مورد استفاده قرار خواهند گرفت.

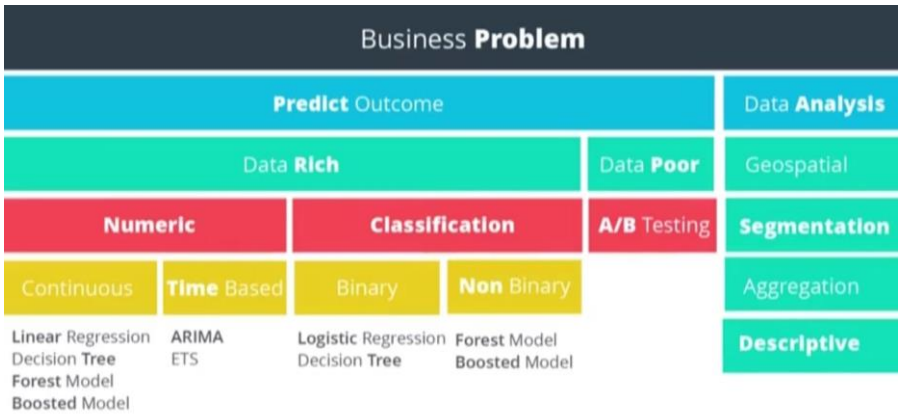
در کنار نرم‌افزارهای موجود، ابزارهای جامعی مانند پروتکل‌های[4] مدیریت و نظارت بر شبکه‌های کامپیوتری ارائه شده است که SNMP یکی از این موارد است. پروتکل‌های پایش شبکه از راه دور[5] (RMON) و پروتکل اطلاعاتی مدیریت مشترک[6] (CMIP) موارد دیگر این ابزارها هستند [۲]. شرکت سیسکو[7] نیز در سال ۱۹۹۶ یک ویژگی به نام نت‌فلو[8] در مسیریاب‌های[9] خود معرفی کرد که وظیفه‌ی آن جمع‌آوری بسته‌های جاری در رابط‌های[10] شبکه است.

در این پروژه از مفاهیم سری زمانی استفاده شده است که در بخش بعدی به طور کامل توضیح داده خواهد شد. ابزارهایی تحت این مفاهیم نیز وجود دارد که به کاربر امکان جمع‌آوری و ذخیره‌سازی داده‌های جاری، نه تنها در

---

[1] Long Short-Term Memory
[2] Deep Learning
[3] Intenet of Things
[4] Protocols
[5] Remote Network Monitoring
[6] Common Management Information Protocol
[7] Cisco
[8] NetFlow
[9] Routers
[10] Interfaces

شبکه‌های کامپیوتری، بلکه در تمامی اجزای سیستم را می‌دهد. به طور مثال، کاربر می‌تواند داده‌های مربوط به پردازنده‌ی سیستم خود را با استفاده از نمودارهای سری زمانی مشاهده کند. ابزارهایی مانند پرومتئوس[1] از این دسته هستند که از پایگاه داده‌های سری زمانی جهت ذخیره‌ی داده‌ی خود استفاده می‌کنند. ابزارهایی نیز مانند گرافانا[2] وجود دارند که به مصور سازی هرچه بهتر داده‌ها با نمودارها کمک می‌کنند. همچنین پایگاه داده‌های سری زمانی مانند اینفلاکس‌دی‌بی[3] توسعه داده شده‌اند تا در صورت بهره‌گیری از الگوریتم‌های سری زمانی، بتوان داده‌ها را در آن‌ها ذخیره کرد. التریکس[4]، یکی از شرکت‌های فعال در زمینه‌ی تولید نرم‌افزار، محصولاتی تولید می‌کند که در علوم داده و تحلیل[5] استفاده می‌شوند. یکی از نرم‌افزارهای این شرکت به همین نام، تمامی امکانات لازم جهت تحلیل، استخراج مدل و در نهایت دریافت خروجی را برای سری‌های زمانی به کاربر، به صورت رابط گرافیکی[6] (GUI)، ارائه می‌دهد.



شکل ۲-۲: کاربردها و الگوریتم‌های استفاده شده در سری زمانی [۱۱]

شکل ۲-۲ کاربردهای سری زمانی و حالت‌های مختلف برای هر الگوریتم سری زمانی را نشان می‌دهد. به طور کلی سری زمانی برای تحلیل آن‌چه که هست و یا پیش‌بینی آن‌چه که رخ خواهد داد استفاده می‌شود. به عنوان مثال مدل جنگل[7] برای دسته‌بندی داده‌های غیرعددی و همچنین برای پیش‌بینی داده‌های عددی استفاده می‌شود.

۲-۳ معرفی سری‌های زمانی

به طور کلی مقادیری از داده که وابسته به زمان هستند و می‌توان طی گذر زمان آن‌ها را جمع‌آوری کرد، تشکیل یک سری زمانی می‌دهند. داده‌های سری زمانی می‌توانند متعلق به تغییرات دما، تغییرات جمعیت یک منطقه، تغییرات

[1] Prometheus
[2] Grafana
[3] InfluxDB
[4] Alteryx
[5] Data Science & Analytics
[6] Graphical User Interface
[7] Forest Model

ارزش سهام در بازار بورس و یا داده‌های جمع‌آوری شده از یک CN باشند پس این داده‌ها، بیانگر تغییرات ایجاد شده در یک پدیده در طول زمان را منعکس می‌کنند.

به دلیل وابسته بودن داده‌های TS به زمان، می‌توان یک بردار مانند X در نظر گرفت و سری زمانی را به صورت زیر معرفی کرد [۳]:

$$X(t), t = 0, 1, 2, \dots$$

در این عبارت t بیانگر زمان و X یک متغیر تصادفی است. همان طور که در عبارت بیان شده است، زمان صفر نیز قابل استفاده است. این زمان، می‌تواند لحظه‌ی شروع یک پدیده و یا لحظه‌ی شروع جمع‌آوری داده‌های یک پدیده‌ی در جریان باشد [۳]. وابسته بودن داده‌های TS به زمان و تغییراتی که منعکس می‌کنند اهمیت ترتیب را در آن‌ها نشان می‌دهد. اگر در هر یک از مراحل جمع‌آوری، تحلیل و پیش‌بینی ترتیب داده‌ها دستخوش تغییر شود، نتایج به دست آمده قابلیت اعتماد ندارند.

شکل ۲-۳: نمونه‌ای از نمودار سری زمانی [۹]

شکل ۲-۱ نشان‌دهنده‌ی نمودار یک سری زمانی استخراج شده از داده‌های یک بازار سهام است. در نهایت می‌توان با استفاده از ابزارهای ترسیم مختلف نمودار، مقادیر داده‌های سری زمانی را به طور پیوسته در زمان ترسیم کرد.

۲-۳-۱ مدل‌های جمع‌آوری داده در سری زمانی

در جمع‌آوری داده‌های TS اگر فقط از یک ویژگی پدیده‌ی موردنظر استفاده شود، متغیر X در عبارت بیان شده یک‌بعدی بوده و مدل سری زمانی را یک متغیره[۱] می‌نامند. ولی اگر از چندین ویژگی برای جمع‌آوری داده استفاده شود، به مدل سری زمانی چند متغیره[۲] گویند. وابستگی داده‌های TS به زمان امری اساسی است. ولی اگر در کنار ویژگی متغیر بودن با زمان تغییرات مکان و مختصات داده‌ها نیز لحاظ شود، مباحث مورد بحث وارد علم آمار فضایی[۳] می‌شوند.

معمولا مرحله‌ی جمع‌آوری داده‌های TS بدون توقف و به صورت پیوسته انجام می‌شود که به آن زمان-پیوسته[۴] گویند؛ در غیر این صورت مدل جمع‌آوری داده را زمان-گسسته[۵] می‌نامند. از مثال‌های نام برده شده در بخش قبل، تغییرات جمعیت یک منطقه مثالی از مدل زمان-گسسته و تغییرات دما مثالی از مدل زمان-پیوسته هستند. در

[1] Univariate
[2] Multivariate
[3] Spacial Statistics
[4] Continuous Time
[5] Discrete Time

تحلیل‌های انجام شده معمولا از روش جمع‌آوری زمان-گسسته انجام می‌شود که مقاطع مشخصی از زمان برای آن‌ها در نظر گرفته می‌شود. به طور مثال جمع‌آوری داده‌ها به صورت بازه‌های ساعتی، روزانه، هفتگی، ماهانه و سالانه انجام می‌شود. در نهایت سری‌های زمان-پیوسته قابلیت تبدیل شدن به سری‌های زمان-گسسته را دارند.

در این پروژه مدل جمع‌آوری داده‌های شبکه به صورت یک‌بعدی و زمان-گسسته است. همچنین داده‌های شبکه فقط از نظر تغییرات زمانی قابل تحلیل و بررسی هستند و مباحث علم آمار فضایی در این پایان‌نامه بررسی نمی‌شوند.

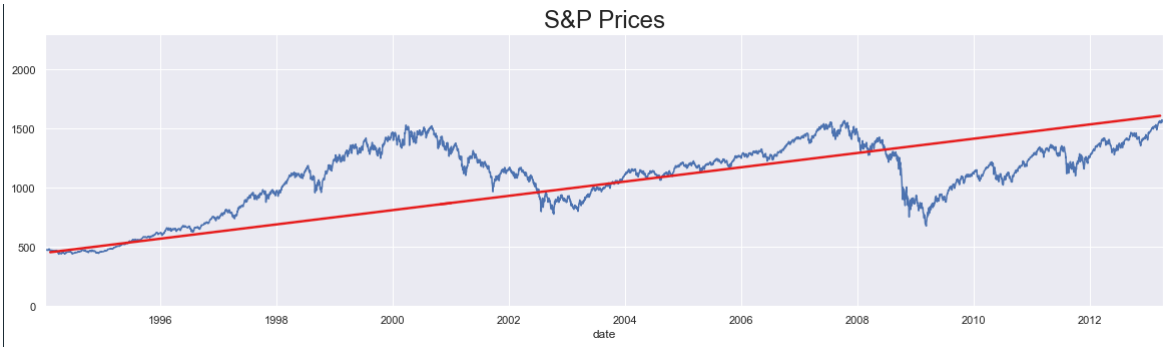
۲-۴ مولفه‌های رفتاری سری زمانی

داده‌های بررسی شده در سری زمانی، داده‌های پیوسته در زمان هستند که با گذر زمان رفتار متفاوتی از خود نشان می‌دهند. همیشه برای شروع تحلیل داده‌های سری زمانی، ابتدا باید ویژگی‌های مشخصی را از رفتار آن‌ها استخراج کرد تا بتوان با توجه به ویژگی‌های رفتاری داده‌ها بهترین مدل تحلیل را انتخاب و در نتیجه دقیق‌ترین پیش‌بینی را ارائه کرد.

در این پروژه، شناسایی این ویژگی‌ها کمک شایانی به استخراج خروجی موردنظر، یعنی بازه‌های زمانی می‌کند. این ویژگی‌ها به طور کلی نشان‌دهنده‌ی روند[1] کلی داده‌ها به صورت صعودی یا نزولی طی گذر زمان، وجود تکرار در رفتار این روند و خطاهای احتمالی هستند که با استفاده از ابزارهای ترسیم نمودار، می‌توان آن‌ها را به صورت مصور نشان داد.

۲-۴-۱ روند (Trend)

اگر داده‌های یک سری زمانی در یک بازه‌ی مشخص از زمان به طور کلی صعودی و یا به طور کلی نزولی باشند، دارای روند مشخص هستند. روند را می‌توان با رسم نمودار سری زمانی تشخیص داد؛ به این صورت که در نمودار رسم شده، اگر نقطه‌ی ابتدایی و انتهایی نمودار سری زمانی به هم وصل شوند، شیب خط به‌دست‌آمده نشان‌دهنده‌ی روند کلی داده‌ها است. شکل ۲-۲ نمونه‌ای از نمودار سری زمانی را به همراه روال کلی آن که با خط قرمز نشان داده شده، نشان می‌دهد.



شکل ۲-۴: نمونه‌ای از روند سری زمانی [۹]

[1] Trend

باید به این نکته توجه داشت که روند دائمی نیست، و هنگامی که افزایش یا کاهش ممتد در داده‌ها رویت می‌شود می‌توان وجود روند را اعلام کرد. به طور کلی می‌توان گفت روند، تغییرات بلندمدت مقادیر سری زمانی را نشان می‌دهد. اگر داده‌ها در یک بازه‌ی زمانی مشخص تغییرات افزایشی و کاهشی نداشته باشند روند وجود ندارد.

علاوه بر ترسیم نمودار زمانی، توابع دیگری در تحلیل سری زمانی وجود دارند که نمودار روند را به طور جداگانه ترسیم می‌کنند. در ادامه به توضیح این توابع خواهیم پرداخت.

۲-۴-۲ فصلی‌بودن (Seasonality)

در مدت زمان یک سال، ۴ فصل با ویژگی‌های منحصر به فرد وجود دارد. زمان شروع و پایان هر فصل از پیش مشخص و تعیین شده است و با فرارسیدن هر کدام، تغییراتی در روند چرخش زمین و خورشید، دما و چرخه‌ی طبیعت دیده می‌شود که به طور کلی این تغییرات در هر سال یکنواخت و مشخص هستند. هیچ‌گاه دیده نمی‌شود که در اواسط سال شاهد سرد و زمستانی شدن هوا باشیم.

ویژگی فصلی بودن در داده‌های سری زمانی، مانند فصول سال، موعد شروع و پایان مشخصی دارد و فاصله‌ی بین هر رخداد آن ثابت و مشخص است. در طول یک فصل مشخص، تغییرات یکسانی بر داده‌های سری زمانی اعمال می‌شود. به طور مثال، اگر ویژگی فصلی بودن داده‌ها را به صورت سالیانه در نظر بگیریم و تغییرات سه ماه ابتدایی سال دارای خاصیت فصلی باشند، در سه ماه ابتدایی سال بعد نیز شاهد همان تغییرات خواهیم بود.

تغییرات فصلی محدود به سال نیستند. فصول تعریف شده می‌توانند محدود به ماه، هفته، روز و حتی ساعت باشند. به عنوان مثال اگر سری زمانی تغییرات میزان خرید کارمندان را در نظر بگیریم، با شروع هر ماه کارمندان حقوق خود را دریافت می‌کنند و ۱۰ روز ابتدایی هر ماه به انجام خریدهای مشخصی مشغول هستند. پس تغییرات مشخصی در ۱۰ روز ابتدایی هر ماه بر داده‌ها اعمال شده است.

نمودار خاصیت فصلی بودن را با استفاده از تابع مخصوصی که در تحلیل سری زمانی ارائه شده می‌توان استخراج کرد که خروجی آن شامل نمودار روند، نمودار فصلی و باقی‌مانده‌هاست که در بخش بعد توضیح داده شده است. شکل ۲-۳ نمونه‌ای از خروجی این تابع را برای قسمت فصلی نشان می‌دهد.

شکل ۲-۵: خروجی نمودار تحلیل فصلی

۲-۴-۳ الگوهای تناوب (Cyclic Patterns)

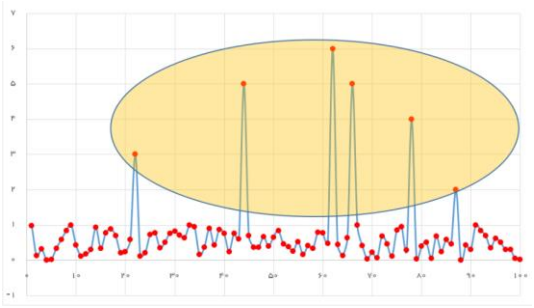
تغییرات چرخه‌ای که در بازه‌های مختلف و نامعلوم در روند داده‌ها دیده می‌شود، الگوهای تناوبی نام دارند. بر خلاف الگوهای فصلی که در زمان مشخص و در بازه‌های تعیین شده امکان حضور داشتند، تغییرات تناوبی در بازه‌های

نامشخص و طولانی مدت رخ می‌دهند که معمولا این بازه‌ها بیشتر از ۲ سال است. به عنوان مثال چرخه‌ی ۴ مرحله‌ای کسب‌وکار که در طی ۳ سال رخ می‌دهد، باعث تکرار شدن روند داده‌های یک سازمان می‌شود و دارای خاصیت تناوبی است [۳]. فاصله‌ی میان هر دو رخداد تناوبی نیز از قبل مشخص نبوده و متغیر است.

۲-۴-۴ خطاها (Errors)

در بعضی سری‌های زمانی ممکن است قسمتی از داده‌ها نه روند خاصی داشته باشند و نه دارای خاصیت تکرارشونده‌ای مانند تناوب و فصلی بودن باشند. در تحلیل ابتدایی سری زمانی به این قسمت از داده‌ها، داده‌های خطا گویند زیرا توسط هیچ الگویی شناسایی نمی‌شوند. در برخی منابع نیز از این نوع داده‌ها به نام تغییرات نامعمول[1] نام برده شده است.

این داده‌ها، که معمولا به صورت فراز شدید[2] و یا سقوط شدید[3] در نمودار سری زمانی قابل رؤیت هستند، باید در مراحل ابتدایی تحلیل شناسایی و حذف شوند. زیرا خطاها هم‌بستگی داده‌ها را این برده و انجام اعمال بیشتر مانند پیش‌بینی سری زمانی توسط این نوع داده‌ها دچار اختلال می‌شود و نتایج گمراه‌کننده‌ای را تولید می‌کند.



شکل ۲-۶: نمونه‌ی داده‌های خطا در سری زمانی [۳]

در شکل ۲-۵ مقادیری که در ناحیه‌ی زردرنگ وجود دارند داده‌های خطا محسوب می‌شوند.

۲-۴-۵ نمودارهای تابع خودهمبستگی[4] (ACF) و خودهمبستگی جزئی[5] (PACF)

جهت افزایش سادگی در پروسه‌ی تحلیل سری زمانی مقادیری از سری به عنوان نماینده انتخاب می‌شوند که به آن‌ها تأخیر[6] گویند. فاصله‌ی میان تأخیرهای انتخاب شده یکسان و مشخص است. به طور مثال می‌توان از مقدار صفر شروع کرده و با فاصله‌ی زمانی بسیار اندکی از سری زمانی نمونه‌برداری کرد. در شکل ۲-۵ چهار سری زمانی قرار دارند که به جز تأخیر شروع، باقی تأخیرها با فاصله‌ی یک روزه از یکدیگر استخراج شده‌اند. موضوعی که در تحلیل سری‌های زمانی اهمیت دارد، بررسی میزان تغیرات میان این تأخیرها است. اگر سری زمانی موردنظر با سری زمانی دیگری

[1] Irregular Changes
[2] Spike
[3] Downfall
[4] Auto Correlation Function
[5] Partial Auto Correlation Function
[6] Lag

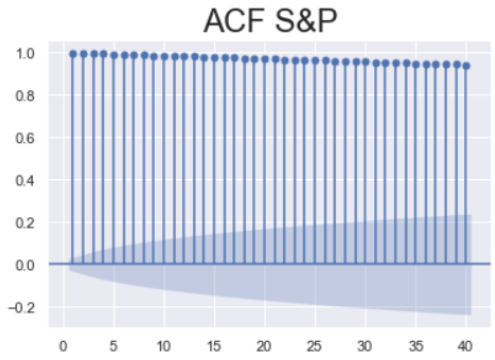
مقایسه شود به آن بررسی میزان همبستگی گویند و اگر مقادیر یک سری زمانی با خودش مقایسه شود به آن بررسی میزان خودهمبستگی گویند.

nikkei	ftse	dax	spx	date	1
18124.01	3445.98	2224.95	469.9	7/1/1994	2
18443.44	3440.58	2225	475.27	10/1/1994	3
18485.25	3413.77	2228.1	474.13	11/1/1994	4
18793.88	3372.02	2182.06	474.17	12/1/1994	5
18577.26	3360.01	2142.37	472.47	13/01/199	6
18973.7	3400.56	2151.05	474.91	14/01/199	7
18725.37	3407.83	2115.56	473.3	17/01/199	8
18514.55	3437.01	2130.35	474.25	18/01/199	9
19039.4	3475.15	2132.52	474.3	19/01/199	10

شکل ۲-۷: تأخیرهای جمع‌آوری شده از چهار سری زمانی [۹]

خودهمبستگی یک سری زمانی به معنی وجود همبستگی میان مقدار هر تأخیر با مقادیر تأخیرهای قبلی همان سری زمانی است. به عنوان مثال اگر x_n مقدار یک تأخیر در سری زمانی باشد میزان همبستگی آن با مقدار x_{n-1} با یک ضریب عددی مشخص می‌شود که این ضریب می‌تواند مقادیر مثبت و منفی میان صفر و یک داشته باشد. اگر ضریب همبستگی بین دو تأخیر از سری زمانی برابر عدد یک باشد به این معنی است که تغییری در مقدار جدید حاصل نشده و میزان همبستگی صد درصد است. ولی اگر تغییرات روی داده‌ها اعمال شده باشد ضریب همبستگی کمتر از عدد یک است. جهت استخراج ضرایب خودهمبستگی در یک سری زمانی، ابتدا تأخیرها به دست می‌آیند سپس مقادیر تأخیرها یک به یک بررسی شده و پس از هر بررسی، به میزان یک واحد تأخیر سری زمانی جابه‌جا[1] می‌شود تا اختلاف مقدار هر تأخیر با مقادیر دیگر به‌دست آید.

همان طور که از تعریف بیان شده برداشت می‌شود، برای تعیین میزان خودهمبستگی یک سری زمانی باید مقادیر تأخیرهای آن را مقادیر تأخیرهای قبلی در همان سری زمانی مقایسه کرد. تابع خودهمبستگی یا ACF سری زمانی به همراه تعداد تأخیرهای موردنیاز جهت بررسی را به عنوان ورودی دریافت کرده و نمودار ACF را رسم می‌کند.



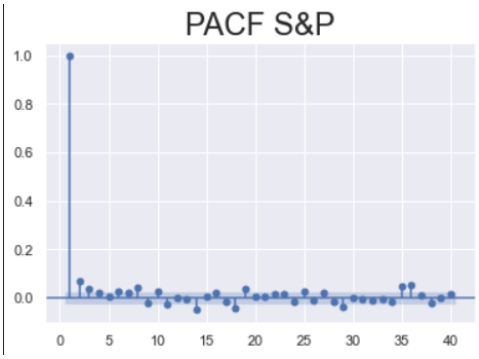
شکل ۲-۸: نمودار ACF [۹]

ماهیت نمودار ACF به صورت نمودار میله‌ای است که هر میله‌ی آن نشان‌دهنده‌ی مقدار یک تأخیر سری زمانی است. در این نمودار، محور افقی نشان‌دهنده‌ی شماره و تعداد تأخیرها و محور عمودی نشان‌دهنده‌ی ضریب همبستگی

[1] Shifts

است. همیشه ضریب همبستگی تأخیر صفر برابر یک است زیرا مقدارش با خودش مقایسه شده و به دلیل عدم وجود تفاوت همبستگی کامل وجود دارد. در نمودار ACF یک قسمت تیره روی محور افقی نمایش داده می‌شود که به آن فاصله‌ی اطمینان[1] گویند. اگر ضریب همبستگی یک تأخیر در این ناحیه قرار گیرد، نشان‌دهنده‌ی ضریب همبستگی نزدیک به صفر است که بیان می‌کند همبستگی میان مقادیر آن تأخیر و مقادیر اطراف آن وجود ندارد. در نقطه‌ی مقابل، اگر مقدار ضریب همبستگی خارج از این محدوده باشد، میزان همبستگی مقادیر سری زمانی در آن تأخیر از نظر آماری قابل توجه است.

نمودار PACF که برای تحلیل‌های پیش‌بینی استفاده می‌شود، خروجی کاملا مشابهی از نظر ظاهری با نمودار ACF دارد. یکی از تفاوت‌های نمودار ACF و PACF این است که در ACF مقادیر تأخیرها از خود سری زمانی برداشته می‌شوند ولی در PACF مقادیر تأخیرها از اختلاف مقادیر پیش‌بینی شده[2] و مقادیر اصلی رؤیت شده[3] در سری زمانی برداشت می‌شوند که به این اختلاف باقی‌مانده[4] می‌گویند.



شکل ۲-۹: نمودار PACF [۹]

از کاربردهای نمودارهای معرفی شده می‌توان به مواردی مثل پیش‌بینی سری زمانی، تشخیص اختلال سفید[5] (WN)، تعیین ایستایی[6] سری زمانی و شناسایی پیاده‌روی تصادفی[7] (RW) اشاره کرد که مفاهیم آن‌ها در ادامه بررسی شده‌اند.

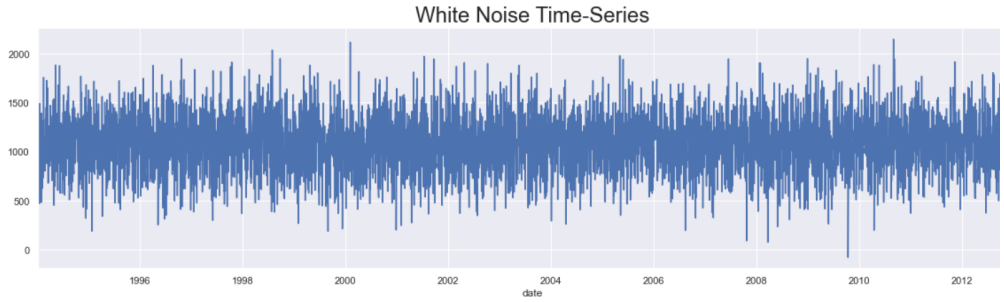
۲-۴-۶ اختلال سفید (White Noise)

یکی از انواع سری‌های زمانی اختلال سفید است. میان مقادیر داده‌های این نوع سری زمانی هیچ‌گونه همبستگی وجود ندارد در نتیجه قابل پیش‌بینی نیست. داده‌های WN به زمان وابستگی نداشته و دارای میانگین صفر و واریانس ثابت (σ^2) هستند [۹، ۱۰، ۱۵]. در صورت تشخیص WN در یک سری زمانی، نمی‌توان تحلیل و بررسی بیشتری انجام داد. شکل ۲-۷ نمونه‌ای از سری زمانی WN را نشان می‌دهد. از دیگر ویژگی‌های WN می‌توان به عدم حضور روند در

[1] Confidence Interval
[2] Forecasted Values – Fitted Values
[3] Observed Values
[4] Residual
[5] White Noise
[6] Stationarity
[7] Random Walk

آن اشاره کرد. همچنین اگر در یک سری زمانی طولانی داده‌های خطا را جدا کرده و با آن‌ها سری زمانی جدیدی تشکیل دهیم، سری حاصل WN است.

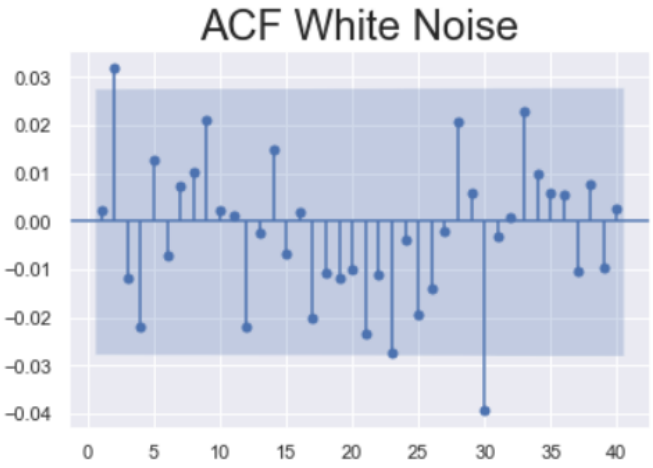
راه‌های مختلفی جهت تشخیص WN بودن یک سری زمانی وجود دارد. به طور مثال می‌توان میانگین را محاسبه کرد که در صورت صفر بودن می‌توان گفت سری زمانی موردنظر WN است.



شکل ۲-۱۰: نمونه‌ای از white noise [۹]

همچنین در صورت محاسبه‌ی واریانس داده‌ها و بررسی تغییرات آن با گذشت زمان، در صورت ثابت بودن می‌توان ادعا داشت که سری زمانی موردنظر WN است. یکی دیگر از ساده‌ترین روش‌های تشخیص WN، رسم نمودار سری زمانی است. در صورت وجود تغییرات در روند سری زمانی، WN نیست و برعکس.

یکی از روش‌های قابل استناد در تشخیص WN، استفاده از نمودار ACF است. در قسمت قبل بیان شد که در صورت قرار گرفتن مقادیر ضریب همبستگی در فاصله‌ی اطمینان، ارتباط و همبستگی میان تأخیرها وجود ندارد و می‌توان گفت که سری زمانی موردنظر تصادفی است. این ویژگی‌ها معرف WN نیز هستند.



شکل ۲-۱۱: نمودار ACF برای سری زمانی White Noise [۹]

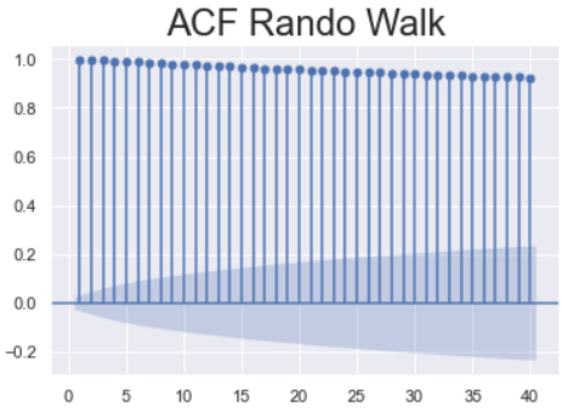
همان طور که در شکل ۲-۸ مشخص شده است، به جز تأخیرهای ۲ و ۳۰ باقی ضرایب همبستگی در فاصله‌ی اطمینان قرار دارند که نشان‌دهنده‌ی عدم وجود همبستگی میان مقادیر این سری زمانی و در نتیجه تصادفی بودن آن است. با توجه به توضیحات ارائه شده، با یک تابع مولد اعداد تصادفی می‌توان یک سری زمانی WN را تولید کرد.

۲-۴-۷ پیاده‌روی تصادفی (Random Walk)

پیاده‌روی تصادفی یا Random Walk نیز نوع دیگری از سری‌های زمانی است. رفتار آینده‌ی این نوع سری زمانی، مانند WN، قابل پیش‌بینی نیست زیرا همان طور که از نام این سری زمانی پیداست، مقادیر آن به صورت تصادفی تولید می‌شوند. تفاوتی که این نوع سری زمانی با WN دارد، در روش تولید مقادیر آن است. در WN هر مقدار تولیدی در سری زمانی بدون وابستگی به زمان و داده‌های دیگر سری است. ولی در پیاده‌روی تصادفی هر مقدار با توجه به مقدار قبلی تولید می‌شود؛ به این صورت که به مقدار کنونی یک مقدار تصادفی اضافه شده و مقدار بعدی تولید می‌شود. می‌توان با شروع از صفر و یک تابع تولیدکننده‌ی عدد رندوم یک سری زمانی RW تولید کرد.

شکل ۲-۱۲: نمودار یک سری زمانی RW در کنار یک سری زمانی نرمال [۹]

برخلاف WN که رسم نمودار یکی از روش‌های تشخیص آن بود، با رسم نمودار RW نمی‌توان آن را تشخیص داد زیرا همان طور که در شکل ۲-۹ نشان داده شده است، نمودار حاصل کاملا شبیه به یک نمودار سری زمانی نرمال است. در این شکل، روند نارنجی‌رنگ نشان‌دهنده‌ی سری زمانی RW و روند آبی‌رنگ نشان‌دهنده‌ی سری زمانی نرمال است. باید توجه داشت که میانگین سری زمانی RW صفر نیست [۱۵]. اگر از نمودار ACF نیز برای تشخیص یک سری زمانی RW استفاده کنیم، نتیجه تفاوتی با یک سری زمانی نرمال ندارد.

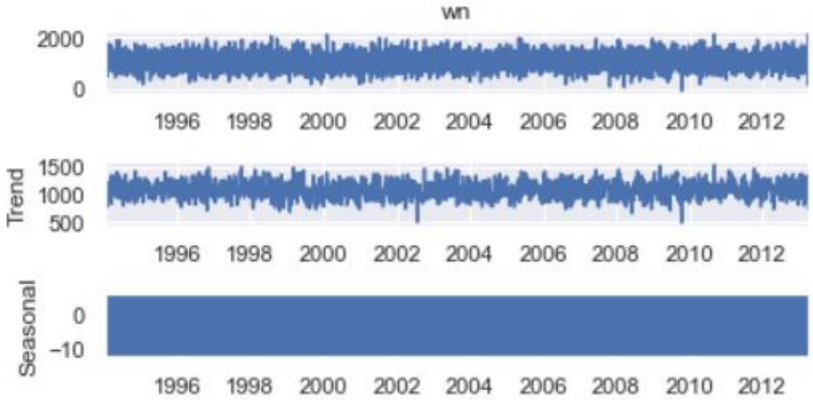


شکل ۲-۱۳: نمودار ACF برای سری زمانی RW [۹]

اگر شکل ۲-۵ را با شکل ۲-۱۰ مقایسه کنیم، تفاوت چندانی مشاهده نمی‌شود. یکی از راه‌های تشخیص سری زمانی RW ایستا کردن سری است که در ادامه به توضیح آن پرداخته شده است.

۲-۴-۸ ایستایی (Stationarity)

ایستایی در سری‌های زمانی به معنی عدم وجود تغییرات در کلیت مقادیر داده‌ها در یک بازه‌ی زمانی مشخص است. این مفهوم به معنی عدم تغییر مقدر نیست. به طول مثال white noise طی زمان در حال تولید مقادیر کاملا تصادفی است ولی نمونه‌ای از سری زمانی ایستا است. سری‌های زمانی ایستا دارای ویژگی‌های مستقل از زمان هستند. میانگین، و واریانس در آن‌ها ثابت است و کوواریانس نیز با گذر زمان دچار تغییر نمی‌شود. از طرفی، وجود خاصیت فصلی و یا وجود روند مشخص در سری زمانی نشان‌دهنده‌ی وابستگی مقادیر به زمان است. در نتیجه سری‌های زمانی ایستا روند مشخص و خاصیت فصلی ندارند. عکس این موضوع نیز صادق است؛ به این صورت که سری‌های زمانی که دارای خاصیت فصلی و یا روند هستند ایستا نیستند. باید به این نکته توجه داشت که سری‌های زمانی ایستا می‌توانند خاصیت تناوبی داشته باشند.



شکل ۲-۱۴: نمودارهای white noise در تحلیل ایستایی [۹]

شکل ۲-۷ نمودارهای تحلیل یک سری زمانی white noise را نشان می‌دهد که دارای روند مشخص و خاصیت فصلی نیست. می‌توان گفت که روند آن به صورت خط افقی است.

روش‌های متعددی برای تشخیص ایستایی یک سری زمانی وجود دارد که روش‌های اصلی شامل رسم نمودار سری زمانی، استفاده از نمودار ACF است. همچنین می‌توان به صورت غیرخودکار به محاسبه‌ی میانگین و واریانس سری زمانی در بازه‌های مختلف پرداخت تا در صورت ثابت بودن ایستایی سری زمانی اعلام شود. ولی این روش از نظر بازده غیرقابل استفاده است. اگر در نمودارهای سری زمانی روند و خاصیت فصلی دیده شد، سری زمانی ایستا نیست. همچنین در نمودار ACF اگر مقادیر تأخیرها در ناحیه‌ی اطمینان حضور داشته باشند، سری زمانی ایستا است. در غیر این صورت، اگر مقادیر تأخیرها خارج از ناحیه‌ی اطمینان باشند سری زمانی ایستا نیست.

روش‌های آماری دیگری نیز برای تشخیص ایستایی سری‌های زمانی وجود دارد. روش دیکی-فولر[1] یکی از این روش‌ها است که جز دسته‌ی آزمون‌های ریشه‌ی واحد[2] محسوب می‌شود. این دسته آزمون‌ها تعیین می‌کنند که یک سری زمانی تا چه حد به یک روند وابسته است. این دسته آزمون‌ها روش‌ها و گرایش‌های زیادی دارند که روش دیکی-

[1] Dickey–Fuller Test

[2] Unit Root Test

فولر یکی از آن‌ها است. این تست سری زمانی را به عنوان ورودی دریافت می‌کند و خروجی خود را تولید می‌کند. شکل ۲-۸ یک خروجی نمونه از تست دیکی-فولر است. در خروجی این تست، درصدهایی به عنوان معیار نشان داده می‌شوند که در شکل ۲-۸ در سطرهای پنج الی هفت خروجی قرار دارند. سطر اول خروجی باید با این معیارها مقایسه شود تا درصدی که به آن اندازه احتمال می‌رود داده‌های سری زمانی ایستا باشند شناسایی شود. در شکل ۲-۸ عدد سطر اول منفی یک است که از مقدار معیارهای درصدهای داده شده بیشتر است. همچنین در سطر دوم عدد به دست آمده حدوداً چهل‌ویک صدم است که از عدد معیار یعنی پنج‌صدم بیشتر است. با توجه به نتایج به دست آمده می‌توان نتیجه گرفت که سری زمانی داده شده ایستا نیست.

```
sts.adfuller(df.market_value)
✓ 1.1s
(-1.7369847452352478,
 0.41216456967706006,
 18,
 5002,
 {'1%': -3.431658008603046,
  '5%': -2.862117998412982,
  '10%': -2.567077669247375},
 39904.880607487445)
```

شکل ۲-۱۵: نتایج تست دیکی-فولر [۹]

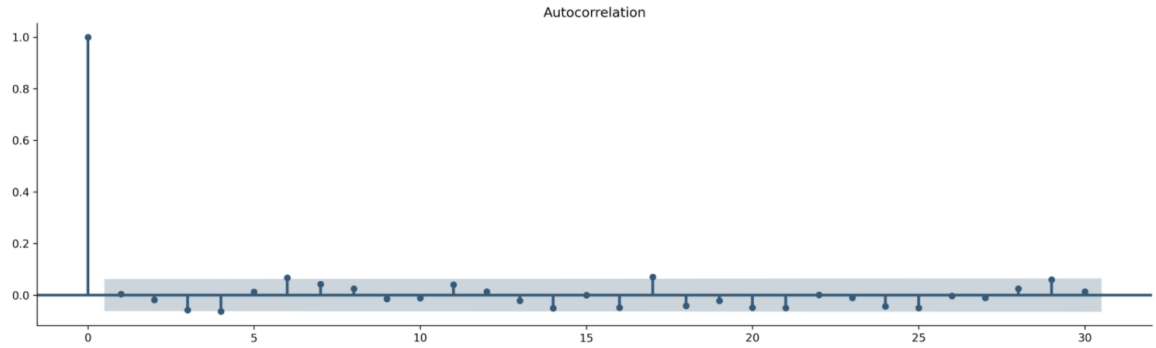
این نکته نیز حائز اهمیت است که روش‌های پیش‌بینی سنتی[1] سری زمانی قادر به استفاده از سری زمانی غیرایستا نیستند. سری‌های زمانی استفاده شده در این روش‌ها باید بدون روند و خاصیت فصلی باشند زیرا تبدیل کردن آن‌ها به مدل‌های پیش‌بینی سری زمانی آسان است. ولی باید توجه داشت که سری‌های زمانی ایستا که دارای مقادیر کاملا تصادفی طی زمان هستند را نمی‌توان پیش‌بینی کرد زیرا مقادیر آن‌ها هیچ‌گونه وابستگی‌ای به زمان ندارند. White noise مثالی از این دست سری‌ها است. در صورت شناسایی شدن white noise از هرگونه تحلیل آتی باید جلوگیری شود.

در بخش قبل بیان شد که سری زمانی RW، که یک سری تصادفی است، به راحتی قابل شناسایی نیست و در نمی‌توان آن را پیش‌بینی کرد. همچنین دیدیم که نمودار ACF آن یک سری زمانی غیرایستا را نشان می‌دهد. به طور کلی سری RW، برخلاف سری WN که در ابتدای روند تحلیل شناسایی و متوقف می‌شود، در مرحله‌ی پردازش‌های قبل از پیش‌بینی شناسایی و متوقف می‌شود.

یکی از روش‌های ایستا کردن سری‌های زمانی اختلاف از میانگین و روش دیگر تولید تفاوت درجه اول[2] (FOD) است [۱۱،۱۲]. در روش اول میانگین مقادیر تأخیرها محاسبه شده و مقدار میانگین از مقدار هر تأخیر کاسته می‌شود. برای تولید FOD، مقادیر تأخیرهای سری زمانی را دو به دو از یک‌دیگر کم می‌کنیم. به عنوان مثال اگر x_1 و x_2 دو تأخیر پشت‌سرهم در یک سری زمانی باشند، $x_2 - x_1$ مقدار اولین تأخیر سری زمانی جدید را تولید می‌کند. اگر FOD

[1] Classic Approaches
[2] First Order Difference

یک سری زمانی RW محاسبه شود و نمودار آن به همراه نمودار ACF آن رسم شود، خروجی‌ها نشان‌دهنده‌ی یک سری ایستا مانند WN خواهند بود.



شکل ۲-۱۶: نمودار ACF سری زمانی RW [۱۲]

همان طور که در شکل ۲-۱۳ نشان داده شده است، به‌جز موارد معدودی، اکثریت مقادیر تأخیرها در فاصله‌ی اطمینان قرار دارند که نشان‌دهنده‌ی عدم وجود همبستگی میان داده‌های سری زمانی است؛ در نتیجه سری زمانی RW ایستا، غیرقابل پیش‌بینی و تصادفی است.

۲-۵ ارتباط میان ویژگی‌های سری زمانی

هرکدام از ویژگی‌های معرفی شده تا کنون به نحوی در معرفی یک سری زمانی دخالت دارند. دو مدل برای بیان این ارتباط وجود دارد. چهار ویژگی اصلی برای معرفی سری زمانی لازم است که این چهار ویژگی روند، تناوب، فصلی بودن و مؤلفه‌ی تصادفی هستند [۳].

- **مدل ضربی[1]:**

در این مدل، چهار ویژگی معرفی شده به عنوان ویژگی‌های اصلی تعریف کننده‌ی یک سری زمانی در یکدیگر ضرب شده و سری زمانی را می‌سازند. آن دسته از سری‌های زمانی که با گذر زمان رفتار صعودی دارند و نرخ رفتارهای فصلی در آن‌ها زیاد است از این مدل تبعیت می‌کنند [۲۰].

- **مدل جمعی[2]:**

در این مدل هر چهار ویژگی اصلی با یکدیگر جمع شده و سری زمانی را معرفی می‌کنند. این مدل هنگامی رخ می‌دهد که واریانس داده‌های سری زمانی با گذر زمان تغییر نکند. به طور کلی، اگر سری زمانی صعودی باشد و میزان افزایش هر تأخیر با میزان افزایش تأخیر گذشته در تناسب باشد سری زمانی موردنظر از مدل جمعی پیروی می‌کند [۲۰].

[1] Multiplicative Model
[2] Additive Model

مدل‌های معرفی شده، نحوه‌ی ایجاد سری‌های زمانی در اثر تجمیع و یا تقویت ویژگی‌های اساسی آن‌ها را معرفی می‌کنند. اگر هدف پیش‌بینی رفتار سری زمانی باشد باید مدل‌های پیش‌بینی شناخته شوند. روش دیگری براش تشخیص مدل ضربی و جمعی، رسم نمودار باقی‌مانده‌های سری زمانی با استفاده از تابع تجزیه سری زمانی است. این تابع مدل موردنظر را به عنوان ورودی دریافت می‌کند و طبق آن، سری زمانی را به مؤلفه‌های سازنده‌اش تجزیه می‌کند. اگر نمودار باقی‌مانده‌ها از الگوی رفتاری خاصی طی زمان تبعیت کند، مدل سری زمانی بررسی شده جمعی است. در مدل جمعی نمودار باقی‌مانده‌ها معمولا رفتار فصلی دارد. برخلاف مدل جمعی، نمودار باقی‌مانده‌های مدل ضربی هیچ‌گونه الگوی خاصی را دنبال نمی‌کند.

۲–۶ پیش‌بینی در سری زمانی (Time Series Forecasting)

با توجه به مفاهیم ارائه شده درمورد سری‌های زمانی، قبل از شروع هرگونه تحلیل بیشتر، باید ویژگی‌های سری زمانی موردنظر شناسایی شوند. اگر سری زمانی به عنوان WN یا RW شناسایی شد، نمی‌توان سری موردنظر را پیش‌بینی کرد. ولی اگر سری موردبررسی یک سری نرمال بود، باید ایستایی آن بررسی شده تا در صورت ایستا نبودن، با استفاده از روش‌های معرفی شده آن را ایستا کرد. سپس با توجه به ویژگی‌های استخراج شده بهترین مدل را جهت ادامه‌ی فرایند پیش‌بینی انتخاب کرد.

یکی از مفاهیم مورد استفاده در پیش‌بینی سری زمانی باقی‌مانده یا residual است. پس از انتخاب یک مدل پیش‌بینی و به دست آوردن نتایج، به اختلاف میان مقادیر پیش‌بینی شده و مقادیر مشاهده شده در سری زمانی باقی‌مانده گویند. میانگین داده‌های باقی‌مانده باید صفر باشد. همچنین سری زمانی باقی‌مانده‌ها خودهمبستگی ندارد. باید به این نکته توجه داشت که در تحلیل‌های پیش‌بینی سری زمانی، نمودار PACF همبستگی باقی‌مانده‌های سری را ارائه می‌دهد. به طور کلی نمودارهای ACF و PACF ابزارهای پرکاربرد در تحلیل و پیش‌بینی سری‌های زمانی هستند.

روش‌های زیادی برای پیش‌بینی سری‌های زمانی وجود دارد که اکثریت آن‌ها از دقت و درستی کافی برخوردار نیستند. روش میانگین[1] داده‌های آینده را میانگین تمامی داده‌های اتفاق افتاده تا کنون می‌داند. روش میانگین متحرک[2]، میانگین تعداد مشخصی تأخیر را مقدار آینده اعلام می‌کند. هنگامی که داده‌ی کافی برای پیش‌بینی وجود نداشته باشد، روش ساده‌لوحانه[3] مقدار آخرین تأخیر رخ داده را به عنوان مقدار آینده اعلام می‌کند.

۲–۶–۱ مدل‌های پیش‌گویی در سری‌های زمانی

در این قسمت دو مدل کلاسیک و پرکاربرد در پیش‌بینی سری زمانی ارائه شده است:

- **رگرسیون خودکار[4] (AR):**

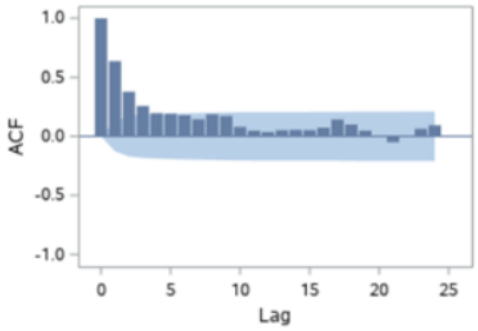
[1] Average
[2] Moving Average
[3] Naive
[4] Auto Regression

درصورتی‌که هر مقدار سری زمانی طبق مقدار تأخیر قبلی به دست آید، آن سری از مدل AR تبعیت می‌کند. به عنوان مثال، هر مقدار جدید میانگین وزن‌دار مقادیر قبلی است [۵، ۷]. باید توجه داشت که سری زمانی در این مدل ممکن است دارای WN باشد. یک فرایند AR از مرتبه‌ی p به صورت زیر معرفی می‌شود [۲، ۵، ۱۷]:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \cdots + \varphi_p y_{t-p} + \varepsilon_t$$

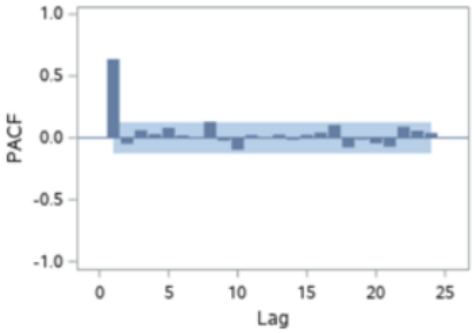
$$\varepsilon_t = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\frac{\tau_1^2}{2\sigma^2}}$$

در اینجا ε_t نشان‌دهنده‌ی WN است و هر y یک تأخیر را مشخص می‌کند. مرتبه‌ی p مقدار تأخیری است که پس از آن نمودار PACF برای اولین بار از حد بالایی فاصله‌ی اطمینان عبور می‌کند [۱۶]. در این مدل نمی‌توان از نمودار ACF بهره برد زیرا در هرصورت همبستگی قابل‌قبولی را نشان می‌دهد [۱۶]. به طور کلی در مدل AR از نمودار ACF یک افت تدریجی انتظار می‌رود ولی در نمودار PACF می‌توان شاهد یک افت ناگهانی پس از p تأخیر بود.



شکل ۲-۱۷: نمودار ACF یک سری زمانی در مدل AR [۱۷]

شکل ۲-۱۶ نشان‌دهنده‌ی نزول تدریجی نمودار ACF در یک سری زمانی مدل AR است.



شکل ۲-۱۸: نمودار PACF یک سری زمانی در مدل AR [۱۷]

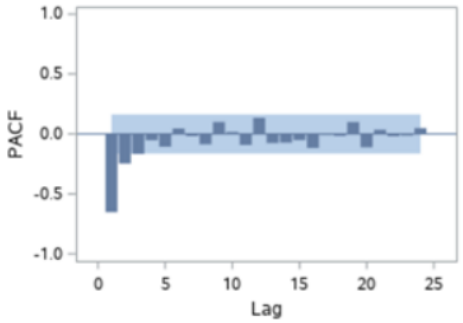
شکل ۲-۱۷ نشان‌دهنده‌ی نزول ناگهانی مقادیر ضرایب همبستگی در نمودار PACF است. قرارگیری تمامی مقادیر در فاصله‌ی اطمینان، نشان‌دهنده‌ی ایستا بودن سری زمانی است.

- **میانگین متحرک:**

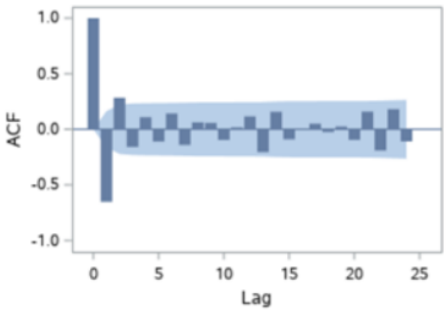
در این مدل، هر مقدار سری زمانی به صورت ترکیبی خطی از داده‌های خطا محاسبه می‌شود. معمولاً فرض می‌شود که داده‌های خطا به صورت مستقل و یکنواخت در طول سری زمانی توزیع شده باشند. یک فرایند MA مرتبه‌ی q به صورت زیر معرفی می‌شود [۵، ۱۶، ۱۷]:

$$y_t = c + \varepsilon_t + \theta_1\varepsilon_{t-1} + \theta_2\varepsilon_{t-2} + \cdots + \theta_q\varepsilon_{t-q}$$

در این عبارت، ε_t نشان‌دهنده‌ی WN است. در این مدل مرتبه‌ی q از نمودار ACF به دست می‌آید. مرتبه‌ی q تأخیری است که پس از آن نمودار ACF برای اولین بار از حد بالایی فاصله‌ی اطمینان عبور می‌کند [۱۶].

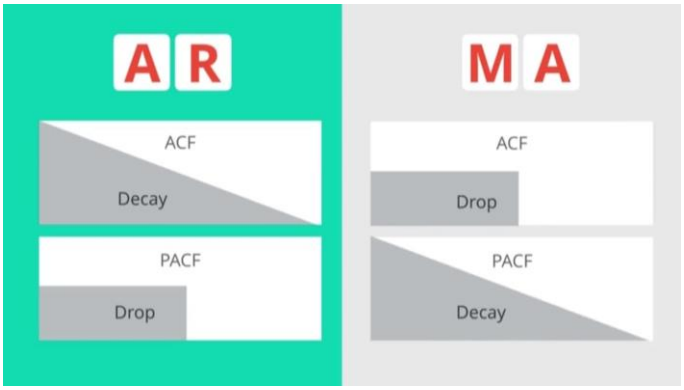


شکل ۲-۱۹: نمودار PACF یک سری زمانی در مدل MA [۱۷]



شکل ۲-۲۰: نمودار ACF یک سری زمانی در مدل MA [۱۷]

شکل ۲-۱۸ و ۲-۱۹ نمودارهای ACF و PACF یک سری زمانی در مدل MA را نشان می‌دهند. در نمودار PACF نزول تدریجی و در نمودار ACF افت ناگهانی مقادیر مشاهده می‌شود که این رفتار، برخلاف رفتار مدل AR است.



شکل ۲-۲۱: تفاوت رفتاری نمودارهای ACF و PACF در مدل‌های AR و MA [۱۱]

مدل‌های دیگری نیز وجود دارد که از ترکیب مفاهیم دو مدل معرفی شده به دست می‌آیند. مدل‌های ARMA و ARIMA از این دسته هستند. این مدل‌ها نه تنها در پیش‌بینی سری‌های زمانی، بلکه در تشخیص ناهنجای و دسته‌بندی داده‌های شبکه نیز کاربرد دارند. در مقالات و پروژه‌های انجام شده، این نتیجه حاصل شده است که مدل AR برای پیش‌بینی رفتار شبکه‌های کامپیوتری مدل بهتری نسبت به باقی مدل‌های سری زمانی است [۲، ۵، ۷]. مدل دیگری به نام خطا، روند، فصلی[1] (ETS) وجود دارد که با استفاده از سه ویژگی سری‌های زمانی یعنی خطا، روند و خاصیت فصلی به پیش‌بینی سری زمانی می‌پردازد [۱۸].

۲-۷ مفاهیم آماری

در این پروژه، جهت انجام پردازش‌های نهایی و رسیدن به خروجی موردنظر، باید با مفاهیم آماری نیز آشنا بود. با بهره‌گیری مستقیم از این مفاهیم و تحلیل‌ها خروجی پروژه حاصل می‌شود. دو مفهومی که در این پروژه استفاده می‌شوند واریانس و به دنبال آن انحراف‌معیار هستند. باید توجه داشت که از مفاهیم دیگری مانند میانگین نیز در این پروژه استفاده می‌شود ولی به دلیل کاربرد روزمره و روشن بودن مفاهیم، از توضیح دادن آن‌ها به تفصیل پرهیز شده است.

۲-۷-۱ واریانس

تعریفی که برای واریانس ارائه می‌شود، توضیح فرمول آن است که عبارت است از مقدار متوسط مربع اختلاف مقادیر از میانگین. درواقع واریانس نمایشی از میزان گستردگی داده‌های یک مجموعه است. روش محاسبه‌ی واریانس به این صورت است که ابتدا میانگین داده‌ها محاسبه می‌شود. سپس مقدار میانگین از هر داده تفریق شده و حاصل به توان دو می‌رسد. در پایان، میانگین مقادیر به‌دست‌آمده محاسبه می‌شود [۱۹].

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2$$

[1] Error, Trend, Seasonal

در فرمول ارائه شده μ معرف میانگین، n معرف تعداد داده‌ها و x_i ها معرف مقادیر داده‌ها هستند. دلیل به توان دو رساندن اختلاف از میانگین این است که ممکن است مجموعه‌ی داده‌ها شامل داده‌ی منفی باشد. در این صورت داده‌های مثبت داده‌های منفی را خنثی می‌کنند. به توان دو رساندن باعث جلوگیری از این اتفاق می‌شود.

۲-۷-۲ انحراف‌معیار

انحراف‌معیار بیان‌کننده‌ی میزان پراکندگی داده‌ها نسبت به میانگین است. برای محاسبه‌ی آن، ابتدا باید واریانس داده‌ها را محاسبه کرد. سپس جذر مقدار حاصل، برابر انحراف‌معیار داده‌ها است [۱۹].

۲-۸ جمع‌بندی

با توجه به گسترش و پیچیده‌تر شدن شبکه‌های کامپیوتری و وجود نیاز به کنترل، محافظت و ارتقاء آن‌ها، ابزارها و مفاهیم مختلفی ارائه شده‌اند تا با رسیدن به اهداف مشخص خود سلامت شبکه‌های کامپیوتری و روند توسعه‌ی آن‌ها را حفظ کنند. علوم مختلف در کنار علم شبکه به‌کار گرفته می‌شوند تا تحلیل‌های مختلف را از دیدگاه‌های مختلف به مدیران و متخصصان شبکه ارائه دهند. نتایج این تحلیل‌ها مسیر رسیدن به اهدافی مثل پیش‌بینی داده‌های شبکه، تشخیص ناهنجاری و دسته‌بندی داده‌های شبکه را هموار می‌کنند.

مفاهیم سری زمانی با استفاده از تحلیل‌های آماری و ترسیم بصری داده‌ها و نتایج، امکان شناسایی الگوهای رفتاری شبکه‌های کامپیوتری و همچنین پیش‌بینی و دسته‌بندی داده‌های جاری در آن‌ها را فراهم می‌کنند. در این پروژه، داده‌های شبکه که در فایل‌های pcap جمع‌آوری شده‌اند به صورت سری زمانی آماده‌سازی شده و با استفاده از مفاهیم معرفی شده تحلیل و بررسی می‌شوند. در نهایت با استفاده از محاسبات آماری، بازه‌های زمانی که هرکدام معرف یک دسته‌ی رفتاری خاص داده‌های شبکه است به عنوان خروجی ارائه می‌شوند.

فصل سوم
روش پیشنهادی

۳–۱ مقدمه

از آنجایی‌که داده‌های شبکه به صورت پیوسته در زمان در جریان هستند،می‌توان با استفاده از مفاهیم ارائه شده درمورد سری‌های زمانی، آن‌ها را مورد تحلیل و بررسی قرار داد. در این فصل ابتدا مسئله‌ی مطرح شده به طور کامل و دقیق شرح داده می‌شود. سپس روش پیشنهادی و گام‌های اصلی آن یعنی آماده‌سازی داده‌ها، تحلیل اولیه و در نهایت پردازش نهایی و دریافت خروجی به ترتیب و با جزئیات بررسی می‌شود.

۳–۲ شرح مسئله

همان‌طور که در بخش‌های گذشته بیان شد، با توجه به گسترش مقیاس کاربرد فناوری شبکه‌های کامپیوتری، نیاز به پایش و نگهداری این شبکه‌ها افزایش یافته است. امروزه ابزارها و پروتکل‌های زیادی جهت فراهم کردن امکانات تحلیلی برای کاربر ارائه شده که امکانات محدودی دارند. به طور مثال نرم‌افزارهای موجود و یا پروتکل‌های رایج پیاده‌سازی شده در بسترهای شبکه‌ای، صرفا به جمع‌آوری داده‌ها و اطلاعات شبکه کفایت کرده و همان داده‌ها را به عنوان نتیجه و بدون پردازش‌های مفهومی به کاربر نمایش می‌دهند. گاهی ابزارهای نام‌برده، از داده‌های جمع‌آوری شده نمودارهایی تهیه کرده و جهت آسان‌تر شدن درک کاربر، به صورت مصور آن‌ها را به نمایش می‌گذارند. در کاربردهای گسترده‌تر، که فقط به شبکه‌های کامپیوتری ختم نمی‌شود، ابزارهای پیاده‌سازی شده از مفاهیم علوم مختلف، مانند علوم آماری بهره می‌گیرند. پایگاه‌داده‌های سری زمانی مثال‌هایی از این قبیل ابزارها هستند. همچنین اگر مدیران شبکه‌های کامپیوتری اهدافی والاتر از پایش شبکه داشته باشند، باید از مفاهیم علوم آماری و هوش مصنوعی استفاده کنند که مدت زمان فراگیری و پیاده‌سازی آن‌ها زیاد است. به طور مثال، پیش‌بینی رفتار شبکه و یا دسته‌بندی داده‌های غیرعددی شبکه از این دسته اهداف هستند. منظور از داده‌های غیرعددی شبکه، داده‌های متعلق به کاربردهای غیرعددی و غیرآماری است. به عنوان مثال داده‌های شبکه‌های اجتماعی، داده‌های بازی‌های کامپیوتری برخط[1] یا داده‌های سرویس‌های چندرسانه‌ای مواردی از داده‌های غیرعددی هستند [۴].

اگر هدف از پایش داده‌های شبکه فقط نگهداری اطلاعات بدون انجام پردازش‌های پیچیده باشد، ابزارهای موجود به نحو احسن نقش خود را در این زمینه ایفا کرده و نیازهای کاربران را براورده می‌کنند. ولی اگر اهدافی مثل پیش‌بینی

[1] Online

رفتار شبکه و یا دسته‌بندی داده‌ها مدنظر باشد، دانستن رفتار کنونی شبکه یکی از مهم‌ترین عوامل است. منظور از رفتار شبکه، الگویی است که داده‌های شبکه در شرایط کاملا عادی طبق آن در جریان هستند. شرایط عادی شرایطی است که داده‌های غیرمعمول در شبکه جریان نداشته باشد. مثلا زمان‌هایی که شبکه تحت حملات مختلف قرار می‌گیرد و یا هرگونه عاملی باعث ایجاد اختلال در داده‌های شبکه می‌شود، شرایط غیرمعمول است. می‌توان بیان کرد که ثبت رفتار شبکه مرحله‌ای میان جمع‌آوری داده‌ها و تحلیل‌های پیچیده مانند پیش‌بینی است، که جمع‌آوری داده‌ها را ابزارهای موجود و تحلیل‌های پیچیده را متخصصان انجام می‌دهند.

در فصل گذشته به طور کامل درمورد سری‌های زمانی صحبت شد و ویژگی‌های آن‌ها مورد بررسی قرار گرفت. مسئله‌ی اصلی‌ای که این پروژه بر آن تمرکز دارد، ارائه‌ی یک روش با بهره‌گیری از مفاهیم آماری و سری زمانی برای تحلیل داده‌های شبکه جهت استخراج الگوی رفتاری آن است. الگوی رفتاری موردنظر در این پروژه شامل بازه‌های زمانی پیوسته‌ای است که به آن‌ها دانه‌ی زمانی[1] گویند. نقاطی که این دانه‌های زمانی از یکدیگر جدا شده‌اند، نشان‌دهنده‌ی نقاطی است که رفتار شبکه از آن نقطه به بعد تغییر کرده است و هر دانه‌ی زمانی دارای ویژگی‌های منحصربه‌فرد خود است. در ادامه، مثالی جهت واضح‌تر شدن هدف پروژه آورده شده است.

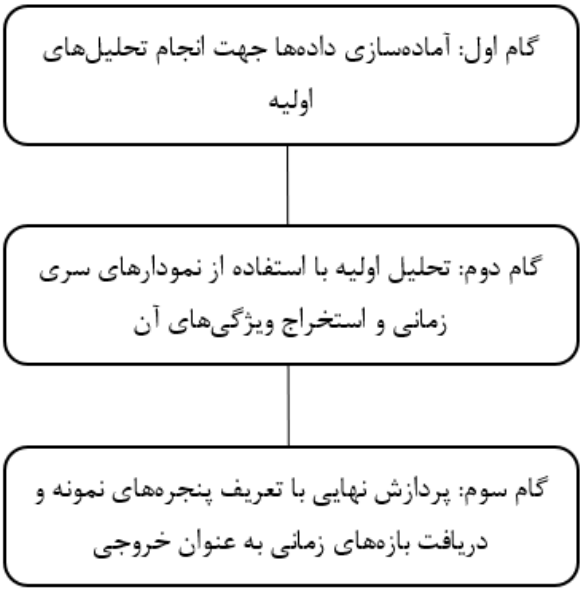
شبکه‌ی سیستم‌های کامپیوتری یک سازمان با شروع ساعت اداری در روز شروع به کار می‌کند و با پایان ساعت اداری تمام فرایندهای خود را متوقف و شروع به انجام پردازش‌های نهایی آن روز و ذخیره‌سازی داده‌ها می‌کند. با فرارسیدن ساعت ۲۰:۰۰ همان روز، هرگونه جریان داده در شبکه متوقف شده و سیستم‌های سازمان مذکور تا شروع ساعت اداری روز بعد بدون فعالیت هستند. از دید ناظر شبکه، میزان جریان داده‌های شبکه‌های این سازمان در طول ساعات اداری بسیار زیاد بوده و پس از ساعت اداری کاهش چشمگیری یافته و در پایان روز به صفر می‌رسد. اگر شروع ساعت اداری را از ساعت هفت صبح در نظر گرفته و پایان ساعت اداری ساعت شانزده بعدازظهر باشد، خروجی موردنظر این پروژه یا همان دانه‌های زمانی در مثال بیان شده باید به این صورت باشند که دانه‌ی زمانی اول مشخص‌کننده‌ی بازه‌ی ساعت هفت صبح الی شانزده بعدازظهر، دانه‌ی زمانی دوم مشخص کننده‌ی بازه‌ی ساعت شانزده بعدازظهر الی هشت شب و دانه‌ی زمانی آخر معرف بازه‌ی ساعت هشت شب الی هفت صبح فردای آن روز است. اگر بازه‌های خروجی به یک روز محدود شوند، می‌توان از ساعت ۰۰:۰۰ هر روز الی هفت صبح، همچنین هشت شب تا ساعت ۰۰:۰۰ را یک دانه‌ی زمانی جدا در نظر گرفت.

خروجی‌های این پروژه به مدیران شبکه الگوی رفتاری داده‌های جاری را در ساعات مشخص شبانه‌روز نشان می‌دهد. هم‌چنین از داده‌های خروجی این پروژه می‌توان جهت توسعه‌ی زیرساخت‌های شبکه در راستای سیستم‌های شناسایی ناهنجاری استفاده کرد. از طرفی می‌توان با استفاده از نتایج به‌دست‌آمده از تحلیل‌های سری زمانی و بهره‌گیری از مدل‌های کلاسیک سری زمانی و یا ابزارهای پیشرفته‌تر مانند شبکه‌های LSTM به پیش‌بینی رفتار شبکه پرداخت.

[1] Time Seeds

۳–۳ روش پیشنهادی

روش پیشنهادی در این پروژه از سه گام اصلی تشکیل شده است. شکل ۳-۱ نشان‌دهنده‌ی این مراحل است که در بخش‌های بعدی به تفصیل بررسی می‌گردند.



شکل ۳-۱: ساختار کلی روش پیشنهادی

۳–۳–۱ گام اول: پردازش اولیه

برای شروع پروژه باید داده‌های خام را آماده‌ی پردازش کرد. از آنجایی که داده‌های ورودی در این پروژه داده‌های جمع‌آوری شده از شبکه هستند، باید ابتدا فایل‌هایی که در نتیجه‌ی پایش و ذخیره کردن بسته‌های شبکه آماده شده‌اند مهیا باشند. همان‌طور که در بخش‌های قبل گفته شد، فایل‌هایی با پسوند pcap وجود دارد که می‌توان آن‌ها را از نرم‌افزارهایی مانند Wireshark به عنوان خروجی دریافت کرد که این نوع فایل‌ها در این پروژه به عنوان ورودی مورداستفاده قرار می‌گیرند. همچنین فایل‌هایی با پسوند csv نیز می‌توانند به عنوان ورودی این پروژه در نظر گرفته شوند. همان طور که در شکل‌های ۲-۱ و ۲-۷ مشاهده شد، هر دو نوع فایل pcap و csv شامل رکوردهایی هستند که درمورد هر تأخیر در سری زمانی اطلاعاتی را ارائه می‌دهند. معمولا فایل‌های pcap حاوی اطلاعات بسته‌های شبکه ازجمله آدرس آی‌پی مبدأ[1]، آدرس آی‌پی مقصد[2]، تعداد بسته‌های منتقل شده در هر تأخیر، حجم بسته‌های منتقل شده در هر تأخیر، زمان ارسال و دریافت بسته‌های هر تأخیر و غیره هستند.

ابتدا باید دقت داشت که اگر فایل ورودی پروژه در فرمت pcap بود، باید به csv تبدیل شود؛ زیرا مدت زمان بارگیری فایل‌های pcap طولانی بوده و سربار زیادی به سیستم تحمیل می‌شود. پس از آماده‌سازی فایل csv، آن را در

[1] Source IP Address
[2] Destination IP Address

پروژه بار کرده و ویژگی‌های موجود در آن بررسی می‌شود. سپس ویژگی‌های مشخص‌کننده‌ی تعداد بسته‌های جاری در شبکه و آدرس‌های آی‌پی مبدأ و مقصد از آن استخراج شده و در ساختمان داده‌هایی به نام دیتافریم[1] ذخیره می‌شوند. دیتافریم ساختار داده‌ای دو بعدی است که هر ستون آن معرف یک ویژگی از داده‌های جمع‌آوری شده است.

در ادامه لازم است تا ویژگی‌های ذکر شده از فایل pcap استخراج شوند. نکته‌ای که باید به آن توجه داشت آن است که اندیس‌گذاری فایل‌های pcap و csv طبق زمان نیست. هدف اصلی این پروژه، بهره‌گیری از مفاهیم سری زمانی برای تحلیل داده‌های شبکه است و سری‌های زمانی وابسته به گذر زمان هستند. پس لازم است که فایل ورودی طبق زمان رویداد هر تأخیر سری زمانی اندیس‌گذاری شود [۸، ۹]. پس از تنظیم کردن اندیس فایل ورودی، اولین گام پروژه به اتمام رسیده و فایل موردنظر برای انجام هرگونه پردازش آماده است.

۳–۳–۲ گام دوم: تحلیل ویژگی‌های سری زمانی

فایلی که در شروع این گام از پروژه در دسترس است، حاوی چهار ستون است. ستون اول، معرف زمان و تاریخ رویداد هر تأخیر، اندیس فایل است و ستون‌های بعدی آدرس آی‌پی مبدا، مقصد و تعداد بسته‌های هر تأخیر است. سری زمانی اصلی تحلیل شده در این پروژه، سری زمانی ایجاد شده توسط تعداد بسته‌های تأخیرها است. آدرس‌های آی‌پی مبدا و مقصد در صورتی کاربرد دارند که هدف، دسته‌بندی و تحلیل داده‌ها بر اساس گره‌های[2] شبکه باشد. در این حالت می‌توان برای هر گره شبکه تحلیل سری زمانی انجام داد و در نهایت داده‌های خروجی مورد انتظار پروژه که دانه‌های زمانی هستند را برای گره موردنظر دریافت کرد.

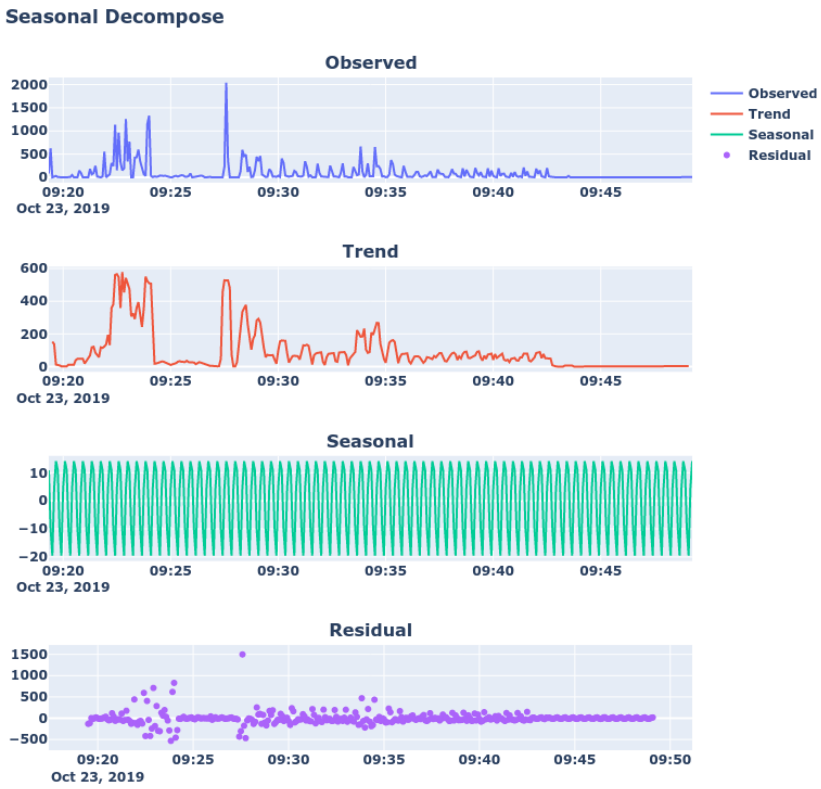
ابتدا با استفاده از ابزارها و کتابخانه‌های رسم نمودار، نمودار سری زمانی داده‌های موردنظر رسم می‌شود. محور افقی نمودار رسم شده معرف زمان و محور عمودی نشان‌دهنده‌ی تعداد بسته‌های دریافت شده از شبکه است. هدف از این کار شناسایی ویژگی‌های موجود در سری زمانی از جمله روند، فصلی بودن، الگوهای تناوبی و در نهایت تشخیص اولیه‌ی white noise است. باید توجه داشت که درصورت white noise بودن و یا random walk بودن داده‌ها، هرگونه پردازش و تحلیل آن‌ها بیهوده خواهد بود. با توجه به اینکه داده‌های ورودی این پروژه اطلاعات جریان داده‌های شبکه‌های کامپیوتری هستند، امکان تصادفی بودن مقادیر تا حدودی وجود ندارد. درصورتی‌که داده‌های جمع‌آوری شده حاوی داده‌های رویدادهای تصادفی مانند حملات و خرابی‌ها باشند، باید داده‌های مذکور از سری زمانی کنار گذاشته شوند؛ زیرا الگوی رفتاری شبکه باید در شرایط عادی به دست آید.

پس از رسم نمودار سری زمانی داده‌ها و بررسی ظاهری آن، باید با استفاده از تحلیل‌های آماری به استخراج ویژگی‌های آن پرداخت. اولین تحلیل، استفاده از تابع دیکی-فولر، جهت تشخیص ایستایی سری زمانی است. با توجه به روش تحلیل این تابع و اعداد خروجی آن، که در فصل گذشته توضیح داده شد، ایستایی سری زمانی بررسی می‌شود. در صورت ایستا نبودن سری زمانی، استفاده از روش‌های ایستاسازی سری زمانی در راستای اهداف این پروژه نیست؛ زیرا در گام بعدی توضیح داده خواهد شد که در این پروژه، به ایستا بودن سراسری سری زمانی لازم نیست و تنها بازه‌های مشخصی از داده‌ها جهت پردازش نهایی نیاز به ایستا بودن دارند. همان‌طور که در فصل گذشته توضیح داده

[1] Dataframe
[2] Nodes

شد، یک سری زمانی ایستا دارای میانگین و واریانس ثابت در تمام بازه‌های زمانی است و روند صعودی یا نزولی در داده‌ها دیده نمی‌شود. از این مهم در پردازش نهایی داده‌ها استفاده می‌شود. باید دقت داشت که ممکن است داده‌های ورودی پروژه از جنس سری زمانی random walk باشند. در فصل گذشته بررسی شد که تشخیص داده‌های RW به راحتی قابل انجام نیست و ابتدا باید سری زمانی ایستا شود. سپس با استفاده نمودار ACF سری زمانی ایستای حاصل، RW بودن آن بررسی می‌شود. در صورت ایستا بودن نمودار ACF، سری زمانی مورد بررسی از نوع RW بوده و امکان انجام پردازش‌های بیشتر بر روی داده‌های آن وجود ندارد.

پس از بررسی ایستایی و تشخیص عدم وجود WN و RW در سری زمانی ورودی، برای به دست آوردن روند و بررسی فصلی بودن داده‌ها از تابعی به نام تجزیه‌ی فصلی[1] استفاده می‌شود که خروجی آن به صورت ترکیبی از چهار نمودار است [۹، ۱۱]. این تابع مدل تجزیه‌ی سری زمانی را به عنوان ورودی دریافت می‌کند. خروجی تابع برای هر دو مدل جمعی و ضربی بررسی می‌شود. همان طور که در فصل قبل توضیح داده شد، در صورت تبعیت کردن نمودار باقی‌مانده‌ها از الگو یا پراکندگی خاص، مدل سری زمانی موردنظر جمعی است و برعکس.



شکل ۳-۲: نمونه‌ای از خروجی تابع تجزیه‌ی فصلی

نمودار اول مقادیر ذخیره شده در سری زمانی و نمودار دوم روند جزئی داده‌ها را به تصویر می‌کشند. نمودارهای سوم و چهارم نیز به ترتیب، الگوی فصلی موجود در سری زمانی و الگوی باقی‌مانده‌ها را نمایش می‌دهند. در صورت

[1] Seasonal Decompose

عدم وجود الگوی فصلی در داده‌ها تصویر نمودار سوم، یعنی نمودار فصلی، به صورت یک بلوک تیره و بدون حفره خواهد بود [۹]. بررسی الگوی فصلی و روند در این پروژه از این جهت موردنیاز است که در صورت وجود یک الگوی فصلی مشخص در داده‌ها و با توجه به خروجی موردنظر، الگوهای فصلی به عنوان بخشی از دانه‌های زمانی در خروجی ارائه شوند.

در نهایت، برای جمع‌بندی و تایید ویژگی‌های استخراج شده، از نمودار ACF کمک گرفته می‌شود. پس از ثبت نتایج به دست آمده از تحلیل‌های اولیه‌ی سری زمانی ورودی، پردازش نهایی و دریافت خروجی، به عنوان گام آخر پروژه، آغاز می‌شود.

۳-۳-۳ گام سوم: پردازش نهایی

پس از ثبت ویژگی‌های سری زمانی ورودی، نوبت به پردازش نهایی و دریافت خروجی می‌رسد. روند پردازش نهایی داده‌های سری زمانی ورودی به این صورت است که در ابتدا، دو پنجره برای حرکت روی سری زمانی و انجام پردازش تعریف می‌شود. پنجره‌ی اول پنجره‌ی مبدأ[1] و پنجره‌ی دوم، پنجره‌ی آزمون[2] نامیده می‌شود [۲، ۶]. اندازه‌ی این پنجره‌ها می‌تواند ثابت یا متغیر باشد. در این پروژه، برای شروع پردازش، پنجره‌هایی با طول ثابت درنظر گرفته شده است. ولی در طول فرایند پردازش طول پنجره‌ها دچار تغییر می‌شود [۲۱، ۲۲]. به جهت به دست آوردن مقدار مناسب اندازه‌ی پنجره‌ها جهت شروع پردازش، برای هر اندازه‌ی مشخص پردازش‌هایی به صورت جداگانه انجام شده و خروجی ثبت می‌شود.

اندازه‌ی پنجره‌های معرفی شده در این پروژه نمایانگر تعداد تأخیرهای سری زمانی است که در محدوده‌ی آن پنجره قرار می‌گیرند. درنتیجه هر پنجره، خود معرف یک سری زمانی است. به دلیل ثابت در نظر گرفتن اندازه‌ی پنجره‌ها هنگام شروع پردازش یک سری زمانی، در این پروژه، باید اندازه‌ی مناسب برای آن‌ها پیدا شود. لازم به ذکر است که اندازه‌ی پنجره‌ی مبدأ و آزمون در شروع پردازش برابر است. اندازه‌های مختلفی برای اولین پنجره‌ها در نظر گرفته می‌شود و با هرکدام پردازش روی چند فایل ورودی انجام می‌شود. سپس نتایج به دست آمده بررسی شده و اندازه‌ی پنجره‌ی مناسب برای پردازش نهایی اعلام می‌شود. اندازه‌های بررسی شده در این پروژه اعداد پنج، ده، دوازده، پانزده، بیست، بیست‌وپنج، سی و سی‌وپنج هستند.

پس از تنظیم اندازه‌ی اولیه‌ی پنجره‌ها، با شروع از ابتدای سری زمانی پنجره‌ها تعریف شده و پردازش آغاز می‌شود. هر پنجره تکه‌ای از سری زمانی ورودی را در بر می‌گیرد. برای شروع پردازش هر پنجره، ابتدا باید ایستایی سری زمانی محدود شده به پنجره‌ی موردنظر بررسی شود. در صورت ایستا نبودن، روش تفاضل میانگین برای ایستا کردن تکه سری زمانی موردنظر استفاده می‌شود؛ به این صورت که مقادیر سری زمانی محدود به پنجره‌ی درحال پردازش از میانگین مقادیر همان سری زمانی تفریق می‌شوند. ایستا بودن سری‌های زمانی محدود به پنجره‌ها برای محاسبات نهایی اهمیت دارد. زیرا سری‌های زمانی ایستا دارای میانگین و واریانس ثابت و مستقل از زمان هستند. باید توجه داشت که برای بررسی ایستایی پنجره‌ها، حداقل طول موردنیاز برای هر پنجره پنج است. دلیل انتخاب این اندازه، این

[1] Source Window
[2] Test Window

است که تابع تحلیل دیکی-فولر به حداقل پنج تأخیر نیاز دارد تا بتواند خروجی موردنظر را تولید کند. هرگاه ایستایی سری‌های زمانی محدود به دو پنجره‌ی مبدأ و آزمون تایید شد، حال نوبت به انجام محاسبات اصلی جهت تصمیم‌گیری برای تفکیک یا پیوند پنجره‌های مذکور است.

ابتدا میانگین مقادیر سری زمانی محدود شده به هر پنجره، در صورت عدم محاسبه در مرحله‌ی بررسی ایستایی، محاسبه می‌شود. سپس واریانس داده‌ها و به دنبال آن، انحراف‌معیار نیز محاسبه می‌گردد. از آنجایی‌که انحراف‌معیار معرف میزان پراکندگی داده‌ها حول میانگین است، برای هر پنجره بازه‌ای با مرکزیت میانگین در نظر گرفته می‌شود. کران بالای بازه‌ی مذکور برابر با حاصل جمع میانگین و انحراف‌معیار و کران پایین برابر حاصل تفریق انحراف‌معیار از میانگین است:

$$[\mu - \sigma, \mu + \sigma]$$

با توجه به ماهیت رفتاری انحراف‌معیار و میانگین، باید انتظار داشت که مقادیر داده‌های بازه‌ی به دست آمده نزدیک به هم بوده و سری زمانی رفتاری یکنواخت داشته باشد. حال که برای دو پنجره بازه‌های موردنظر محاسبه شده است، باید هم‌پوشانی این بازه‌ها بررسی شود. منظور از هم‌پوشانی وجود مقادیر مشترک در این بازه‌ها است. به طور مثال اگر بازه‌ی به دست آمده برای پنجره‌ی مبدأ بازه‌ی بسته‌ی [۳۶۰ + ،۳۶۰−] و بازه‌ی حاصل برای پنجره‌ی آزمون بازه‌ی بسته‌ی [۶۸۰ + ،۶۸۰−] باشد، نتیجه گرفته می‌شود که این دو بازه هم‌پوشانی دارند. زیرا مقادیر بازه‌ی اول به طور کامل در بازه‌ی دوم قرار می‌گیرند. ذکر این نکته لازم است که هم‌پوشانی کامل بازه‌ها نیاز نیست. اگر بخشی از یک بازه با بخشی از بازه‌ی دیگر مقادیر مشترک داشته باشد، باز هم نتیجه معرف وجود هم‌پوشانی میان بازه‌ها است.

در این نقطه از فرایند پروژه، دو رویکرد متفاوت به طور کلی بررسی می‌شود. رویکرد اول به این صورت است که در صورت وجود هم‌پوشانی میان بازه‌های محاسبه شده برای هر پنجره، صرفاً پنجره‌ها بدون تغییر اندازه و به اندازه‌ی یک پنجره رو به جلو حرکت کنند و پردازش ادامه یابد. در صورت عدم وجود هم‌پوشانی، هر پنجره به عنوان یک دانه‌ی زمانی ثبت شده و زمان پایان هر دو پنجره در لیست خروجی ثبت می‌شود. در رویکرد دوم، در صورت وجود هم‌پوشانی میان بازه‌های به دست آمده، رفتار سری‌های زمانی محدود شده به دو پنجره یکسان در نظر گرفته می‌شود دو پنجره با یکدیگر ادغام شده و تشکیل یک پنجره‌ی بزرگتر را دهند و یک پنجره‌ی جدید با اندازه‌ی پنجره‌های شروع ایجاد شده که پنجره‌ی بزرگ حاصل، با پنجره‌ی جدید مقایسه می‌شود و پردازش ادامه می‌یابد. این روند تاجایی که هم‌پوشانی میان پنجره‌ها نباشد ادامه دارد. در صورت عدم وجود هم‌پوشانی میان بازه‌های به دست آمده، نتیجه بر این است که رفتار سری زمانی در هر پنجره متفاوت بوده و از آنجایی‌که سری زمانی معرف رفتار شبکه در آن بازه‌ی زمانی است، هر پنجره به عنوان یک دانه‌ی زمانی منحصربه‌فرد تلقی شده و زمان نشان‌دهنده‌ی کران بالای هر پنجره به لیست دانه‌های زمانی اضافه می‌شود. در این پروژه، هر دو رویکرد مورد آزمون قرار گرفته و رویکرد دوم به عنوان رویکرد برتر انتخاب شده است. در نهایت، پس از اتمام پردازش پنجره‌های فعلی و ثبت نتیجه، پنجره‌ها به اندازه‌ی طول یک پنجره رو به جلو حرکت می‌کنند.

دلیل ادغام پنجره‌های هم‌پوشان و تشکیل پنجره‌ای بزرگتر این است که ممکن است در سری زمانی ورودی، تغییرات مقادیر به قدری تدریجی باشد که هیچ‌گاه دو پنجره دچار عدم هم‌پوشانی نشوند. به عنوان مثال، سری زمانی ورودی ممکن است روندی تدرجی و آهسته رو به بالا داشته باشد. در این صورت همیشه بازه‌های محاسبه شده برای

پنجره‌ها دارای هم‌پوشانی خواهند بود و ممکن است تمام سری زمانی به عنوان یک دانه‌های نهایی از پردازش خارج شود. متغیر بودن اندازه‌ی پنجره‌های پردازش از این امر جلوگیری می‌کند زیرا از یک حد مشخص به بعد، میانگین پنجره‌ی جدیدی که در صورت عدم هم‌پوشانی بازه‌ها در نظر گرفته می‌شود قطعا کمتر یا بیشتر از میانگین پنجره‌ی قدیمی است.

فرایند پردازش نهایی تا زمانی که نتوان دو پنجره با حداقل اندازه‌ی پنج ایجاد کرد که بتوانند روی سری زمانی به حرکت خود ادامه دهند ادامه می‌یابد. در نهایت، به ازای هریک از زمان‌های ثبت شده در لیست خروجی، یک خط جداکننده روی نمودار سری زمانی ورودی رسم می‌شود که تکه‌های محدود شده میان هر دو خط نشان‌دهنده‌ی یک دانه‌ی زمانی هستند و رفتار شبکه در هر دانه‌ی زمانی منحصربه‌فرد است.

۳–۴ جمع‌بندی

با توجه به اینکه امروزه تحلیل و پردازش داده‌ها در زمینه‌های علمی و کسب‌وکارهای مختلف اهمیت زیادی دارد، تحلیل داده‌های شبکه‌های کامپیوتری از این قائده مستثنا نیستند. ابزارها و روش‌های زیادی جهت پایش شبکه و ثبت داده‌های آن وجود دارد که محققان و متخصصان شبکه با استفاده از روش‌های پیش‌بینی و دسته‌بندی داده‌ها، از آن‌ها استفاده می‌کنند. این پروژه با هدف ارائه‌ی روشی برای تحلیل رفتار شبکه و ثبت آن انجام شد. خروجی‌های به‌دست‌آمده در این پروژه دانه‌های زمانی هستند که هرکدام بازه‌ای از زمان است که شبکه در آن زمان رفتار منحصربه‌فردی دارد.

پس از دریافت داده‌های خام شبکه و آماده‌سازی آن‌ها برای تحلیل سری زمانی، ویژگی‌های اولیه سری‌های زمانی از آن‌ها استخراج می‌شود. سپس گام نهایی، پردازش اصلی روی آن‌ها انجام می‌شود. در ابتدا، دو پنجره با اندازه‌ی مشخص روی داده‌های سری زمانی ورودی حرکت می‌کنند و پس از بررسی ایستایی هر بازه، میانگین و انحراف‌معیار داده‌های آن بازه محاسبه می‌شوند. بازه‌ای با مرکزیت میانگین که حد پایین آن از اختلاف میانگین و انحرف معیار و حد بالای آن از جمع میانگین و انحراف‌معیار به دست می‌آید، برای هر پنجره در نظر گفته می‌شود. در صورت وجود هم‌پوشانی میان بازه‌های به دست آمده برای هر پنجره، دو پنجره ادغام می‌شوند. پنجره‌ی حاصل با پنجره‌ی جدیدی به اندازه‌ی پنجره‌های شروع پردازش مقایسه شده و این روند تا جایی که بازه‌ی به دست آمده‌ی پنجره‌ها هم‌پوشانی نداشته باشد ادامه می‌یابد. در صورت عدم وجود هم‌پوشانی، محل اتصال دو پنجره ثبت شده و هر پنجره یک دانه‌ی زمانی را تشکیل می‌دهد.

در فصل بعد، ابزارهای استفاده شده در راستای این پروژه و نتایج به دست آمده در اثر پردازش نهایی داده‌های ورودی بررسی می‌شوند.

فصل چهارم
نتایج

۴–۱ مقدمه

پس از انجام پردازش‌های نهایی و دریافت خروجی‌ها، نوبت به تحلیل و بررسی آن‌ها می‌رسد. پردازش نهایی با اندازه‌های مختلف برای پنجره‌های مبدأ و آزمون انجام شد و خروجی موردنظر نمودارهای سری زمانی هستند که خطوطی عمودی، دانه‌های زمانی را روی آن‌ها جدا کرده‌اند.

در این فصل، ابتدا به بررسی و معرفی ابزارهای استفاده شده در این پروژه پرداخته می‌شود. سپس قسمت‌هایی از کدهای پروژه به همراه خروجی‌های پردازش‌های انجام شده روی داده‌های ورودی ارائه و بررسی می‌شود.

۴–۲ معرفی ابزارها

برای انجام این پروژه ابزارها و پیش‌نیازهای مختلفی استفاده شده است. به طور مثال، فایل‌های pcap که به عنوان ورودی پروژه هستند از نرم‌افزارهای پایش شبکه به دست می‌آیند که این فایل‌ها با استفاده از زبان‌های برنامه‌نویسی در محیط‌های توسعه‌ی مختلف قابلیت پردازش دارند. پردازش‌های مختلف با استفاده از کتابخانه‌هایی که زبان برنامه‌نویسی استفاده شده در اختیار کاربر قرار می‌دهد امکان‌پذیر است. در ادامه زبان برنامه‌نویسی و کتابخانه‌های استفاده شده از آن به همراه محیط‌های توسعه‌ی به کار رفته در این پروژه معرفی شده‌اند. در پیوست تصاویری از محیط‌های ابزارهای نام‌برده شده ارائه شده است.

۴–۲–۱ زبان برنامه‌نویسی و کتابخانه‌ها[1]

برای انجام این پروژه از زبان برنامه‌نویسی پایتون[2] استفاده شده است. زبان سطح بالای پایتون به دلیل در اختیار داشتن کتابخانه‌های متنوع در راستای تحلیل و پردازش داده یکی از بهترین زبان‌های استفاده شده در این زمینه است. دلیل استفاده از زبان پایتون در این پروژه نیز ارائه‌ی توابع مخصوص تحلیل سری‌های زمانی است.

کتابخانه‌هایی که در این پروژه استفاده شده‌اند به صورت زیر هستند:

[1] Libraries
[2] Python

- **کتابخانه‌ی** datetime:

این کتابخانه، یکی از کتابخانه‌های داخلی[1] پایتون است که برای کار با مهرهای زمانی[2] استفاده می‌شود. مهرهای زمانی رشته‌ای از کاراکترها هستند که زمان رخ دادن یک اتفاق را نشان می‌دهند. باید توجه داشت که قالب زمانی مشخصی برای کار با توابع سری زمانی در نظر گرفته شده است. این قالب به صورت **سال/ماه/روز ساعت:دقیقه:ثانیه** تعریف می‌شود. در این پروژه، با استفاده از این کتابخانه، قالب مهرهای زمانی فایل ورودی به قالب قابل قبول توابع سری زمانی تبدیل می‌شوند که پس از آن با استفاده از تابع تنظیم اندیس[3]، ستون زمان به عنوان اندیس فایل ورودی تنظیم می‌شود.

- **کتابخانه‌های** numpy **و** pamdas:

این دو کتابخانه به صورت شخص ثالث[4] در پایتون ارائه شده‌اند. استفاده‌ی اصلی این کتابخانه‌ها برای کار با اعداد و ساختمان داده‌ها است. لازم به ذکر است که کتابخانه‌ی pandas از numpy به صورت درونی بهره می‌برد. در این پروژه، از کتابخانه‌ی numpy برای محاسبات آماری از جمله میانگین، واریانس و انحراف‌معیار استفاده شده است. همچنین کتابخانه‌ی pandas برای کار با فایل‌های csv و پردازش روی آن‌ها به کار گرفته شده است.

- **کتابخانه‌ی** scapy:

این کتابخانه به عنوان یک کتابخانه‌ی شخص ثالث در پایتون، برای کار با داده‌های شبکه در نظر گرفته شده است. در این پروژه، به دلیل استفاده از فایل‌های pcap به عنوان ورودی و تبدیل آن‌ها به فایل‌های csv، از این کتابخانه استفاده شده است.

- **کتابخانه‌ی** statsmodels:

این کتابخانه نیز که به صورت شخص ثالث در پایتون ارائه شده است، دارای توابع محاسباتی برای سری‌های زمانی است. در این پروژه تابع adfuller از این کتابخانه برای انجام تست دیکی-فولر استفاده شده است.

- **کتابخانه‌ی** plotly:

این کتابخانه، یک کتابخانه‌ی شخص ثالث متن‌باز[5] برای پایتون است که بیش از چهل نوع نمودار را، برای ترسیم، پشتیبانی می‌کند. این کتابخانه در زمینه‌های مختلف علمی از جمله ترسیم نمودارهای جغرافیایی،

---

[1] In-Built

[2] Timestamps
[3] Set Index
[4] Third Party
[5] Open Source

ریاضی و آماری کاربرد دارد. در این پروژه برای ترسیم نمودارهای سری زمانی از این کتابخانه استفاده شده است.

۴-۲-۲ محیط‌های توسعه

از آنجایی‌که زبان برنامه‌نویسی استفاده شده در این پروژه پایتون است، محیط‌های زیادی برای کدنویسی و توسعه با استفاده از این زبان پراستفاده وجود دارند. در این پروژه، ابتدا جهت یادگیری و آزمون توابع سری زمانی با استفاده از پایتون و کتابخانه‌های مربوطه، از محیط ژوپیتر نوت‌بوک[۱] استفاده شد. این محیط توسعه هم به صورت جداگانه قابل نصب و استفاده است و هم با استفاده از نصب توسعه‌ی آناکندا[۲]، روی سیستم‌های مختلف قابل اجرا است. آناکندا یک توسعه از زبان‌های برنامه‌نویسی پایتون و آر[۳] است که برای محاسبات علمی به کار رفته و کتابخانه‌های زیادی را در اختیار کاربر قرار می‌دهد. در روند این پروژه، پس از نصب و اجرا، محیط توسعه‌ی ژوپیتر در محیط ویرایشگر متن وی‌اس‌کد[۴] یکپارچه‌سازی شد. برای انجام این کار، از افزونه‌ی ژوپیتر برای وی‌اس‌کد استفاده شد که یک سرور محلی[۵] اجرا شده و برای اجرای هر سلول از کدها[۶]، یک درخواست[۷] سمت آن سرور ارسال می‌شود.

پس از گذراندن گام‌های یادگیری مفاهیم سری زمانی، از گوگل کولب[۸] برای توسعه‌ی کدهای این پروژه استفاده شده است. گوگل کولب یک محیط توسعه تحت ژوپیتر است که زیرساخت آن کاملا براساس فضای ابری[۹] است. برای استفاده از این ابزار، به انجام هرگونه نصب و تنظیمات روی سیستم محلی نیاز نیست.

۴-۳ توابع پیاده‌سازی شده

جهت افزایش خوانایی کد و راحتی در استفاده، هر قسمت از کد که پرتکرار بوده و کار مشخصی را انجام می‌دهد به تابعی تبدیل شده است که در ادامه، هر تابع معرفی و بررسی شده است.

۴-۳-۱ تابع read_csv

وظیفه این تابع خواندن داده‌های فایل‌های csv و انجام پردازش اولیه روی آن‌ها است. این تابع برای استفاده از فایل‌های csv متفاوت، موارد زیر را استفاده می‌کند:

- از csv_path برای شناسایی فایل csv در مسیر مشخص شده استفاده می‌کند.
- با استفاده از cols می‌توان نام ستون‌هایی که پردازش بر روی آن‌ها انجام می‌شود را مشخص کرد.
- برای کار با زمان نیاز به مشخص کردن نام ستون شامل مقادیر زمانی است.

---

[1] Jupyter Notebook
[2] Anaconda Distribution
[3] R Programming Language
[4] VSCode
[5] Local Server
[6] Code Cells
[7] Request
[8] Google Colab
[9] Based-Cloud

به دلیل اینکه زمان می‌تواند در قالب خاصی وجود داشته باشد، برای تبدیل آن به یک شی تاریخ-زمان، باید الگوی رشته‌ی مهر زمانی مشخص شده باشد.

در پایان، اگر فایل csv موردنظر ستون‌هایی برای تعداد بسته‌های ارسالی و دریافتی داشته باشد، نام این ستون‌ها در قالب یک لیست مشخص می‌شود تا مقادیر این دو ستون با هم جمع شده و در قالب داده‌ی موردنظر ذخیره می‌شود. توجه شود که اگر مقداری برای این پارامتر درنظر گرفته نشود، در cols باید ستونی با نام pktCount وجود داشته باشد.

پس از ساخت قالب داده، ورودی تاریخ-زمان مشخص شده به قالب شی numpy.datetime64 درآمده تا بتوان با آن همانند زمان رفتار کرد و سپس همین ستون به عنوان شاخص قالب داده معرفی می‌شود.

```python
def read_csv(csv_path, cols, datetime_col, datetime_format='%d/%m/%Y%H:%M:%S', pkt_cols=None):
  """
  Read csv file and create a dataframe based on the given params.
  ## Parameters
  - csv_path: The csv file path
  - cols: Name of columns that have datetime, Forward and backward packets, and IP addresses.
  - datetime_col: Name of the column which contains datetime.
  - datetime_format: The format of the datetime_col
  - pkt_cols: If dataset contains columns for forward and backward packets, specify them as a list.

  If the dataset doesn't have pkt_cols, it must has column named as pkCount which contains
  total packets.
  """

  df = pd.read_csv(csv_path)
  if pkt_cols:
    df['pktCount'] = df[pkt_cols[0]] + df[pkt_cols[1]]

  df.loc[:, cols]
  df[datetime_col] = pd.to_datetime(df[datetime_col], format=datetime_format)
  df.set_index(datetime_col, inplace=True)
  return df
```

شکل ۴-۱: تابع read_csv

۴-۳-۲ تابع resample_df

از آنجایی‌که کار با واحد زمانی کمتر از ثانیه، مانند میلی ثانیه، سربار زیادی را برای سیستم ایجاد می‌کند، این تابع با توجه به پارامتر rule، با مقدار پیشفرض 1T که به معنای یک دقیقه است، برای ستون pktCount فایل ورودی عمل جمع را در بازه‌ی مشخص شده توسط rule اجرا می‌کند.

```python
def resample_df(df, rule='1T'):
  return df.resample(rule).agg({'pktCount': 'sum'})
```

شکل ۴-۲: تابع resample_df

۴-۳-۳ تابع to_stationary

وظیفه‌ی این تابع، بررسی کردن ایستا بودن قالب داده‌ای است که با استفاده از win_df معرفی شده است که در صورت برآورده نکردن شرط موجود، از تمامی نمونه‌های موجود در قالب داده win_df، مقدار win_mean که میانگین پنجره win_df است، کم می‌شود و مقادیر جدید در یک لیست ذخیره شده و بازگردانده می‌شوند.

```python
def to_stationary(win_df):
  """
  Check if the given dataframe is stationary or not
  and make it stationary if needed.

  ## Prameters
  - win_df: The window's dataframe to check its stationarity.
  - win_mean: The window's mean value
  - win_start: The window's start index in main dataframe
  - win_end: The window's end index in main dataframe
  - df: The main dataframe
  """

  win_mean = round(win_df['pktCount'].mean(), 2)
  arr = win_df['pktCount'].values.tolist()

  if sts.adfuller(win_df['pktCount'])[1] > 0.05:
    arr = (win_df['pktCount'] - win_mean).values.tolist()
  return arr
```

شکل ۴-۳: تابع to_stationary

۴-۳-۴ تابع extract_time_nodes

در این تابع، فرایند پیدا کردن دانه‌های زمانی انجام می‌شود. همان طور که در فصل قبل اشاره شد، از دو پنچره‌ی مبدأ و آزمون استفاده می‌شود که پنجره آزمون در هر مرحله به اندازه win_size، که تعداد نمونه‌های موجود در هر پنجره را نمایش می‌دهد، به سمت جلو حرکت می‌کند و پنجره پایه در صورت نبود شرط همپوشانی، جایگزین پنجره آزمون می‌شود. پس از بررسی ایستا بودن دو پنجره، مقدار میانگین و انحراف‌معیار هر دو پنجره به‌دست می‌آید.

ازآنجایی‌که به صفر رسیدن تعداد بسته‌های شبکه نشانه‌ای از وجود یک دانه‌ی زمانی است، بررسی می‌شود که این اتفاق افتاده است یا خیر. در صورت روی دادن آن، دانه‌ی زمانی جدیدی ثبت می‌شود. در غیر این صورت، شرط همپوشانی بررسی می‌شود و در صورت برقرار بودن شرط، دو پنجره به عنوان پنجره مبدأ، با هم ادغام می‌شوند. رویکرد دنبال شده توسط این تابع، رویکرد دوم است که در فصل گذشته توضیح داده شد.

در نهایت، تمامی دانه‌های زمانی در لیستی به اسم time_nodes ذخیره می‌شوند.

```python
def extract_time_nodes(df, win_size=10):
  time_nodes = []
  test_win_end = win_size
  ref_win_start = 0
  ref_win_end = win_size

  while test_win_end < len(df):

    test_win_start = ref_win_end
    test_win_end = test_win_start + win_size
    ref_win = df.iloc[ref_win_start:ref_win_end]
    test_win = df.iloc[test_win_start:test_win_end]

    if not (ref_win.size < 5 or test_win.size < 5):

      ref_arr = to_stationary(ref_win)
      test_arr = to_stationary(test_win)

      ref_win_mean = round(np.mean(ref_arr), 2)
      ref_win_std = round(np.std(ref_arr), 2)
      test_win_mean = round(np.mean(test_arr), 2)
      test_win_std = round(np.std(test_arr), 2)

      if (ref_win_std == 0 and test_win_std > 0) or (ref_win_std > 0 and test_win_std == 0):
        time_nodes.append(ref_win.index.values[-1])
        ref_win_start = test_win_start
      elif ref_win_std == 0 and test_win_std == 0:
        ref_win_start = test_win_start
      else:
        test_up_braket = ref_win_mean-ref_win_std <= test_win_mean+test_win_std <= ref_win_mean+ref_win_std
        test_bottom_braket = ref_win_mean-ref_win_std <= test_win_mean-test_win_std <= ref_win_mean+ref_win_std
        ref_up_braket = test_win_mean-test_win_std <= ref_win_mean+ref_win_std <= test_win_mean+test_win_std
        ref_bottom_braket = test_win_mean-test_win_std <= ref_win_mean-ref_win_std <= test_win_mean+test_win_std

        if not (ref_up_braket or ref_bottom_braket or test_up_braket or test_bottom_braket):
          time_nodes.append(ref_win.index.values[-1])
          ref_win_start = test_win_start

      ref_win_end = test_win_end

  return time_nodes
```

شکل ۴-۴: تابع extract_time_nodes

۴-۳-۵ تابع plot

همان طو که از نام این تابع برمی‌آید، وظیفه‌ی رسم نمودار و خطوط جداکننده‌ی دانه‌های زمانی به دست آمده از تابع extract_time_nodes را بر عهده دارد.

```python
def plot(df, nodes, title):
  """Plot given dataframe with its time's nodes"""

  fig = px.line(x=df.index, y=df['pktCount'])
  for node in nodes:
    time = pd.to_datetime(node)
    fig.add_vline(x=time, line_dash='dash', line_color='green')
  fig.update_layout(title=title, xaxis_title='DateTime', yaxis_title='Packets')
  fig.show()
```

شکل ۴-۵: تابع plot

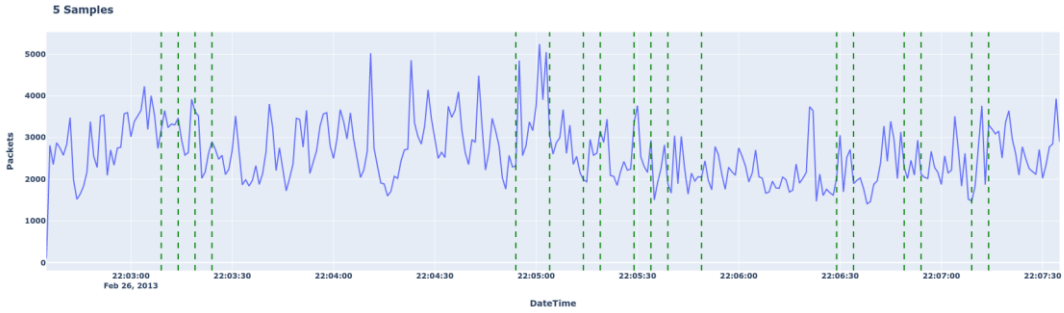
۴-۴ تحلیل نتایج

همان‌طور که در فصل گذشته توضیح داده شد، دو رویکرد متفاوت در گام پردازش نهایی این پروژه در نظر گرفته شده است. رویکرد اول عدم ادغام پنجره‌های هم‌پوشان و رویکرد دوم ادغام پنجره‌های هم‌پوشان و تشکیل پنجره‌ای بزرگتر است. در این قسمت نتایج به دست آمده از هر دو رویکرد ارائه شده و با یکدیگر مقایسه می‌شوند.

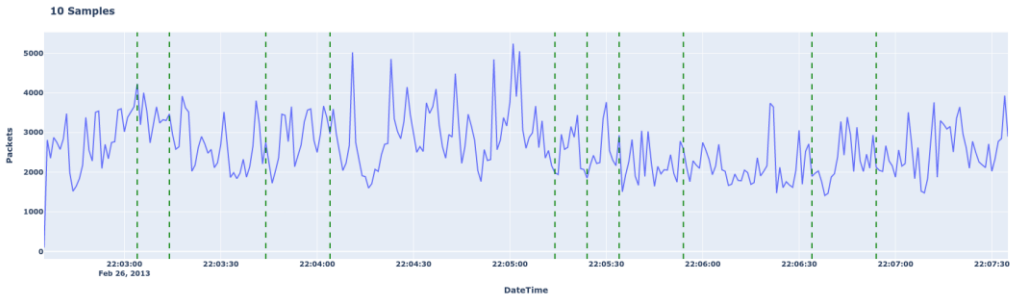
بیان شد که برای شروع پردازش، اندازه‌ای ابتدایی برای پنجره‌ها در نظر گرفته می‌شود. پس از پایان هر پردازش، عدد جدیدی به عنوان اندازه‌ی پنجره‌ها تنظیم شده و این روند تا جایی که اندازه‌ای مناسب یافت شود ادامه می‌یابد. در هر رویکرد، به ازای هر اندازه‌ی پنجره، تصاویری از نتایج پردازش ارائه می‌شود و در نهایت، بهترین اندازه‌ی بررسی شده اعلام می‌شود.

۴-۴-۱ رویکرد اول

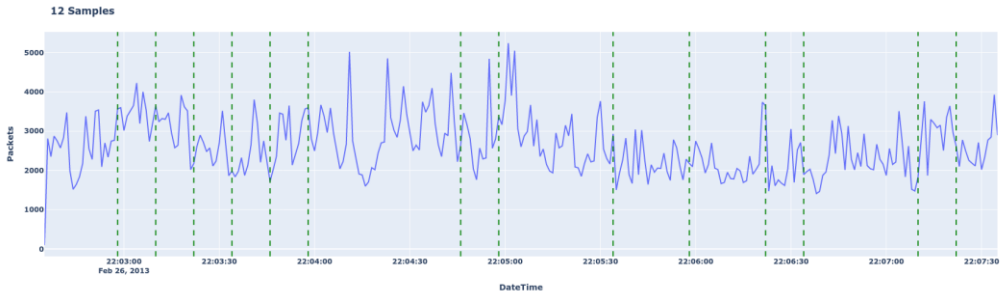
این رویکرد، به ازای اندازه‌ی پنجره‌های پنج، ده، دوازده، پانزده، بیست، بیست‌وپنج، سی و سی‌وپنج داده‌ها مورد بررسی قرار گرفته است. نتایج در ادامه ارائه شده‌اند:



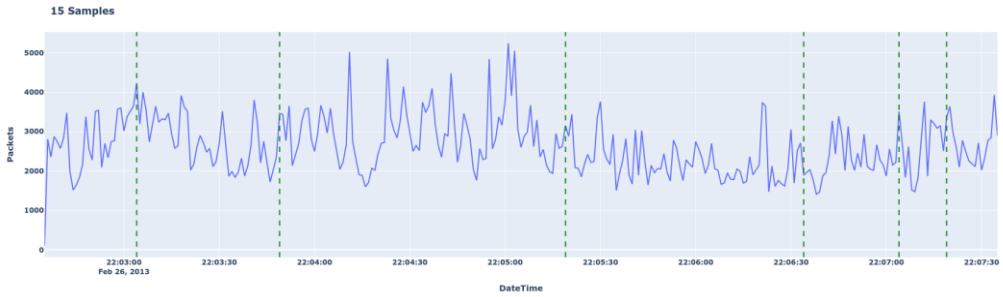
شکل ۴-۶: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی پنج



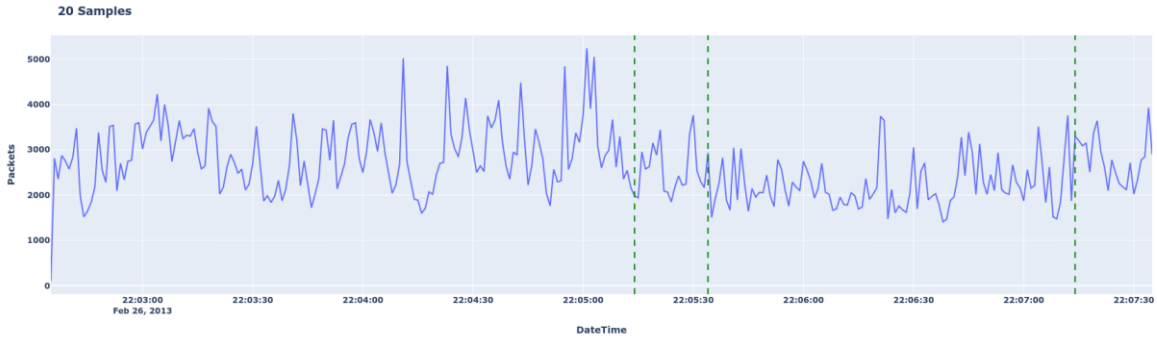
شکل ۴-۷: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی ده



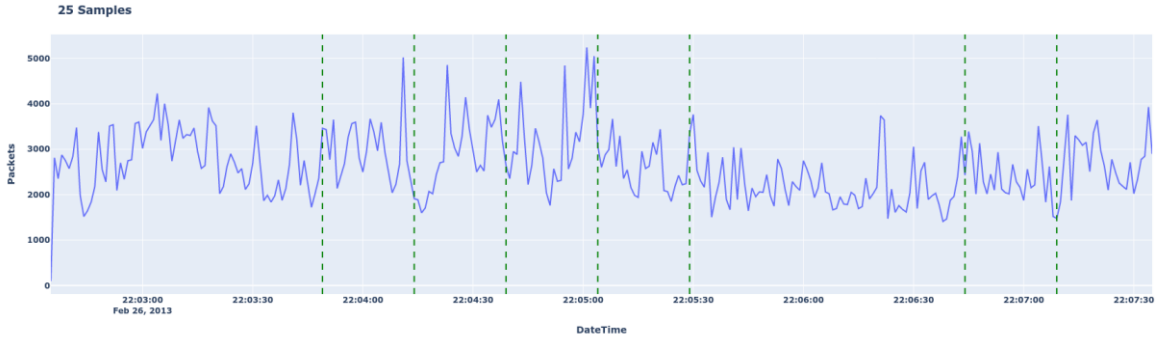
شکل ۴-۸: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی دوازده



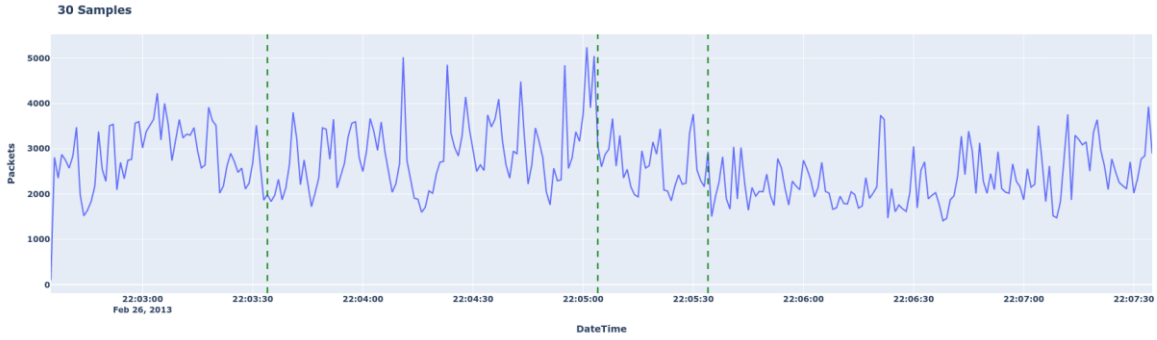
شکل ۴-۹: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی پانزده



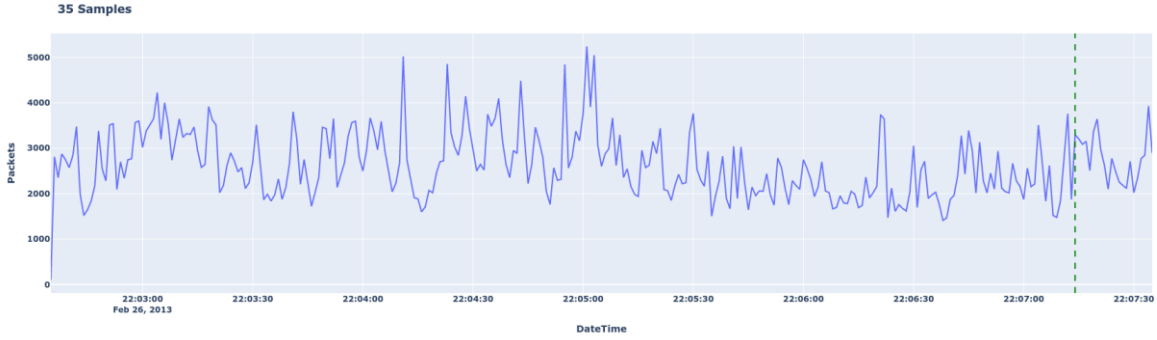
شکل ۴-۱۰: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی بیست



شکل ۴-۱۱: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی بیست‌وپنج



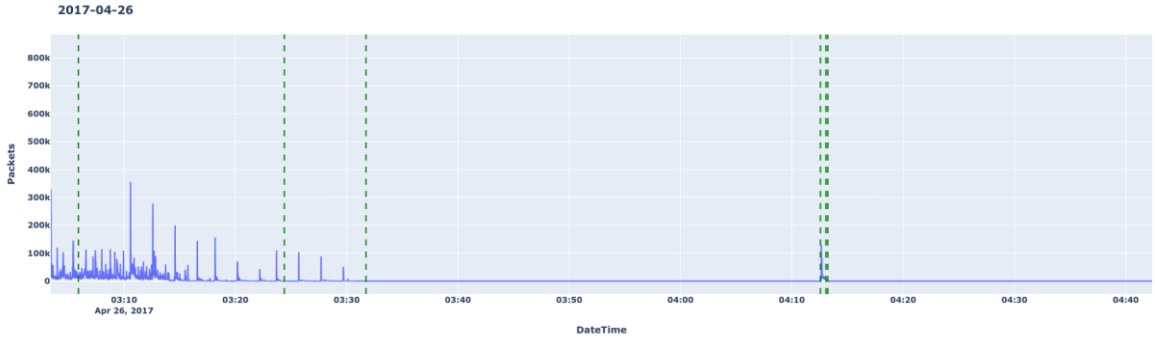
شکل ۴-۱۲: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی سی



شکل ۴-۱۳: خروجی رویکرد اول به ازای اندازه‌ی پنجره‌ی سی‌وپنج

از بررسی خروجی‌های این رویکرد این نتیجه به دست می‌آید که اندازه‌ی پنجره‌ی سی مناسب‌ترین اندازه‌ی پنجره‌ی است. اندازه‌های پنج، ده و دوازده به میزان زیادی دانه‌های زمانی نامناسب استخراج کرده‌اند. دلیل نامناسب بودن دانه‌های استخراج شده، فاصله‌های بسیار نزدیک یا بسیار دور دانه‌های ایجاد شده در مدت زمانی است که رفتار شبکه دارای روند یکسان بوده است. اندازه‌ی پانزده در ابتدای سری زمانی، روند صعودی، نزولی و نوسانات را تشخیص داده است ولی در انتهای سری زمانی دانه‌هایی را که روند یکسانی دارند، جدا کرده است. اندازه‌ی بیست نیز به اندازه‌ی کافی دانه‌بندی انجام نداده است. اندازه‌ی بیست‌وپنج، قسمت‌هایی از سری زمانی که دچار نوسان، صعود یا نزول شده را از وسط تقسیم کرده است که نتیجه‌ی اشتباهی برای دانه‌های زمانی است. در نهایت، اندازه‌ی پنجره‌ی سی، قسمت شروع سری زمانی، قسمت نوسانی، قسمت نزولی و قسمت پایانی سری زمانی، که روند تدریجی رو به رشدی دارد را به عنوان دانه‌های زمانی تولید کرده است که طبق ویژگی‌های سری‌های زمانی، کاملا نرمال و طبیعی است.

اندازه‌ی پنجره‌ی ده برای فایل ورودی دیگری نیز آزمایش شد.

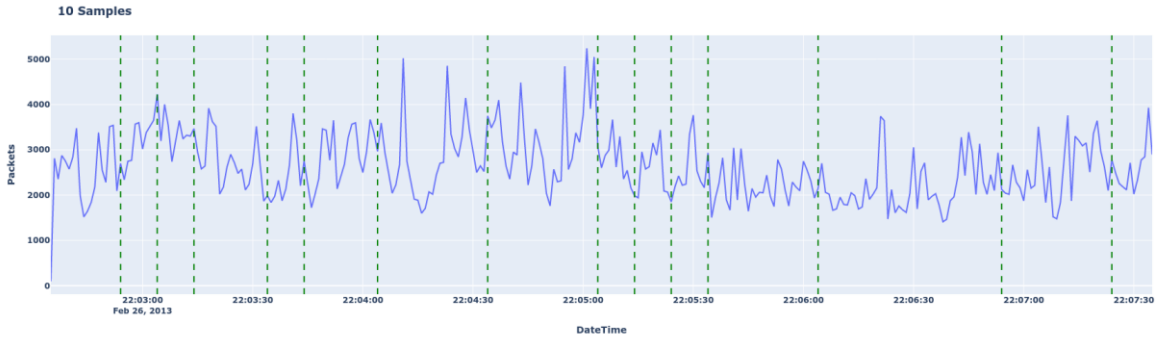


شکل ۴-۱۴: خروجی رویکرد اول با اندازه‌ی پنجره‌ی ده برای داده‌های جدید

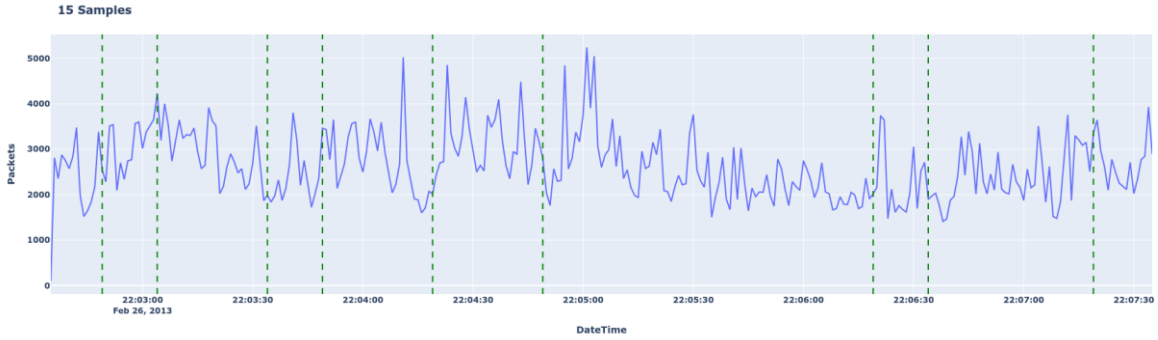
همان‌طور که در شکل ۴-۹ می‌توان مشاهده کرد، اندازه‌ی ده تقسیم بندی نامناسبی انجام داده است زیرا دانه‌ی سوم و انتهای دانه‌ی دوم دارای ویژگی‌های یکسان سری زمانی هستند و باید در یک دانه قرار گیرند.

۴-۴-۲ رویکرد دوم

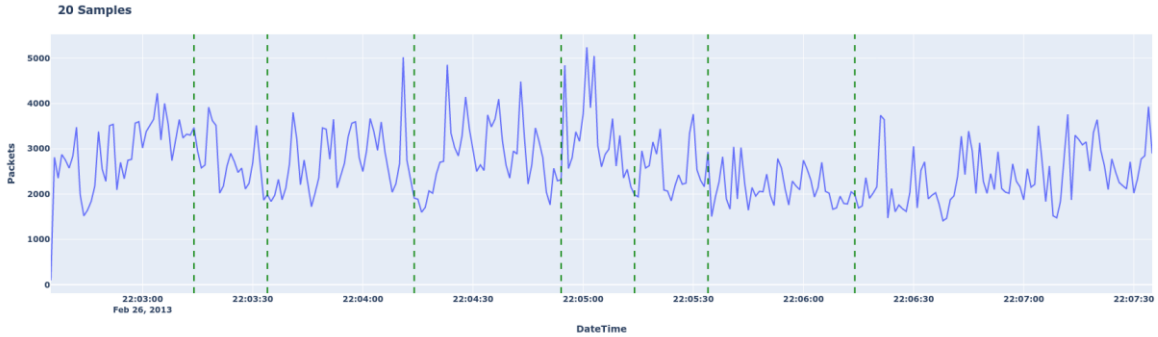
در این رویکرد، هنگام شناسایی پنجره‌های دارای هم‌پوشانی، دو پنجره ادغام شده و با پنجره‌ی جدیدی به اندازه‌ی ابتدایی پنجره‌ها مقایسه می‌شوند. این روند تا پیدا شدن پنجره‌های غیرهم‌پوشان ادامه دارد. اندازه پنجره‌هایی که خروجی آن‌ها در رویکرد اول بسیار از هدف دور بود، در این رویکرد آزمایش نشدند. این رویکرد با اندازه پنجره‌های ده، پانزده، بیست و سی آزمایش شده است. نتایج در ادامه ارائه شده‌اند:



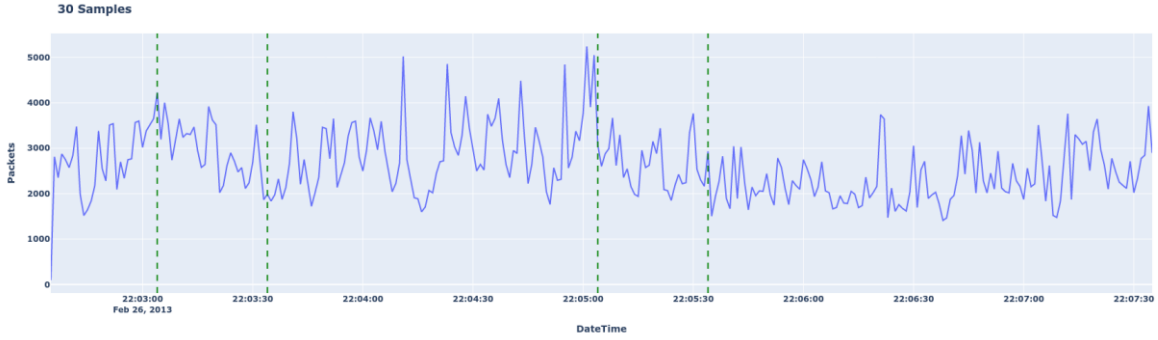
شکل ۴-۱۵: خروجی رویکرد دوم برای اندازه‌ی پنجره‌ی ده



شکل ۴-۱۶: خروجی رویکرد دوم برای اندازه‌ی پنجره‌ی پانزده



شکل ۴-۱۷: خروجی رویکرد دوم برای اندازه‌ی پنجره‌ی بیست

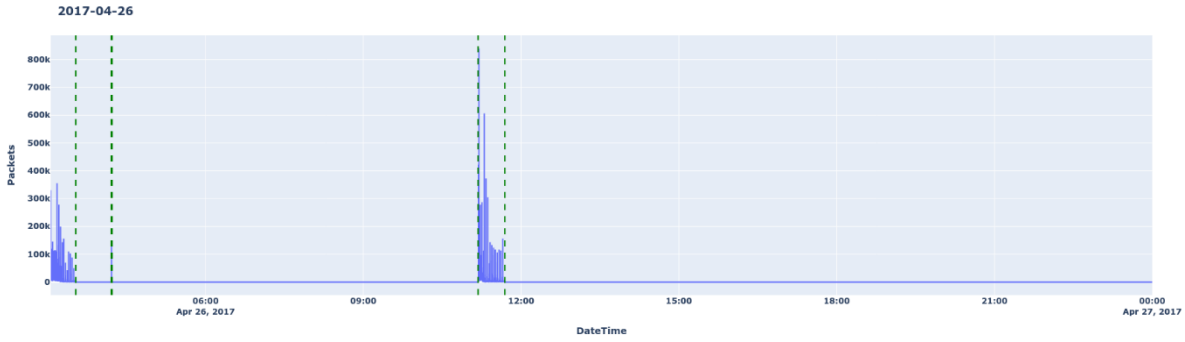


شکل ۴-۱۸: خروجی رویکرد دوم برای اندازه‌ی پنجره‌ی بیست‌وپنج

به طور کلی، تعداد دانه‌های تولید شده در رویکرد دوم بیشتر از رویکرد اول است. اندازه‌ی ده، مانند رویکرد اول، دانه‌های زیادی را تولید کرده است که از نظر ویژگی‌های سری زمانی بسیار نزدیک به هم هستند. اندازه‌ی پانزده هم دارای دانه‌هایی است که روند سری زمانی آن‌ها یکسان بوده و می‌توانند در یک دانه قرار گیرند. دو اندازه‌ی بیست و سی از نظر دقت و تعداد دانه‌های خروجی، اندازه‌های بهتری هستند ولی اندازه‌ی بیست دارای دانه‌های زیادتری در قسمت نوسان داده‌ها است. در این رویکرد، مانند رویکرد اول، اندازه‌ی سی بهترین اندازه‌ی پنجره انتخاب می‌شود.

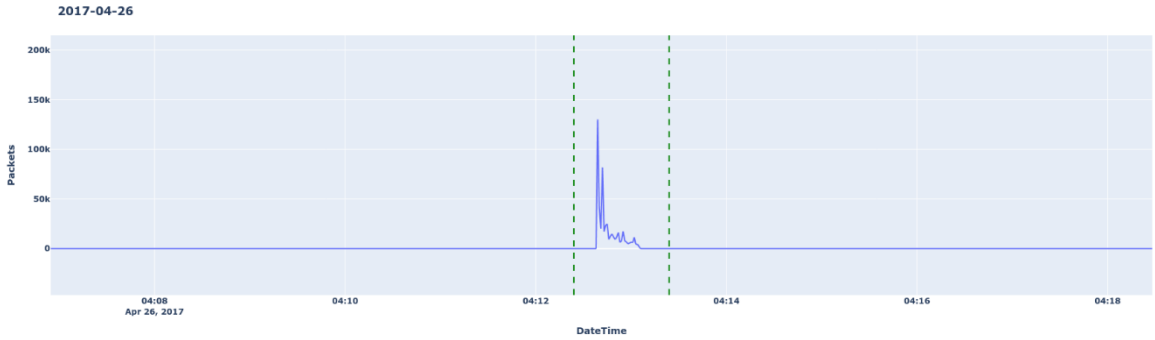
مورد دیگری که باید به آن توجه داشت این است که تفاوت خروجی‌های اندازه‌ی سی در دو رویکرد، فقط در یک دانه‌ی زمانی است. در رویکرد دوم، شروع سری زمانی که با صعود و نزول متوالی است، به دو دانه‌ی زمانی تقسیم شده که منطقی‌تر است. این رویکرد برای کاربر مشخص می‌کند که در بازه‌ی زمانی مشخص شده انتظار افت جریان داده‌ی شبکه را داشته باشد.

فایل داده‌های دومی که در رویکرد اول بررسی شد، در این رویکرد نیز با اندازه‌ی سی بررسی شده است.



شکل ۴-۱۹: خروجی رویکرد دوم با اندازه‌ی پنجره‌ی سی برای داده‌های جدید

بر خلاف رویکرد اول که شروع داده‌ها را به سه دانه‌ی زمانی تقسیم کرده بود، رویکرد دوم به درستی چهار دانه‌ی زمانی تشکیل داده که دقت این رویکرد را نشان می‌دهد.



شکل ۴-۲۰: دانه‌ی زمانی تشخیص داده شده توسط رویکرد دوم

شکل ۴-۱۵ یک دانه‌ی زمانی که توسط رویکرد دوم با اندازه‌ی سی تشخیص داده شده است را نشان می‌دهد. در تمام زمانی که داده‌های جاری در شبکه صفر بوده است، یک نوسان چند دقیقه‌ای دیده است. از آنجایی‌که داده‌های

بررسی شده در شکل ۴-۱۵ داده‌های یک حمله به شبکه هستند، تشخیص هرگونه افزایش داده میان جریان داده‌ی صفر حیاتی است.

۴–۵ جمع‌بندی

این پروژه با استفاده از زبان پایتون و با بهره‌گیری از کتابخانه‌های رسم نمودار و توابع محاسباتی سری زمانی پیاده‌سازی شده است. محیط‌هایی که کدهای پروژه در آن‌ها نوشته شده است شامل ویرایشگر متن وی‌اس‌کد، ژوپیتر نوت‌بوک و گوگل کولب است.

دو رویکرد معرفی شده در فصل سوم، با استفاده از اندازه‌های مختلف برای پنجره‌های ابتدایی آزمایش و خروجی‌های مربوطه ارائه و تحلیل شده‌اند. با توجه به دانه‌های زمانی تولید شده در هر رویکرد و اندازه‌ی پنجره‌ها، رویکرد دوم در دانه‌بندی دقت بیشتر داشته و دانه‌های صحیح‌تری طبق رفتار سری زمانی ورودی و ویژگی‌های آن تولید می‌کند. همچنین میان اندازه‌هایی که برای پنجره‌های ابتدایی در نظر گرفته شد، اندازه‌ی سی بهترین و دقیق‌ترین دانه‌بندی را در تمامی آزمایش‌ها ارائه داد.

به طور کلی می‌توان نتیجه گرفت که روش ارائه شده جهت دانه‌بندی سری زمانی جریان داده‌های شبکه از نتیجه‌ی قابل‌قبولی برای شبکه‌های هدف که شبکه‌های اداری و سازمانی هستند، برخوردار است.

فصل پنجم
جمع‌بندی

امروزه، در زمینه‌های علمی و کاری مختلف، جمع‌آوری، پردازش و تحلیل داده‌ها از اساسی‌ترین امور به شمار می‌رود. از آنجایی‌که فناوری شبکه‌های کامپیوتری یکی از گسترده‌ترین و پرکاربردترین فناوری‌ها در عصر امروز است، حفظ و نگهداری زیرساخت‌های این فناوری از اهمیت بسیاری برخوردار است. در این راستا، مختصصا و ناظران این فناوری، در هر کسب‌وکار و سازمانی، به طور مداوم به پایش این زیرساخت‌ها مشغول هستند. داده‌های جمع‌آوری شده از شبکه‌های کامپیوتری نیز با استفاده از علوم تحلیل داده‌های امروز قابل تحلیل و بررسی هستند.

ابزارهای زیادی جهت جمع‌آوری و ثبت داده‌های شبکه و تحلیل آن‌ها وجود دارد. همچنین روش‌های زیادی از علوم مختلف در کنار شبکه‌های کامپیوتری قرار گرفته است که امکان پیش‌بینی رفتار شبکه و همچنین اموری مثل دسته‌بندی داده‌ها و تشخیص ناهنجاری را برای کاربران متخصص این فناوری ایجاد می‌کند. یکی علوم تحلیل داده‌ها که در کسب‌وکارهای زیادی استفاده می‌شود، علم آمار است. مفاهیم سری زمانی با استفاده از مفاهیم آماری می‌توانند داده‌هایی را که وابسته به زمان بوده و به صورت پیوسته در حال تغییر هستند، تحلیل کنند. در این پروژه، جریان داده‌های شبکه که متغیری وابسته به زمان است، به عنوان یک سری زمانی بررسی و تحلیل می‌شود.

هدف از انجام این پروژه، رسیدن به هدف ثبت رفتار شبکه‌های سازمانی است. به این صورت که ابتدا داده‌های شبکه جمع‌آوری شده و به عنوان ورودی به این پروژه داده شده‌اند. سپس با استفاده از پردازش‌های انجام شده، الگوری رفتاری مشخصی برای شبکه‌ی مذکور ثبت می‌شود. ثبت الگوی رفتاری یک شبکه به ناظران و متخصصان آن کمک می‌کند تا بتوانند یک طرح اولیه‌ی ذهنی از آنچه که قرار است در شبکه اتفاق افتد داشته باشند. این امر در راستای توسعه‌ی سیستم‌های تشخیص ناهنجاری و یا پیش‌بینی رفتاری شبکه بسیار تأثیرگذار است.

الگوی رفتاری شبکه در این پروژه به این صورت ثبت می‌شود که خروجی پردازش‌های انجام شده، دانه‌های زمانی هستند. هر دانه‌ی زمانی بازه‌ای مشخص از زمان است که در آن بازه، شبکه از خود رفتاری منحصربه‌فرد و متفاوت نسبت به دیگر بازه‌ها نشان می‌دهد. به طور مثال، میزان داده‌های جاری در شبکه ممکن است در یک دانه‌ی زمانی رو به افزایش و در بازه‌ی دیگر رو به کاهش باشند.

پردازش‌های انجام شده در این پروژه، با استفاده از مفاهیم آماری مانند میانگین، واریانس و انحراف‌معیار انجام می‌شود. ابتدا فایل داده‌های شبکه برای تحلیل سری زمانی آماده شده و ویژگی‌های موردنظر از آن استخراج و

اندیس‌گذاری می‌شوند. سپس با رسم نمودارهای سری زمانی موردنظر، ویژگی‌های آن استخراج شده و برای پردازش نهایی آماده می‌شود. پردازش نهایی با تعریف دو پنجره‌ی متحرک روی سری زمانی با اندازه‌ی ثابت تعریف می‌شوند. برای هر پنجره میانگین و انحراف‌معیار محاسبه و بازه‌ی خطای داده‌ها از میانگین با استفاده از انحراف‌معیار محاسبه می‌شود. بازه‌ی مذکور برای هر دو پنجره انجام شده و وجود هم‌پوشانی برای این بازه‌ها بررسی می‌شود. در صورت وجود هم‌پوشانی، پنجره‌های در حال پردازش دارای رفتار یکسان بوده و ادغام می‌شوند. در غیر این صورت پنجره‌ها از یکدیگر تفکیک شده و هرکدام به عنوان یک دانه‌ی زمانی به خروجی اضافه می‌شود. سپس پنجره‌های مذکور روی سری زمانی رو به جلو حرکت می‌کنند تا داده‌ها به اتمام برسند.

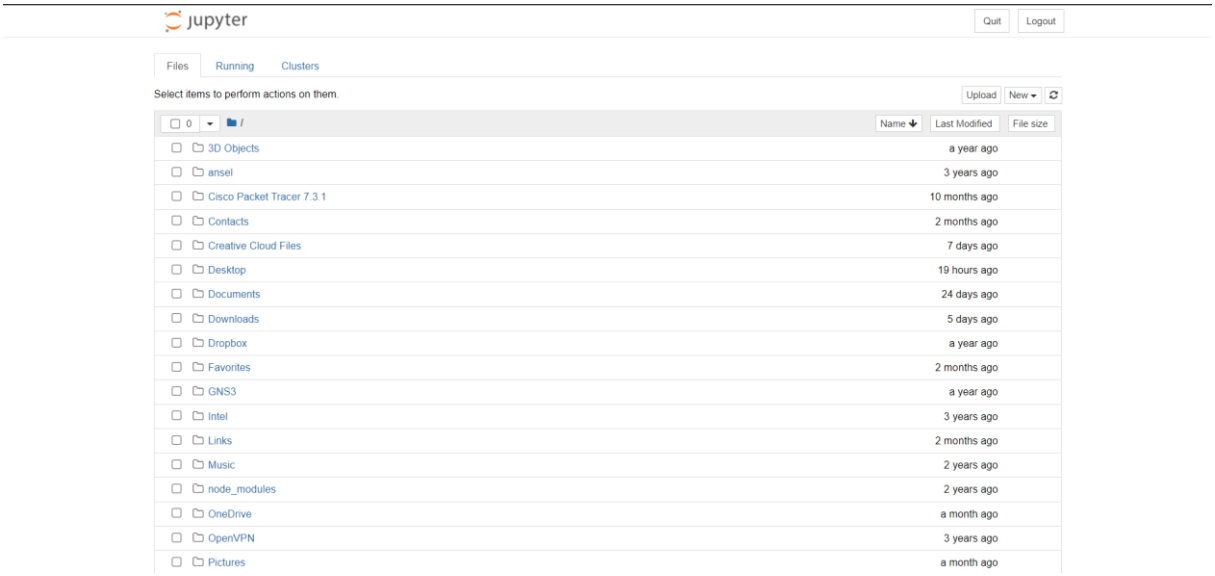
در این پروژه، برای سه دسته داده‌ی ورودی، هفت اندازه‌ی مختلف پنجره تعریف و پردازش شد. روش استفاده شده در این پروژه، به دلیل استفاده از مفاهیم آماری، می‌تواند دارای دقت زیادی در اندازه‌گیری ماهیت داده‌های سری زمانی داشته باشد.

به عنوان پیشنهاد جهت انجام تحقیقات بیشتر و افزایش دقت دانه‌بندی زمانی، می‌توان اندازه‌ی پنجره‌های اولین پردازش را با استفاده از انجام پیش‌پردازش‌هایی و با استفاده از داده‌های آماری به دست آورد. در نهایت می‌توان مفاهیم آماری استفاده شده در این پروژه را به صورت دقیق‌تر و با داده‌های بیشتر اندازه‌گیری کرد و آزمون‌های بیشتر در راستای افزایش دقت خروجی موردنظر انجام داد. همچنین می‌توان این پروژه را با استفاده از مفاهیم هوش مصنوعی، یادگیری ماشین[1] (ML) و یادگیری عمیق، به‌جای مفاهیم سری زمانی، انجام داد. شبکه‌های LSTM نمونه‌ای از این مفاهیم در یادگیری عمیق هستند.

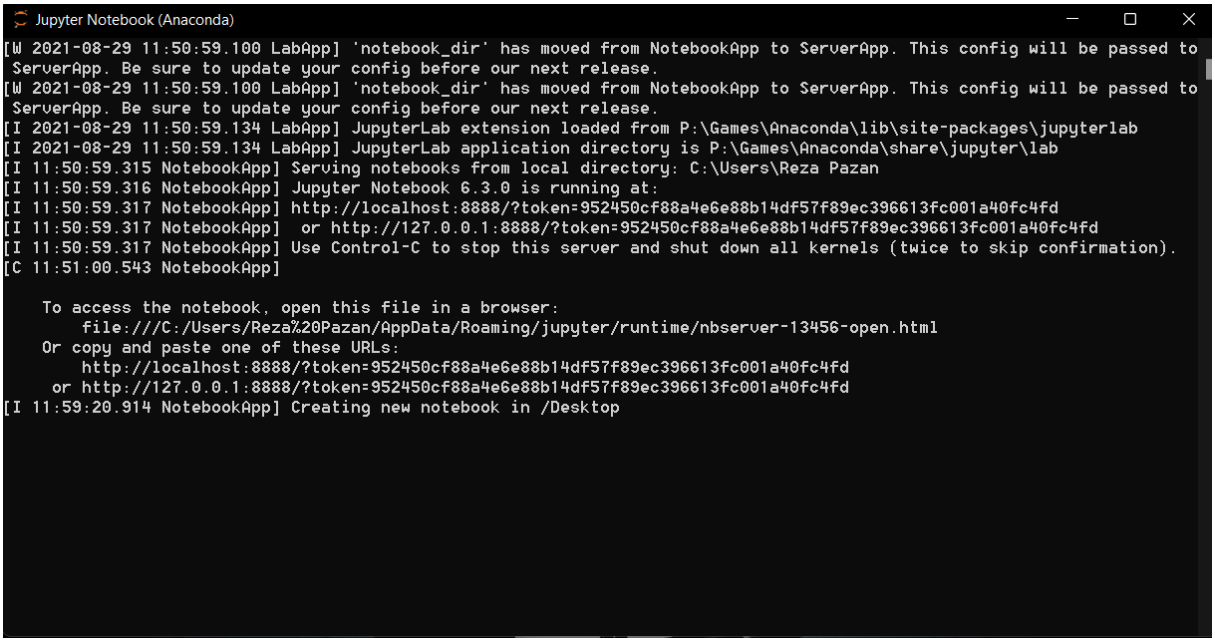
از نتایج این پروژه نیز می‌توان برای دسته‌بندی داده‌های شبکه، توسعه‌ی سیستم‌های تشخیص ناهنجاری و پیش‌بینی رفتار شبکه‌های کامپیوتری استفاده کرد. این تحقیقات هم با استفاده از مدل‌های سری زمانی و هم با استفاده از مفاهیم علم هوش مصنوعی قابل انجام هستند.

[1] Machine Learning

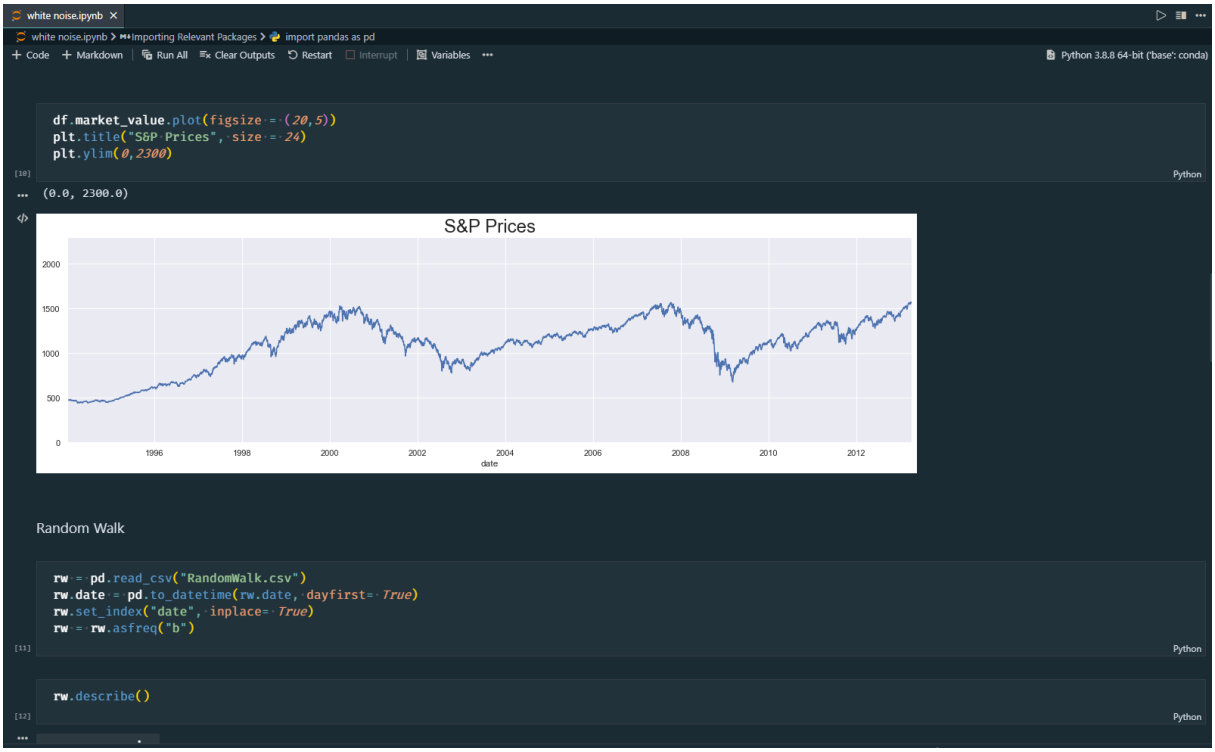
پیوست



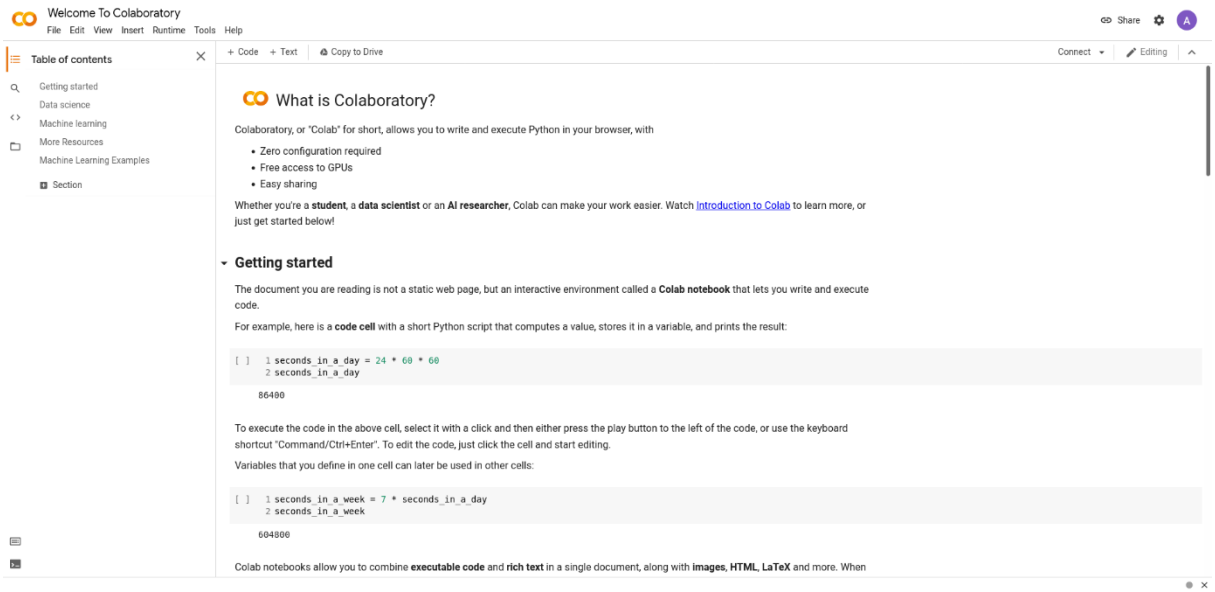
شکل ۶-۱: صفحه‌ی ابتدایی Jupyter Notebook



شکل ۶-۲: سرور اجرا شده برای Jupyter Notebook



شکل ۶-۳: محیط یکپارچه‌سازی شده‌ی Jupyter در Visual Studio Code



شکل ۶-۴: صفحه‌ی ابتدایی Google Colab

منابع

[1] Jung, Sangjoon, Chonggun Kim, and Younky Chung. "A prediction method of network traffic using time series models." In International Conference on Computational Science and Its Applications, pp. 234–243. Springer, Berlin, Heidelberg, 2006.

[2] Wu, Qingtao, and Zhiqing Shao. "Network anomaly detection using time series analysis." In Joint international conference on autonomic and autonomous systems and international conference on networking and services–(icas–isns' 05), pp. 42–42. IEEE, 2005.

[3] https://blog.faradars.org/time–series, Accessed August 22nd , 2021.

[4] Joshi, Manish, and Theyazn Hassn Hadi. "A review of network traffic analysis and prediction techniques." arXiv preprint arXiv:1507.05722 (2015).

[5] Grebennikov, A., Y. Krukov, and D. Chernyagin. "A prediction method of network traffic using time series models." Grebennikov, Y. Krukov, D. Chernyagin.–2011 (2011).

[6] Kiran, Mariam, Cong Wang, George Papadimitriou, Anirban Mandal, and Ewa Deelman. "Detecting anomalous packets in network transfers: investigations using PCA, autoencoder and isolation forest in TCP." Machine Learning 109, no. 5 (2020).

[7] Ntlangu, Mbulelo Brenwen, and Alireza Baghai–Wadji. "Modelling network traffic using time series analysis: A review." In Proceedings of the International Conference on Big Data and Internet of Thing, pp. 209–215. 2017.

[8] Derek Banas, Time Series Analysis, https://www.youtube.com/playlist?list=PLGLfVvz_LVvSVgVCsPWLr961id7kRv1wt, Accessed August 11th, 2021.

[9]Time Series Analysis in Python 2020 by 365 Careers, https://www.udemy.com/course/time–series–analysis–in–python, Accessed August 1st, 2021.

[10] https://machinelearningmastery.com/white–noise–time–series–python, Accessed August 30th, 2021.

[11]Tony Moses, Udacity, Business Analyst Nanodegree, Time Series Forecasting, https://git.ir/udacity–time–series–forecasting.

[12]Time Series From Scratch, https://towardsdatascience.com/time–series–from–scratch–white–noise–and–random–walk–5c96270514d3, Accessed August 27th, 2021.

[13] Aldhyani, Theyazn HH, Melfi Alrasheedi, Ahmed Abdullah Alqarni, Mohammed Y. Alzahrani, and Alwi M. Bamhdi. "Intelligent hybrid model to enhance time series models for predicting network traffic." IEEE Access 8 (2020): 130431–130451.

[14] Zhao, Zheng, Weihai Chen, Xingming Wu, Peter CY Chen, and Jingmeng Liu. "LSTM network: a deep learning approach for short-term traffic forecast." IET Intelligent Transport Systems 11, no. 2 (2017): 68-75.

[15] https://towardsdatascience.com/how-to-detect-random-walk-and-white-noise-in-time-series-forecasting-bdb5bbd4ef81, Accessed August 28th, 2021.

[16] https://towardsdatascience.com/significance-of-acf-and-pacf-plots-in-time-series-analysis-2fa11a5d10a8, Accessed August 28th, 2021.

[17] https://towardsdatascience.com/identifying-ar-and-ma-terms-using-acf-and-pacf-plots-in-time-series-forecasting-ccb9fd073db8, Accessed August 28th, 2021.

[18] Jofipasi, Chesilia Amora. "Selection for the best ETS (error, trend, seasonal) model to forecast weather in the Aceh Besar District." In IOP conference series: materials science and engineering, vol. 352, no. 1, p. 012055. IOP Publishing, 2018.

[19] https://blog.faradars.org/standard-deviation-and-variance, Accessed September 2nd, 2021.

[20] https://www.r-bloggers.com/2017/02/is-my-time-series-additive-or-multiplicative, Accessed September 2nd, 2021.

[21] Mozaffari, Ladan, Ahmad Mozaffari, and Nasser L. Azad. "Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: A case study on San Francisco urban roads." Engineering science and technology, an international journal 18, no. 2 (2015): 150-162.

[22] Wagner, Neal, and Zbigniew Michalewicz. "An analysis of adaptive windowing for time series forecasting in dynamic environments: Further tests of the DyFor GP model." In Proceedings of the 10th annual conference on Genetic and evolutionary computation, pp. 1657-1664. 2008.