|  |  |  |
| --- | --- | --- |
| **­**  **دانشگاه صنعتي اميركبير**  **(پلي تكنيك تهران)** | **باسمه تعالي**  فرم تعريف **پروژه**  فارغ التحصيلي دوره كارشناسي | **دانشكده مهندسي كامپيوتر** |

**تاريخ: ...........................**

**شماره: ..........................**

|  |
| --- |
| **عنوان پروژه: طراحی و پیاده‌سازی سیستم پیش‌بینی ترافیک بر پایه شبکه‌های عصبی** |
| **استاد راهنماي پروژه: دکتر رضا صفابخش امضاء:** |
| **مشخصات دانشجو:**  **نام و نام خانوادگي: الهه داستان[[1]](#footnote-1)**  **شماره دانشجویي: ۹۶۳۱۰۲۵ ترم ثبت نام پروژه: دوم ۰۰-۹۹** |
| **داوران پروژه:**   1. **امضاء داور:** 2. **امضاء داور:** |
| **شرح پروژه** (در صورت مشترك بودن بخشي از كار كه بعهده دانشجو مي باشد مشخص شود)**:**  **در این پروژه قصد داریم ترافیک را در شبکه‌ای از خیابان‌ها و چهارراه‌ها با استفاده از وابستگی‌های زمانی و مکانی پیش‌بینی کنیم. برای اینکار به جای استفاده از روش‌های سنتی که کارآیی خوبی نداشته‌اند از پیچش روی گراف‌ها بهره خواهیم جست.** |
| **وسائل مورد نياز:**  **- امکان دسترسی به مقالات مرتبط**  **- یک سرور جهت اجرا و ارزیابی مدل‌ها (نیازمندی‌های سخت‌افزاری سرور در ضمیمه آورده شده است.)** |
| **محل انجام پروژه: دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر تاريخ شروع: اسفند ۱۳۹۹** |

اين قســمت توســط دانشــكده تكميــل مي‌گــردد:

|  |
| --- |
| **تاريخ تصويب در گروه: اسم و امضاء:**  **تاريخ تصويب در دانشكده: اسم و امضاء:**  **اصلاحات لازم در تعريف پروژه:** |

**توجه**: پروژه حداكثر يك‌ماه و نيم پس از شروع ترمي كه در آن در درس پروژه ثبت نام به عمل آمده است بايد به تصويب برسد.

|  |  |  |
| --- | --- | --- |
| نسخه ۱- دانشكده | نسخه ۲- استاد راهنما | نسخه ۳- دانشجو |

## تعریف مسئله

پیش‌بینی سرعت ترافیک برای کنترل و هدایت آن ضروری است. به علت پیچیدگی و غیرخطی بودن جریان ترافیک روش‌های قدیمی نمی‌توانند ترافیک را برای سفرهایی با زمان طولانی و متوسط به خوبی پیش‌بینی کنند. به دقت و بی درنگ پیش‌بینی کردن سرعت ترافیک برای افراد و سازمان‌های ارائه دهنده‌ی خدمات حمل و نقل و حتی دولت موضوع مهمی است چراکه حمل و نقل نقش مهمی در زندگی هر فرد ایفا می کند و یکی از اصلی‌ترین توانایی‌ها در سیستم ترابری هوشمند (ITS)به شمار می‌رود. در این پروژه سعی داریم با استفاده از آموزش دادن شبکه‌های عصبی عمیق به پیش‌بینی سرعت ترافیک در برخی نقاط مشخص (مانند چهاراه‌ها و میدان ها) بپردازیم. امروزه با استفاده از داده‌های عظیمی که در دسترس است و همچنین پیشرفت سخت‌افزار می‌توانیم شبکه‌های عمیقی که در گذشته قابل آموزش نبودند را آموزش دهیم و از توانایی بالای آن‌ها در پیش بینی مسائل پیچیده استفاده کنیم.

## مرور پیشینه

پیش‌بینی سرعت ترافیک بر مبنای طول سفر به دو دسته‌ی کلی کوتاه (۵ تا ۳۰ دقیقه) ‌و متوسط-طولانی (بیشتر از ۳۰ دقیقه) تقسیم می‌شود. اکثر روش‌های آماری مانند رگراسیون خطی در سفرهای کوتاه مدت عملکرد خوبی دارند، اما به علت عدم قطعیت و پیچیدگی جریان ترافیک این روش‌ها در سفرهای طولانی مدت دقت خوبی ندارند. [1]

مطالعات قبلی بر روی پیش‌بینی سرعت ترافیک در سفرهای بلند مدت را می‌توان به دو دسته‌ی مدل کردن پویا و روش‌های داده محور تقسیم کرد. در روش‌های مدل کردن پویا از ابزار ریاضیاتی، مانند معادلات دیفرانسیل و دانش فیزیکی، برای فرموله کردن مسائل ترافیک استفاده می‌شود [2] . در این روش برای رسیدن به حالت پایدار در پروسه‌ی شبیه‌سازی به سیستم‌های پیچیده و توانایی محاسباتی بسیار بالا نیاز است و فرضیه‌ها برای ساده‌سازی، دقت پیش‌بینی را کاهش می‌دهند. از این رو و به دلیل وجود داده‌ی زیادی که امروزه قابل دسترس است، علاقه به سمت روش‌های داده محور بیشتر است. در روش‌های داده محور مدل‌های آماری کلاسیک و مدل‌های یادگیری ماشین دو نماینده‌ی اصلی هستند.

روش مدل خودهمبسته میانگین متحرک[[2]](#footnote-2) و انواع آن یکی از روش های تجزیه و تحلیل سری زمانی است که مبتنی بر آمار کلاسیک[[3]](#footnote-3) است و در طول زمان بسیار مورد بحث قرار گرفته است [3]، اما این مدل محدود به فرضیات ثابتی درباره‌ی توالی‌های زمانی است و نمی‌تواند ارتباطات زمانی-مکانی را به حساب بیاورد. اخیرا روش‌های یادگیری ماشین در پیش بینی سرعت ترافیک توانسته‌اند مدل‌های آماری کلاسیک را به چالش بکشند.

امروزه روش های یادگیری عمیق با موفقیت بر روی مسايل ترافیکی اعمال شده‌اند. پیشرفت‌های زیادی در این زمینه انجام شده است، مانند شبکه ی باور عمیق[[4]](#footnote-4) [4] و کدکننده خودکار پشته‌ای[[5]](#footnote-5) [5]. اما برای این شبکه‌های متراکم سخت است که بتوانند ویژگی‌های زمانی و مکانی را به طور مشترک از ورودی استخراج کنند. همچنین در هنگام محدودیت و یا غیبت ویژگی‌های مکانی توانایی این شبکه ها تحت تاثیر قرار میگیرد.

## راه حل پیشنهادی

برای درک بهترموضوع همزمان با توضیح راه حل یک مثال را به طور موازی پیش می‌بریم.

هدف از این پروپوزال بیان راه حلی برای پیش بینی سرعت ترافیک دربرخی مختصات ها (مانند تعدادی میدان) در چند گام زمانی آینده است که بدین جهت از سرعت ترافیک در این محل ها در گام های زمانی پیشین استفاده می کنیم. این مساله سری زمانی را به زبان ریاضی به شکل زیر توصیف می‌کنیم:

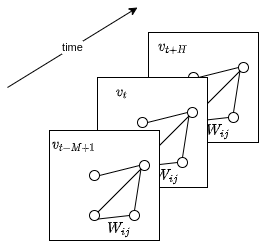
|  |  |  |
| --- | --- | --- |
| (1) |  |  |

|  |  |
| --- | --- |
| یک بردار به طول تعداد نقاطی که قصد داریم سرعت ترافیک را در آن ها پیش بینی کنیم که هر المان شامل سرعت ترافیک در یکی از مختصات های مورد نظر در زمان است. |  |
| تعداد گام های زمانی آینده که می خواهیم سرعت ترافیک را پیش بینی کنیم. |  |
| تعداد گام های زمانی پیشین که برای پیش بینی استفاده می کنیم. |  |

برای مثال فرض کنیم می خواهیم سرعت ترافیک را در میدان فاطمی و میدان فلسطین (دو میدان معروف در تهران) پیشبینی کنیم یک بردار به طول دو خواهد بود که یک عضو آن سرعت ترافیک در میدان فاطمی در زمان مانند ساعت دو بعد از ظهر امروز و عضو دیگر آن شامل همین اطلاعات برای میدان فلسطین خواهد بود. در این مثال را برابر یک و را برابر سه در نظر می گیریم. منظور از (1) این است که می خواهیم سرعت ترافیک در قدم زمانی بعدی را با دانستن قدم زمانی قبلی پیش‌بینی کنیم در این مثال اگر اندازه ی گام زمانی را برابر یک ساعت فرض کنیم، می‌خواهیم سرعت ترافیک در یک ساعت بعدی را با توجه به سه ساعت قبل پیش‌بینی کنیم.

برای پیش‌بینی از هر دو نوع ویژگی زمانی و مکانی بهره می‌بریم. برای حداکثر استفاده از اطلاعات مکانی به جای بخش بخش دیدن شبکه‌ی ترافیک (به عنوان مثال شبکه‌های شطرنجی[[6]](#footnote-6) یا بخش‌ها[[7]](#footnote-7)) آن را به وسیله‌ی یک گراف مدل می‌کنیم، در (1) نشان دهنده‌ی سرعت ترافیک در تمامی گره‌های این گراف در زمان است گره‌های این گراف نقاطی مشخص هستند که سرعت ترافیک در آن‌ها را داشته باشیم که در این مثال نقاط ما میدان فاطمی و میدان فلسطین هستند، سرعت در این نقاط به طور مثال می‌تواند از طریق دوربین‌های سرعت سنج یا از طریق سیستم موقعیت‌‌یاب‌جهانی[[8]](#footnote-8) رانندگان تشخیص داده شود. در نهایت سرعت‌های پیش‌بینی شده با نمایش داده می‌شوند.

(1) را به صورت شهودی می‌توان به وسیله‌ی شکل زیر نشان داد.



شکل 1: گراف مسیر یکسان است و هر دیاگرام نشان دهنده ی جریان ترافیک درگره های این گراف در یک لحظه متفاوت است [1]

در شکل 1 نشان دهنده‌ی میزان ارتباط و همبستگی بین گره‌های و است در این مثال گره‌های و همان میادین یاد شده هستند و نشان می‌دهد که به طور مثال اگر سرعت ترافیک در میدان فاطمی کاهش پیدا کند این موضوع چقدر می‌تواند بر روی سرعت ترافیک در میدان فلسطین تاثیر بگذارد که در این پروژه از رابطه‌ی زیر برای به دست آوردن آن‌ها استفاده می‌کنیم [1]:

|  |  |  |
| --- | --- | --- |
| (2) |  |  |

در (2( فاصله بین تقاطع‌های و می‌باشد که در مثال ما برابر 1.7 کیلومتر است و و پارامترهای ثابت هستند. به وسیله‌ی و تنک بودن[[9]](#footnote-9) ماتریس را کنترل می‌کنیم هرچه را افزایش دهیم باعث می‌شود تنها ارتباطات قوی‌تر را به حساب بیاوریم و ماتریس تنک‌تر می‌شود همچنین هرچه را کاهش دهیم عددی که به عنوان میزان ارتباط محاسبه می‌کنیم کاهش و ماتریس تنک‌تر می‌شود.

با توجه به توضیحات بالا برنامه به دو فایل ورودی احتیاج دارد که اطلاعات درج شده در شکل 2 را در اختیار برنامه قرار دهند. در فایل اول گراف یا به عبارتی وزن یال‌ها شرح داده می‌شود و در دیگری سرعت ترافیک در نودها در بازه‌های زمانی متوالی قرار می‌گیرد . در خروجی جریان ترافیک پیش بینی شده در نودها در قدم زمانی بعدی به ما گفته می‌شود . در مثال ما فایل‌های ذکر شده فرمتی مانند زیر دارند البته اعداد زیر به صورت تصادفی و فرضی هستند.

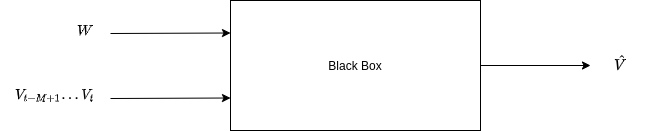
|  |  |  |  |
| --- | --- | --- | --- |
| 2 بعد از ظهر | 1:55 بعد از ظهر | 1:50 بعد از ظهر | زمان مکان |
| km/h 36 | km/h 32 | km/h40 | میدان فلسطین |
| km/h 17 | km/h 27 | km/h 10 | میدان فاطمی |

*مثالی از فرمت فایل ورودی برنامه که شامل سرعت ترافیک در میادین فلسطین و فاطمی در سه قدم زمانی است.*

|  |  |  |
| --- | --- | --- |
| میدان فاطمی | میدان فلسطین |  |
| 316 | 0 | میدان فلسطین |
| 0 | 316 | میدان فاطمی |

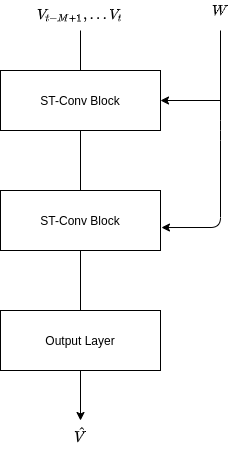
*مثالی از فرمت فایل ورودی برنامه که شامل وزن یالها برای نشان دادن میزان ارتباط مکانی است.*

برای سادگی گراف را بدون جهت در نظر می‌گیریم و در نتیجه فایل ماتریسی متقارن است.



شکل 2: در یک نگاه سطح بالا برنامه با دانستن گراف و جریان ترافیک در گره های این گراف در M واحد زمانی گذشته جریان ترافیک در واحد زمانی بعدی را تخمین زده و برمی‌گرداند.

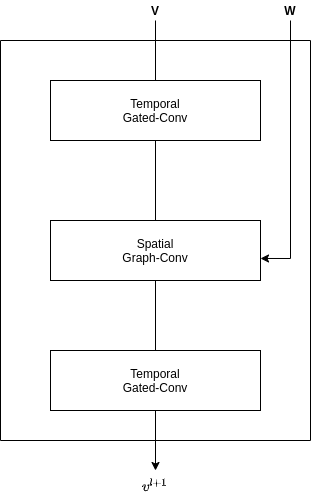
بلوک‌های تشکیل دهنده‌ی نرم افزار را در شکل3 مشاهده می‌کنید**،** ساختار برنامه از دو بلوک کانولوشنی زمانی-مکانی و یک لایه‌ی خروجی کاملا متصل در انتها تشکیل شده است.



شکل 3: شبکه‌ی کانولوشنی زمانی-مکانی بر روی گراف [1]

هر بلوک کانولوشنی زمانی-مکانی شکل 4 از دو لایه‌ی کانولوشنی زمانی تشکیل شده که در ورودی و خروجی قرار گرفته‌اند و یک لایه‌ی کانولوشنی گراف مکانی مانند پلی بین آن دو قرار گرفته است، که می‌تواند با سرعت خوبی اطلاعات مکانی را پس از اعمال پیچش روی گراف به کانولوشن‌های زمانی انتشار دهد.

ارتباطات باقی‌مانده[[10]](#footnote-10) داخل هر بلوک اعمال شده‌اند. لایه‌ی خروجی ویژگی‌­­های جامع به دست آمده را ادغام می‌کند تا پیش بینی نهایی تولید شود.



شکل 4:ساختار بلوک کانولوشنی زمانی-مکانی. گراف مسیرها تنها در لایه‌ی کانولوشنی گراف مکانی استفاده می شود. [1]

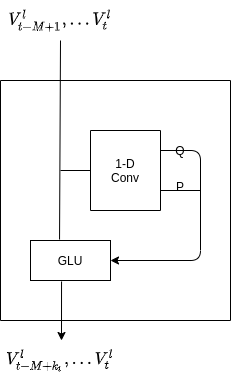
در این پروژه برای استخراج ویژگی‌های مکانی از پیچش[[11]](#footnote-11) روی گراف استفاده می‌کنیم و بدیهی است پیچش استاندارد که معمولا در بحث پردازش تصویر روی تصاویر اعمال می‌کنیم در این مساله قابل استفاده نیست و باید از مدل عمومی‌تری استفاده کنیم. در این پژوهش از کانولوشن روی گراف طیفی[[12]](#footnote-12) استفاده می‌کنیم. [6] با استفاده از )3( می‌توانیم پیچش روی گراف انجام دهیم تا الگوها و ویژگی‌های با‌ معنی را در دامنه‌ی فضا پیدا کند.

|  |  |  |
| --- | --- | --- |
| )3( |  |  |

منظور از عامل کانولوشن روی گراف است. سیگنال گراف و کرنل است. L ماتریس لاپلاسین نرمال شده‌ی گراف و ماتریس بردار ویژه‌های L است. همچنین ماتریس قطری مقدار ویژه‌های L است.

در مثال ذکر شده و و می‌شود المان‌های ماتریس در ابتدا به طور تصادفی انتخاب می‌شوند و خروجی اولین لایه‌ی کانولوشنی زمانی است که در ادامه توضیح می‌دهیم.

در مسائل سری زمانی**،** شبکه‌های عصبی بازگشتی بسیار رایج اند اما در مسایل ترافیک استفاده از این شبکه‌ها بسیار زمان‌بر است. همچنین در ترافیک تغییرات به صورت پویا می‌باشند که این شبکه دیر به این تغییرات جواب می دهد در نتیجه در برای استخراج ویژگی‌های زمانی در این پروژه با الهام از [7]از ساختارهای کاملا پیچشی بر روی محور زمان استفاده می‌کنیم تا بتوانیم رفتار پویای ترافیک بر حسب زمان را دنبال کنیم.



شکل 5:ساختار لایه‌ی کانولوشنی زمانی. [1]

شکل 5 لایه‌ی پیچشی زمانی را نشان می‌دهد که شامل یک پیچش علّی یک بعدی[[13]](#footnote-13) با یک فیلتر به عرض است که پس از آن یک GLU قرار دارد. برای هر گره در گراف g کانولوشن زمانی بر روی تمامی همسایه‌ی ورودی اعمال می‌شود. در این کانولوشنpadding وجود ندارد‌‌، در نتیجه در هر مرتبه باعث کوتاه‌تر شدن توالی‌ها به اندازه‌ی می شود. با این توضیحات می‌توانیم رابطه‌ی زیر را بیان کنیم:

|  |  |  |
| --- | --- | --- |
| )4( |  |  |

کرنل کانولوشن ورودی را به خروجی نگاشت می‌کند، خروجی به دو نیمه قسمت می‌شود (تعداد کانال ها ثابت می‌ماند( و ورودی‌های GLU هستند. نمایشگر عملیات ضرب درایه‌ای است. در اینجا نیز مانند کانولوشن مکانی المان‌های در ابتدا به طور تصادفی انتخاب می‌شوند

در مثالی که با آن پیش می‌رویم ابعاد را می‌توانیم به صورت در نظر بگیریم چراکه بعد اول برابر تعداد کل داده‌هایی است که در اختیار داریم (در اینجا سه) ، بعد دوم برابر تعداد گام‌های زمانی گذشته که قصد داریم برای پیش‌بینی استفاده کنیم مثلا در اینجا آن را دو فرض کردیم و بعد سوم برابر تعداد گره‌های گراف در این مثال 2 و بعد آخرتعداد کانال‌های ورودی لایه که آن را یک فرض کردیم. حال اگر عرض را در این مثال برابر 2 در نظر بگیریم پس از اعمال کانولوشن بعد دوم که برابر با تعداد گام‌های زمانی است به اندازه‌ی 1 واحد کوچک‌تر و برابر با 1 می‌شود. در این مثال برای سادگی فرض کنید اندازه‌ی کانال ورودی و خروجی این لایه با هم برابرند و نیاز به padding وجود ندارد. مقادیر کرنل به طور تصادفی انتخاب می‌شوند و نوشتن اعداد از این مرحله به بعد هیچ سندیتی ندارد در نتیجه آن را متوقف می‌کنم.

در نهایت بلوک‌های کانولوشنی زمانی-مکانی از رابطه ی زیر پیروی می‌کنند:

|  |  |  |
| --- | --- | --- |
| )5( |  |  |

و به ترتیب کرنل لایه‌های زمانی بالایی و پایینی در بلاک هستند کرنل پیچش روی گراف است.

بعد از روی هم قرار دادن دو بلوک پیچشی زمانی-مکانی یک لایه‌ی کاملا متصل به عنوان لایه‌ی خروجی در انتها قرار می دهیم.

## ابزارها و امکانات مورد نیاز

مرتبه زمانی روش پیچش روی گراف به این دلیل که امکان استفاده از تبدیل سریع فوریه[[14]](#footnote-14) وجود ندارد،

می‌باشد و اعمال این روش بدون داشتن تجهیزات گران قیمتی مانند پردازنده‌های گرافیکی زمان قابل توجهی است. حداقل امکانات مورد نیاز برای اجرای این پروژه عبارت اند از:

* CPU: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
* GPU: NVIDIA GeForce GTX 1080

## ارزیابی

در نهایت به منظور ارزیابی پروژه از داده‌های ترافیکی واقعی PeMSD7 استفاده می‌کنیم که توسط اداره‌ی حمل و نقل ایالت کالیفرنیا[[15]](#footnote-15) جمع آوری شده است. این داده مربوط به ۲۲۸ ایستگاه می‌باشد که به صورت تصادفی از میان ۳۹ هزار ایستگاه‌ انتخاب شده‌اند و شامل ۱۲ هزار سطر است که نشان دهنده‌ی جریان ترافیک در این ۲۲۸ ایستگاه با توالی زمانی پنج دقیقه می‌باشد. داده‌ها در ماه‌های می تا جون سال ۲۰۱۲ جمع آوری شده‌اند و جهت پاک کردن داده‌های ترافیکی بی‌قاعده روزهای غیرکاری حذف شده‌اند. برای مقایسه‌ی عملکرد این روش با روش‌های دیگر از معیارهای میانگین مطلق خطا[[16]](#footnote-16)، میانگین مطلق درصد خطا[[17]](#footnote-17) و جذر میانگین مربعات خطا[[18]](#footnote-18) استفاده می‌کنیم. این روش را با روش‌های پایه‌ی شبکه‌ی عصبی پیش‌خور[[19]](#footnote-19)، مدل خودهمبسته میانگین متحرک و حافظه‌ی کوتاه مدت ماندگار کاملا متصل[[20]](#footnote-20) [8] مقایسه خواهیم کرد.

## زمان بندی انجام پروژه

|  |  |
| --- | --- |
| مدت زمان | فعالیت |
| سه ماه | پیاده سازی |
| یک ماه | آموزش مدل و ارزیابی |
| دو ماه | بهبود‌های احتمالی ( از نظر مرتبه‌ی زمانی) |
| دو ماه | جمع بندی |

## لیست مراجع

|  |  |
| --- | --- |
| [1] | B. Yu, H. Yin and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," *arXiv preprint arXiv:1709.04875,* 2018. |
| [2] | E. I. Vlahogianni, "Computational intelligence and optimization for transportation big data:challenges and opportunities," in *Engineering and Applied Sciences Optimization*, Springer, 2015, pp. 107-128. |
| [3] | B. M. Williams and L. A. Hoel, "Modeling and forecasting vehicular traffic flow as a seasonal arima process: Theoretical basis and empirical results," *Journal of transportation engineering,129(6):664–672,* 2003. |
| [4] | Y. Jia, J. Wu and Y. Du, "Traffic speed prediction using deep learning method," in *IEEE International Conference on Intelligent Transportation*, 2016. |
| [5] | Q. Chen, X. Song, H. Yamada and R. Shibasaki, "Learning deep representation from big and heterogeneous data for traffic accident inference," in *AAAI Conference on Artificial Intelligence*, 2016. |
| [6] | J. Bruna, W. Zaremba, A. Szlam and Y. LeCun, "Spectral networks and locally connected networks on graphs," *arXiv preprint arXiv:1312.6203,* 2013. |
| [7] | J. Gehring, M. Auli, D. Grangier, D. Yarats and Y. N. Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122,* 2017. |
| [8] | I. Sutskever, O. Vinyals and Q. V. Le, "Sequence to sequence learning with neural networks," 2014. |

1. Email: [elahe.dstn@gmail.com](mailto:elahe.dstn@gmail.com) [↑](#footnote-ref-1)
2. Auto-Regressive Integrated Moving Average [↑](#footnote-ref-2)
3. Classical statistics [↑](#footnote-ref-3)
4. deep belief network [↑](#footnote-ref-4)
5. stacked auto encoder [↑](#footnote-ref-5)
6. grids [↑](#footnote-ref-6)
7. segments [↑](#footnote-ref-7)
8. GPS [↑](#footnote-ref-8)
9. sparsity [↑](#footnote-ref-9)
10. Residual connection [↑](#footnote-ref-10)
11. Convolution [↑](#footnote-ref-11)
12. spectral graph convolution [↑](#footnote-ref-12)
13. 1-D causal convolution [↑](#footnote-ref-13)
14. FFT [↑](#footnote-ref-14)
15. https://pems.dot.ca.gov/ [↑](#footnote-ref-15)
16. MAE [↑](#footnote-ref-16)
17. MAPE [↑](#footnote-ref-17)
18. RMSE [↑](#footnote-ref-18)
19. Feed-Forward Neural Network [↑](#footnote-ref-19)
20. Full-Connected LSTM [↑](#footnote-ref-20)