

Building Energy Efficient Traffic Congestion Prediction System

Speaker

MARRIMANU Charan Teja (m5282019)
Department of Computer Science
The University of Aizu

Supervisor

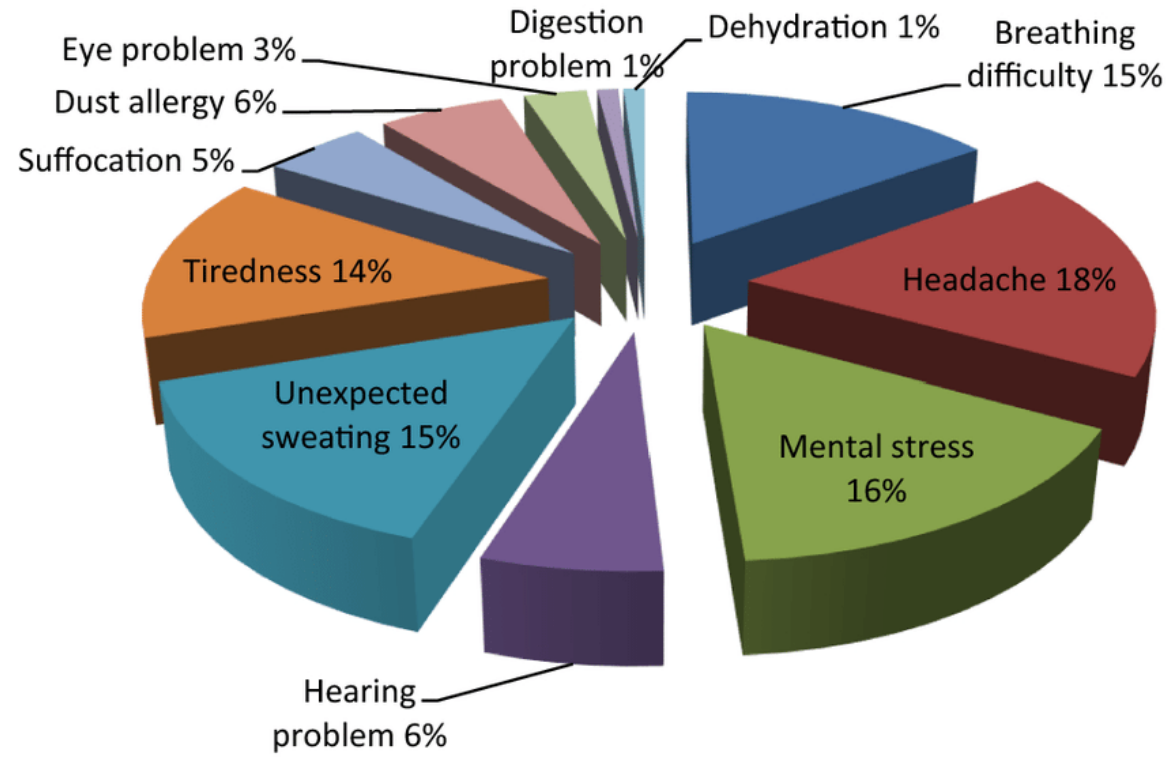
RAGE UDAY KIRAN
Associate Professor
The University of Aizu

Outline

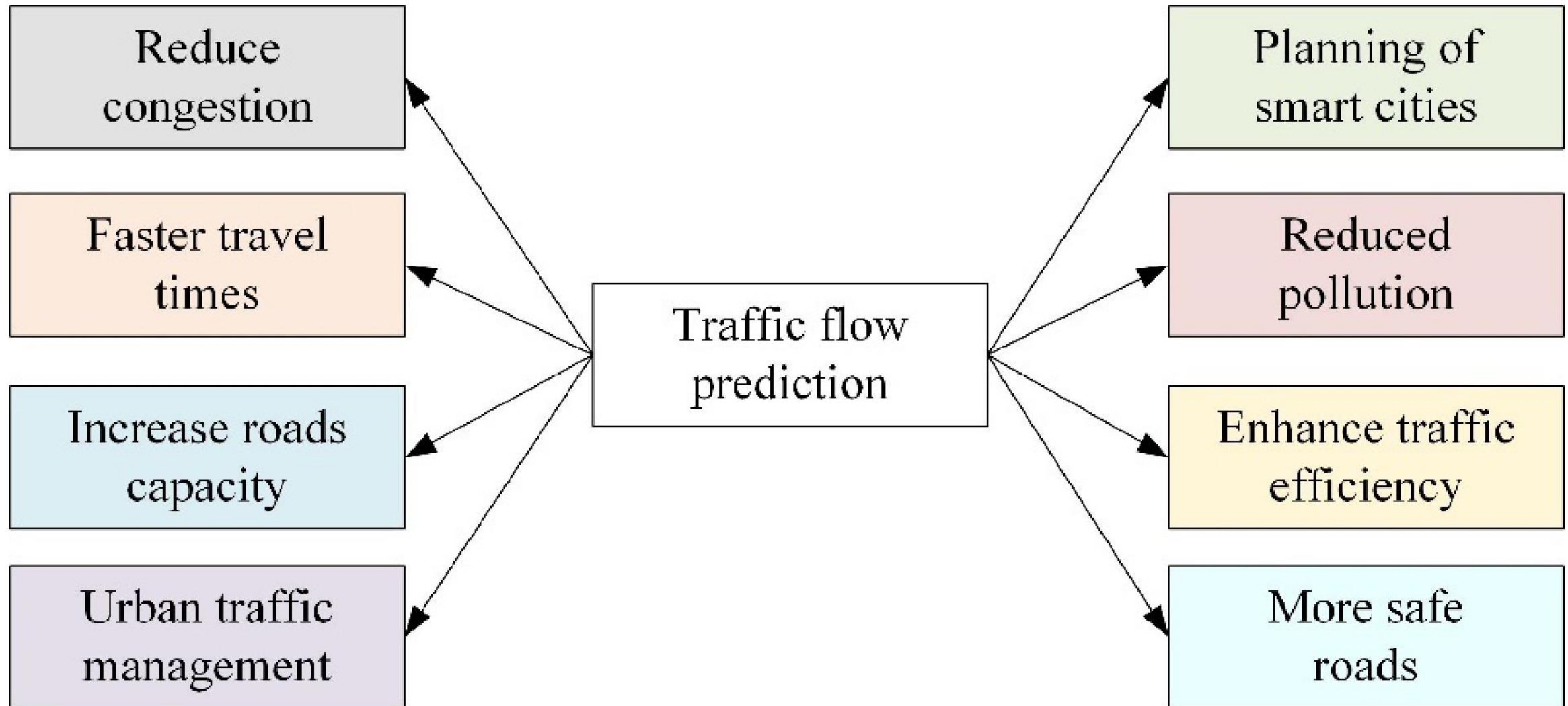
- Importance of Predicting Traffic Congestion
- Problem Definition
- Related Work
- Limitations
- Current Progress
- Future Work

Importance

- Traffic Congestion Prediction Systems (TCPS) are crucial to
 - Achieve Sustainable Developmental Goals (SDGs)
 - Reduce loss of life
 - Improve health
 - Faster travel
 - Reduce pollution
 - Reduce transportation costs



Applications

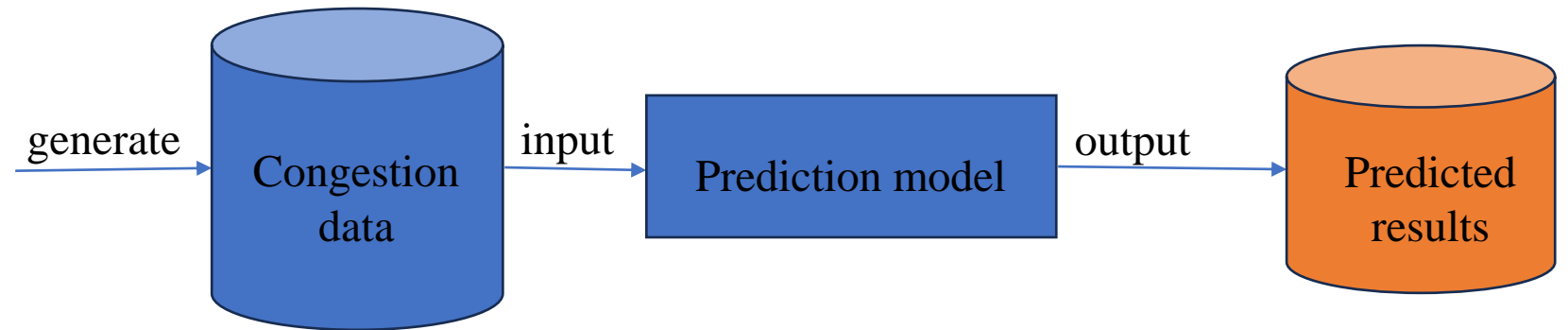


Problem Definition

- From the historical congestion data of a large transportation network, predict the future traffic congestion effectively.
 - Low Error values
 - Fast results



Large scale transportation network



Related Work

S.No	Author Name	Year	Methodology	Model	Dataset	Result	Limitation
1	Ismail Abiodun Sulaimon et al.	2021	Traffic-Related Air Pollutant (TRAP) Prediction using Big Data and Machine Learning	Extra Trees Regressor, Hist Gradient Boosting Regressor	Meteorological data	Extra Trees Regressor- 93% Hist Gradient Boosting Regressor – 92%	Performance is nothing close to the best performing ML algorithm in this category
2	Moe Myint Mo et al.	2023	Measuring Traffic Congestion Based on the Taxi Operations of Traditional and On-Demand Taxis in Yangon	Aggregated model	Collection of GPS Data	The result shows that the “traditional taxis” stay in one place longer than the “on demand taxis.”	It is difficult to validate the occupied, vacant, intersection, and waiting times.
3	Wei Cheng et al.	2022	Combination predicting model of traffic congestion index in weekdays based on Light GBM-GRU	Gated Recurrent Unit	Gaia Open Dataset	An accuracy of more than 90%	The limitation of collection technology makes the lack of traffic characteristics, which leads to the error of prediction.
4	Navin Ranjan et al.	2021	Large-Scale Road Network Congestion Pattern Analysis and Prediction Using Deep Convolutional Autoencoder	Convolution Autoencoder	dataset from Seoul city	GRU – 98.32%	we remove the data from some particular day due to missing data caused by an error or failure of the web crawling program

Related Work

S.No	Author Name	Year	Methodology	Model	Dataset	Result	Limitation
5	Honglei Ren et al.	2018	A Deep Learning Approach to the Citywide Traffic Accident Risk Prediction	LSTM Model	Big traffic accident data	LSTM - 90.3%	One limitation of these works is that, they did not incorporate several importance factors such as traffic flow, weather condition, air quality into their model.
6	Chantakarn Pholpol et al.	2021	Traffic Congestion Prediction using Deep Reinforcement Learning in Vehicular Ad-Hoc Networks (VANETS)	Reinforcement Learning	Road Traffic Data	Reinforcement Learning – 95%	VANET itself has a limitation and also lack of flexibility to find the congested route and to reroute to the alternative paths.
7	Mahmuda Akhtar et al.	2021	A Review of Traffic Congestion Prediction Using Artificial Intelligence	Hidden Markov Model	Stationary data and Probe data	HMM – 82%	No study has provided any reasonable logic on selecting the membership function, which is a significant limitation of fuzzy logic models.
8	Kadda Beghdad bey et al.	2024	Improving Road Traffic Speed Prediction Using Data Augmentation: A Deep Generative Models-based Approach	Deep Generative Model	Road Traffic Dataset	Deep Generative Model – 87%	Traffic datasets are often small

Limitations of Existing Studies

- Existing ML/DL models are based on single task learning:
 - build model for each road segment independently
 - Limited ability to share knowledge across related tasks or road segments
- Problems of Single Task Learning
 - Overfitting
 - High Computational Cost
 - Scalability Issues
- Proposed Solution: Multi-Task Learning

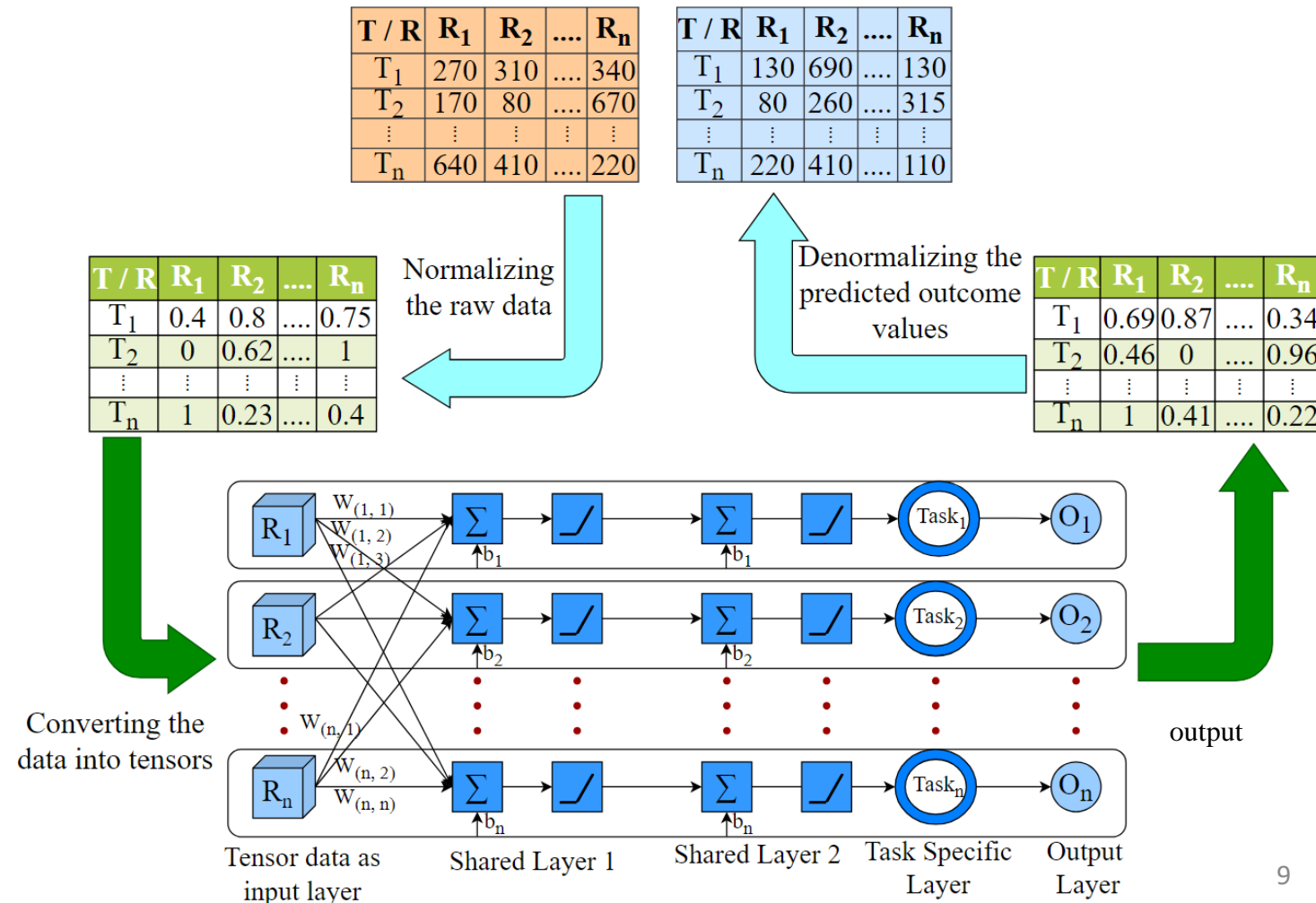
Proposed Multi-Task Learning Model

MTL uses shared information across multiple tasks to improve performance.

Input: Tensor data of normalized traffic congestion lengths.

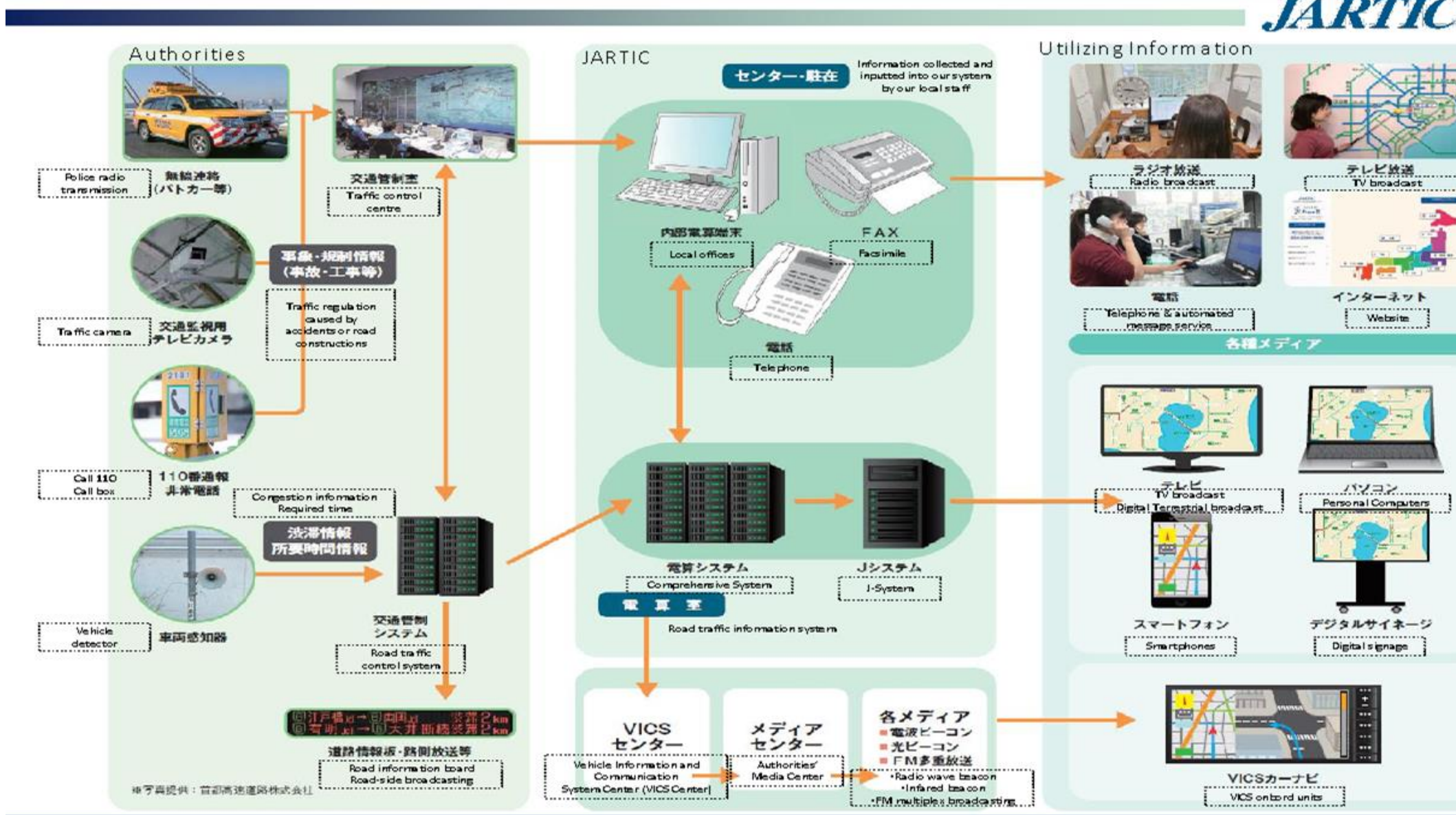
Layers:

- Two shared hidden layers with Linear + ReLU activation.
- Task-specific layers generating predictions for each road segment.
- Output layer for normalized predictions.



About JARTIC system

JARTIC

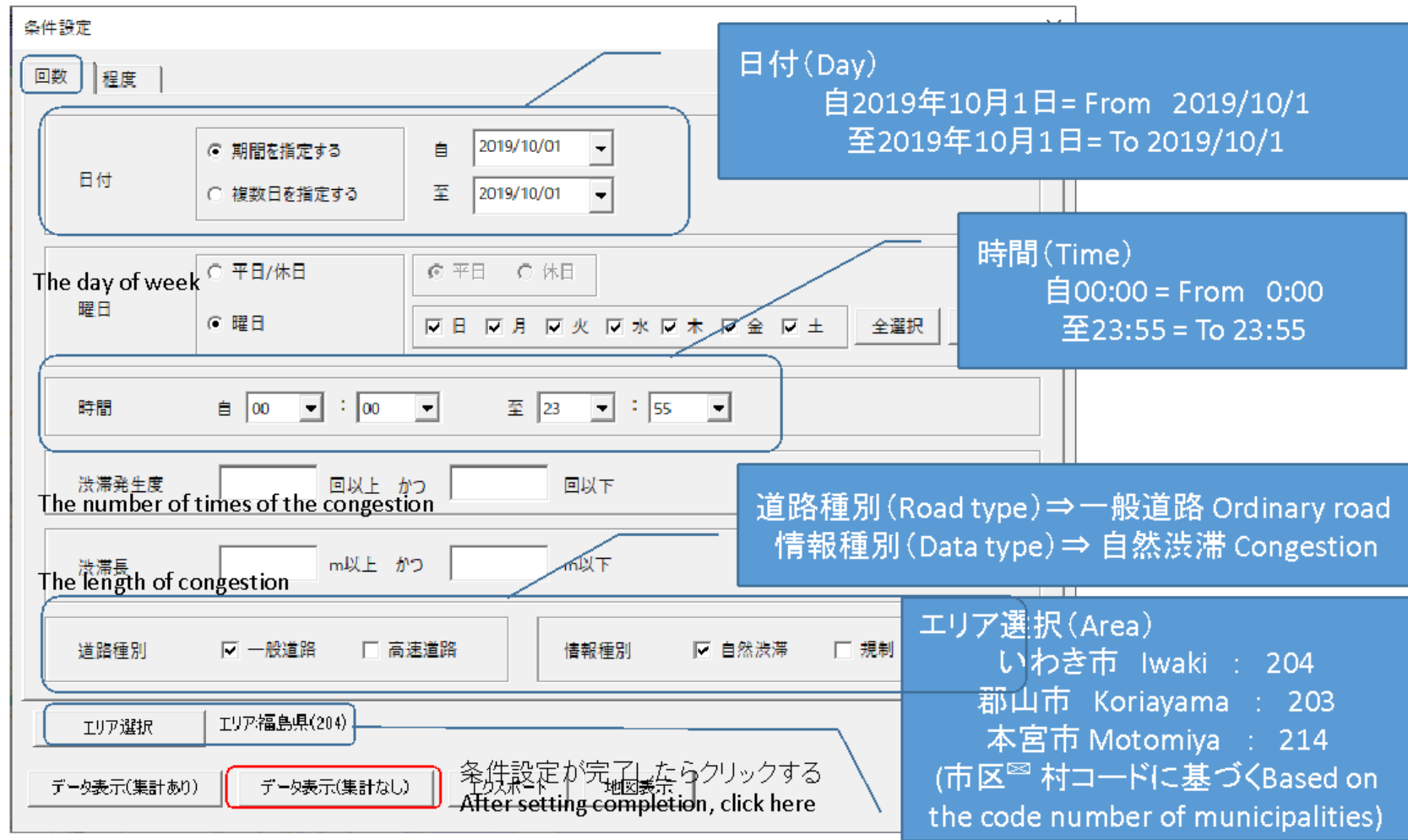


System usage example

To search total amount of congestion for 288 cycle (0:00, 0:05, ..., 23:55) in Iwaki on 1-Oct-2019, setting the setting screen as in the below drawing.

※渋滞量… 渋滞の規模を表す指標。渋滞長×渋滞発生時間で求める

※Amount of congestion… An index expressing the scale of congestion, multiplying the length of congestion and duration time of congestion



The screenshot shows the '条件設定' (Condition Setting) screen. It includes tabs for '回数' (Frequency) and '程度' (Degree). The '日付' (Date) section has radio buttons for '期間を指定する' (Specify period) and '複数日を指定する' (Specify multiple days), with date pickers for '自' (From) and '至' (To). The '曜日' (Day of week) section has radio buttons for '平日/休日' (Weekday/Weekend) and '曜日' (Day of week), with checkboxes for each day of the week. The '時間' (Time) section has time pickers for '自' (From) and '至' (To). The '渋滞発生度' (Congestion occurrence degree) section has input fields for '回数以上' (Number of times above) and '回数以下' (Number of times below). The '渋滞長' (Congestion length) section has input fields for 'm以上' (m above) and 'm以下' (m below). The '道路種別' (Road type) section has checkboxes for '一般道路' (Ordinary road) and '高速道路' (Expressway). The '情報種別' (Data type) section has checkboxes for '自然渋滞' (Natural congestion) and '規制' (Regulation). The 'エリア選択' (Area selection) section has a dropdown menu for 'エリア' (Area) and a text input for '福島県(204)'. At the bottom, there are buttons for 'データ表示(集計あり)' (Data display (with summary)), 'データ表示(集計なし)' (Data display (without summary)), and '条件設定が完了したらクリックする' (Click here after setting completion).

条件設定

回数 | 程度

日付

期間を指定する
複数日を指定する

自 2019/10/01 至 2019/10/01

日付(Day)
自2019年10月1日= From 2019/10/1
至2019年10月1日= To 2019/10/1

The day of week

平日/休日
曜日

平日 休日

日 月 火 水 木 金 土 全選択

時間

自 00 : 00 至 23 : 55

時間(Time)
自00:00 = From 0:00
至23:55 = To 23:55

渋滞発生度

回数以上 回数以下

The number of times of the congestion

渋滞長

m以上 m以下

The length of congestion

道路種別

一般道路 高速道路

情報種別

自然渋滞 規制

道路種別(Road type)⇒一般道路 Ordinary road
情報種別(Data type)⇒自然渋滞 Congestion

エリア選択(Area)

いわき市 Iwaki : 204
郡山市 Koriyama : 203
本宮市 Motomiya : 214
(市区 村コードに基づくBased on the code number of municipalities)

エリア福島県(204)

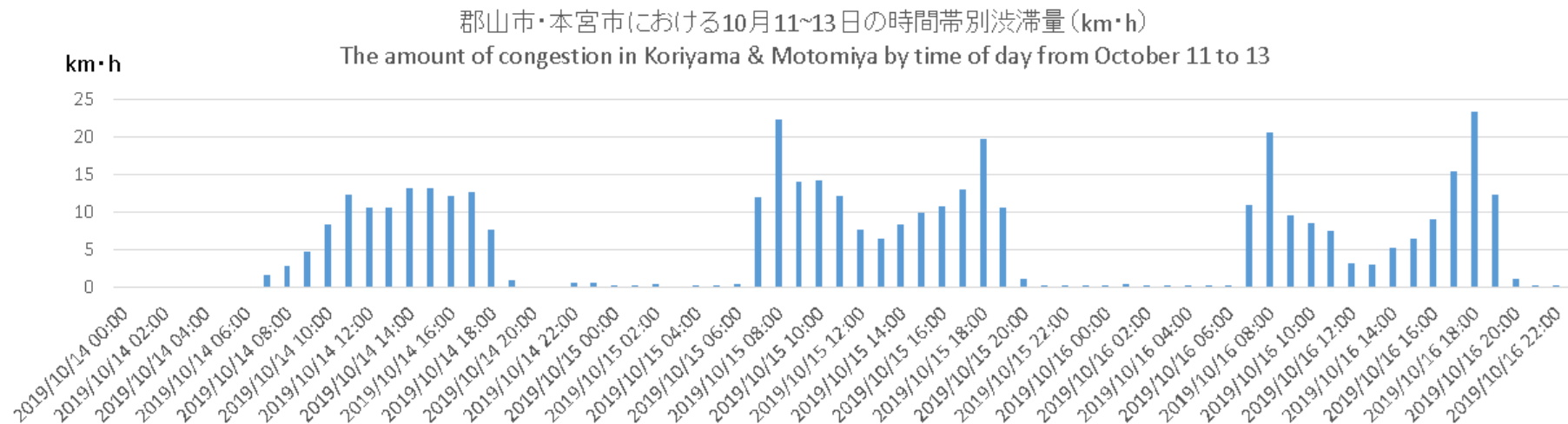
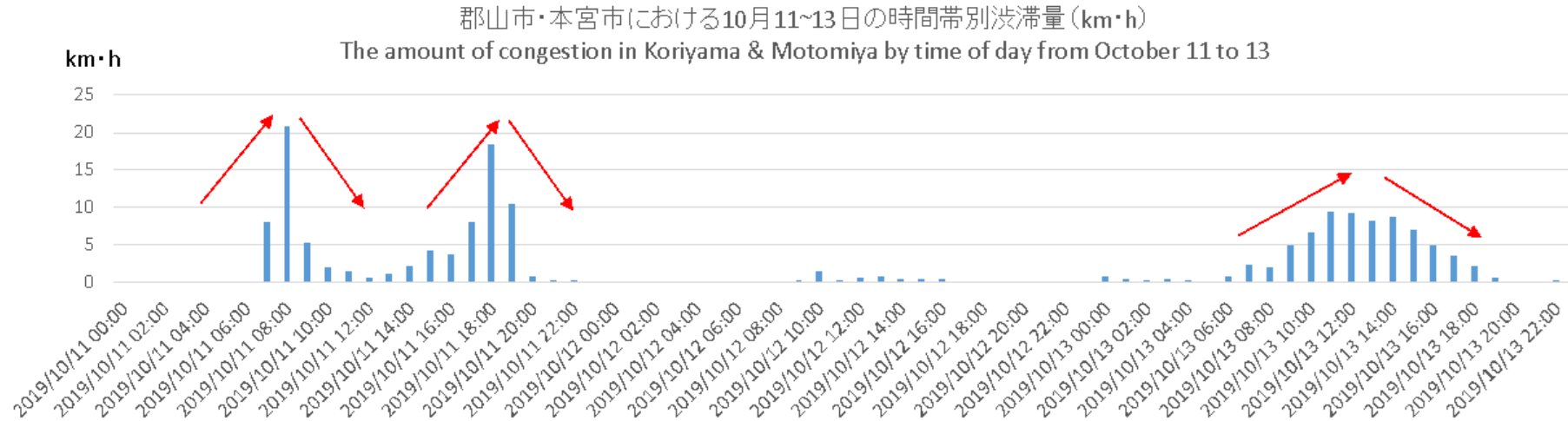
データ表示(集計あり) データ表示(集計なし)

条件設定が完了したらクリックする
After setting completion, click here

各検索条件はAND条件です
=
And condition

System usage example

The graph below shows the change over time about the total amount of congestion in Koriyama city and Motomiya city at each time of day from Oct 11 to 16 in 2019.

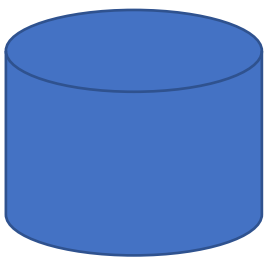


Real Time Data Collection



(a) Road sensor network of Fukushima prefecture

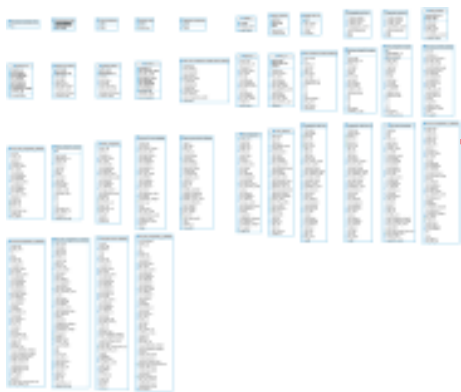
generates



(b) 5 min. interval traffic congestion data

104 K rows

storage



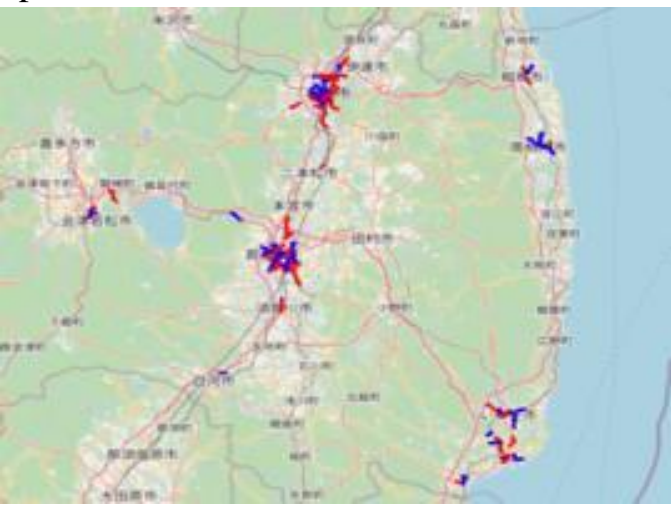
(c) JARTIC schema

Issue



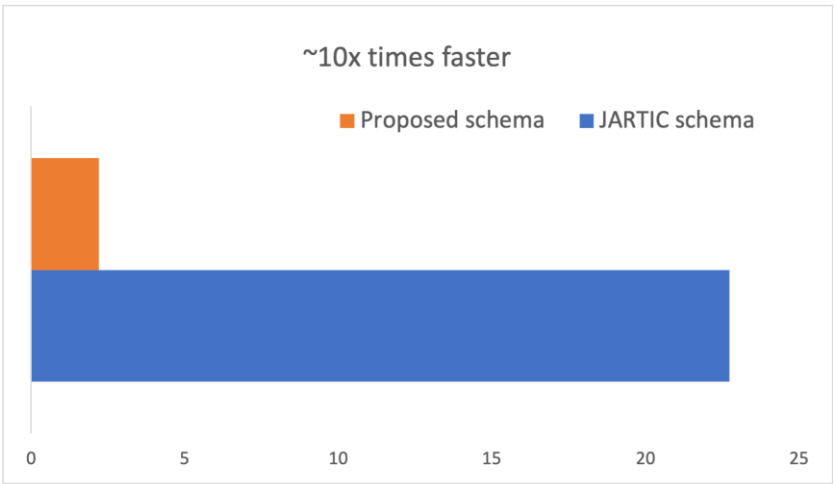
(d) Slow and cannot handle big data app requirements

Our Solution



benefits

(g) Faster discovery of competitive information in big data at low cost. Our AI system was able to predict traffic congestion in Fukushima with an accuracy of 83.5%

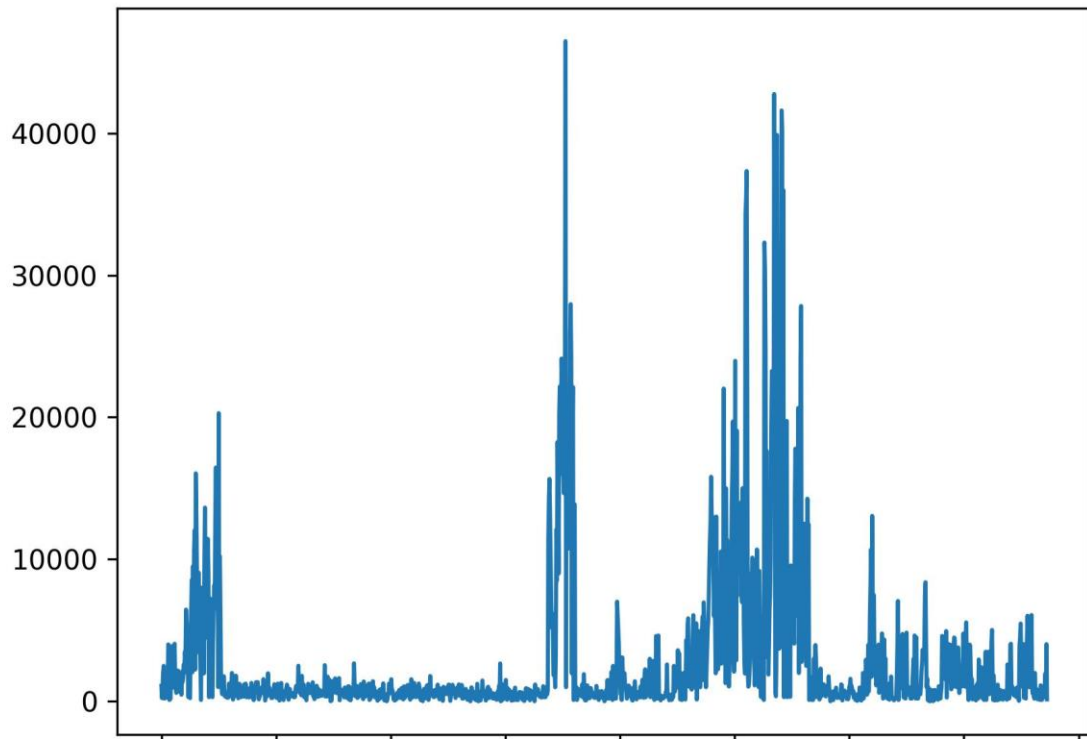


(f) Transactional databases for machine learning tasks can be generated 10 times faster

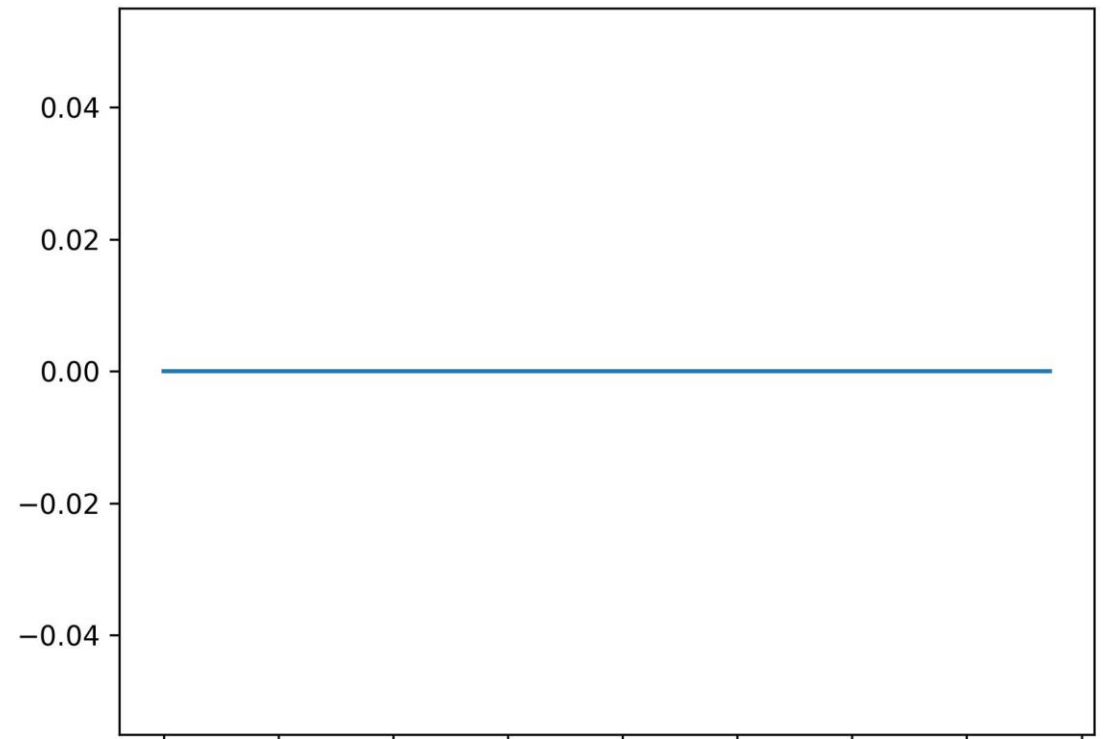
Performance evaluation



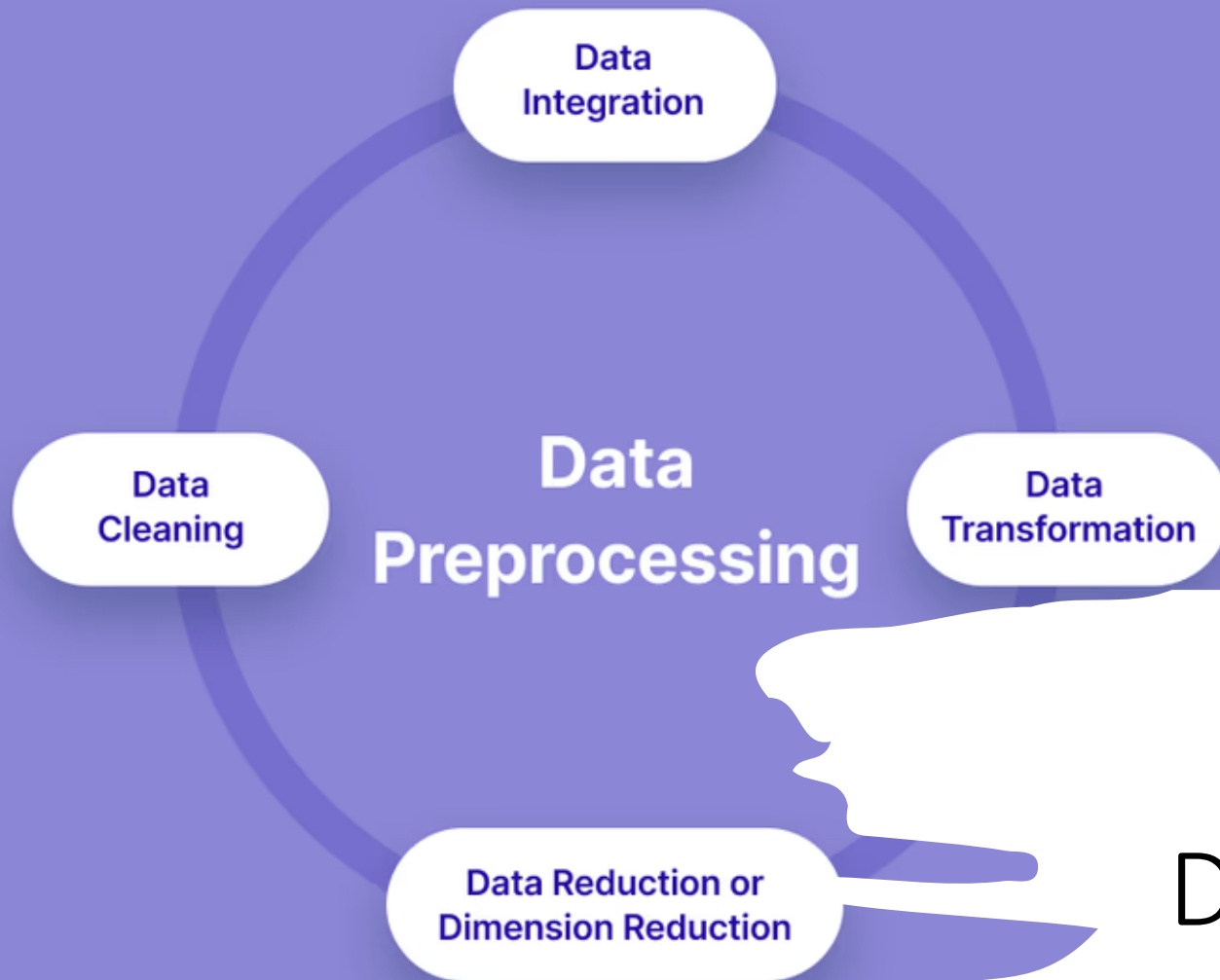
Outliers Identification in Dataset:



Maximum value in each column



Minimum value in each column



```
imputation
  BackwardFill.py
  ForwardFill.py
  HotDeck.py
  Interpolation.py
  KNNImputation.py
  MICEImputation.py
  MatrixFactorizationKNN.py
  MeanImputation.py
  MedianImputation.py
  ModelImputation.py
  Multiple.py
  README.md
  SoftImputation.py
  ZeroImputation.py
  __init__.py
normalization
  DecimalScalingNormalization.py
  LogTransformationNormalizati...
  MaxAbsScalerNormalization.py
  MinMaxNormalization.py
  PowerTransformerNormalizatio...
  QuantileTransformerNormaliza...
  README.md
  RobustScalingNormalization.py
  RootTransformationNormalizat...
  UnitVectorNormalization.py
  ZScoreNormalization.py
  __init__.py
```

```
1 # This code is written by Charan as part of internship at the university of aizu
2
3
4 class BackwardFill:
5
6     def getResult(self, sourceDF):
7         if sourceDF[sourceDF.columns[0]].dtype not in ['int64', 'float64']:
8             df = sourceDF.set_index(sourceDF.columns[0])
9             elif all(sourceDF.iloc[:, 0].diff().dropna() == sourceDF.iloc[:, 0].diff().dropna().iloc[0]):
10                 df = sourceDF.set_index(sourceDF.columns[0])
11             else:
12                 df = sourceDF.copy()
13
14         getImputedDataFrame = df.fillna(method='bfill')
15         getImputedDataFrame = df.fillna(method='ffill')
16         return getImputedDataFrame
17
18     def smartResult(self, sourceDF, Timestamp=True):
19         if Timestamp:
20             df = sourceDF.set_index(sourceDF.columns[0])
21         else:
22             df = sourceDF.copy()
23
24         getImputedDataFrame = df.fillna(method='bfill')
25         getImputedDataFrame = df.fillna(method='ffill')
26         return getImputedDataFrame
27
28
29 if __name__ == '__main__':
30     obj = BackwardFill()
31     obj.getResult(dataFrame)
32     obj.smartResult(dataFrame)
```

Data Preprocessing:

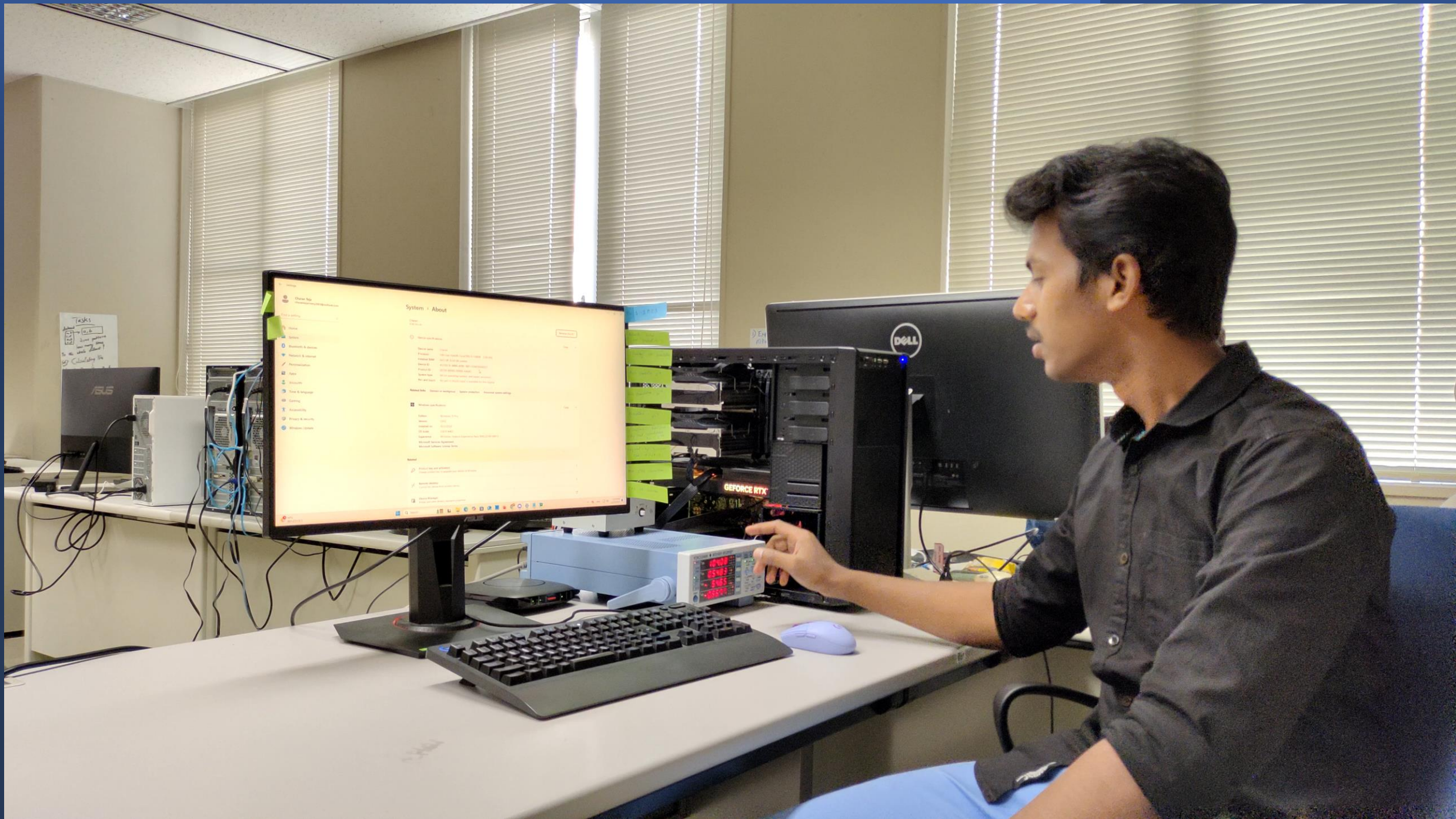
Imputation and Normalization

Experimental Setup

Power Calculation device: Yokogawa WT310EH Digital Power Meter

Edition	Windows 11 Pro
Processor	13th Gen Intel(R) Core(TM) i5-13400F 2.50
GHz	
Installed RAM	64.0 GB
System type	64-bit operating system
Shared GPU	32 GB
SSD	2TB
Required Internet	Minimum 100 MBPS





Problem Statement:

- Traditional models predict congestion independently for each road segment, which is inefficient for large networks.
- We need an energy-efficient, scalable system that accurately forecasts traffic across multiple road segments.

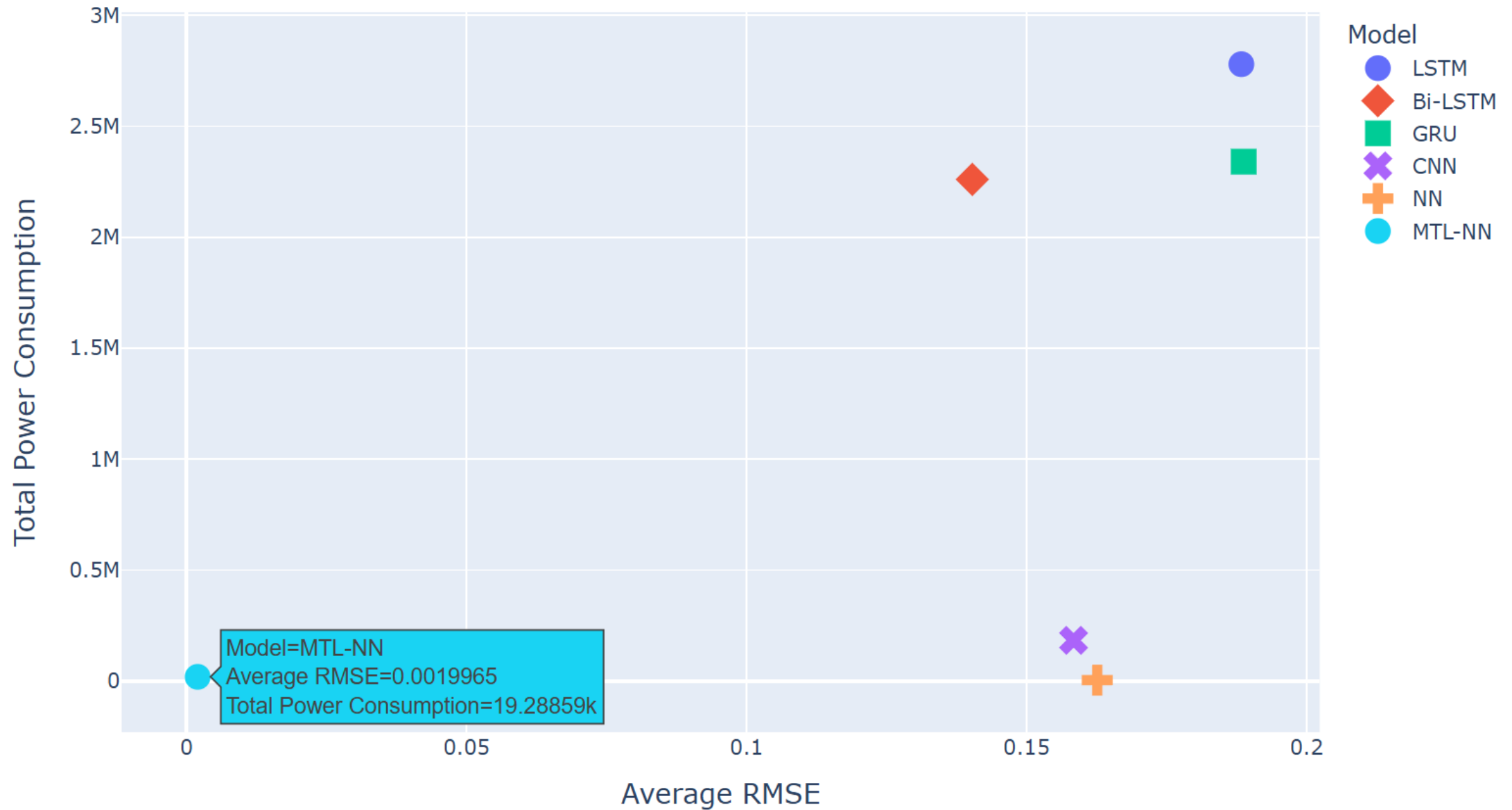
Objectives:

- Predict traffic congestion for large-scale road networks.
- Improve prediction accuracy using shared information across tasks.
- Develop a model with reduced computation time, memory and power consumption.
- **Novelty:** Use of shared and task-specific layers for different road segments.

Results:

	Model	Average_RMSE	Average_MAE	Aveerage_R2	Total_Time	Total_Memory	Total_Power_Consumption
0	LSTM	0.188374	0.087861	0.244536	5948.2	240.9800	2780044.50
1	Bi-LSTM	0.140328	0.064542	0.626795	4,543.90	350.5700	2260605.70
2	GRU	0.188786	0.105214	0.435177	4,578	260.7200	2340839.20
3	CNN	0.158414	0.083124	0.697999	395.66	238.3800	183652.39
4	NN	0.162634	0.082682	0.481399	4.6	86.7500	4461.43
5	MTL-NN	0.001997	0.000801	0.999880	27.4368	0.0009	19288.59

Power Consumption vs RMSE by Model



Our Route Recommender System

私たちのルート推奨システム



Road congestion data
道路渋滞情報



Road Network
道路網



Google Geocoding and Routes
Googleのジオコーディングとルート



Yahoo! Weather data
Yahoo!の気象情報

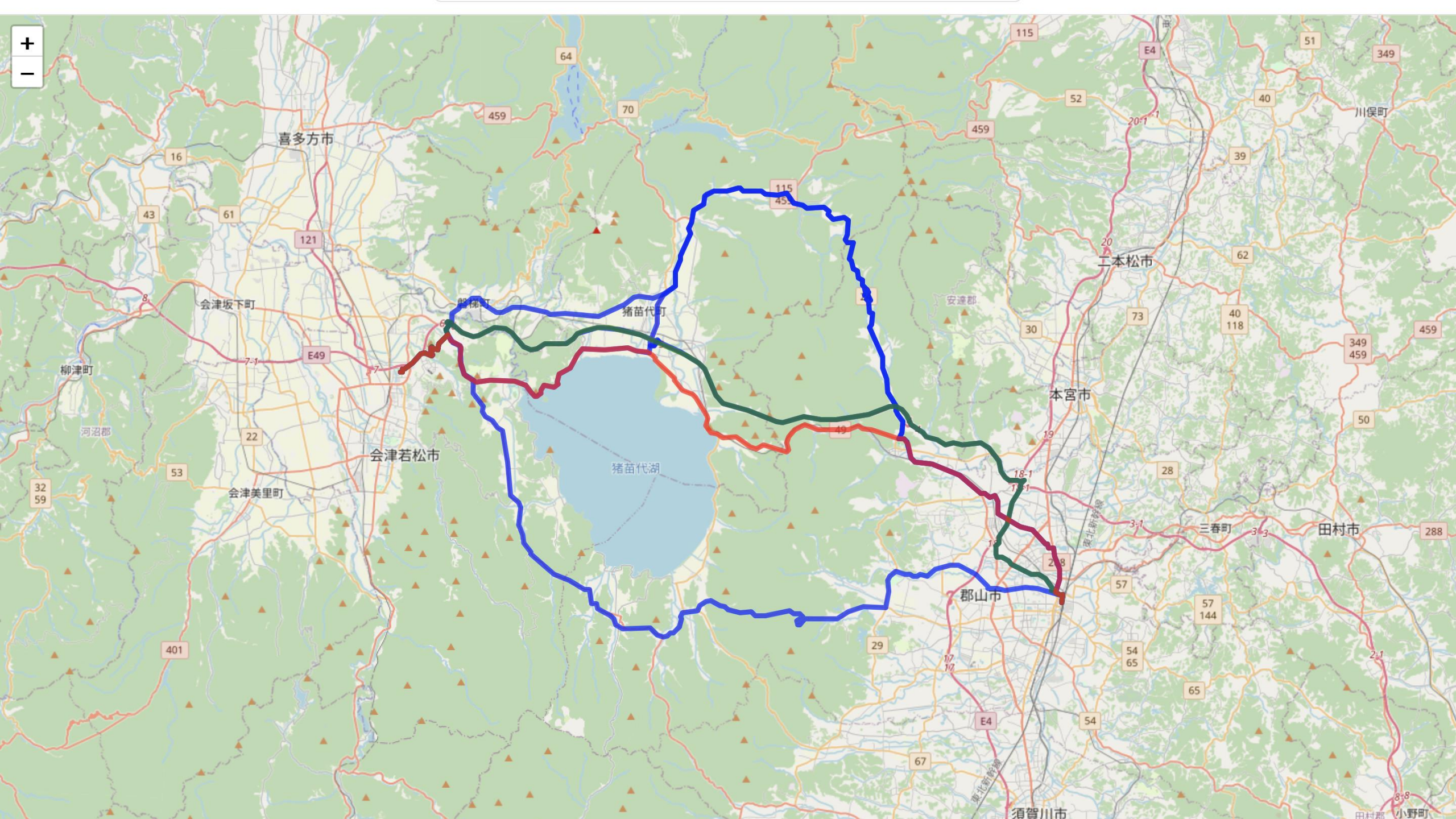


Wind & Pollution
風と公害









$\int (Congestion, RoadLayouts, Routes, Weather, WindSpeed, Pollution)$

Recommends Route with minimal human driving

人の運転を最小限に抑えたルートを推奨



Schedule

			1 st year												2 nd year														
			1 st sem						2 nd sem						3 rd sem						4 th sem								
			10	11	12	1	2	3	4	5	6	7	8	9	10	11	12	1	2	3	4	5	6	7	8	9			
Phase1	learning prerequisite courses																												
	reviewing literatures																												
	special reaserch																												
	Paper writing																												
																													
Phase2	proposing novel methods																												
	Ananlysis of results and comparing																												
	Discussion with guides																												
	Thesis write-up																												
																													



Any Questions

The image features the words 'THANK YOU!' in a bold, 3D sans-serif font. Each letter is a different color: 'T' is blue, 'H' is purple, 'A' is pink, 'N' is orange, 'K' is yellow, 'Y' is green, and 'O' is teal. The letters have a slight shadow, giving them a three-dimensional appearance. Surrounding the text are numerous small, colorful dots in shades of blue, yellow, orange, and purple, resembling confetti or stars. The entire graphic is set against a white background.

THANK YOU!