

Data management plan (DMP)

Predicting Urban Traffic Congestion Levels using Multi-Layer Perceptrons on SDCC SCOOT Data

SCOOT-MLP

Version	Effective date	Description of document/changes
2	28/05/2025	First version of the DMP – created for the start of the project

Level of distribution		This DMP is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). It is publicly available under DOI to be assigned upon publication in the TU Wien Research Data Repository.
-----------------------	---	---

FWF Data Management Plan (DMP)

I General Information						
I.1 Administrative information	PI: FWF project number: Internal Project ID: DMP version: 2, 28.05.2026 Contributors: Habib Ahmad, e12532822@student.tuwien.ac.at, ORCID: 0009-0002-0332-2702, TU Wien, Project Member Johannes Held, e11705340@student.tuwien.ac.at, ORCID: 0009-0003-3806-507X, TU Wien, Project Member Kyzer Gerez, e12427543@student.tuwien.ac.at, ORCID: 0000-0003-4463-3929, Oregon State University, Project Member Gevorg Zakaryan, e11941370@student.tuwien.ac.at, ORCID: 0009-0002-8980-8948, TU Wien, Project Member					
I.2 Data management responsibilities and resources	Person responsible for data management and DMP: Kyzer Gerez, e12427543@student.tuwien.ac.at, ORCID: 0000-0003-4463-3929, Oregon State University Co-ordination of data management responsibilities across partners: Kyzer Gerez, e12427543@student.tuwien.ac.at, ORCID: 0000-0003-4463-3929, Oregon State University Resources costed in for RDM: There are no costs dedicated to data management and ensuring that data will be FAIR.					
II Data Characteristics						
II.1 Data description and collection or re-use of existing data	Produced datasets:					
	dataset ID	title	type	format	estimated volume	contains sensitive data
	P1	Processed SDCC Traffic Database	Databases	DBRepo relational database	1 - 5 GB	no
						Normalized 3NF relational representation of the SDCC traffic dataset stored in DBRepo with semantic metadata annotations,

			accessed via REST API, SQL			ontology mappings, and SQL views for ML workflows.
P2	FAIR Metadata Records	Structured text, Plain text	JSON-LD, JSON, Markdown	100 - 1000 MB	no	RO-Crate, Croissant, FAIR4ML, CodeMeta, CITATION.cff, and Model Card metadata describing datasets, software, workflows, and models.
P3	Machine Learning Models	Other	PKL	100 - 1000 MB	no	Trained machine learning model artefacts generated during congestion prediction experiments, including associated FAIR4ML metadata and Model Cards.
P4	Prediction and Evaluation Outputs	Structured text, Images	CSV, JSON, PNG	100 - 1000 MB	no	Generated predictions, evaluation metrics, confusion matrices, and visualizations produced during model evaluation.

Reused datasets:

dataset ID	title	source	rights (e.g. license)	contains sensitive data	description
R1	Traffic Flow Data Jan to June 2022 SDCC	https://data-sdublincoco.opendata.arcgis.com/datasets/sdublincoco::traffic-flow-data-jan-to-june-2022-sdcc		no	...

Methods and software used for data generation and reuse

This project reuses an existing dataset obtained from the SDCC Open Data Portal. The dataset contains traffic sensor measurements recorded at 15- minute intervals, including traffic flow, degree of saturation, and congestion indicators, as well as temporal and spatial attributes. The dataset comprises approximately 1,048,575 records and has a size of approximately 79 MB. The original dataset is ingested into a relational database infrastructure implemented in DBRepo. A normalized relational schema in Third Normal Form (3NF) was designed and implemented using SQL CREATE statements executed from Jupyter notebooks through the DBRepo REST API. The database schema, entity–relationship diagrams, and descriptive metadata are maintained within the repository to support citation, discoverability, and reuse. Existing data are therefore reused rather than newly collected, while additional derived datasets and machine learning outputs are generated during experimentation. To improve interoperability and FAIR compliance, dataset attributes

	are semantically annotated using domain-relevant ontologies, and numeric attributes are linked to formal units-of-measurement concepts using the SI Digital Framework and related ontologies where appropriate. Metadata mappings are stored directly in DBRepo via the REST API. Additional denormalized SQL VIEW definitions are created to provide query-ready feature tables and aggregated datasets suitable for downstream machine learning workflows. The experimental pipeline was reimplemented to retrieve all input data exclusively through the DBRepo REST API rather than local CSV files. This includes the use of REST endpoints for both normalized tables and derived SQL views, with robust error handling and reproducibility checks to ensure equivalence with the original local-file implementation. Machine learning models are trained on the processed data and generate derived artefacts including predictions, evaluation metrics, confusion matrices, and visualization outputs. Metadata for datasets, models, and software components are documented using multiple interoperable standards, including RO-Crate, CodeMeta, FAIR4ML, Croissant JSON-LD, and Model Cards. Trained models and generated datasets are deposited in the TU Wien Research Data Repository, while software releases are archived through GitHub-Zenodo integration with DOI assignment. The project uses the following software and technologies: Python and Jupyter Notebooks for data processing and experimentation, SQL for schema design and view definitions, DBRepo REST API for database management and data access, GitHub for version control and collaboration, Zenodo and the TU Wien Research Data Repository for archival and publication, JSON-LD metadata standards including RO-Crate, CodeMeta, FAIR4ML, and Croissant. All software dependencies, runtime requirements, licences, and metadata specifications are documented in the project repository to ensure transparency and reproducibility of the research workflow.
III Documentation and Data Quality	
III.1 Metadata and documentation	Data organisation, metadata, and documentation: The project data are structured according to a normalized relational database schema designed in Third Normal Form (3NF). The original CSV dataset from the SDCC Open Data Portal is treated as immutable raw input data and retained in its original form within the repository for reproducibility purposes. The dataset is ingested into DBRepo through automated Python/Jupyter notebook workflows using the DBRepo REST API. The schema separates entities such as traffic measurements, temporal information, and spatial attributes into relational tables with explicit primary and foreign key relationships. SQL VIEW definitions are additionally provided to create denormalized, query-ready representations for downstream machine learning tasks, including aggregated feature tables and balanced training datasets. Metadata and semantic annotations are versioned together with the data infrastructure. Attribute mappings to ontologies, unit-of-measurement mappings, and metadata records are stored programmatically through the DBRepo API to ensure consistency and traceability. Structured metadata standards including RO-Crate, CodeMeta, FAIR4ML, Croissant JSON-LD, and Model Cards are maintained alongside the datasets and software artefacts. Versioning is managed through Git and GitHub using tagged releases. All schema definitions, notebooks, metadata files, SQL scripts, and machine learning workflows are tracked in the Git repository. Major project milestones are archived as GitHub releases and linked to Zenodo to mint persistent DOIs for reproducibility and citation. Separate versioned deposits are maintained for: source/input datasets, trained machine learning models, generated prediction outputs and evaluation artefacts. The project also distinguishes between raw, processed, and generated data: Raw data: original SDCC CSV dataset, Processed data: normalized database tables and SQL views in DBRepo, Generated data: predictions, evaluation metrics, figures, confusion matrices, and trained ML model artefacts. To ensure reproducibility, all dependencies and runtime requirements are version-pinned and documented in requirements.txt, environment files, and codemeta.json. The repository additionally includes release metadata (CITATION.cff), provenance metadata (RO-Crate), and FAIR-compliant metadata descriptions for all major research outputs. The project provides descriptive, structural, and semantic metadata to support the identification, discovery, interoperability, and reuse of both the input data and all generated research outputs. Descriptive metadata are maintained in DBRepo, Zenodo, and the TU Wien Research Data Repository, and include titles, descriptions, keywords, creators/authors with ORCIDs, licences, related identifiers (DOIs, GitHub URLs, DBRepo URLs), resource types, and provenance information. Metadata records explicitly reference the original SDCC Open Data Portal dataset and preserve information about the original

	publisher and licensing conditions. To support interoperability and FAIR compliance, dataset attributes are semantically annotated using domain-relevant ontologies. Attribute meanings are mapped to established ontology concepts, while units of measurement for numeric attributes are linked to formal unit ontologies including the SI Digital Framework and, where necessary, Quantities and Units (QUDT/Commons). These mappings are stored as metadata within DBRepo through the REST API. Several community-recognized metadata standards are used throughout the project: RO-Crate metadata describing datasets, software, models, licences, creators, and relationships between artefacts CodeMeta metadata for software description, dependencies, runtime requirements, repository location, and licensing FAIR4ML metadata for trained machine learning models, including hyperparameters, evaluation metrics, intended use, limitations, and training dataset references Croissant JSON-LD metadata describing dataset structure, field types, units, distributions, and licences Model Cards documenting model purpose, intended and out-of-scope uses, evaluation results, limitations, ethical considerations, and licensing. The project also includes technical and provenance metadata such as: SQL schema definitions and entity–relationship diagrams SQL VIEW definitions used in the ML pipeline API endpoint documentation for DBRepo access dependency and environment specifications release metadata through GitHub and Zenodo citation metadata via CITATION.cff. This will help others to identify, discover and reuse our data. Additionally, we will provide common metadata such as title, description, or keywords when publishing data in open access repositories. In such a case, we will follow the default template provided by the repository, such as Data Cite Metadata or Dublin Core. As far as possible, we will use controlled vocabularies for our data to allow inter-disciplinary interoperability and machine-actionability. Documentation required to validate the data analysis and facilitate reuse is provided through a combination of repository documentation, structured metadata standards, reproducible workflows, and archived research outputs. The GitHub repository contains a comprehensive README.md describing the project objectives, dataset sources, software requirements, installation instructions, workflow execution steps, input and output descriptions, licensing information, contributor information (including ORCIDs), and instructions for reproducing the experiments. Step-by-step reproduction instructions are provided to enable independent validation of the full analysis pipeline. All data-processing and machine learning workflows are implemented in version-controlled Python scripts and Jupyter notebooks. The notebooks document the complete experimental process, including data ingestion, schema creation, semantic annotation, data loading into DBRepo, feature preparation, model training, evaluation, and generation of derived outputs. The final implementation retrieves all data through the DBRepo REST API rather than local files, ensuring that the workflow remains reproducible and traceable. The database infrastructure is documented through: SQL schema definitions in Third Normal Form (3NF)
--	---

	entity–relationship diagrams SQL VIEW definitions used for downstream ML tasks metadata stored in DBRepo API endpoint documentation and authentication details. To support reuse and interoperability, the project includes several machine-readable metadata standards: RO-Crate for describing relationships between datasets, software, models, and outputs CodeMeta for software metadata and dependencies FAIR4ML metadata for trained machine learning models Croissant JSON-LD metadata for dataset structure and distributions Model Cards documenting intended use, evaluation results, limitations, and ethical considerations. Validation and reproducibility are further supported by: version-controlled releases through GitHub DOI-minted releases via Zenodo archived deposits of trained models and generated datasets in the TU Wien Research Data Repository pinned software dependencies and runtime requirements explicit licensing information for input data, software, and generated outputs provenance metadata linking datasets, code, models, and derived artefacts.
III.2 Data quality control	Data quality control: The following data quality checks will be done: data entry validation, peer review of data and representation with controlled vocabularies.
IV Data Storage, Sharing, and Long-Term Preservation	
IV.1 Data storage and backup during the research process	Storage and backup facilities: For the duration of the project, storage and backup of data will be ensured by Kyzer Gerez (acting as the person responsible for data management and DMP) in cooperation with the system operator. The data will be stored on the servers of TU Wien.

	P2 (FAIR Metadata Records), P3 (Machine Learning Models), P4 (Prediction and Evaluation Outputs), R1 (Traffic Flow Data Jan to June 2022 SDCC), P1 (Processed SDCC Traffic Database) will be stored on TUfiles: TUfiles is a central and readily available network drive with daily backups and regular snapshots provided by Campus IT. It is suitable for storing data with moderate access requirements, but high availability demands that allows full control of allocating authorisations. Only authorized staff members will have access. Data security and protection of sensitive data: We pay strict attention to compliance with the relevant institutional and national data protection policies. At this stage, it is not foreseen to process any sensitive data in the project. If this changes, advice will be sought from the data protection specialist at TU Wien, and the DMP will be updated. Access to data during research: <table><tr><th>dataset ID</th><th>selected project members</th><th>all other project members</th><th>the public</th></tr><tr><td>P1</td><td>writing</td><td>writing</td><td>reading only</td></tr><tr><td>P2</td><td>writing</td><td>writing</td><td>reading only</td></tr><tr><td>P3</td><td>writing</td><td>writing</td><td>reading only</td></tr><tr><td>P4</td><td>writing</td><td>writing</td><td>reading only</td></tr><tr><td>R1</td><td>writing</td><td>writing</td><td>reading only</td></tr></table> If incidents occur, the project contributors will attempt to address them where feasible.	dataset ID	selected project members	all other project members	the public	P1	writing	writing	reading only	P2	writing	writing	reading only	P3	writing	writing	reading only	P4	writing	writing	reading only	R1	writing	writing	reading only
dataset ID	selected project members	all other project members	the public																						
P1	writing	writing	reading only																						
P2	writing	writing	reading only																						
P3	writing	writing	reading only																						
P4	writing	writing	reading only																						
R1	writing	writing	reading only																						
IV.2 Data sharing and long-term preservation	Data publication and access conditions: As far as possible, obtained datasets will be published in repositories. Details on access conditions, reuse licenses, reasons for restrictions, etc. are collected in the table below. <table><tr><th>dataset ID</th><th>access conditions</th><th>estimated publication date</th><th>location for publication (repository)</th><th>PID</th><th>license</th></tr><tr><td>P1</td><td>Open</td><td>2026-03-30</td><td>TU Wien Research Data</td><td>DOI</td><td>CC-BY-4.0</td></tr><tr><td>P2</td><td>Open</td><td>2026-05-30</td><td>TU Wien Research Data</td><td>DOI</td><td>CC-BY-4.0</td></tr></table>	dataset ID	access conditions	estimated publication date	location for publication (repository)	PID	license	P1	Open	2026-03-30	TU Wien Research Data	DOI	CC-BY-4.0	P2	Open	2026-05-30	TU Wien Research Data	DOI	CC-BY-4.0						
dataset ID	access conditions	estimated publication date	location for publication (repository)	PID	license																				
P1	Open	2026-03-30	TU Wien Research Data	DOI	CC-BY-4.0																				
P2	Open	2026-05-30	TU Wien Research Data	DOI	CC-BY-4.0																				

P3	Open	2026-05-30	TU Wien Research Data	DOI	MIT
P4	Open	2026-05-30	TU Wien Research Data	DOI	CC-BY-4.0

Repository description:

TU Wien Research Data is an institutional repository of TU Wien to enable storing, sharing and publishing of digital objects, in particular research data. It facilitates the funders' requirements for open access to research data and the FAIR principles by making research output findable, accessible, interoperable, and reusable. A DOI is assigned to each dataset published in TU Wien Research Data. This service is developed by the TU Wien Center for Research Data Management and hosted by TU.it. <https://researchdata.tuwien.ac.at/>

Methods or software needed to access and use data: Yes. Reuse of the project outputs requires commonly available tools and software including Python, Jupyter Notebooks, SQL-compatible database tools, and access to the DBRepo REST API. Users may also require Git/GitHub for accessing version-controlled resources and standard JSON-LD compatible tools for working with metadata formats such as RO-Crate, FAIR4ML, and Croissant.

Long-term preservation and deletion of data:

dataset ID	location for long-term storage	minimum retention period (≥ 10 years)	foreseeable research uses and/or users
P1	TU Wien Research Data	10 years	The primary target audience includes researchers and students in data science, machine learning, transportation engineering, and urban analytics who are interested in traffic prediction, congestion analysis, and FAIR research data management practices. The project may also be relevant to public-sector organizations, smart-city initiatives, and transportation planners seeking reproducible approaches for analysing urban traffic patterns and evaluating congestion mitigation strategies.
P2	TU Wien Research Data	10 years	
P3	TU Wien Research Data	10 years	
P4	TU Wien Research Data	10 years	

V.1 Legal aspects	Personal data: At this stage, it is not foreseen to process any personal data in the project. If this changes, advice will be sought from the data protection specialist at TU Wien, and the DMP will be updated. Intellectual property rights and rights of use: The following individual(s) hold rights and control access to the project data: The original SDCC Open Data Portal dataset remains under the control and licensing terms of the original data publisher/rights holder (CC BY 4.0) the project team reuses and republishes the data in accordance with the source license. The project team members (repository contributors/authors) are responsible for managing and maintaining the generated artefacts, including processed datasets, SQL views, trained machine learning models, predictions, evaluation outputs, metadata records, and documentation. The GitHub repository is publicly accessible, allowing open access to the source code, notebooks, metadata files, workflows, and documentation under the MIT License. Deposited datasets and model artefacts in Zenodo and the TU Wien Research Data Repository are publicly accessible under their assigned licences, while the corresponding project contributors manage the deposit records and metadata.
V.2 Ethical aspects	Ethical issues: No particular ethical issue is foreseen with the data to be used or produced by the project. This section will be updated if issues arise.