UTILIZING NEURAL TEMPORAL POINT PROCESSES TO FORECAST SUBSEQUENT EVENTS IN DIABETIC RETINOPATHY AND ASSOCIATED MEDICATIONS

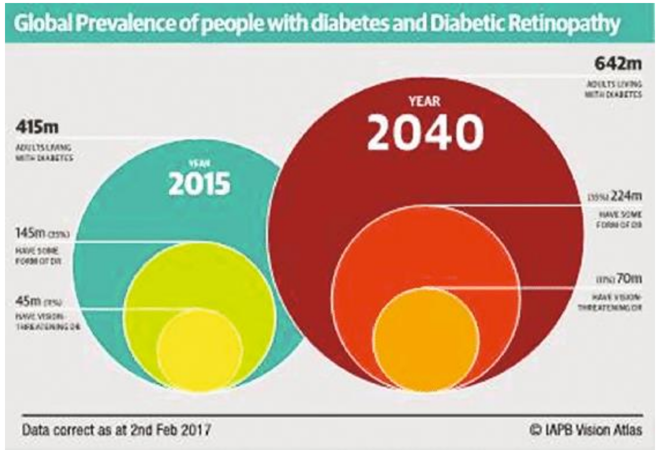
A Novel Approach for Forecasting Diabetic Retinopathy Events /Medications using Neural Temporal Point Processes

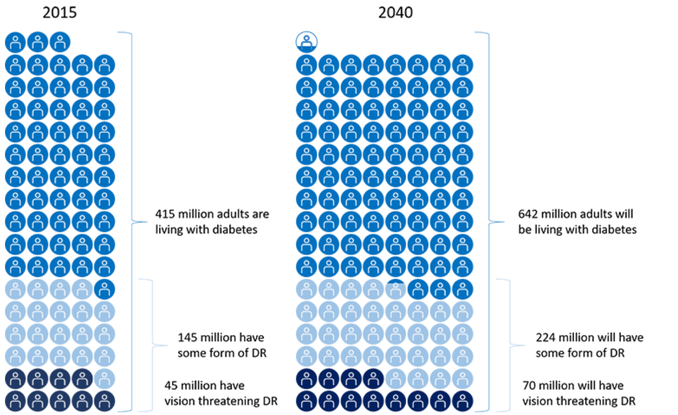
Neeraj Yadav

Supervisor: Dr. Michael Barnes

MSc. Big Data Science

DIABETIC RETINOPATHY: A GLOBAL CONCERN





DIABETIC RETINOPATHY (DR) IS A DOMINANT MICROVASCULAR COMPLICATION OF DIABETES, STANDING AS A PRIMARY FACTOR IN GLOBAL VISION IMPAIRMENT

TIMELY DETECTION OF DR IS PARAMOUNT; EARLY INTERVENTION CAN SIGNIFICANTLY PREVENT OR REDUCE VISION LOSS, SAFEGUARDING QUALITY OF LIFE.

TEMPORAL POINT PROCESS (TPP) AND ELECTRONIC HEALTH RECORDS (EHR)

EHRs have revolutionized patient care by digitizing comprehensive health data. Central to modern healthcare, these records not only streamline patient management but also introduce intricate temporal patterns, demanding advanced analytical approaches

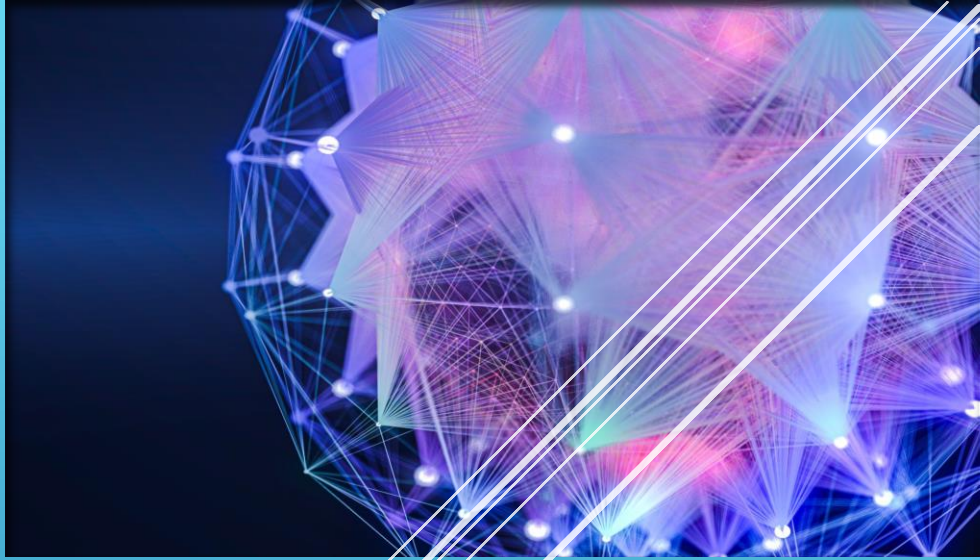
TPPs are statistical models that capture the intricacies in the timing of event sequences. Historically applied in fields like seismology and finance, they've recently risen to prominence in healthcare analytics, optimizing the modeling of event data within Electronic Health Records

Temporal Point Process is defined by its conditional intensity function λ(t) , which represents the instantaneous event rate at time t, given the history of events up to time t. Formally, the conditional intensity function λ(t) is defined as:

$$\lambda(t) = \lim_{\Delta t \to 0} \frac{P(\text{Event in } [t, t+\Delta t) \mid \text{History up to } t)}{\Delta t}$$

where P(·) represents the probability, and the history up to time t is denoted by the sequence of past event times

NEURAL TEMPORAL POINT PROCESSES: BRIDGING TPPS & NEURAL NETWORKS



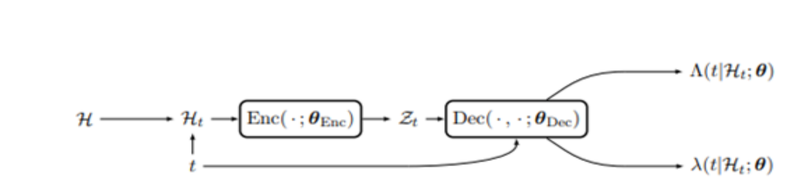
Core Concept:

Merging traditional TPP models with neural network architectures, Neural TPPs masterfully capture intricate, nonlinear temporal dependencies in event data. Beyond healthcare, they've proven pivotal in Natural Language Processing tasks.

Encoder-Decoder Architecture:

Encoder: Processes event sequences, compressing temporal data into a context vector. Consists of an Embedding Layer and an RNN Layer.

Decoder: Generates contextually and temporally coherent output sequences, driven by the context vector. Comprises the Conditional Intensity Function Estimation Layer and the Event Generation Mechanism.



Overview of Previous Work on Neural TPPs

CLOSED FORM LIKELIHOOD:
Du et al. (2016) The Hawkes process stands out, with RMTPP by as an exemplar, although with some limitations in capturing exponential dependencies."

ANALYTIC CONDITIONAL INTENSITY:
MEI AND EISNER (2017) AND ZHU ET AL. (2020) This approach leverages NNs and embraces flexibility without needing closed-form cumulative intensity.

ANALYTIC CONDITIONAL CUMULATIVE INTENSITY:
OMI ET AL. (2019) Utilized an MLP for a monotonic decoder, emphasizing positivity through activation functions.

ATTENTION-BASED NEURAL TPPS:
ENGUEHARD ET AL. (2020) Presented a deep dive into encoder-decoder choices for Neural TPPs in EHRs, underscoring the nuances and challenges of decoder selection.

SYNTHEA DATASET OVERVIEW

Synthea Dataset: A Synthesis of Real-World Medical Scenarios, crafted through human-expert-curated Markov processes. Rich in diagnoses, treatments, and healthcare event timelines, it offers a granular look into the world of Diabetic Retinopathy.

Dataset Features:

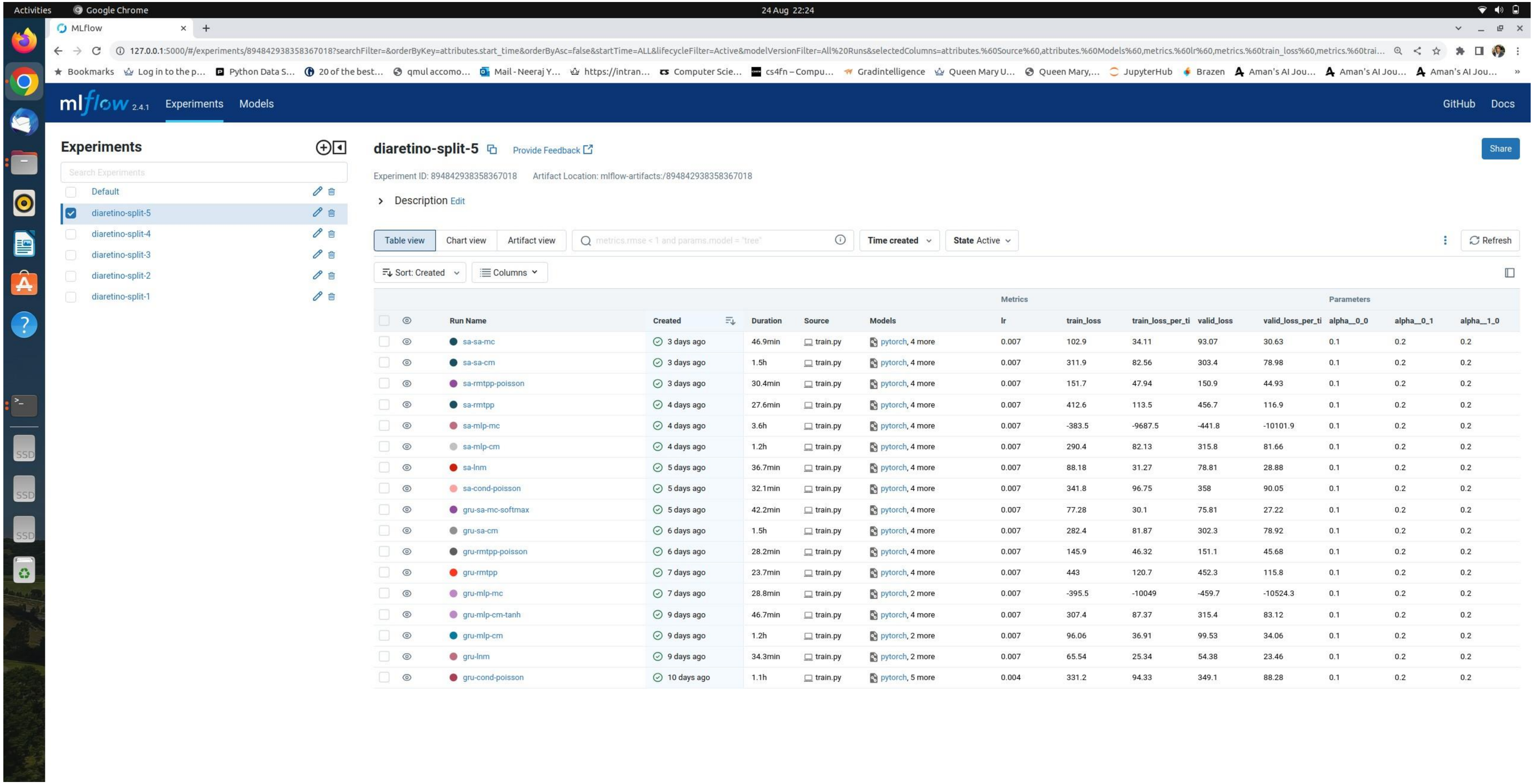
- Authenticity: Mimicking real-world medical standards, Synthea guarantees realistic data, echoing genuine healthcare scenarios.
- Patient Profiles: From age to specific encounter details, the dataset paints a full picture of each patient.
- Clinical Chronicles: In-depth records span blood glucose levels, eye check-ups, and treatments, providing a clinical narrative for every patient.
- Temporal Insights: Time-stamped events allow us to dissect and model the chronological progression of clinical events linked to Diabetic Retinopathy

NEURAL TEMPORAL POINT PROCESS MODELS USED

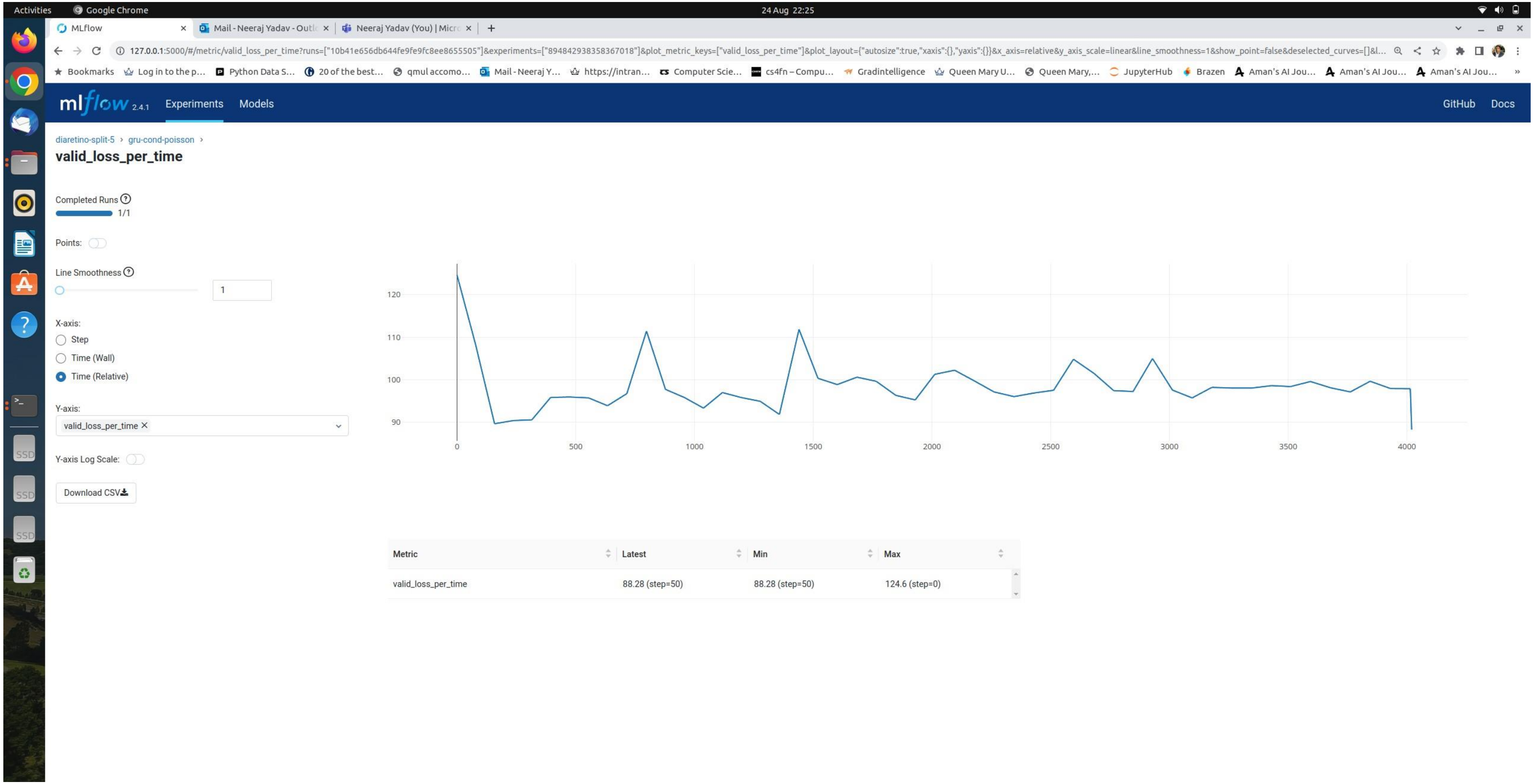
- **Encoder:**
  - **Recurrent Neural Networks (RNNs):** "Processes sequences, capturing temporal EHR dependencies."
- **Decoder Approaches:**
  - **Closed Form Likelihood:** "Employs models with closed-form likelihood e.g., Hawkes Process."
  - **Analytic Conditional Intensity:** "Uses Neural Network to approximate conditional intensity."
  - **Analytic Conditional Cumulative Intensity:** "Designs decoder to approximate conditional cumulative intensity."
- **Base Intensity:**
  - **Description:** "Characterizes inherent event rate, either constant or dynamic."
- **Selected Models:**
  - **GRU-Cond-Poisson:** "GRU integrated with a conditional Poisson process."
  - **GRU-LNM:** "Combines GRU with a Log-Normal Mixture."
  - **GRU-MLP-CM:** "GRU with MLP, employs a conditional mean approach."
  - **GRU-RMTPP:** "Pairs GRU with the RMTPP process."
  - **GRU-SA-CM:** "GRU with self-attention and a conditional mean approach."
  - **SA-Cond-Poisson:** "Self-attention mechanism combined with a conditional Poisson process."
  - **SA-LNM:** "Merges self-attention with a Log-Normal Mixture."
  - **SA-MLP-CM:** "Self-attention and MLP with a conditional mean approach."
  - **SA-RMTPP:** "Self-attention mechanism paired with the RMTPP process."

- Activate environment: conda env create
- source activate env_tpp
- Install dependencies: pip install -e .
- Running the Model:
- Without MLflow: python scripts/train.py --no-mlflow
- With MLflow: Set directories:
- MLFLOW_BACKEND_STORE
- MLFLOW_DEFAULT_ARTIFACT_ROOT
- Set host location:
- MLFLOW_HOST=0.0.0.0, MLFLOW_PORT=5000
- Start MLflow server:
- mlflow server
- Run with remote server URI:
- --remote-server-uri
- General Usage:
- Training:
- Scripts/train.py
- Select encoder/decoder types and configurations.
- Evaluation:
- scripts/evaluate.py
- Specify --model-dir MODEL_DIR
- Python environment to be correctly set up.
- Wide flexibility in encoder/decoder choices.

INSTALLATION AND RUNNING MODELS



Run Name	Created	Duration	Source	Models	lr	train_loss	train_loss_per_ti	valid_loss	valid_loss_per_ti	alpha__0_0	alpha__0_1	alpha__1_0
sa-sa-mc	3 days ago	46.9min	train.py	pytorch, 4 more	0.007	102.9	34.11	93.07	30.63	0.1	0.2	0.2
sa-sa-cm	3 days ago	1.5h	train.py	pytorch, 4 more	0.007	311.9	82.56	303.4	78.98	0.1	0.2	0.2
sa-rmtpp-poisson	3 days ago	30.4min	train.py	pytorch, 4 more	0.007	151.7	47.94	150.9	44.93	0.1	0.2	0.2
sa-rmtpp	4 days ago	27.6min	train.py	pytorch, 4 more	0.007	412.6	113.5	456.7	116.9	0.1	0.2	0.2
sa-mlp-mc	4 days ago	3.6h	train.py	pytorch, 4 more	0.007	-383.5	-9687.5	-441.8	-10101.9	0.1	0.2	0.2
sa-mlp-cm	4 days ago	1.2h	train.py	pytorch, 4 more	0.007	290.4	82.13	315.8	81.66	0.1	0.2	0.2
sa-lnm	5 days ago	36.7min	train.py	pytorch, 4 more	0.007	88.18	31.27	78.81	28.88	0.1	0.2	0.2
sa-cond-poisson	5 days ago	32.1min	train.py	pytorch, 4 more	0.007	341.8	96.75	358	90.05	0.1	0.2	0.2
gru-sa-mc-softmax	5 days ago	42.2min	train.py	pytorch, 4 more	0.007	77.28	30.1	75.81	27.22	0.1	0.2	0.2
gru-sa-cm	6 days ago	1.5h	train.py	pytorch, 4 more	0.007	282.4	81.87	302.3	78.92	0.1	0.2	0.2
gru-rmtpp-poisson	6 days ago	28.2min	train.py	pytorch, 4 more	0.007	145.9	46.32	151.1	45.68	0.1	0.2	0.2
gru-rmtpp	7 days ago	23.7min	train.py	pytorch, 4 more	0.007	443	120.7	452.3	115.8	0.1	0.2	0.2
gru-mlp-mc	7 days ago	28.8min	train.py	pytorch, 2 more	0.007	-395.5	-10049	-459.7	-10524.3	0.1	0.2	0.2
gru-mlp-cm-tanh	9 days ago	46.7min	train.py	pytorch, 4 more	0.007	307.4	87.37	315.4	83.12	0.1	0.2	0.2
gru-mlp-cm	9 days ago	1.2h	train.py	pytorch, 2 more	0.007	96.06	36.91	99.53	34.06	0.1	0.2	0.2
gru-lnm	9 days ago	34.3min	train.py	pytorch, 2 more	0.007	65.54	25.34	54.38	23.46	0.1	0.2	0.2
gru-cond-poisson	10 days ago	1.1h	train.py	pytorch, 5 more	0.004	331.2	94.33	349.1	88.28	0.1	0.2	0.2

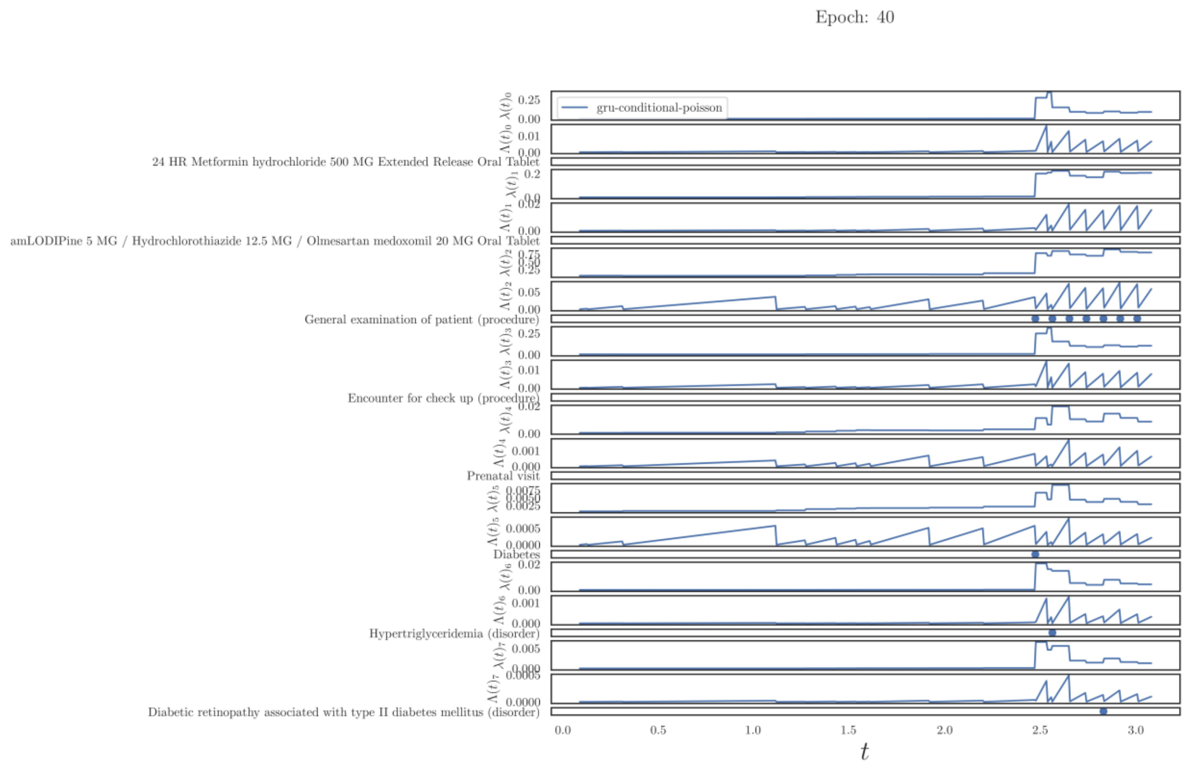

MLflow
127.0.0.1:5000/#/metric/valid_loss_per_time
mlflow 2.4.1
Experiments
Models
GitHub
Docs
diaretino-split-5 › gru-cond-poisson ›
valid_loss_per_time
Completed Runs
1/1
Points:
Line Smoothness
1
X-axis:
Step
Time (Wall)
Time (Relative)
Y-axis:
valid_loss_per_time
Y-axis Log Scale:
Download CSV
120
110
100
90
0
500
1000
1500
2000
2500
3000
3500
4000
Metric
Latest
Min
Max
valid_loss_per_time
88.28 (step=50)
88.28 (step=50)
124.6 (step=0)

ANALYZING MODEL RESULTS: EVALUATING PERFORMANCE METRICS

Table 2: Evaluation on task

Encoder GRU	ROC-AUC	NLL/time
Decoder		
CP	**0.859**	90.252
RMTPP	0.759	126.499
LNM	0.768	**25.644**
MLP-CM	0.504	86.553
MLP-MC	0.702	46.67
ATTN-CM	0.506	83.111
ATTN-MC	0.793	36.704
Encoder SA		
CP	0.828	92.936
RMTPP	0.728	113.703
LNM	0.752	32.566
MLP-CM	0.504	86.135
MLP-MC	0.697	48.90
ATTN-CM	0.506	83.849
ATTN-MC	**0.776**	**43.491**

- We evaluated multiple models on the Synthea dataset, benchmarking them on two pivotal metrics: ROC-AUC and NLL.
- ROC-AUC Weighted: "This metric offers insights into a model's classification prowess, taking into account class prevalence in the dataset.
- NLL/Time: A measure of the model's efficiency in event timing prediction, while maintaining a low log-likelihood loss.
- The GRU-Cond-Poisson model emerged as a leader in the ROC-AUC Weighted score with an impressive 0.859, highlighting its exceptional classification abilities. Models like SA-Cond-Poisson and GRU-SA-MC also fared well with scores of 0.828 and 0.793. However, models such as GRU-MLP-CM and SA-MLP-CM, with scores around 0.50, indicate room for refinement.
- The GRU-LNM model shined in NLL/Time with a score of 25.644, showcasing its prediction efficiency. The SA-LNM closely followed with 32.566. Yet, the GRU-RMTPP, with a high score of 126.499, points to challenges in accurate event timing predictions."
- Attention-driven Neural Temporal Point Processes stand out for their interpretability with similar results. These models, while being interpretable, showcase results that are comparable to other top-performing models
- While many models showcased prowess in specific areas, an ideal model balances classification accuracy with prediction efficiency. In this light, the GRU-Cond-Poisson and GRU-LNM models stand out as top performers.



INTERPRETING THE MODEL'S PREDICTIONS

- **Disease Intensities Over Time:** The spikes in the curves signal the predicted likelihood of each disease or medication event at specific intervals.
- **Comparison with Ground Truth:** Actual observed events align alongside predicted intensities, allowing a direct model accuracy assessment."
- **Interplay of Events:** Simultaneous intensity spikes in multiple events hint at possible correlations or causal relationships."
- **Clinical Implications:** High predicted intensities for adverse events spotlight periods demanding enhanced monitoring or strategic treatment shifts.
- **Model Insights:** Peaks or troughs shed light on the model's perception of patterns, while discrepancies between predictions and reality indicate potential refinement areas.

The model's visual output provides deep insights into the progression of conditions linked to diabetic retinopathy. From diabetes onset, through Hypertriglyceridemia development, culminating in Diabetic Retinopathy, the sequence highlights the interconnectedness of these ailments. A curious correlation emerges between neonatal visits and diabetes, hinting at neonatal events influencing diabetes risks. Medication intensity spikes echo the recurring nature of treatment, possibly hinting at symptom flare-ups

THANK YOU