Tallinn University of Technology

Khamidjon Khamidov 246075

Software Engineering

Data Imputation and Animated Visualization of Sound Pressure Level in Tallinn Based on IoT Sensor Data

Master's thesis

Supervisor: Jaanus Kaugerand, Senior Researcher

Tallinn, 2026

Tallinna Tehnikaülikool

Khamidjon Khamidov 246075

Tarkvaratehnika

Tallinna Helirõhutaseme Andmete Imputatsioon ja Animeeritud Visualiseerimine Asjade Interneti Andurite Andmete Põhjal

Magistritöö

Juhendaja: Jaanus Kaugerand, vanemteadur

Tallinn, 2026

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of other have been referred to. This thesis has not been presented for examination anywhere else.

Author: Khamidjon Khamidov

07.05.2026

Abstract

Urban noise is a critical environmental health concern in Tallinn, where approximately 40% of residents are exposed to traffic noise exceeding recommended safety thresholds. The city's IoT-based acoustic monitoring network of 471 low-cost sensors offers high-resolution spatiotemporal coverage but exhibits a fleet-wide missing data rate of approximately 26%, limiting the reliability of any analysis built on the raw dataset.

This thesis develops and evaluates a reproducible imputation pipeline for reconstructing missing hourly Sound Pressure Level (SPL) measurements. Four methods are implemented: a historical median, a K-Nearest Neighbours, an inverse-variance weighted combination of the two, and Google TimesFM 2.5. The methods are evaluated using a stratified masking design across 35 carefully selected test devices structured in three groups — spatially connected, spatially isolated, and short-history.

To support exploration and analysis of the reconstructed data, an interactive web-based dashboard is developed using React, MapLibre GL, and Recharts. The dashboard enables sensor health monitoring through colour-coded missing-data indicators, animated SPL playback, spatial noise pattern analysis, and side-by-side comparison of imputation methods with quantitative accuracy metrics.

The results demonstrate that Google TimesFM foundation model approach can achieve sub-perceptual error levels across a structurally diverse sensor fleet, and that the accompanying visualisation dashboard enables diagnostic analysis that would not be accessible from static tables or aggregate statistics alone.

The thesis is in English and contains 93 pages of text, 8 chapters, 22 figures, 8 tables.

Annotatsioon

Linnamüra on Tallinnas oluline keskkonnaalane terviseprobleem, kus ligikaudu 40% elanikest puutub kokku liiklusmüraga, mis ületab soovituslikke ohutuspiire. Linna IoT-põhine akustilise seire võrgustik, mis koosneb 471 madala hinnaga sensorist, võimaldab kõrge eraldusvõimega ruumilis-ajalist katvust, kuid kogu sensorivõrgustikus esineb ligikaudu 26% puuduvate andmete määr, mis piirab toorandmestikul põhineva analüüsi usaldusväärsust.

Käesolevas magistritöös töötatakse välja ja hinnatakse reprodutseeritav imputatsioonipipeline puuduvate tunnipõhiste helirõhutaseme (SPL) mõõtmiste taastamiseks. Rakendatakse ja võrreldakse nelja meetodit: ajalooline mediaan, K-lähimate naabrite meetod (K-Nearest Neighbours), nende kahe pöördvariatsiooniga kaalutud kombinatsioon ning Google TimesFM 2.5 mudel. Meetodeid hinnatakse kihistatud maskeerimisskeemi abil 35 hoolikalt valitud testseadmel, mis on jaotatud kolme gruppi — ruumiliselt seotud, ruumiliselt isoleeritud ja lühikese ajalooga seadmed.

Rekonstrueeritud andmete uurimiseks ja analüüsimiseks arendatakse interaktiivne veebipõhine juhtpaneel, kasutades Reacti, MapLibre GL-i ja Rechartsit. Juhtpaneel võimaldab sensorite töökindluse jälgimist värvikoodidega tähistatud puuduvate andmete indikaatorite kaudu, animeeritud SPL-andmete taasesitust, ruumiliste müramustrite analüüsi ning imputatsioonimeetodite kõrvutivõrdlust koos kvantitatiivsete täpsusmõõdikutega.

Tulemused näitavad, et Google TimesFM-i vundamendimudelil põhinev lähenemine suudab saavutada alla tajupiiri jääva vea taseme struktuurselt mitmekesises sensorivõrgustikus ning et kaasnev visualiseerimise juhtpaneel võimaldab diagnostilist analüüsi, mida ei oleks võimalik teha ainult staatiliste tabelite või agregeeritud statistika põhjal.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 93 leheküljel, 8 peatükki, 22 joonist, 8 tabelit.

List of Abbreviations and Terms

KNN	International Electrotechnical Commission
LCS	K-Nearest Neighbours — a spatial interpolation method using geographically proximate sensors
LoRaWAN	Low-Cost Sensor
MAE	Long Range Wide Area Network — a low-power wireless communication protocol used in IoT deployments
MAR	Mean Absolute Error — average of absolute differences between predicted and true values
MCAR	Missing at Random — missingness that depends on observed data but not on the missing values themselves
MEMS	Missing Completely at Random — missingness independent of both observed and missing values
RMSE	Micro-Electro-Mechanical Systems — miniaturised sensor technology used in low-cost acoustic devices
SPL	Root Mean Square Error — square root of the average squared differences between predicted and true values

Table of Contents

Table of Figures

Table of Tables

1 Background

Tallinn, as a rapidly developing European capital, faces a complex acoustic environment. Urban noise is no longer viewed merely as an annoyance but as a critical environmental pollutant that impacts urban livability and real estate development. According to Statistics Estonia, approximately 40% of the city's population (roughly 160,000 residents) is currently exposed to traffic noise levels that exceed recommended safety thresholds (Tallinn City Government, 2024).

High SPL, reaching peaks of 75 dB along major transit arteries such as Liivalaia and Pärnu mnt, significantly exceed the 55 dB night-time limit set for residential tranquility (ERR, 2025). This discrepancy has led to tangible consequences for city growth, including the denial of building permits in central districts where noise levels fail to meet legal safety standards (ERR, 2025).

1.1 The Evolution of Acoustic Monitoring and the IoT Shift

Historically, Tallinn relied on static noise mapping — mathematical simulations derived from infrequent manual measurements or outdated datasets (e.g., from 2019). However, there are signs that in future city might look forward to include dynamic real-time monitoring via IoT based sensor networks (Figure 1 – Smart City Sensors). Supported by initiatives like "Test in Tallinn," this shift aims to provide a high-resolution, hourly understanding of how noise pulses through the city (Green Tallinn, 2024).

Figure 1 - Tallinn Smart City Sensors.

While IoT networks offer unprecedented granularity, they introduce significant technical vulnerabilities. Unlike high-grade industrial stations, low-cost IoT sensors are prone to periodic missingness caused by network packet loss (e.g. LoRaWAN/WiFi drops), power failures, or environmental interference (Smart Cities World, 2025). These gaps prevent the city from achieving a truly continuous and reliable acoustic profile.

1.2. The Critical Role of Data Imputation

Proper and correct analysis of Tallinn's noise levels is impossible if the underlying data is fragmented. If a sensor fails during a peak traffic hour, a simple average of the remaining data would mathematically underestimate the true noise pollution. To mitigate this, missing data mechanisms, such as data imputation becomes a prerequisite for any meaningful urban analysis.

By implementing and optimizing a hybrid approach — combining KNN (K-Nearest Neighbors) for spatial gaps, Self-Imputation for historical patterns, and Google's TimesFM for complex temporal sequences — this research aims to restore the integrity of the dataset.

Beyond statistical recovery, the visual analysis of SPL data serves as the bridge between raw data and urban planning. Static tables of decibel values are difficult for stakeholders to interpret; however, animated visualization reveals the temporal flow of noise, making patterns like noise waves or rush-hour spikes visible and intuitive (Siigur, 2025).

Furthermore, the visualization tool on top of data imputation methods serves a dual purpose as a diagnostic interface:

- Identification of Faulty Sensors: By comparing animated trends, analysts can visually spot sensors that are "stuck" or reporting impossible values.
- Mapping Uncertainty: By explicitly marking imputed data and uncertainty levels, the tool communicates the reliability of the analysis to the user.
- Strategic Expansion: Areas where the visualization shows high uncertainty or "blind spots" serve as direct recommendations for where more sensors should be placed to optimize the city's monitoring infrastructure.

2 Problem Statement

The unit of study for this research is Tallinn's IoT-based acoustic monitoring network, with a particular focus on the raw SPL data streams generated by low-cost sensors deployed along major transit routes. The study analyses the data on an hourly basis to observe how urban noise levels change over time across the city.

The monitoring network provides extensive spatial and temporal coverage, but the collected data contains periodic missing values caused by hardware malfunctions, connectivity interruptions, and other operational failures. These gaps reduce the reliability of the dataset and affect the accuracy of further analysis. Incomplete measurements can lead to distorted averages and unreliable conclusions, making it difficult to accurately identify the city's most noise-polluted areas or evaluate the effectiveness of implemented noise-reduction measures. For this reason, there is a clear technical need for a reliable imputation framework capable of reconstructing missing SPL measurements while preserving realistic temporal patterns in the data.

In addition to the technical challenges, the problem also has practical significance for urban planning and policy-making in Tallinn. Accurate noise mapping is important for environmental assessment and development planning, yet current decision-making processes may still rely on outdated static simulations from 2019 (ERR, 2025). As a result, inaccurate or incomplete noise data can negatively affect planning decisions, including the approval or rejection of building permits. Improving the completeness and reliability of the monitoring data can therefore support more accurate environmental analysis and better-informed urban development decisions.

2.1. Research Goal and Questions

The primary goal is to validate hybrid spatiotemporal imputation methods and an interactive visual analysis tool that transforms fragmented IoT data into a continuous, animated, and interpretable acoustic profile for Tallinn. The main questions the thesis aims to answer:

- RQ1: To what extent does a hybrid approach (Self-Imputation, KNN, and TimesFM) reduce the Root Mean Square Error (RMSE) in reconstructed SPL datasets compared to using only KNN or self-imputation?

- RQ2: How does the hybrid approach (combining KNN and Historical Median) compare to Google TimesFM in terms of accuracy and computational efficiency for SPL data reconstruction?

- RQ3: How can an interactive visual dashboard support the analysis of urban SPL data, including the identification of faulty sensors, spatial noise patterns, and data quality across the sensor network?

2.2. Relevant Concepts and Theory

This research is grounded in the following key concepts and existing theoretical frameworks:

- SPL: The logarithmic measure of the effective pressure of a sound relative to a reference value, measured in decibels (dB).

- Data Imputation: The statistical process of replacing missing data with substituted values. This research builds on Spatiotemporal Correlation Theory, which posits that noise at a specific location is correlated both to its own historical patterns (Temporal) and to the readings of its geographic neighbors (Spatial) (Cressie & Wikle, 2011).

- Foundation Models for Time-Series: The study utilizes Google TimesFM, a state-of-the-art decoder-only transformer model. Unlike traditional models, it leverages "Zero-Shot" learning to predict complex patterns in time-series data without requiring extensive local training (Das et al., 2024).

- Visual Analytics: A field of data science that focuses on analytical reasoning supported by interactive visual interfaces. This research applies Temporal Heatmap Animation to translate raw SPL values into intuitive, moving maps for non-technical stakeholders.

2.3. Research Implementation and Methodology

To address the challenges of noise monitoring in Tallinn, this research implements a two-phase technical workflow:

1. Data Reconstruction Phase: A multi-model imputation pipeline is developed in Python to process raw IoT SPL data. This involves the systematic identification of missing patterns and the application of an optimized mixture of models to reconstruct a high-fidelity dataset.

2. Analytical Visualization Phase: An interactive React-based dashboard is developed to consume the processed data. This system generates animated noise maps, enabling the visual identification of temporal trends, the detection of faulty sensors, and the determination of areas where additional monitoring nodes are required.

The study utilizes a Hybrid Quantitative (Amina Ahmed, Lucas Pereira and Kimberly Jane, 2024) and Visual Analytic approach (G Kokk and S Jönsson, 2013). The technical framework consists of three distinct imputation methods:

- Self-Imputation (using historical): This method fills data gaps by analyzing an individual sensor's historical patterns, comparing specific timestamps (e.g., Tuesday at 08:00) against previous recorded periods to maintain temporal consistency.

- Weighted KNN: This technique leverages spatial correlation. By utilizing real-time readings from geographically adjacent sensors, weighted by the inverse of their distance (e. g. Weight = 1/Distance), the system estimates missing values based on the local acoustic environment.

- Google TimesFM (Foundation Model): A pre-trained Time Series Foundation Model handles long-duration gaps. This provides a learning capability that manages complex sequences where simple historical or spatial patterns may prove insufficient.

The efficency of these methods is validated through Synthetic Gap Analysis (Josse & Husson, 2016). By intentionally removing a subset of known data, the Root Mean Square Error (RMSE) and Mean Absolute Error (MAE) are calculated to determine the optimal mixture of models required to approach the "ground truth."

2.4. Justification of the Proposed Framework

The proposed methodology is considered adequate for the Tallinn urban environment for the following reasons:

- Resilience to IoT volatility: By integrating spatial (KNN), temporal (Self-Imputation), and deep-learning (TimesFM) methods, the framework ensures that the analysis remains robust even during significant network outages or system failures.

- Advanced pattern recognition: The inclusion of a Foundation Model is superior to imputing only via KNN or adhoc self-imputation mehtods, as it accounts for the non-linear changes of urban noise caused by irregular events, such as roadworks or seasonal traffic shifts.

- Actionability through visualization: While raw SPL data is dense and difficult to interpret, animated visual analysis provides a necessary layer of clarity for urban planning. It allows for the immediate recognition of changes and sensor drift, fulfilling the requirement for a proper and correct analysis to support the city's strategic noise reduction goals.

2.5. Validation

The primary method for validating the imputation algorithm is synthetic gap testing. This involves taking a portion of the Tallinn IoT dataset where all values are known and intentionally removing a percentage of that data at random (e.g., 10% or 20%). By creating these artificial gaps, a ground truth is established that the algorithm must attempt to reconstruct.

To measure the effectiveness of the proposed hybrid model, the results of the hybrid algorithm will be compared against:

- Historical average (temporal): This method fills data gaps by calculating the mean or median value of the specific sensor’s historical records for the corresponding time of day and day of the week. It assumes that urban noise follows a strict periodic cycle. While effective for routine patterns, it fails to account for real-time anomalies or spatial shifts in the city's acoustic environment.

- KNN: This method utilizes spatial correlation by estimating missing values based on the readings of geographically adjacent sensors. The baseline assumes that noise levels at a specific coordinate are highly correlated with its immediate neighbors. This method is effective for capturing localized "noise blooms" but is vulnerable if multiple neighboring sensors fail simultaneously.

The accuracy of the reconstruction will be measured using two standard mathematical formulas:

1. Mean Absolute Error (MAE) (Chai & Draxler, 2014): This measures the average distance between the imputed value and the real value.

2. Root Mean Square Error (RMSE) (Chai & Draxler, 2014): This penalizes larger errors more heavily, which is critical for noise monitoring where a 5 dB mistake is much more significant than a 1 dB mistake.

If the error is low, the algorithm is ready to be used on real unknown gaps in Tallinn network. Otherwise, if the error is high in specific areas, it indicates where the city needs more sensors because the current neighbors (KNN) are too far away to provide a good estimate.

3 Literature Review

Acoustic monitoring in contemporary urban environments is a multidisciplinary field that integrates classical acoustics with distributed computing systems. The reliability of any noise assessment framework is fundamentally dependent on adherence to established SPL standards, as well as the accuracy and stability of the underlying sensing infrastructure.

The primary metric used in urban noise assessment is SPL, defined as a logarithmic measure of effective sound pressure relative to a reference value of 20 μPa (International Organization for Standardization, 2015). In order to ensure compliance with international research and regulatory frameworks, monitoring systems must adhere to the IEC 61672-1:2013 standard (International Electrotechnical Commission, 2013), which defines the performance requirements and classification of sound level meters.

A key aspect of this standard is the application of frequency weighting. Since human auditory perception is not uniformly sensitive across all frequencies, the A-weighting filter (dB(A)) is applied to emphasize frequencies in the approximate range of 500 Hz to 10 kHz. Within the European Union, these measurements are further consolidated into the Lden (Day-Evening-Night Level) metric, as specified by the Environmental Noise Directive (2002/49/EC) (European Parliament and Council of the European Union, 2002). This directive provides the regulatory and conceptual foundation for this study, as it mandates that member states, including Estonia, produce continuous and accurate noise maps for the protection of public health against long-term environmental noise exposure.

The literature identifies a transition from Class 1 Sound Level Meters — high-precision instruments typically cost several thousand dollars (Picaut et al., 2020) — to LCS (Low-Cost Sensor) networks. These IoT-based sensor nodes commonly employ Micro-Electro-Mechanical Systems (MEMS) microphones, enabling high-density spatiotemporal monitoring of urban environments.

Although Class 1 instruments provide superior measurement accuracy, they cost noticeably higher than low-cost sensors which would limit their scalability for constrained budget. However, as highlighted by Murphy and King (2022), the adoption of IoT-based sensing networks introduces a trade-off in reliability. These low-cost sensors are susceptible to intermittent data loss due to network latency (e.g., LoRaWAN or NB-IoT

packet loss), power instability, and environmental degradation. This results in the so-called “dirty data” problem, which necessitates advanced computational techniques for data reconstruction.

3.1. Data Imputation Techniques in Time-Series Analysis

Data imputation refers to the process of estimating and reconstructing missing values within a dataset. In acoustic time-series applications, simplistic statistical imputation methods are often insufficient due to the inherently non-linear and dynamic nature of urban soundscapes.

Spatial and Temporal Methodologies:

- KNN (Picaut et al., 2020) is a widely applied non-parametric technique that exploits spatial correlation structures. In dense urban sensor networks, noise levels recorded at a given location are often strongly correlated with measurements from geographically proximate sensors. Existing literature indicates that a weighted KNN formulation, in which the influence of neighboring sensors is inversely proportional to distance (1/d), significantly enhances the reconstruction of localized noise phenomena, often referred to as “noise blooms.”

- A recent and significant advancement in this domain is the emergence of foundation models for time-series forecasting and imputation. TimesFM (Oreshkin et al., 2020) represents a state-of-the-art approach in this category. As a decoder-only transformer model pre-trained on large-scale real-world temporal datasets, it exhibits zero-shot learning capabilities. In contrast to traditional models that require task-specific training for each individual sensor or dataset, TimesFM is capable of modeling complex, non-linear temporal dynamics. This includes capturing structured patterns such as the temporal signature of rush-hour traffic congestion, even in scenarios involving extended missing data intervals where conventional statistical methods based on historical averages are insufficient.

Recent research suggests that no single imputation technique is universally optimal across all missing data scenarios. Consequently, a hybrid methodological framework is required to address heterogeneous patterns of data loss.

This study contributes to this emerging research direction by proposing an optimized hybrid pipeline that integrates self-imputation based on historical temporal continuity, KNN-based spatial interpolation, and TimesFM-based global temporal pattern modeling. The integration of these complementary approaches enables robust reconstruction under varying conditions, including isolated sensor failures as well as large-scale network outages.

3.2. Comparative Analysis of Similar Studies

Review of Siigur (2025)

The foundational work in the context of Estonian acoustic monitoring is "Sound Pressure Level Analysis and Visualization in Tallinn Urban Environment Based on Low-Cost IoT Sensor Data" (Siigur, 2025). This study made a significant contribution by demonstrating the feasibility of deploying distributed IoT sensor networks within Tallinn and establishing an initial framework for data acquisition and cloud-based storage. Siigur provided a comprehensive characterization of Tallinn's acoustic environment and produced initial spatial visualizations of SPL along major transportation corridors.

Despite the contributions of prior work, the study mainly focused on low processing required algorithms which mainly focused on in-device processing with lightweight algorithms. In addition, no existing studies in the Tallinn context have incorporated time-series foundation models such as TimesFM for enhancing imputation performance.

This thesis addresses the identified gap by mainly focusing on server-side processing algorithms with higher processing power devices which focuses on accuracy of imputation data. By developing a hybrid imputation pipeline implemented in Python alongside an interactive visualization interface built in React, this research directly addresses the issues of data inconsistency and interpretability identified in the problem statement.

The main objective is to provide urban planners in Tallinn with a continuous, statistically validated representation of the acoustic environment, thereby enabling evidence-based decision-making in domains such as urban planning, building permit regulation, and noise mitigation strategy development.

3.3. Research Novelty and Contributions

The originality of this research is characterized by a methodological shift from passive observation toward active data reconstruction and system-level diagnostics. The study introduces several novel contributions to the analysis of the Tallinn IoT-based acoustic environment:

1. The Hybrid-Optimized Pipeline

Integration of Zero-Shot Foundation Models. While prior studies have primarily relied on conventional statistical approaches, this research aims to incorporate TimesFM into the domain of urban acoustic analysis. The key innovation lies in leveraging the model's zero-shot capabilities, enabling the reconstruction of complex temporal noise patterns without the need for extensive task-specific training, as typically required by traditional models such as RNNs or LSTMs (Schuster & Paliwal, 1997; Hochreiter & Schmidhuber, 1997).

Spatiotemporal Weighting Mechanism. The proposed methodology introduces a hybrid framework that dynamically combines spatial (KNN-based), temporal (self-imputation), and deep learning (TimesFM-based) components. The weighting of these components is adapted based on the characteristics of the missing data segment. For example, spatial interpolation is prioritized for short-range gaps, while foundation model predictions are utilized for longer temporal discontinuities. This adaptive weighting mechanism represents a level of methodological optimization not present in existing Tallinn-focused acoustic studies.

2. Quantitative Validation of Ground Truth

The study systematically compares the performance of the proposed hybrid model against established baseline methods, including KNN-based imputation and historical averaging. This enables a clear quantification of performance improvements and highlights the added value of integrating modern machine learning approaches into time-series reconstruction tasks.

3. Diagnostic Visual Analytics

While existing visualization tools primarily serve descriptive purposes (e.g. displaying noise levels), this research introduces a diagnostic-oriented visualization framework. The developed React-based interface is designed to support system monitoring by enabling the identification of anomalies such as sensor drift, stagnant ("frozen") readings, and spatial data gaps. Additionally, the visualization incorporates representations of imputation uncertainty to enhance interpretability.

The study advances beyond static heatmap representations by introducing temporally dynamic visualizations. These allow for the identification of short-duration acoustic events, referred to as "noise pulses," which are typically obscured in aggregated hourly or daily data. This approach provides a more detailed understanding of the temporal propagation of noise across Tallinn's urban infrastructure.

4 Methodology & Data Acquisition

The approach for this study combines data engineering, statistical analysis, and machine learning techniques to enable both reconstruction and evaluation of missing time-series observations. The selected methods are appropriate to the research problem, as they account for both spatial and temporal dependencies inherent in environmental noise data, while also enabling quantitative validation of reconstruction accuracy.

The primary dataset is obtained from the urban acoustic monitoring network deployed in Tallinn. The network comprises 471 low-cost IoT sensors equipped with MEMS microphones, distributed across key urban locations including transportation corridors, residential areas, and public spaces. Each sensor provides:

1. SPL measurements in decibels (dB)
2. Timestamped observations in UTC format
3. Static geographic coordinates (latitude and longitude)

The dataset covers the period from 01.09.2021 to 31.12.2021, with minutely sampling. Prior to analysis, the raw dataset are pre-processed to ensure data consistency and analytical reliability with following steps:

1. Device-level records are extracted and consolidated to form a consistent catalogue of sensors. In cases of minor coordinate inconsistencies, the initial recorded position is retained to preserve spatial integrity.
2. Raw timestamps are standardised and converted from UTC to the local timezone (Europe/Tallinn), ensuring alignment with human activity cycles. All observations are aggregated to hourly intervals, providing a consistent temporal resolution for subsequent analysis.
3. Within each hourly interval, multiple readings are aggregated using the median, which provides robustness against transient noise spikes. Invalid or non-numeric values are excluded during processing.
4. Data completeness is evaluated at the device level, revealing significant variability in sensor reliability. This step is essential for understanding the extent and structure of missing data prior to imputation.

4.1. Imputation Framework and Visualization

The study explicitly distinguishes between different types of missingness:

- **Missing Completely at Random (MCAR):** Short-term gaps caused by transient network or power issues, assumed independent of the underlying acoustic signal.
- **Systematic Gaps (MAR):** Extended outages associated with device-specific factors such as hardware failure, where missingness is assumed to depend on sensor reliability rather than environmental conditions.

This classification informs the selection of appropriate imputation techniques, ensuring methodological alignment with the statistical properties of the data with four imputation methods selected with increasing levels of analytical complexity:

1. Historical Median Method: Utilises temporal patterns by imputing missing values based on historical observations at the same hour of day. This captures daily periodicity but is limited in cold-start scenarios (Xie et al., 2017).
2. KNN: Exploits spatial correlation by estimating missing values from nearby sensors using geographic proximity. This method performs well in densely monitored areas.
3. Hybrid Statistical Model: Combines temporal and spatial estimates using inverse-variance weighting, enabling adaptive balancing between the two sources based on data reliability.
4. Google TimesFM: A pre-trained time-series foundation model is employed for neural re-imputation. The model leverages recent temporal context to capture complex, non-linear patterns in the data. Its zero-shot capability eliminates the need for dataset-specific training, making it suitable for large-scale IoT applications.

The selection of these models allows for a comprehensive comparative analysis between traditional statistical approaches and modern machine learning techniques.

To support interpretation and validation, a visualization framework is developed using a combination of backend and frontend technologies. For backend, a Python-based API

(FastAPI) framework with a SQLite database is selected to enable efficient querying of time-indexed sensor data. An interactive dashboard is implemented using React frontend technology, using map-based visualization (MapLibre GL) for spatial analysis, time-series charts (Recharts) for temporal exploration and dynamic filtering based on imputation method. The visualization stack enables both descriptive and diagnostic analysis, allowing users to explore spatial patterns, temporal trends, and the effects of different imputation techniques in real time.

4.2. Methodological Justification

The chosen methodology is appropriate for the research objectives for several reasons:

1. It integrates both spatial and temporal dimensions of urban acoustic data.
2. It combines interpretable statistical methods with advanced machine learning models.
3. It enables reproducible and quantitative evaluation through standard error metrics.
4. It provides a practical framework for handling real-world IoT data challenges, including missingness and sensor unreliability.

Overall, the methodology supports a comprehensive investigation into data reconstruction techniques and their applicability to urban noise monitoring systems.

5 Implementation

This chapter describes the design and implementation of the system developed to address the research objectives. The implementation consists of four stages. The first stage covers data ingestion and pre-processing: 3 million row CSV export from Tallinn's IoT sensors is parsed, timestamps are normalised to Tallinn local time and truncated to hourly boundaries, sub-minute readings are aggregated into hourly medians using the median to reduce sudden noise spikes, and missing hours are counted across all 471 devices - revealing a fleet-wide gap rate of approximately 26%. The second stage implements four independent imputation methods: a historical median baseline that exploits the strong daily periodicity of urban noise, a spatial K-Nearest Neighbours method that draws on concurrent readings from geographically proximate sensors, an inverse-variance weighted combination of the two that adapts dynamically to whichever source is more reliable for each individual slot, and Google TimesFM 2.5, a pre-trained time series foundation model to re-impute slots where sufficient temporal context is available. The third stage presents the results through an interactive dashboard built with React, MapLibre GL, and Recharts, allowing both spatial and temporal exploration of the imputed data, as well as a quantitative comparison of model accuracy. The final stage evaluates the imputation methods using a stratified masked approach. Specifically, 20% of the original readings are withheld from 35 selected test devices, which are grouped into three categories: spatially connected, spatially isolated, and short history. This design helps reveal the distinct weaknesses of each method. Performance is assessed using MAE and RMSE in decibels

5.1. Data Ingestion and Pre-processing

The primary data source is a single CSV file exported from Tallinn's IoT acoustic monitoring platform, containing approximately 3 million individual sensor readings collected from devices over a four-month period from September to December 2021. Each row in the file represents a single measurement event and carries the sensor name, the recorded SPL in decibels, a production timestamp with UTC offset, and the geographic coordinates of the sensor. Throughout the project, this file is treated as immutable and read-only. All transformations are performed exclusively on derived data written to the database, ensuring that the original export can be re-processed from scratch

at any point without risk of data corruption or inconsistency. This design ensures full reproducibility: given the same source file, the entire pipeline can be re-executed to produce an identical database.

The ingestion stage is implemented through a sequence of three dedicated Python scripts, each with a clearly scoped responsibility. Rather than loading the full dataset into memory at once, each script processes the CSV in a single streaming pass, which keeps memory usage bounded regardless of file size and allows the pipeline to be run on modest hardware without specialised infrastructure.

The first script, *csv_to_sql.py*, constructs the device registry. It processes the raw CSV file to identify unique sensor names and records the first observed geographic coordinates for each sensor as its canonical location. Since the dataset contains millions of rows with repeated sensor entries—one per reading rather than per device—this deduplication step is required to assign stable identifiers for downstream processing. Devices are assigned sequential integer IDs sorted alphabetically by name, ensuring consistency across multiple pipeline runs. The resulting devices table is the main reference for all later data, linking every reading, imputed value, and evaluation result to a device using its ID.

A small number of devices show slight differences in their recorded locations, likely caused by GPS errors or inconsistent metadata. As sensors are fixed installations, the first observed coordinates are retained, and any differences exceeding a tolerance of 1×10^{-5} degrees (approximately one metre at Tallinn's latitude) are flagged rather than overwriting the reference location.

The second and most computationally intensive script, *csv_to_sp_levels.py*, converts raw readings into hourly aggregated SPL. For each valid record, the script parses the timestamp—accounting for two format variations—converts it from UTC to local time (Europe/Tallinn), and truncates it to the start of the corresponding hour. This local time alignment is essential, as urban noise patterns are driven by human activity cycles such as traffic peaks, working hours, and nighttime periods, which follow local time rather than UTC and shift during daylight saving transitions.

Readings sharing the same device and hour are grouped and reduced to a single value. Formally, let R(d,h) denote the set of readings for device d during hour h. The aggregated value is:

$$v_{d,h} = round(median(R_{d,h}))$$

The third script, *compute_missing_hours.py*, runs after the imputation tables have been generated and serves a diagnostic and metadata function. Rather than re-reading the raw CSV, it operates entirely on the already-processed *sp_levels* table, querying the minimum and maximum Unix timestamps per device to determine each sensor's operational window. The theoretical total number of hourly slots within this window is:

$$hours_{total} = (max(d) - min(d)) / 3600 + 1$$

The number of hours with at least one observed reading is obtained by counting distinct hourly timestamps in *sp_levels* for that device. Missing hours are then derived as:

$$missing_{hours} = max(0, total_{hours} - withData_{hours})$$

Together, the three scripts produce two database tables — **devices**, containing one record per sensor with its identifier, coordinates, and coverage statistics, and **sp_levels**, containing one record per device-hour pair with the aggregated SPL value — from which all imputation and evaluation steps read. The pipeline design ensures strict separation between the raw source and the derived database: no script modifies the CSV, and each script writes exclusively to its own designated portion of the schema.

This modularity means that individual stages can be re-executed independently if parameters change without requiring a full restart.

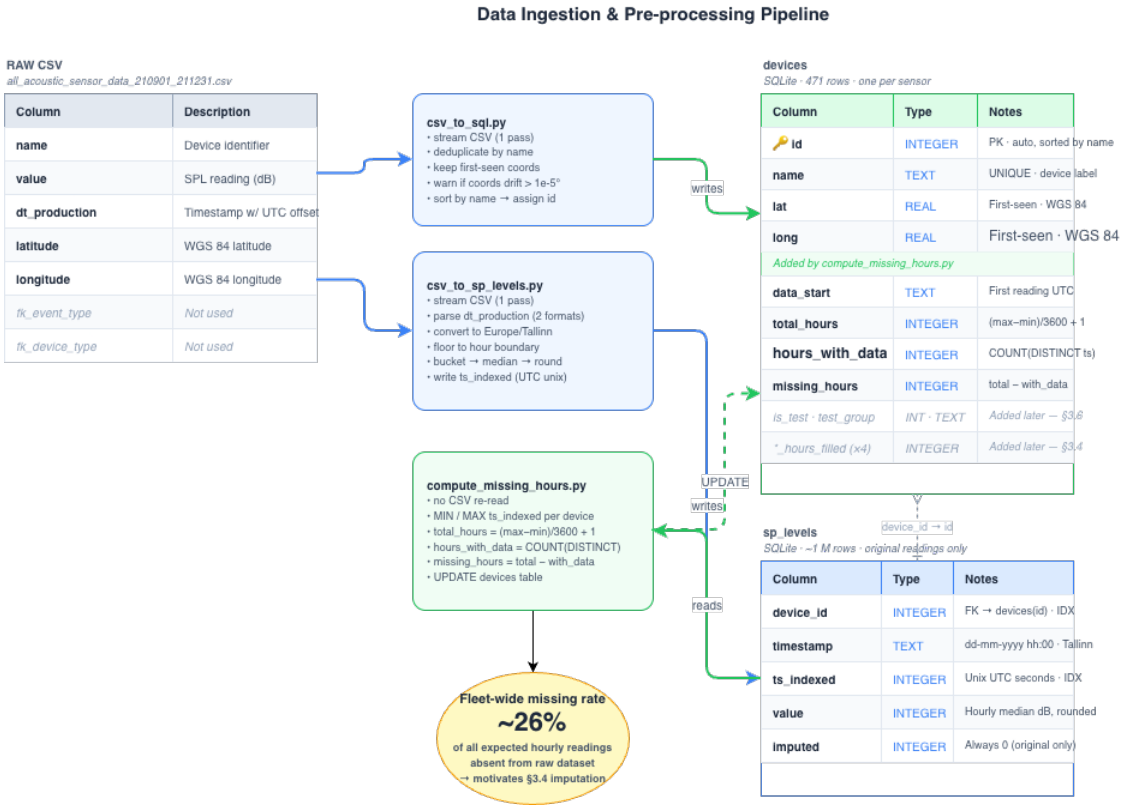


Figure 2 - Data Ingestion & Pre-processing Pipeline.

The ingestion process shows an overall missing rate of approximately 26%. However, this missingness is not uniform across devices. Some sensors have nearly complete records, with missing rates below 1%, while others lack data for most of their expected operational period.

5.2. Database Design

The system stores all sensor data, device metadata, imputed values, and coverage statistics in a single SQLite database (spl.db). This section describes the table schema and relationships between tables and documents the indexing strategy that ensures API query latency remains within interactive bounds.

The database consists of six core tables: one device registry, one table of original aggregated readings, and four tables corresponding to different imputation methods:

- **devices -** Stores one record per sensor device. Includes metadata such as device identifier, name, geographic coordinates, and coverage statistics (e.g. *data_start, data_end, missing_hours*).

- **sp_levels -** Contains the original hourly aggregated SPL. Each row corresponds to a (*device_id, hour*) pair and includes both timestamp representations (*timestamp* and *ts_indexed*) alongside the measured *value*. This table is treated as immutable after pipeline execution.

- **Imputation tables (4x) -** Each imputation method is stored in its own table with identical schema to *sp_levels*, including the *imputed* flag. This ensures uniform query patterns across methods and eliminates schema-level branching in the backend.

The schema prioritises query simplicity and consistency over normalisation depth. While a fully normalised design could separate timestamps, measurements, and imputation metadata into multiple relational layers, such decomposition would introduce unnecessary join operations for a read-heavy workload. Given that the system is primarily analytical and serves precomputed or indexed data, denormalisation improves performance without compromising correctness.

Separating imputation methods into distinct tables also ensures method isolation. Each imputation pipeline writes independently to its own table, eliminating the risk of overwriting or contaminating results between methods. This design further simplifies evaluation, as comparisons can be performed by switching table references rather than filtering rows by method identifiers.

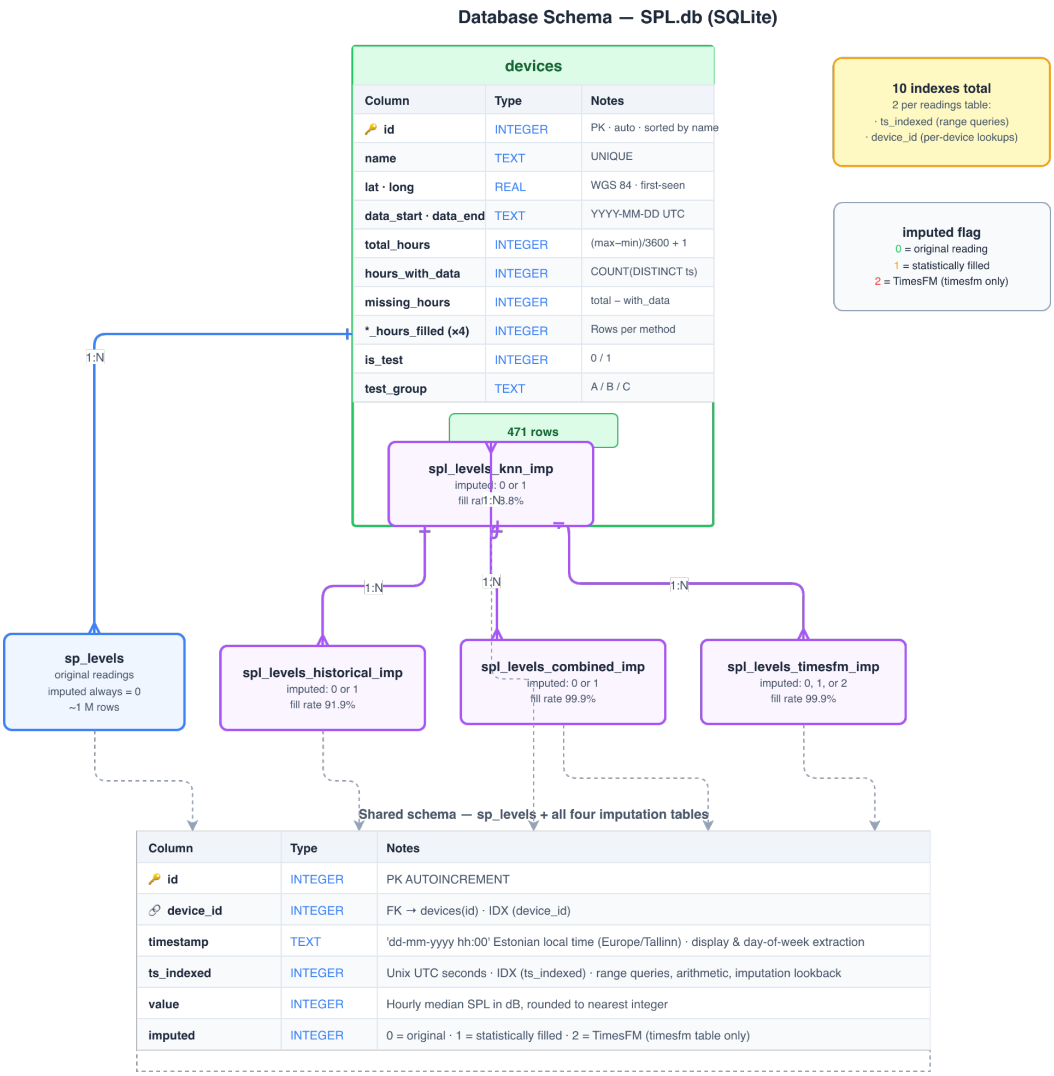


Figure 3 - Database Schema.

The device registry contains one row per physical sensor and is initially populated by csv_to_sql.py, which creates the core schema consisting of id, name, lat, and long. Subsequent stages extend this table incrementally:

- **Coverage statistics** (added by compute_missing_hours.py): data_start, data_end, total_hours, hours_with_data, missing_hours

- **Imputation coverage metrics** (added by compute_imputation_coverage.py): hist_hours_filled, knn_hours_filled, combined_hours_filled, timesfm_hours_filled — each representing the number of imputed records generated per device by the corresponding method

- **Evaluation metadata:** (added by select_test_devices.py): is_test, test_group — used to identify the 35 evaluation devices and assign them to stratified groups for analysis

Imputation Tables. The system defines four imputation tables: spl_levels_historical_imp, spl_levels_knn_imp, spl_levels_combined_imp and spl_levels_timesfm_imp**.** Each of these tables mirrors the schema of sp_levels exactly, ensuring structural consistency across all methods. They contain a complete set of device-hour records covering each device's active time span, including both original observations (copied from sp_levels) and generated values produced by the respective imputation algorithm.

5.3. Imputation Methods

Missing values in the sensor dataset are not randomly distributed. IoT devices tend to fail in bursts - for example due to power outages or network failures - resulting in contiguous gaps of several hours. In addition, some devices exhibit structural missingness caused by delayed deployment or early decommissioning.

Four imputation methods are implemented in increasing order of complexity: historical median, spatial KNN, an inverse-variance weighted combination of both, and a neural time series foundation model (TimesFM). Each method produces a fully independent imputed table; the methods are not layered on top of one another. This section describes their algorithms, implementation details, and resulting fill rates.

All methods operate on the same input representation: a per-device hourly timeline spanning each device's active period from *data_start* to *data_end*. Each hourly slot is either observed (present in *sp_levels*) or missing. The objective of imputation is to assign a value to every missing slot.

5.3.1. Historical Median

The historical median method exploits the strong periodic structure of urban noise, where traffic patterns, commercial activity, and human behaviour exhibit consistent daily and weekly cycles. As a result, observations from the same hour in previous days serve as reliable predictors for missing values.

For a missing value at device d and timestamp t, corresponding to hour-of-day h, the method retrieves up to 10 prior observations from the same device and hour. The most recent 10 values are retained, and the imputed value is computed as:

$$\hat{y} = round(median(L(d,h,t)[-10:]))$$

A window of 10 observations is sufficient to smooth short-term anomalies (e.g., isolated quiet or noisy days), while remaining small enough to avoid mixing distinct seasonal effects such as holidays or daylight saving transitions. If no prior observations exist for a given device–hour combination, the value cannot be imputed. This primarily affects early stages of device operation and is handled by skipping such cases. The historical method achieves a fill rate of 91.9%, with failures concentrated in cold-start scenarios and early device activity periods.

5.3.2. Spatial KNN

The spatial KNN method exploits spatial autocorrelation in urban noise: nearby sensors are exposed to similar acoustic environments.

For a missing slot at device *d* and time *t*:

$$Nr_d = \{d' \mid d' \neq d, dist(d,d') \leq r$$
$$V_{(d,t,r)} = \{ v(d',t) \mid d' \in Nr(d) \}$$
$$\hat{y} = \text{round}(\text{median}(V(d, t, r)))$$

A two-stage radius strategy is used:

1. Primary radius (500 m): used if at least 3 neighbours have data
2. Fallback radius (1000 m): used if insufficient neighbours exist
3. Skip: if no neighbours have data at time *t*

The median is used instead of a distance-weighted mean to avoid sensitivity to outliers and because sensors within short urban distances often exhibit near-identical values. To reduce runtime complexity, Pairwise Haversine distances (Sinnott, 1984) are computed once for all 471 devices, producing a 471×471 distance matrix. From this, two lookup

tables are created: neighbours_500 and neighbours_1000. The spatial KNN method achieves a fill rate of 98.8%, with remaining gaps occurring when no neighbouring device has data at the same timestamp.

5.3.3. Combined (Inverse-Variance Weighted Blend)

Historical and spatial methods fail in complementary cases: historical fails for cold-start devices, while KNN fails for spatially isolated or temporally sparse configurations. The combined method integrates both sources when available.

The combined method values are computed as:

$$\hat{y}_{hist} = median(L)$$
$$\hat{y}_{knn} = median(V)$$

$$w = 1 \,/\, max(Var(samples), 1)$$
$$\hat{y} = round((w_{hist} \cdot \hat{y}_{hist} + w_{knn} \cdot \hat{y}_{knn}) \,/\, (w_{hist} + w_{knn}))$$

L is a set of up to 10 past readings from the same device taken at the same hour of day. V is a set of readings from nearby sensors at the same timestamp. For each set, the median is calculated to produce two independent estimates: one based on the device's own historical behaviour, and one based on spatial information from neighbouring sensors. These two estimates are calculated separately and do not influence each other.

Next, the reliability of each estimate is assessed. This is done by measuring how much the values in each set vary. If the readings in a set are very similar, the estimate is considered more reliable. If the readings vary widely, the estimate is considered less reliable. To avoid extreme cases where very low variation could distort the result, a minimum variance threshold is applied.

Finally, the two estimates are combined based on their reliability. The more consistent (lower-variance) estimate is given a higher influence in the final result, while the more uncertain estimate contributes less. This means that if historical data is more stable than spatial data, the result will lean more toward the historical estimate, and vice versa. If

both sources are equally reliable, they contribute equally. The final value is rounded to match the resolution of the sensor measurements.

The combined method achieves 99.9% coverage, with remaining gaps due to simultaneous absence of both spatial and temporal information.

5.3.4 TimesFM

TimesFM 2.5 is a transformer-based time series foundation model developed by Google Research and released under Apache 2.0. The implementation used is the 200M-parameter PyTorch variant (google/timesfm-2.5-200m-pytorch).

Unlike statistical methods, TimesFM does not rely on explicit spatial structure or handcrafted periodicity assumptions. Instead, it learns temporal dynamics directly from historical sequences.

TimesFM is applied on top of the combined imputation output. It does not modify raw data but re-estimates selected imputed slots where sufficient context exists. A three-state flag is used in the output in Table 1.

Table 1 - Imputed flag description

Imputed	**Meaning**
0	original measured value
1	retained statistical estimate
2	replaced by TimesFM prediction

If fewer than 72 prior observations exist, the combined (KNN and historical) estimate is retained.

To improve efficiency, inference is performed in batches rather than per-slot calls. Batch size is determined by model configuration (default 32). This reduces runtime from impractical serial execution to ~13 hours on CPU.

Final coverage remains 100%, as TimesFM only replaces already-imputed slots rather than expanding coverage. Its contribution is primarily accuracy improvement rather than increased completeness.

5.6. Visualization Dashboard

The visualization layer of this project is implemented as a React single-page application that provides interactive access to the outputs of the imputation pipeline. Its primary objective is to enable exploratory analysis of processed sensor data without requiring direct database interaction. This includes inspection of individual sensor time series, spatial distributions of SPL data at arbitrary timestamps, animated temporal evolutions, and comparative evaluation of imputation methods. This section presents the system architecture, the global state mechanism linking the frontend to multiple imputation tables, the design of each application view, and the API contract between frontend and backend components. It outlines four architectural layers (storage, pipeline, backend and frontend layers) which ensure separation of concern between different components.

The core data repository is a single SQLite database (data/SPL.db), which serves as the central storage component of the system. It contains device metadata in the *devices* table, processed and aggregated sensor readings in the sp_levels table, and four additional tables corresponding to the outputs of the different imputation methods.

Following the completion of the data processing pipeline, the database assumes a read-only role during system operation. The backend interacts with the database exclusively through query operations, with no modifications performed at runtime.

In addition to the database, two CSV files (evaluation_results.csv and evaluation_summary.csv) are used to store evaluation outputs. These files are generated once during the evaluation stage and subsequently accessed in a read-only manner by the backend. Storing evaluation results in flat-file format ensures independence from the database schema and facilitates straightforward inspection and sharing of results without requiring database-specific tools.

The pipeline layer is composed of a collection of standalone Python scripts located in the *scripts/* and *imputation/* directories. Each script performs a distinct processing task, such

as data ingestion, temporal aggregation, imputation, or evaluation, and writes its results to the storage layer.

The scripts are executed sequentially in a predefined order and operate independently of both the backend and frontend components. This separation enables partial re-execution of the pipeline, allowing individual stages to be rerun without restarting the entire system. It also supports independent validation and testing of each processing step.

Imputation methods are designed to be largely independent of one another. Each method reads from the base tables (devices and sp_levels) and produces its own output table. The only exception is the TimesFM-based method, which builds upon the combined imputation output. This dependency is explicitly defined, while all other methods remain fully decoupled.

The backend is implemented as a FastAPI-based RESTful service that exposes the stored data through structured JSON endpoints. It follows a stateless design: each incoming request establishes a new connection to the SQLite database, executes the required query, and returns the result without maintaining session state.

All endpoints accept a *source* parameter, which determines the selected imputation method. This parameter is resolved centrally through a dedicated mapping function, ensuring consistent handling of data source selection across all endpoints.

The frontend is implemented as a React-based single-page application that provides an interactive interface for data exploration. It communicates exclusively with the backend via HTTP requests and has no direct interaction with the underlying database or storage structures. This strict separation ensures that changes to backend implementation details, such as database schema or technology stack, do not affect the frontend application (Figure 4).

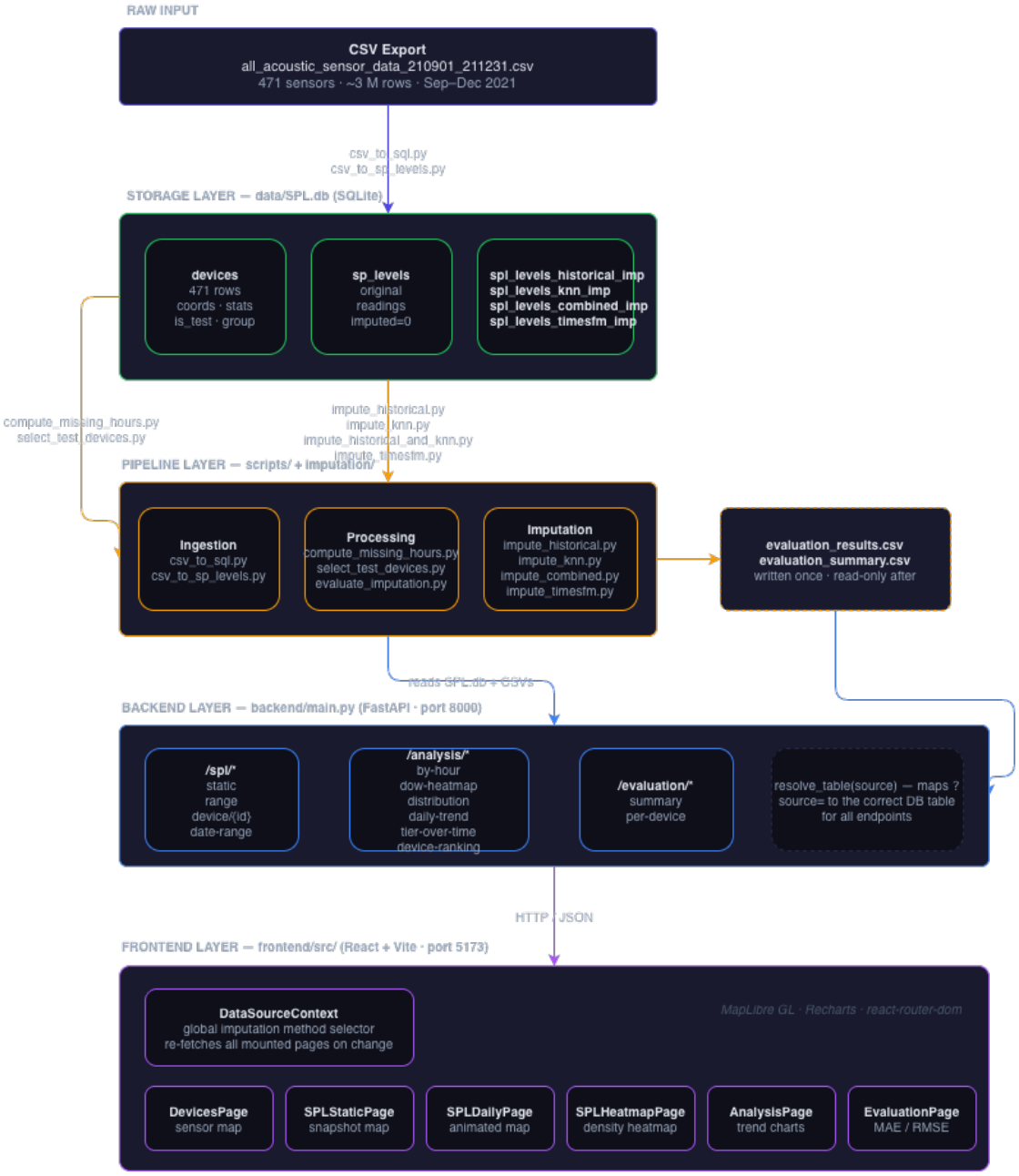


Figure 4 - Visual dashboards architecture.

5.6.1. Technology Choices and Justification

SQLite was selected over client–server database systems such as PostgreSQL or MySQL due to its suitability for the specific requirements of this research.

The complete processed dataset—including device metadata, several million hourly observations across 471 sensors, and multiple imputation tables — can be stored within a single *.db* file of approximately 500 MB. This eliminates the need for a dedicated database server, authentication mechanisms, or configuration files. As a result,

reproducing the system on another machine requires only the transfer of a single file, significantly simplifying deployment and ensuring reproducibility.

Following the execution of the data processing pipeline, the database operates in a read-only capacity. SQLite is well-suited to this scenario, as it efficiently supports concurrent read operations without the overhead associated with more complex database management systems. Features such as connection pooling, replication, and advanced transaction management are unnecessary in this context and would introduce avoidable complexity.

The largest table in the database contains approximately 3.5 million rows. With appropriate indexing (e.g., on *device_id* and *ts_indexed*), query response times remain within interactive thresholds, typically below 100 milliseconds for standard queries and under 300 milliseconds for aggregate operations. This level of performance is sufficient for real-time data exploration without requiring additional optimisation layers such as caching.

Python was chosen as the primary development language for both the data processing pipeline and the backend API due to its extensive ecosystem for data analysis and machine learning. Libraries such as NumPy, pandas, and scikit-learn provide efficient implementations of the statistical methods used in this study. Additionally, integration with TimesFM necessitates the use of Python, as the model is accessible through a Python-based interface.

FastAPI was selected as the backend framework due to its lightweight design and developer-oriented features. It provides automatic generation of API documentation via the OpenAPI standard, facilitating testing and debugging. Furthermore, its declarative approach to request validation and parameter handling reduces boilerplate code and improves maintainability. The resulting backend implementation remains concise while providing all required functionality.

React was selected for frontend development due to its component-based architecture, which aligns well with the modular structure of the dashboard. Each analytical view is implemented as an independent component, while shared state—such as the selected imputation method—is managed centrally through a context mechanism. This approach simplifies state propagation and avoids unnecessary coupling between components.

Vite was chosen as the build tool in place of traditional alternatives due to its fast development server and efficient hot-module replacement. This significantly improves the development workflow, particularly in scenarios involving frequent interface updates.

5.6.2. Application Architecture

The frontend is developed using React with the Vite build toolchain and structured around React Router for client-side navigation. The application defines seven routes, each corresponding to a distinct analytical view. A persistent navigation bar is rendered across all pages and includes both route navigation and a global imputation method selector, which determines the active data source for all data-dependent views.

The backend is implemented using FastAPI (Python). All frontend requests are handled through RESTful endpoints that query a SQLite database and return JSON responses. No object-relational mapping (ORM) layer is used; instead, SQL queries are executed directly via Python's *sqlite3* module. The backend is stateless in the sense that each request is independent and self-contained.

All primary data views (Devices, SPL Static, SPL Daily, SPL Chart, SPL Heatmap, and Analysis) support switching between five data sources: the original measurements and four imputation variants. This selection is controlled via a global dropdown in the navigation bar and implemented using React Context (Figure 5).

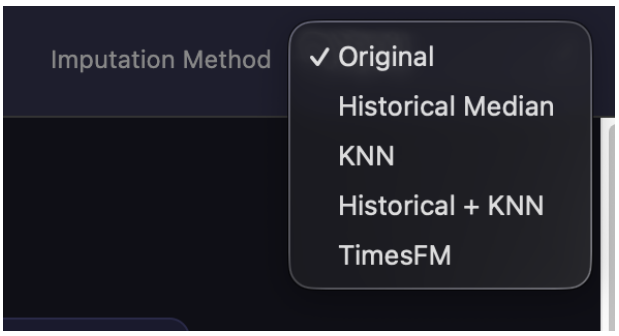


Figure 5 - Imputation method selections.

Each data endpoint accepts a *source* query parameter, which is resolved to a corresponding database table via a mapping function as shown in Figure 6.

```
def resolve_table(source: str) -> str:
    tables = {
```

```
        "original": "sp_levels",
        "historical": "spl_levels_historical_imp",
        "knn": "spl_levels_knn_imp",
        "combined": "spl_levels_combined_imp",
        "timesfm": "spl_levels_timesfm_imp",
    }

    if source not in tables:
        raise HTTPException(
            status_code=400,
            detail=f"Unknown source: {source}"
        )
```

Figure 6 - Endpoint to database table mapping.

This design centralises table resolution in a single function, avoiding duplication across endpoint implementations. When the selected source is changed in the frontend, all dependent views automatically refetch data, resulting in immediate updates of map layers, charts, and tables.

5.6.3. Page Designs

Devices Page (/devices)

Purpose. This page provides an inventory of all 471 sensors and visualises data completeness as a measure of device health. It serves as the primary entry point for assessing data coverage across the sensor network.

Map layer. Each device is rendered as a rectangular marker on a MapLibre map. Marker colour encodes the completeness ratio, defined as the proportion of *total_hours* for which a value is available (measured or imputed, depending on the selected source). Green indicates completeness above 95%, yellow 85–95%, and red below 85%.

For the original dataset, completeness is computed as *hours_with_data* / *total_hours*. For imputation-based sources, the numerator corresponds to precomputed fill counts stored

in the *devices* table (e.g., *hist_hours_filled, knn_hours_filled*). Switching the imputation method updates marker colours dynamically without reloading the page.

Device panel. Selecting a marker shows a detailed information panel displaying metadata such as device identifier, temporal coverage (*data_start, data_end*), total hours, and per-method completeness metrics. Test devices (*is_test = 1*) are highlighted and annotated in blue with their respective evaluation group (*test_group*) (Figure 7).

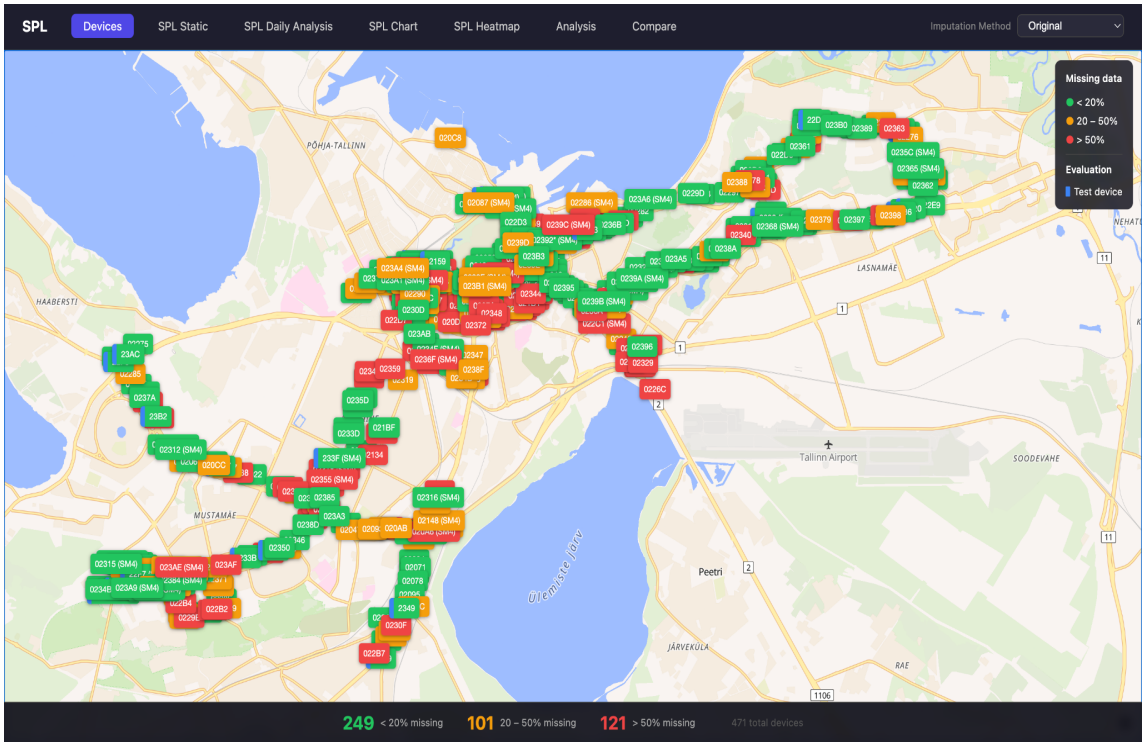


Figure 7 - Devices page.

SPL Static Page (/spl-static)

Purpose. This view provides a snapshot representation of spatial SPL distribution at a user-defined timestamp.

Controls. A date picker and hourly selector (0–23) allow specification of the target timestamp. The valid range is dynamically retrieved from the backend via the */data/date-range* endpoint.

Map layer. Devices with available readings at the selected timestamp are displayed as coloured markers according to the WHO noise classification: < 45 dB - green (low exposure), 45–54 dB - lime, 55–64 dB - yellow, 65–74 dB - orange and ≥ 75 dB - red

(high exposure). Devices without available values are rendered in grey, ensuring that spatial gaps in coverage are visually explicit. In cases where imputation is enabled but a value is still unavailable, the device remains greyed out, indicating unresolved missingness.

A legend is provided to map colour values to WHO categories. For the TimesFM-based imputation source, an additional provenance indicator (pink if imputed) is displayed in the pin to distinguish between observed, statistically estimated, and model-predicted values (Figure 8).

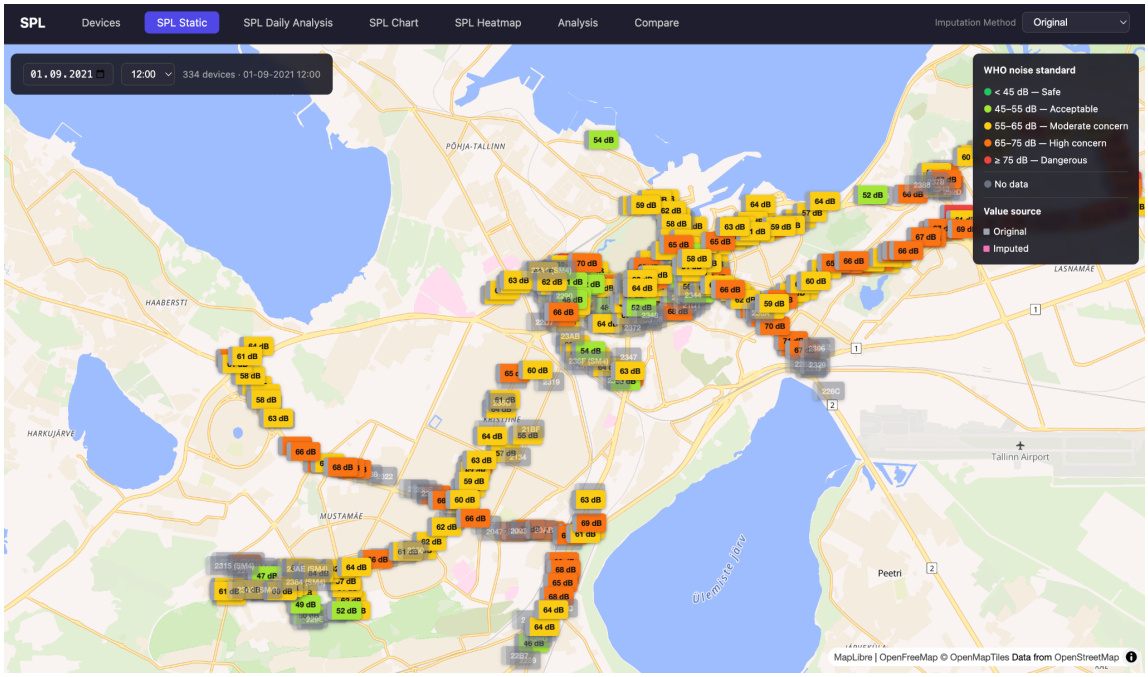


Figure 8 - SPL Static page.

SPL Daily Page (/spl-daily)

Purpose. This page provides a temporal animation of SPL variation across all devices over a selected time interval, advancing in hourly increments (Figure 9).

Controls. Users define a start and end date using constrained date inputs. Playback controls include play/pause, adjustable speed (1x–10x), and manual stepping. A timestamp overlay indicates the current frame in Tallinn local time, and a tier distribution counter summarises the number of devices in each WHO category.

Implementation. Upon activation, all required frames are preloaded via a single request to */data/daily-range*. The resulting dataset is stored in memory and iterated using a timer-

driven index. This design eliminates runtime data-fetching during animation, ensuring stable performance even at higher playback speeds (up to approximately 10 frames per second for a 30-day interval).

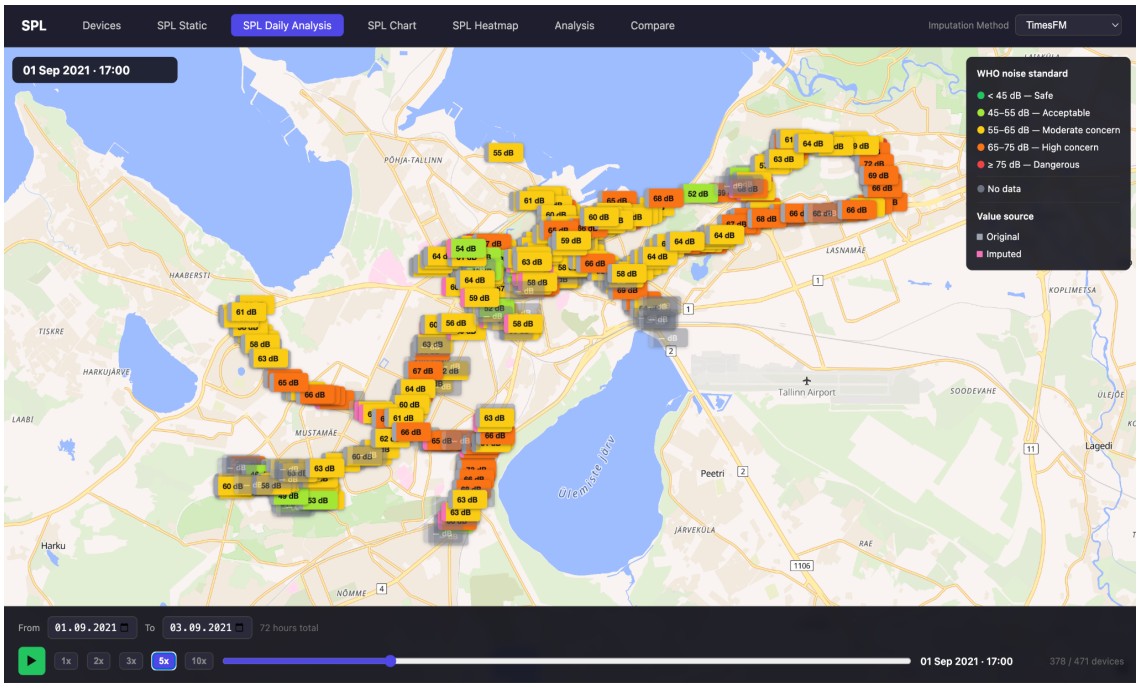


Figure 9 - SPL Daily Analysis page.

SPL Chart Page (/spl-chart)

Purpose. This view presents a detailed time series for an individual device across the full observation period. Devices may be selected via a dropdown menu (Figure 10).

Chart. A Recharts LineChart is used to visualise hourly SPL values. Per-point colouring is applied based on WHO tier classification, either through segmented line rendering or custom point components. A brush component enables interactive zooming without requiring additional data fetching.

Missing observations in the original dataset appear as discontinuities in the time series. When an imputed dataset is selected, these gaps are filled, resulting in a continuous trajectory. A legend differentiates observed values (blue) from imputed values (grey), where applicable.

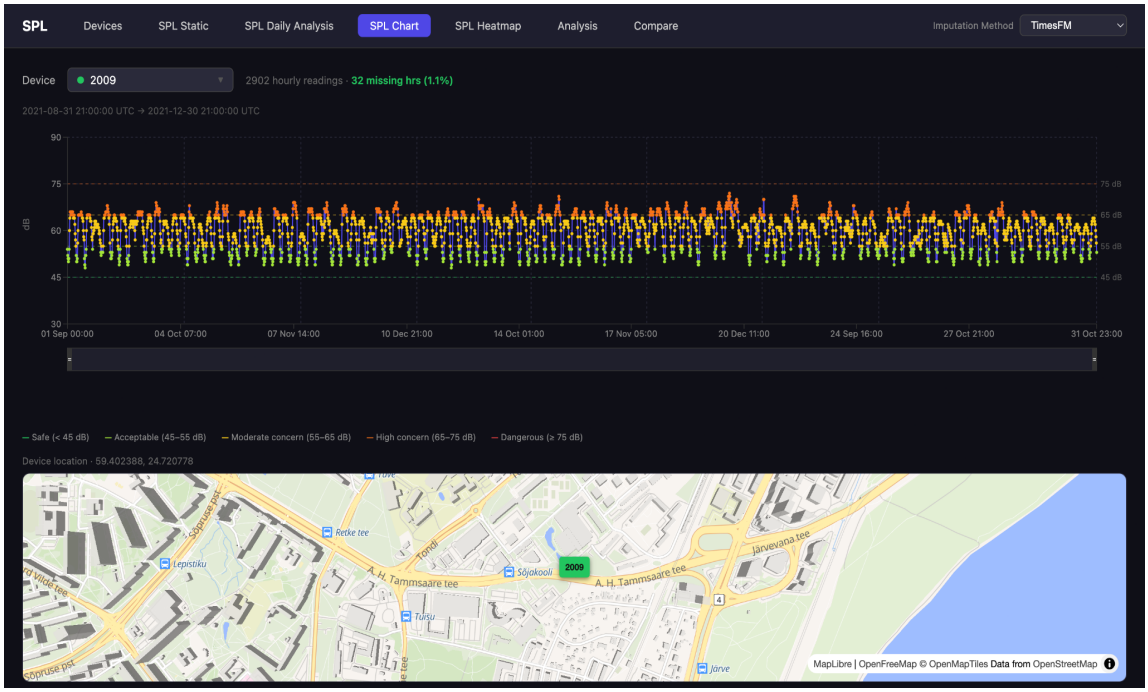


Figure 10 - SPL Chart page.

SPL Heatmap Page (/spl-heatmap)

Purpose. This view visualises spatial intensity distributions of SPL as a continuously evolving heatmap over time (Figure 11).

Rendering. Data is provided as a GeoJSON FeatureCollection (Butler et al., 2016), where each feature represents a sensor location with an associated SPL value. These values are used as weights in a MapLibre heatmap layer, producing a continuous spatial density field. The colour scale transitions from transparent (low intensity) through green, yellow, and orange to red, consistent with WHO categorisation (World Health Organization, 2018).

Animation. Temporal progression is implemented using the same preloading mechanism as the SPL Daily view. Each frame updates the underlying GeoJSON source via *setData()*, enabling efficient GPU-accelerated re-rendering without reinitialisation of map layers.

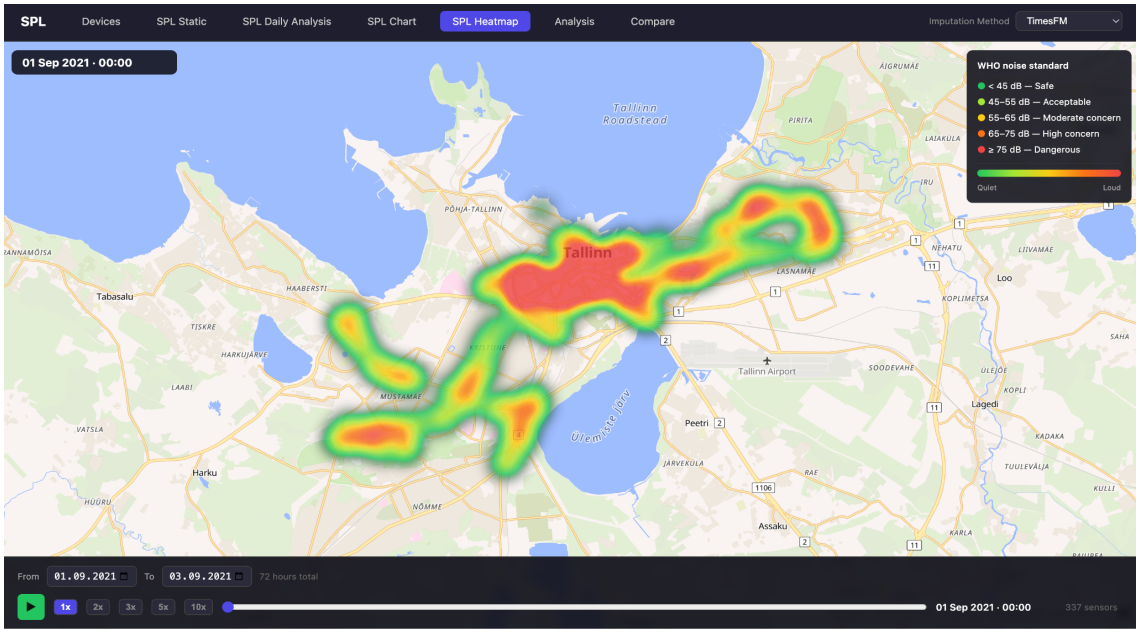


Figure 11 - SPL Heatmap page.

Analysis Page (/analysis)

Purpose. This page provides aggregated statistical summaries of the dataset, enabling analysis of temporal and spatial patterns in environmental noise exposure (Figure 12).

The interface is structured into several analytical modules:

WHO Tier Distribution. Displays the proportion of device-hours falling within each WHO noise category, aggregated across the entire dataset and optionally segmented by imputation method.

Figure 12 - WHO tier and SPL value distribution.

Hourly Profile. Illustrates the diurnal pattern of noise exposure using median SPL values per hour of day, supplemented with interquartile ranges.

Day-of-Week Profile. Presents average SPL values grouped by weekday, highlighting systematic weekly variation (Figure 13).

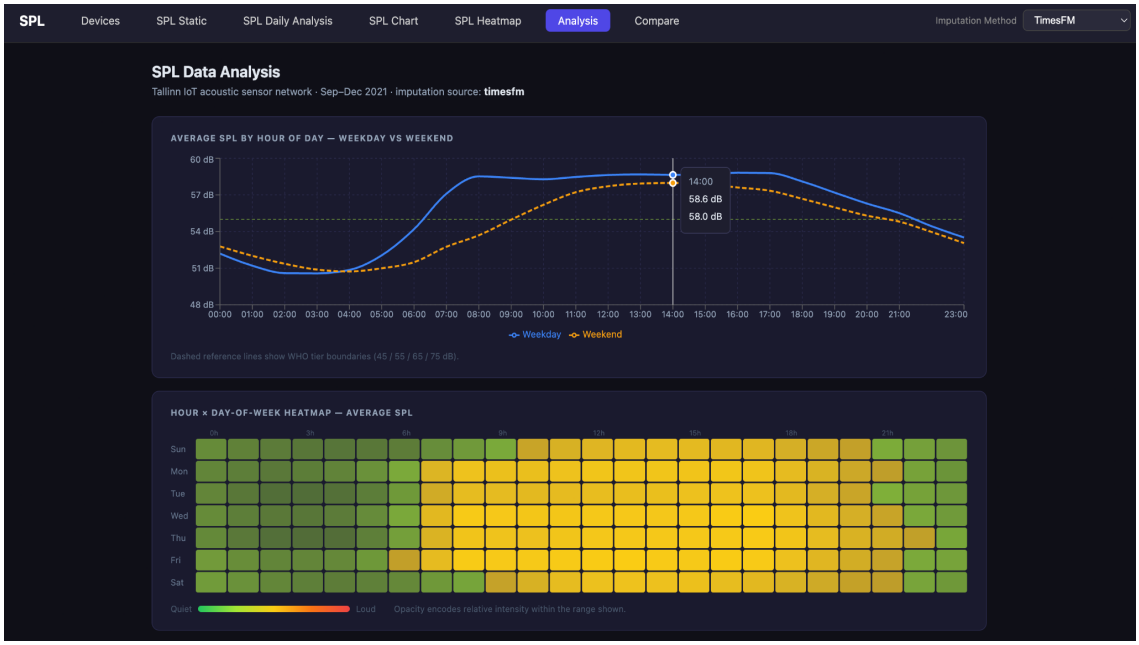


Figure 13 - SPL data over week and weekend.

Device Ranking. Identifies the 15 highest- and 15 lowest-exposure devices based on mean SPL. A complementary map visualises these devices spatially, enabling identification of geographic clustering (Figure 14).

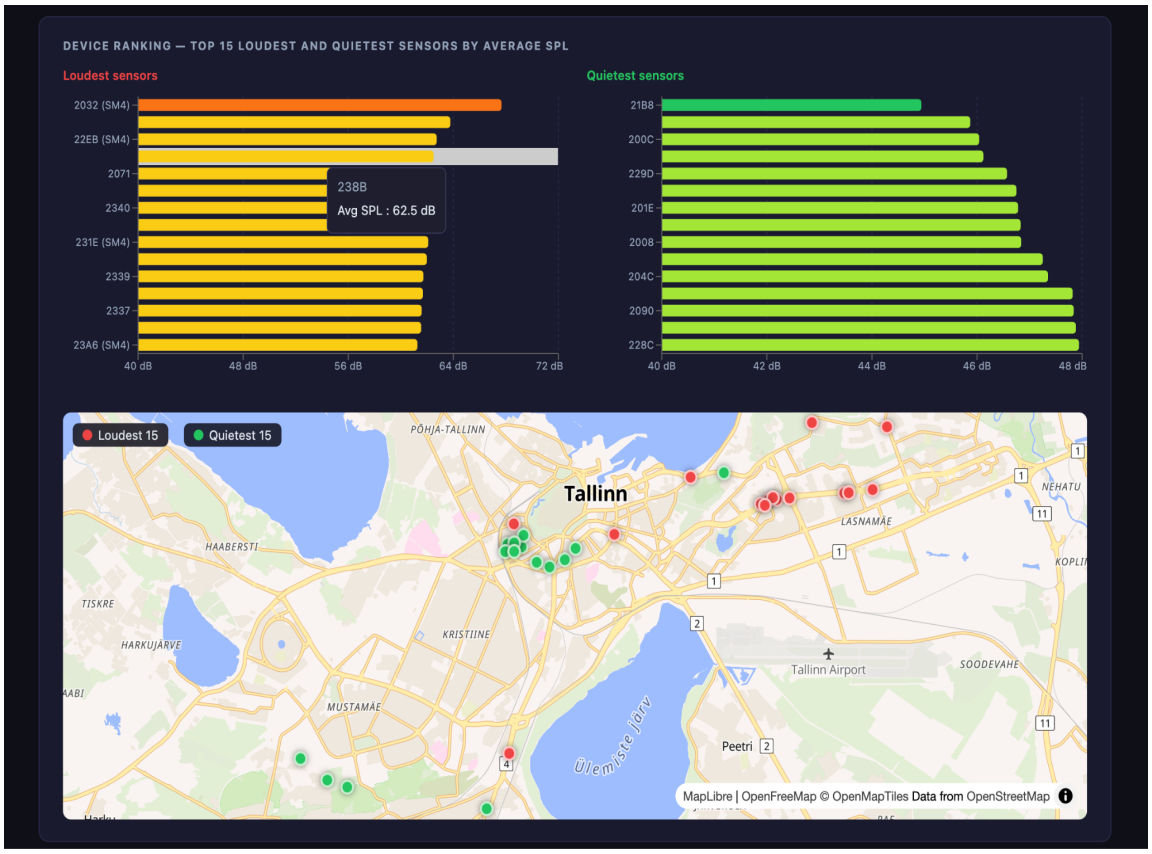


Figure 14 - Loudest and quietest sensors.

Evaluation Page (/compare)

Purpose. This view provides a quantitative comparison of the four imputation methods using precomputed evaluation metrics (Figure 15 and Figure 16).

Data sources. The page consumes two backend endpoints: */evaluation/summary*, which provides aggregated MAE and RMSE values by method and device group, and */evaluation/per-device*, which contains device-level error metrics.

Summary visualisation. A grouped bar chart presents comparative performance across different device groups (Overall, Group A, Group B, Group C), enabling assessment of method robustness under varying data availability conditions. A sortable table lists

individual device-level errors across all methods. Devices are annotated with their evaluation group to support interpretation of group-level trends.

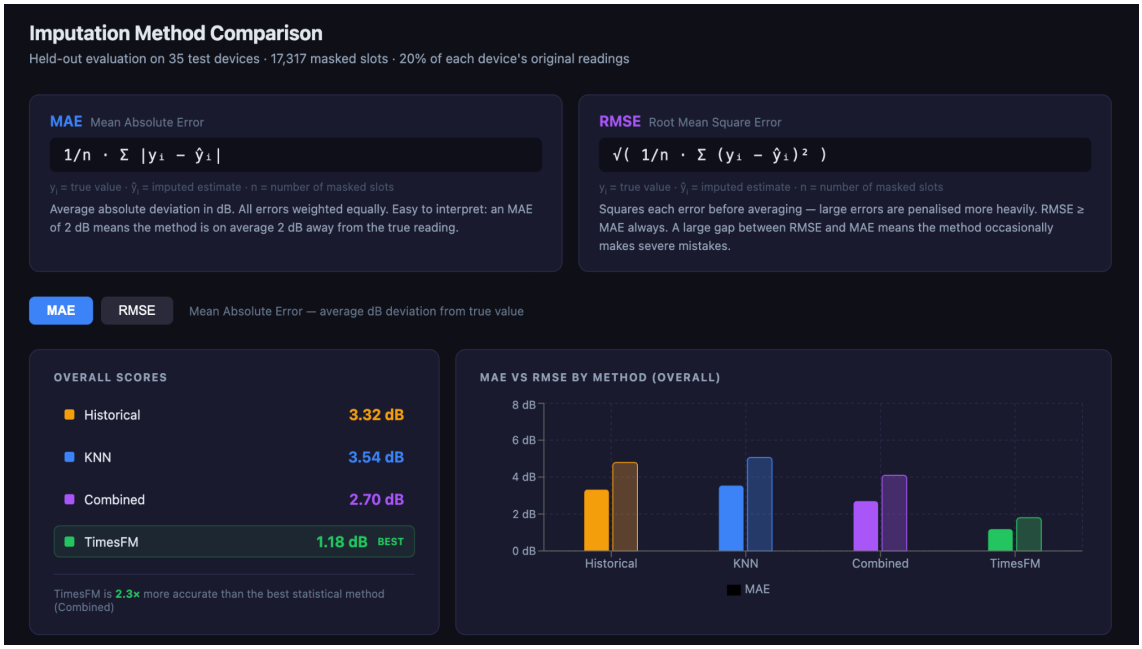
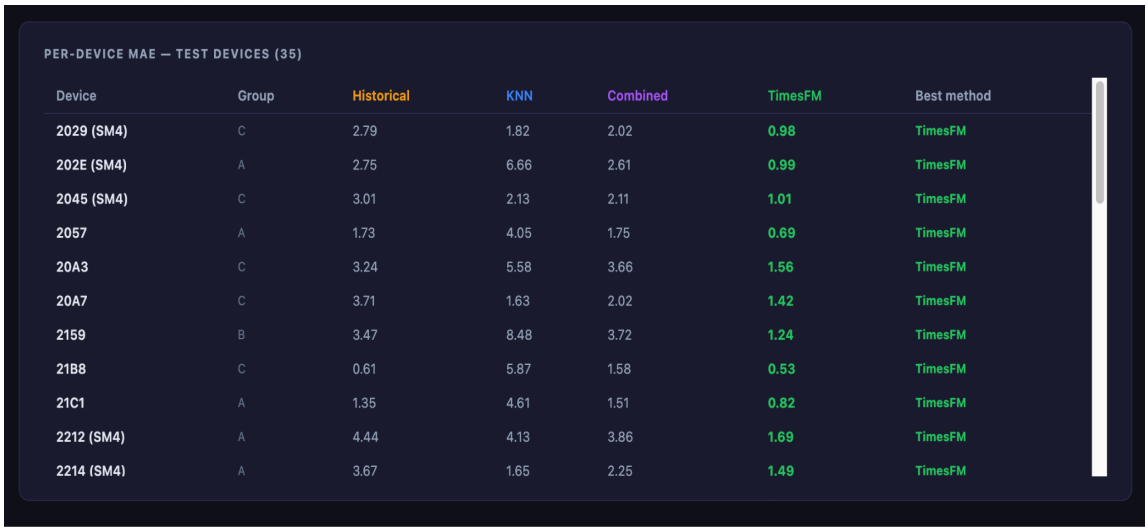


Figure 15 - Imputation method comparison.

Figure 16 - Test device groups.

Design rationale. This view is intentionally decoupled from the global imputation selector, as simultaneous comparison across all methods is required. Coupling it to a single active source would preclude meaningful comparative analysis.



PER-DEVICE MAE — TEST DEVICES (35)

Device	Group	Historical	KNN	Combined	TimesFM	Best method
2029 (SM4)	C	2.79	1.82	2.02	0.98	TimesFM
202E (SM4)	A	2.75	6.66	2.61	0.99	TimesFM
2045 (SM4)	C	3.01	2.13	2.11	1.01	TimesFM
2057	A	1.73	4.05	1.75	0.69	TimesFM
20A3	C	3.24	5.58	3.66	1.56	TimesFM
20A7	C	3.71	1.63	2.02	1.42	TimesFM
2159	B	3.47	8.48	3.72	1.24	TimesFM
21B8	C	0.61	5.87	1.58	0.53	TimesFM
21C1	A	1.35	4.61	1.51	0.82	TimesFM
2212 (SM4)	A	4.44	4.13	3.86	1.69	TimesFM
2214 (SM4)	A	3.67	1.65	2.25	1.49	TimesFM

Figure 17 - Individual device test results

6 Evaluation

Evaluation is performed by selecting a subset of devices, intentionally masking a portion of their observed values, and imputing them using the aforementioned methods. The imputed values are then compared against the original observations. Moreover, a detailed comparison of imputation methods requires the evaluation data to be clean, representative and free from selection bias. This section outlines the preprocessing steps applied prior to evaluation, including sensor eligibility filtering, outlier handling, temporal coverage alignment, the use of imputation flags to define ground truth, and reproducibility controls to ensure consistent results across runs.

6.1. Sensor Eligibility Filtering

Not all devices in the fleet are suitable for evaluation. Devices with excessive missingness provide limited ground truth, while devices with very short operational histories do not support statistically stable estimates. Two hard criteria are therefore applied before a device can enter the evaluation pool.

Criterion 1 — Missing rate < 5%.

$$CAST(missing_{hours}\ AS\ REAL)\ /\ total_{hours}\ <\ 0.05$$

Devices exceeding this threshold would contribute more imputed than observed values once evaluation masking is applied. This would introduce circularity, since imputed values could end up being validated against other imputed values, artificially inflating performance for methods that are internally self-consistent. The 5% bound ensures that the evaluation is dominated by genuine sensor observations.

As an illustrative case, a device at the threshold (5% missing, e.g. 150 missing hours out of 3000) retains approximately 2850 observed readings. With a 20% evaluation mask, roughly 570 ground-truth points remain available, which is sufficient for stable MAE and RMSE estimation.

Criterion 2 — At least 300 observed hours (hours_with_data ≥ 300).

Three hundred hours corresponds to roughly 12.5 days of continuous operation. Below this threshold, the historical method cannot reliably populate its MAX_LOOKBACK = 10 same-hour history requirement, since full coverage of all hour-of-day slots requires at least 10 days of observations. Such devices would therefore place the historical baseline in a structural cold-start regime inconsistent with its intended operating conditions. Moreover, a 20% mask on 300 observations yields only 60 evaluation points, which is borderline for stable error statistics.

From all devices in the database, those satisfying both constraints form the eligible pool. From this subset, 35 devices are selected for evaluation via the stratification procedure. Excluded devices fall into two categories: (i) high-missingness sensors (often connectivity-limited deployments), and (ii) short-lived deployments, typically late-deployed or early decommissioned sensors during Sep–Dec 2021.

6.2. Outlier and Anomaly Handling

During ingestion, non-parsable values are discarded. No explicit lower or upper dB bounds are applied. Observed values across the fleet span approximately 28–94 dB, consistent with MEMS-based urban acoustic sensing (Nencini et al. 2020). While values below 30 dB and above 90 dB are rare, they are physically plausible in extreme but valid conditions (e.g. near-quiet environments or construction-heavy locations) and are therefore retained.

The hourly aggregation step computes the median of all raw readings within each one-hour window per device. After aggregation, no further range-based filtering is applied. This avoids introducing bias by excluding extreme but valid environmental conditions. The evaluation is explicitly designed to cover the full observed SPL distribution, including tail events. Masked evaluation points are sampled uniformly from the full set of observed values, ensuring that both typical and extreme acoustic conditions are represented in the scoring distribution.

6.3. Temporal Coverage Alignment

For the evaluation to be fair, every method must be assessed on exactly the same set of time slots. If one method were tested on a slightly different set of timestamps than another, differences in their scores could reflect the different difficulty of their respective slots

rather than a genuine difference in accuracy. To prevent this, all four methods are applied to the same hourly range for each device — from that device's first recorded reading to its last — and the evaluation script recomputes each method's estimate fresh for every masked slot, rather than reading from the pre-generated imputation tables. This means all methods see identical input data for every slot being tested, and no timing or context mismatch can creep in between them.

There are cases where a method is genuinely unable to produce an estimate. The historical method cannot impute a slot if the device has no prior readings for that hour of day — a situation that occurs at the very beginning of a device's operational life. The KNN method cannot impute a slot if none of the device's spatial neighbours have a reading at that same timestamp. In both cases the result is recorded as missing, and those slots are simply excluded when computing the accuracy metrics for that method. The number of slots each method was actually able to estimate is reported alongside its MAE and RMSE, so any coverage gaps between methods are visible rather than hidden in the aggregate scores.

6.4. Reproducibility Controls

Two steps in the evaluation involve randomness: selecting which 35 devices to use as test devices, and selecting which of each device's readings to mask as held-out ground truth. Both are made fully deterministic by setting the random seed to 42 at the start of each script. This means that re-running the evaluation on the same database will always produce the same test devices, the same masked slots, and therefore the same accuracy scores.

The mask itself is constructed by selecting 20% of each device's observed readings at random, with a minimum of one slot per device. Across all 35 test devices, this produces 17,317 masked evaluation points in total. The selection is stable not only because of the fixed seed but also because the readings are always presented to the sampler in the same order — the database query that retrieves them is deterministic, and the underlying table was populated in a consistent order during ingestion.

TimesFM inference is also reproducible: the model weights are fixed by the Hugging Face (Hugging Face, 2023) checkpoint and produce identical outputs for identical inputs,

so there is no stochastic element in the neural model's predictions either. Table 2 shows the source of randomness in the evaluation.

Table 2 - The sources of randomness in the evaluation.

Step	**Source of randomness**	**Control mechanism**
Device selection	Random shuffle	Fixed seed (42)
Mask sampling	Random sample	Fixed seed + deterministic input order
Metric computation	None	Deterministic arithmetic
TimesFM inference	Pretrained weights	Fixed Hugging Face checkpoint

6.5. Test Device Selection and Stratification

A naive random sample of test devices from the 471-sensor fleet would not yield a fair evaluation of the four imputation methods. The fleet is heterogeneous: most sensors are deployed in dense commercial areas with strong spatial connectivity and long, stable operational histories. Random sampling would therefore overrepresent these “easy” conditions, leading to inflated performance estimates and limited insight into failure modes.

Each imputation method is sensitive to different structural difficulties:

- **Historical median** fails under cold-start conditions, where insufficient same-hour history is available to populate the lookback window reliably.
- **Spatial KNN** fails under spatial isolation, where no nearby sensors are available at a given timestamp or neighbouring sensors are simultaneously missing.
- **TimesFM** requires a minimum temporal context (≥ 72 hours of prior observations) and is therefore sensitive to short operational histories.

- **Combined method** inherits both failure modes but benefits from partial redundancy between spatial and temporal signals.

Without stratification, the evaluation would be dominated by well-connected, long-history devices, where all methods perform comparatively well. This would obscure differences precisely in the regimes where method selection matters most. Stratification ensures that all major difficulty modes are explicitly represented in the test set.

6.5.1. Isolation Metric

To stratify devices by spatial difficulty for KNN, a quantitative measure of isolation is required. The goal is to capture the degree to which KNN is unable to recover missing values due to lack of spatial support.

A direct definition such as:

$$unfilled = missing_{hours} - newly\_filled_{hours}$$

is unsuitable because it conflates two distinct properties: overall missingness and spatial connectivity. Devices with high missingness would dominate this metric regardless of whether KNN performs well or poorly, biasing the ranking toward data quality rather than spatial structure.

Adopted formulation: relative isolation.

The isolation score is defined as the fraction of missing hours that KNN cannot recover:

$$newly\_filled = max(0, filled_{hours} - with\_data_{hours})$$

$$isolation = 1.0 - (newly\_filled_{hours} / missing_{hours})$$

Here, **knn_hours_filled** denotes the total number of rows produced by the KNN imputation pipeline, including both original observations and filled values. Subtracting **hours_with_data** isolates the number of genuinely imputed slots attributable to KNN. Normalising by **missing_hours** yields a bounded score in [0, 1]. Table 3 shows isolation score meaning.

Table 3 - Isolation score meaning.

Score	Meaning
0.0	KNN fills all missing slots → fully spatially connected
0.5	KNN fills half of missing slots → partially connected
1.0	KNN fills none → fully spatially isolated

For devices with **missing_hours = 0**, the score is defined as 0 by convention, since isolation is not meaningful in the absence of missingness.

6.5.2. Three Evaluation Groups

The 35 test devices are partitioned into three disjoint groups, each targeting a distinct failure mode.

Group A (Connected, 15 devices). Eligible devices with **total_hours > 2,000**, sorted in ascending order of ***knn_isolation*.** Devices are sampled randomly (seed 42). They have long operational history with strong spatial connectivity. KNN has abundant neighbouring observations, and historical methods have well-populated lookback windows. This establishes baseline performance under favourable conditions, approximating an upper bound for each method.

Group B (Isolated, 10 devices). Eligible devices with **total_hours > 2,000** and **missing_hours ≥ 10**, sorted in descending order of **knn_isolation**, excluding Group A devices. Random sampling is performed with seed 42. They have long history but weak spatial connectivity. Neighbouring devices are frequently missing at the same timestamps or lie beyond effective radius thresholds. Thi ssolates the effect of spatial sparsity, providing a targeted stress test for KNN. Also evaluates whether temporal methods (historical median, TimesFM) remain stable when spatial information is unavailable.

Group C (Short History, 10 devices). Eligible devices with **total_hours ≤ 2,000**, excluding all devices in Groups A and B. Random sampling is performed with seed 42. They have short operational windows (typically < 3 months). Early lifecycle slots

frequently lack sufficient same-hour historical observations. TimesFM also operates under reduced context for early timestamps. KNN remains largely unaffected, as it depends only on contemporaneous spatial neighbours. This stress tests for temporal methods under limited history conditions and evaluates robustness of historical median and TimesFM under cold-start regimes.

6.5.3. Selection Procedure

The selection pipeline consists of five steps.

Step 1 — Eligibility filtering

```
SELECT
    id,
    name,
    total_hours,
    hours_with_data,
    missing_hours,
    knn_hours_filled,
    CAST(missing_hours AS REAL) / total_hours AS missing_rate
FROM devices
WHERE total_hours IS NOT NULL
    AND hours_with_data >= 300
    AND CAST(missing_hours AS REAL) / total_hours < 0.05
ORDER BY id;
```

Figure 18 - SQL eligibility filtering script.

This produces the eligible device pool, provided in source code in Figure 18.

Step 2 — Isolation score computation

For each eligible device, knn_isolation is computed in Python. Devices with missing_hours are assigned 0.

Step 3 — Pool construction

- Long-history pool: total hours > 2,000
- Short-history pool: total_hours ≤ 2,000
- Connected pool: long-history sorted by ascending isolation
- Isolated pool: long-history with missing_hours ≥ 10, sorted by descending isolation

Step 4 — Group sampling (deterministic)

```
random.seed(42)
chosen_A = random.sample(group_connected, 15)
pool_B = [
    d for d in group_isolated
    if d["id"] not in selected_ids
]
chosen_B = random.sample(pool_B, 10)
pool_C = [
    d for d in group_short
    if d["id"] not in selected_ids
]
chosen_C = random.sample(pool_C, 10)
```

Figure 19 - Group sampling.

Groups are mutually exclusive by construction and by explicit exclusion logic as given in Figure 19 python source code.

Step 5 — Persistence

Selected devices are marked in the database:

- is_test = 1
- test_group ∈ {A-Connected, B-Isolated, C-ShortHistory}

These flags are used consistently in evaluation scripts and frontend visualisation.

6.5.4. Final Composition

The resulting test set contains 35 devices as shown in Table 4.

Table 4 - Test device groups.

Group	Devices	Description	Total hours	Missing rate	KNN isolation
A — Connected	15	Long history, spatially well-connected	> 2,000	< 5%	Low
B — Isolated	10	Long history, spatially sparse	> 2,000	< 5%	High
C — Short History	10	Short operational window	≤ 2,000	< 5%	Mixed

All selected devices satisfy the global eligibility constraints: missing rate below 5% and at least 300 observed hours. With a 20% evaluation mask, this ensures a minimum of 60 evaluation points per device, with most devices contributing several hundred points.

6.6. Evaluation Methodology

Measuring imputation accuracy requires a held-out ground truth that is not accessible to any method during prediction. This section describes the construction of that ground truth via a masking procedure, explains how each method structurally prevents leakage of masked values, defines the evaluation metrics and their rationale, and specifies the evaluation scope.

6.6.1. Masking Procedure

The masking procedure simulates missingness by withholding a random subset of observed measurements for each test device. Each method is then tasked with predicting these withheld values, which are compared against the original measurements.

Mask fraction. For each device, 20% of its observed readings are sampled uniformly at random without replacement:

$$n_{mask} = \max(1, round(hours_{data} \cdot \text{MASK\_FRACTION})$$

$$chosen = random.sample(original, n_{mask})$$

The 20% rate balances two competing constraints. Lower rates (e.g. 5%) yield insufficient evaluation points per device, particularly for Group C. Higher rates (e.g. 50%) significantly degrade input context, causing historical and temporal models to operate under an unrealistic data regime. At 20%, each device retains sufficient context (e.g. 400 readings for a 500-hour device) while providing enough held-out points for stable statistical estimates.

Masked values are not removed from in-memory structures (sp_levels, indices, or lookup tables). This design preserves full context availability while ensuring that prediction logic excludes the target value internally. Physically removing masked entries would unnecessarily degrade contextual information for other slots.

6.6.2. Natural Exclusion of Mask Slots

For the evaluation to be valid, none of the four methods must be allowed to see the true value of the slot it is trying to predict. If a method could access the answer it is supposed to estimate, its accuracy score would be artificially inflated and the comparison would be meaningless. One way to prevent this is to explicitly delete each masked slot from the data before calling the method. However, a simpler and more reliable approach is possible here: each method is designed in a way that makes it structurally impossible to access the target slot's own value, without any special removal step.

Historical Median builds its estimate from readings that occurred on previous days at the same hour. Only observations with a timestamp strictly earlier than the slot being predicted are considered. Since the target slot's timestamp is never strictly less than itself, the slot is automatically excluded from its own lookback window — regardless of whether it is physically present in the data or not.

Spatial KNN builds its estimate from readings taken by other sensors at the same timestamp. The list of neighbours used for any device is precomputed to contain only

other device IDs, never the device itself. Since the target slot belongs to the device being predicted, and that device is never in its own neighbour list, the target value is structurally unreachable by the KNN estimator.

The Combined method calls the Historical and KNN estimators internally and blends their outputs. Because both components already exclude the target slot by their own design, the combined method inherits this property automatically without needing any additional logic.

TimesFM builds its forecast from a window of recent values for the same device, but only values with a timestamp strictly earlier than the slot being predicted are included in that window. Since each hourly slot has a unique timestamp, this strict condition precisely excludes the target slot and nothing else.

The practical benefit of this approach is reliability. An explicit masking step — manually removing a slot before prediction — can be accidentally bypassed by a bug that forgets to apply the filter in some code path. Structural exclusion cannot be bypassed in this way: the logic that excludes the target slot is the same logic that defines how the method works, so it applies automatically every time. Table 5 shows the summary of masking restrictions.

Table 5 - Masking restrictions.

Method	**How the target slot is excluded**
Historical	Only reads timestamps strictly before the target — never the target itself
KNN	Only reads from other devices — never the device being predicted
Combined	Inherits exclusion from both Historical and KNN
TimesFM	Only reads timestamps strictly before the target — never the target itself

6.6.3. Metric Definitions

Two standard regression metrics are used: Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE). Both are computed in decibels over valid prediction pairs.

Mean Absolute Error (MAE)

$$MAE = (1 / n) \times \Sigma |y_i - \hat{y}_i|$$

MAE measures the average absolute deviation between predicted and observed values. It is directly interpretable in physical units: an MAE of 3 dB corresponds to an average prediction error of 3 dB.

MAE is robust to outliers, as errors contribute linearly to the final score.

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{((1 / n) \times \Sigma (y_i - \hat{y}_i)^2)}$$

RMSE penalises large errors disproportionately due to squaring. A small number of large deviations can therefore dominate the final score.

By construction:

- RMSE ≥ MAE
- Small gap → uniform error distribution
- Large gap → presence of rare but large failures

Complementarity of Metrics

MAE captures typical-case performance, while RMSE captures tail risk.

A model with uniform 3 dB errors and a model with mostly 1–2 dB errors but occasional 15 dB failures may achieve similar MAE, but RMSE will clearly distinguish them. Conversely, relying only on RMSE would over-emphasise rare anomalies.

6.6.4. Evaluation Scope

With MASK_FRACTION = 0.20 and random.seed(42), the 35 test devices yield 17,317 held-out evaluation slots which is shown in Table 6.

Table 6 - Masked values for each test device group.

Group	**Devices**	**Approx. slots**	**Characteristics**
A — Connected	15	~9,500	high connectivity, long history
B — Isolated	10	~5,000	sparse spatial connectivity
C — Short History	10	~2,800	limited temporal history
Total	**35**	**17,317**	

These results are reported at three levels:

1. Global performance across all 17,317 masked slots (primary ranking metric)
2. Group-level performance (A, B, C) to analyse sensitivity to difficulty regimes
3. Per-device results, exported to evaluation_results.csv

The frontend additionally exposes per-device breakdowns to enable inspection of method performance heterogeneity across sensor conditions.

However, the evaluation is restricted to the September–December 2021 window and therefore does not capture seasonal variation outside this period. Additionally, the 35-device stratified sample may not fully represent extreme deployment environments (e.g. tunnels, industrial zones, parks). These factors may influence generalisation beyond the evaluated domain.

6.7. Comparative Analysis of Imputation Methods

This section presents the quantitative results of the masked evaluation across the four imputation methods and the three device groups, followed by a structured analysis of performance differences, failure modes, and implications for practical deployment.

6.7.1. Overall Results

Several observations can be made from the overall results shown in Table 7 and Table 8. With an MAE of 1.18 dB, TimesFM reduces error by approximately 2.3x compared to the Combined method and 2.8x compared to the Historical baseline. Its RMSE (1.80 dB) is also substantially lower than that of the statistical approaches, indicating a consistently lower magnitude of error across evaluation slots. TimesFM produces predictions for 16,839 out of 17,317 mask slots (97.2%), excluding 478 cases where sufficient historical context (at least 72 hours) is not available. The RMSE/MAE ratio is relatively consistent (approximately 1.43–1.52), suggesting comparable relative dispersion of errors across methods, despite differences in absolute error magnitude. While KNN slightly underperforms the historical method overall, this relationship changes when results are disaggregated by device group.

Table 7 - Overall MAE and RMSE results.

Method	**N slots**	**MAE (dB)**	**RMSE (dB)**	**RMSE / MAE**
Historical Median	17,170	3.32	4.79	1.44
Spatial KNN	17,317	3.54	5.07	1.43
Combined (Hist + KNN)	17,317	2.70	4.10	1.52
TimesFM	**16,839**	**1.18**	**1.80**	**1.52**

Table 8 - MAE and RMSE by evaluation group.

Group	Method	N	MAE (dB)	RMSE (dB)
A — Connected	Historical	8,502	3.39	4.97
	KNN	8,563	4.12	5.78
	Combined	8,563	2.98	4.54
	TimesFM	8,358	1.20	1.85
B — Isolated	Historical	5,667	3.43	4.70
	KNN	5,708	3.20	4.53
	Combined	5,708	2.53	3.67
	TimesFM	5,582	1.18	1.74
C — Short History	Historical	3,001	2.92	4.42
	KNN	3,046	2.56	3.69
	Combined	3,046	2.23	3.53
	TimesFM	2,899	1.14	1.75

6.7.2. Method-Level Analysis

The relative performance of the Historical and KNN methods varies across device groups. In **Group A (Connected devices)**, KNN performs worse than the Historical method (MAE 4.12 vs 3.39 dB). This may be explained by correlated missingness patterns: in densely connected areas, missing values may occur simultaneously across neighbouring devices (e.g. due to shared infrastructure or network outages). In such cases, spatial neighbours may also be unavailable at the same timestamp, limiting the effectiveness of

KNN-based imputation. In contrast, the Historical method relies on past observations from the same device, which are not affected by concurrent missingness.

In **Group B (Isolated devices)**, KNN performs slightly better than the Historical method (3.20 vs 3.43 dB). Although isolated devices have fewer spatial connections for general imputation tasks, the evaluation mask is applied to timestamps where the target device has observed data. At these times, neighbouring devices are more likely to also have valid readings, allowing KNN to operate effectively. Historical performance remains relatively stable across Groups A and B, suggesting that it is less sensitive to spatial structure.

In **Group C (Short history devices)**, both methods achieve their lowest errors. This may reflect differences in the underlying temporal variability of these devices or differences in operational conditions during their active periods.

The Combined method consistently outperforms both constituent methods across all groups. In Group A, it reduces MAE from 4.12 (KNN) and 3.39 (Historical) to 2.98 dB. Similar improvements are observed in Groups B and C.

This behaviour suggests that combining spatial and temporal estimators provides complementary information. The inverse-variance weighting approach allows the model to place higher weight on the estimator with lower uncertainty for a given slot, rather than relying on a fixed combination.

The performance gains are most pronounced in Group A, where the individual methods exhibit larger discrepancies, and smallest in Group C, where both methods already perform relatively well.

TimesFM shows consistently low error across all groups, with MAE values ranging from 1.14 to 1.20 dB. The variation across groups is minimal compared to the statistical methods, suggesting relatively stable performance across different device conditions.

The difference in MAE between TimesFM and the Combined method (approximately 1.52 dB overall) indicates a substantial reduction in average error. In practical terms, this level of difference may be relevant for applications where small changes in decibel levels affect downstream classification or interpretation.

6.7.3. Error Distribution and Tail Behaviour

In addition to mean error metrics, the RMSE/MAE ratio provides a coarse indication of error dispersion across methods. Values between approximately 1.43 and 1.52 are observed for all methods, suggesting broadly similar relative variability in errors.

While this ratio does not directly characterise tail behaviour, it is consistent with error distributions that include occasional larger deviations from the mean.

In absolute terms, however, the methods differ substantially. TimesFM exhibits much lower RMSE values (1.80 dB) compared to the statistical methods (4.10–5.07 dB), indicating a lower magnitude of large errors.

The higher absolute error levels in the statistical methods imply that, although most predictions may be relatively accurate, occasional larger deviations can occur. These deviations are more likely to be operationally relevant when imputed values are used for threshold-based classification.

6.7.4. Spatial Analysis

The grouped results suggest differences in how each method responds to spatial structure.

KNN performance varies with device connectivity, particularly in Group A, where correlated missingness across neighbouring devices may reduce the availability of usable spatial information.

In contrast, the Historical method shows relatively stable performance across groups, indicating limited sensitivity to spatial configuration and stronger dependence on temporal stability.

TimesFM shows minimal variation across all groups (approximately 0.06 dB MAE range), suggesting that its performance is relatively robust to differences in spatial connectivity and device history length within the evaluated dataset.

From a deployment perspective, this suggests that statistical methods may introduce location-dependent variation in performance, whereas TimesFM provides more uniform behaviour across heterogeneous sensor conditions.

7 Discussion

The comparative evaluation conducted on 17,317 masked observations across 35 stratified test devices yields three key findings that are both individually clear and collectively consistent. This section interprets these findings in detail, progressing from quantitative results to the underlying structural explanations that account for the observed performance differences.

7.1. Why TimesFM Dominates

TimesFM achieves an overall mean absolute error (MAE) of 1.18 dB, outperforming the next-best method (Combined, 2.70 dB) by 1.52 dB and the weakest-performing statistical method (KNN, 3.54 dB) by 2.36 dB. This improvement is not incremental but substantial, as it reduces the average error from above the perceptual threshold of approximately 3 dB — where differences become clearly noticeable — to below the just-noticeable difference of around 1 dB. Consequently, the improvement represents a qualitative enhancement in imputation accuracy rather than a minor refinement.

This performance gap can be explained by the representational capabilities of the respective methods. The statistical approaches — namely the historical median and spatial KNN — are based on predefined assumptions about the structure of urban noise. Specifically, they assume that sound levels follow consistent hourly patterns across days and that measurements are correlated with those of nearby sensors at the same time. While these assumptions hold on average, they break down in precisely the scenarios where imputation is most critical. For example, during network outages, missing data often coincide with atypical events such as public gatherings, road disruptions, or extreme weather conditions. In such cases, historical averages fail to capture the anomaly, and KNN becomes ineffective when neighbouring sensors are simultaneously unavailable.

In contrast, TimesFM does not rely on such explicit assumptions. As a decoder-only transformer pre-trained on a large and diverse collection of real-world time series, it learns general representations of temporal dynamics, including trends, seasonality, structural shifts, and short-term fluctuations. These patterns are not domain-specific and can be transferred across different application areas. Urban SPL data exhibits temporal characteristics — such as daily cycles, weekly periodicity, and occasional abrupt changes

— that are structurally similar to those found in domains like energy consumption, retail demand, or meteorological data. This allows the model to generalise effectively to the acoustic domain without task-specific training.

Importantly, the superiority of TimesFM is consistent across all evaluation groups. Its MAE varies by only 0.06 dB (from 1.14 to 1.20 dB) across Groups A, B, and C. In contrast, the statistical methods show considerably larger variation, with differences of 0.47 dB for the historical method and 1.56 dB for KNN across the same groups. This stability across varying structural conditions — such as spatial isolation and limited historical data — provides strong evidence that the model's performance is not dependent on specific dataset characteristics but reflects a more general robustness.

It is also important to note that these results are obtained in a zero-shot setting. TimesFM was not fine-tuned on the Tallinn SPL dataset; its performance is entirely based on generalisation from its pre-training data, which does not include acoustic sensor measurements. This suggests that the reported MAE of 1.18 dB represents a conservative estimate of the model's potential performance. Further improvements could likely be achieved through domain-specific fine-tuning, particularly for sensors located in atypical acoustic environments, such as tunnels or parks, where temporal patterns may differ from those commonly observed in the pre-training corpus.

7.1. Why KNN Underperforms in Group A

One of the most unexpected findings in the evaluation is that the spatial KNN method performs worse in Group A (Connected) than in Group B (Isolated). Specifically, KNN records a MAE of 4.12 dB in the connected group compared to 3.20 dB in the isolated group. This result is counterintuitive, as Group A devices were selected to represent favourable conditions for KNN, with dense spatial neighbourhoods and long operational histories, whereas Group B was intended to challenge the method.

This apparent contradiction can be explained by distinguishing between *spatial isolation* and *correlated missingness*. The isolation metric used to define the groups captures how often a device lacks available neighbours during its missing periods. Group A devices have low isolation scores, indicating that, in general, neighbouring sensors are available when data is missing. However, the evaluation procedure masks values from timestamps

where the device originally had valid readings. As a result, the evaluation focuses on moments when the network is functioning normally, rather than when real missingness occurs.

In dense urban deployments such as Group A, sensors are often connected through shared infrastructure, including common gateways or network segments. When a failure occurs — such as a gateway outage, firmware update, or system-level disruption — it typically affects multiple nearby devices simultaneously. Consequently, at the exact timestamps when one device loses data, its neighbours are also likely to be unavailable. This leads to a situation where KNN cannot retrieve sufficient neighbouring values, despite the overall spatial density of the network.

In contrast, the historical method is not affected by this issue. Since it relies on past observations from the same device, drawn from different time periods, it remains robust to simultaneous outages. The observed 0.73 dB performance advantage of the historical method over KNN in Group A supports the conclusion that correlated missingness, rather than spatial sparsity, is the primary factor limiting KNN performance in this context.

This finding has broader implications for sensor network design. While spatial redundancy is often assumed to improve data recovery, this assumption holds only when failures are independent. In real-world IoT systems, where devices frequently share infrastructure, failures tend to be correlated. Under such conditions, spatial interpolation methods such as KNN inherit this dependency and may degrade significantly. The results from Group A provide empirical evidence of this limitation in a real urban sensing environment.

7.2. Why the Combined Method Outperforms Its Components

The combined imputation method consistently achieves lower error than both the historical median and spatial KNN across all evaluation groups. For example, in Group A it reduces MAE to 2.98 dB, outperforming the historical method (3.39 dB) and KNN (4.12 dB). Similar improvements are observed in Group B (2.53 dB vs. 3.20/3.43 dB) and Group C (2.23 dB vs. 2.56/2.92 dB).

This improvement is explained by the use of inverse-variance weighting, which enables adaptive combination of the two methods. Rather than applying fixed weights, the

combined approach evaluates the internal consistency of each method's estimates for every individual slot. The method with lower variance — indicating more stable and reliable input data — is assigned greater weight in the final prediction.

As a result, the combined method dynamically adjusts to local conditions. When historical readings exhibit low variability, the historical estimate dominates. Conversely, when neighbouring sensors provide consistent values, the KNN estimate is prioritised. This slot-level adaptivity allows the method to leverage the strengths of both approaches while mitigating their individual weaknesses.

Importantly, this adaptive mechanism operates not only across devices but also within the time series of a single device. The reliability of historical and spatial information can vary significantly depending on time of day, environmental conditions, or network behaviour. A fixed selection of one method cannot account for this variability, whereas the combined approach continuously adjusts its weighting.

The magnitude of improvement varies across groups. The gain is largest in Group A (0.84 dB compared to the best individual method), where KNN performance is degraded by correlated outages and the combined method effectively shifts weight toward the historical component. In Group C, where both methods perform relatively well and produce similar estimates, the benefit of combining them is smaller (0.33 dB). In such cases, the two methods provide redundant information, limiting the potential for improvement.

Overall, the results demonstrate that adaptive combination is more effective than selecting a single method, particularly in heterogeneous environments where data reliability varies across both space and time.

7.3. Visual Dashboard as an Analytical Tool

The interactive dashboard serves not only as a presentation layer for the imputation results but as an analytical instrument in its own right, enabling observations about sensor health, urban noise patterns, and spatial inequality that would not be accessible from raw data tables alone.

7.3.1. Sensor Health and Faulty Device Detection

The Devices page provides an immediate fleet-wide overview of data completeness. Each sensor is colour-coded by its missing data rate: green for sensors with less than 20% missing hours, orange for those with 20–50% missing, and red for those exceeding 50%. This visual encoding makes faulty or unreliable devices immediately identifiable without inspecting individual records. Sensors that appear red are either experiencing persistent connectivity failures, were deployed for only a short portion of the study period, or suffered hardware issues that rendered them largely inactive. The completeness map allows a network operator to identify which sensors require maintenance or replacement at a glance, and to assess the geographic distribution of unreliable nodes across the city.

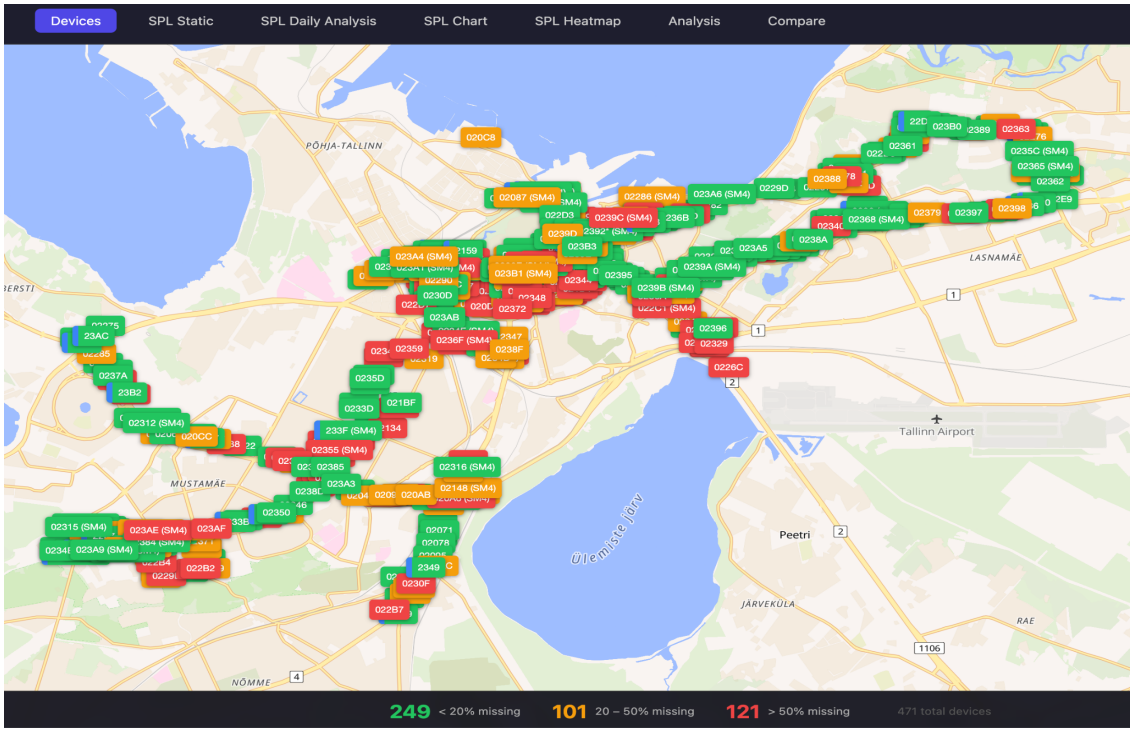


Figure 20 - Devices grouped by missing data.

7.3.2. Spatial and Temporal Noise Patterns

The animated SPL playback and heatmap reveal clear spatial and temporal patterns in Tallinn's urban acoustic environment. Daytime noise levels are consistently higher than nighttime levels across the fleet, reflecting the expected relationship between human activity and acoustic exposure. However, the spatial distribution of noise is far from uniform. Several locations stand out as persistently loud: Pallasti and Kalaranna register

elevated SPL levels across most hours, reflecting their proximity to high-traffic corridors and the waterfront area respectively. Linnamäe tee similarly shows noticeably higher SPL than the surrounding sensors, consistent with its role as a major arterial road. In contrast, Üliõpilaste tee registers among the lowest SPL readings in the network, likely due to its predominantly residential and academic surroundings with limited through-traffic.

At a broader geographic scale, the device ranking map on the Analysis page reveals an east–west asymmetry in noise exposure: western Tallinn tends to be quieter than the eastern parts of the city, which are more exposed to transit corridors and industrial activity. The 15 loudest and 15 quietest devices cluster in spatially distinct zones, suggesting that noise inequality in Tallinn has a clear geographic structure rather than being randomly distributed across the network.

Figure 21 - Loudest and quietest devices.

7.3.3. Weekly and Hourly Profiles

The Analysis page's hourly and day-of-week charts surface the temporal rhythm of urban noise in detail. On weekdays, SPL peaks between 08:00 and 18:00, driven by morning commute traffic, commercial activity, and the afternoon peak hour. The pattern is broad and sustained across the working day. On weekends, the daytime peak narrows considerably, appearing around 14:00 rather than spanning the full working day — reflecting later waking patterns and reduced commuter traffic. Weekends also show a slightly elevated SPL between 00:00 and 04:00 compared to weekdays, consistent with nighttime social activity concentrated on Friday and Saturday nights. Overall, weekend daytime SPL is lower than the corresponding weekday levels, confirming that traffic and

commercial activity are the dominant noise sources in the monitored areas rather than residential or leisure-driven noise.

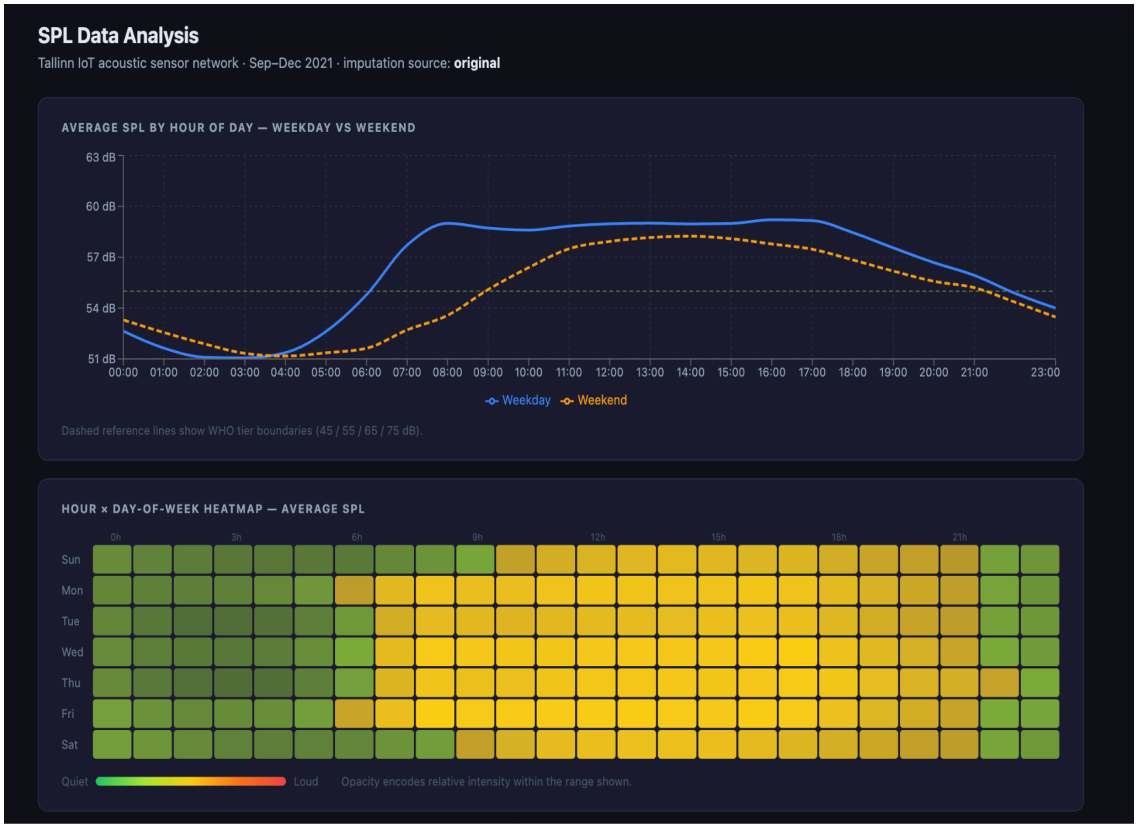


Figure 22 - SPL WHO tiers during the day of week.

7.4. Practical Implications for Urban Noise Monitoring

7.4.1. WHO Tier Classification Accuracy

The World Health Organization environmental noise guidelines define health-based exposure thresholds and dose–response relationships rather than discrete categorical tiers, with increasing SPL associated with progressively higher health risks (World Health Organization, 2018). In applied urban noise mapping and decision-support systems, these continuous values are often discretised into a small number of bands (commonly five classes) for interpretability in spatial visualisation and policy workflows (European Environment Agency, 2020). Consequently, errors in imputed SPL that lead to misclassification between these bands may have direct implications for urban planning decisions, regulatory compliance, and public health assessment.

The statistical methods evaluated in this study produce average errors in the range of 2.70–3.54 dB. This magnitude is comparable to the distance between many real-world observations and the nearest tier boundary. For instance, a reading of 53 dB lies only 2 dB below the 55 dB threshold. Under an imputation method with a mean absolute error of approximately 2.70 dB, such a value has a substantial probability of being misclassified into an adjacent tier. The risk is even higher for KNN, which exhibits a larger average error. Furthermore, the tail behaviour of these methods indicates that occasional large deviations (in the range of 7–9 dB) are possible, potentially shifting values across multiple tier boundaries.

In contrast, TimesFM achieves a substantially lower MAE of 1.18 dB. This reduction significantly decreases the likelihood of crossing tier thresholds during imputation. Even at higher error percentiles, the expected deviation (approximately 2–3 dB) remains well below the 10 dB width of the WHO tiers. As a result, TimesFM provides a level of accuracy that is more suitable for policy-oriented applications, where maintaining correct tier classification is critical. The statistical methods, while computationally efficient, introduce a level of uncertainty that may distort aggregated assessments of environmental noise exposure.

7.4.2. Equity Across the Sensor Fleet

Beyond accuracy, urban noise monitoring systems must also ensure equitable data quality across all monitored areas. If imputation performance varies systematically by location, the resulting noise maps may be more reliable in some regions than others, potentially introducing bias into decision-making processes.

The evaluation results indicate that the statistical methods are sensitive to structural characteristics of the sensor network. Historical imputation tends to perform worse for recently deployed sensors, where limited past data restricts the reliability of temporal patterns. Spatial KNN, on the other hand, is affected by both low sensor density (leading to insufficient neighbours) and correlated outages in dense deployments, where multiple nearby devices may fail simultaneously. These factors are unevenly distributed across the city, meaning that certain areas—such as peripheral zones or newly instrumented regions—are more likely to receive lower-quality imputations.

TimesFM, by contrast, demonstrates consistently stable performance across all evaluation groups, with MAE values ranging narrowly between 1.14 and 1.20 dB. This indicates that its accuracy is largely independent of spatial density and historical data availability. Such uniformity is particularly important for ensuring that noise assessments are comparable across different urban contexts. In practical terms, this means that sensors located in less-developed or recently monitored areas can provide data of comparable reliability to those in well-established, high-density zones.

From a policy perspective, this consistency supports more balanced and representative urban noise mapping. It reduces the risk that certain communities are systematically underrepresented due to lower data quality, thereby strengthening the fairness and credibility of noise-related analyses and decisions.

7.4.3. Deployment Cost Trade-Off

While TimesFM offers clear advantages in accuracy, it also introduces higher computational requirements. Processing approximately 310,000 imputed slots requires around 13 hours on standard CPU hardware, whereas the statistical methods complete the same task within minutes. This difference necessitates consideration of the trade-off between computational cost and imputation quality.

The practical impact of this trade-off depends on the deployment context. In a research setting, where imputation is performed once on a fixed historical dataset, the computational cost is incurred only during initial processing. Once the results are stored, subsequent analysis and visualisation involve no additional inference overhead. In this scenario, the longer runtime is acceptable given the substantial improvement in accuracy.

In contrast, operational monitoring systems often require continuous or near-real-time data updates. In such environments, a 13-hour processing window may be impractical. Several implementation strategies can address this limitation. One approach is to use statistical methods for immediate gap filling and apply TimesFM retrospectively in scheduled batch processes, replacing earlier estimates with more accurate predictions. Alternatively, deploying the model on GPU hardware can significantly reduce inference time, enabling faster turnaround. A third option is to reserve TimesFM for periodic offline analyses, such as regulatory reporting, while relying on statistical methods for day-to-day monitoring.

The architecture developed in this project already supports a hybrid approach, where statistical imputation provides initial coverage and TimesFM operates as a refinement layer. This design allows flexibility in balancing computational efficiency and accuracy, depending on the operational requirements of the monitoring system.

7.5. Relation to Existing Literature

7.5.1. Correlated Missingness in Sensor Network Literature

The limitations of spatial imputation observed in this study are consistent with broader findings in the sensor network literature. In domains such as traffic monitoring, failures of loop detectors are often caused by shared infrastructure issues—for example, communication outages or controller failures—that simultaneously affect multiple sensors within a region. Similar patterns are reported by Murphy and King, who show that low-cost IoT deployments frequently exhibit correlated failure modes due to network-layer disruptions, packet loss in communication protocols such as LoRaWAN or NB-IoT, and power instability.

These observations align with the behaviour identified in this evaluation. The reduced performance of KNN in dense urban areas (Group A) can be attributed to the same underlying phenomenon: sensors that are spatially close are also likely to share infrastructure, and therefore tend to fail simultaneously. As a result, the very neighbours that spatial methods depend on may be unavailable at critical moments.

This leads to a broader methodological implication. Any imputation technique that relies primarily on spatial neighbours implicitly assumes that neighbouring observations are available and independent. When this assumption is violated, as in many real-world IoT deployments, the method inherits the failure structure of the network. In contrast, temporal approaches—such as historical imputation and TimesFM—draw on past observations and are therefore unaffected by concurrent outages. This makes them inherently more robust in environments with correlated missingness.

7.5.2. Neural vs Statistical Methods for Irregular Gaps

The strong performance of TimesFM relative to statistical baselines is consistent with recent developments in time series modelling, particularly the emergence of foundation models based on transformer architectures. These models are designed to learn general

representations of temporal dynamics—such as trends, seasonality, and abrupt level changes—from large and diverse datasets, enabling them to generalise across domains without task-specific training.

Similar patterns have been reported in areas such as weather forecasting, energy demand prediction, and financial time series reconstruction, where pre-trained models demonstrate strong zero-shot or few-shot performance. The results of this study extend this evidence to the domain of urban acoustic monitoring.

The advantage of neural methods becomes especially pronounced under challenging conditions. In Group B (isolated devices), where spatial information is limited, TimesFM achieves an MAE of 1.18 dB compared to 3.20 dB for KNN and 3.43 dB for the historical method. This represents a substantial improvement in accuracy under precisely the conditions where statistical approaches are expected to struggle.

Such behaviour is consistent with findings in related domains, particularly air quality monitoring in sparse networks, where model-based approaches tend to outperform statistical methods most significantly when sensor density is low. The common explanation is that neural models rely primarily on learned temporal structure rather than local spatial correlations, allowing them to maintain performance even when neighbouring data is unavailable.

Overall, the results reinforce the view that **ne**ural time series models offer a robust alternative to traditional statistical imputation, particularly in scenarios characterised by irregular and correlated missing data.

7.6. Limitations

7.6.1. Temporal Scope

The evaluation is based on a single four-month period, spanning September to December 2021. This interval includes several relevant seasonal transitions, such as the change from summer time to winter time (EEST to EET), decreasing temperatures, and evolving traffic patterns during autumn, as well as the lead-up to the winter holiday season. However, it does not include the summer months (June–August), which are typically associated with the highest levels of outdoor noise exposure in Tallinn and are therefore particularly important for policy-related assessments.

Seasonal differences are likely to influence the behaviour of imputation methods. Summer conditions in northern European cities involve increased pedestrian activity, outdoor events, tourism, and extended daylight hours, all of which introduce greater temporal variability in acoustic patterns (Murphy & King, 2016; Picaut et al., 2020). Methods such as the historical median, which rely on consistency across days at the same hour, may perform worse under these conditions due to reduced regularity. In contrast, a model such as TimesFM, which captures broader temporal dynamics, may generalise more effectively if similar high-variability patterns are present in its pre-training data.

As a result, it remains uncertain whether the relative performance rankings observed in the September–December period would hold under summer conditions. Extending the evaluation to a full annual cycle would provide a more comprehensive understanding of method robustness across seasonal regimes.

7.6.2. Zero-Shot vs Fine-Tuned TimesFM

The TimesFM model is evaluated in a zero-shot setting, meaning that it is applied directly to the Tallinn SPL dataset without any domain-specific fine-tuning. This has two important implications.

First, the reported MAE of 1.18 dB should be interpreted as a conservative estimate of the model's potential performance. Fine-tuning the model on a subset of the local dataset would allow it to adapt to the specific characteristics of the sensor network, including the behaviour of MEMS microphones, the structure of Tallinn's traffic system, and location-specific acoustic patterns. Such adaptation would likely reduce the error further.

Second, the zero-shot setting is practically significant, as it reflects the performance that can be achieved in a new deployment without requiring labelled training data or additional computational resources for model training. This is particularly relevant for cities or organisations that wish to deploy similar systems without an existing historical dataset.

The difference between zero-shot and fine-tuned performance is expected to be most pronounced in acoustically atypical environments, such as tunnels, parks, industrial zones, or areas affected by construction activity. These environments may exhibit temporal patterns that deviate from those commonly represented in the model's pre-training data, and therefore benefit most from domain-specific adaptation.

7.6.3. Mask Representativeness

The evaluation methodology employs a uniform random masking strategy, in which 20% of each device's observed readings are randomly removed and used as ground truth for validation. This approach ensures reproducibility and avoids bias toward specific times of day or days of the week. However, it does not fully reflect the characteristics of real-world missing data.

In practice, missing data in IoT sensor networks tends to occur in clusters rather than as isolated points. Network failures, maintenance events, or power interruptions typically result in consecutive gaps spanning multiple hours. By contrast, the random masking approach is more likely to produce isolated missing points and to sample more frequently from typical operating conditions than from anomalous periods.

This difference may influence the evaluation results. Methods may perform differently when imputing continuous gaps compared to isolated missing values, and their behaviour during unusual or high-variance periods may be underrepresented in the current setup. Consequently, the reported performance metrics may not fully capture real-world operational conditions.

A more realistic evaluation could be conducted using actual missing segments from a separate dataset with known ground truth, for example through controlled redeployment or reference measurements. However, such an approach would require significantly greater resources and was therefore beyond the scope of the present study.

8 Conclusion

8.1. Summary of Contributions

This thesis addressed the challenge of incomplete and fragmented data in Tallinn's urban acoustic monitoring network. The starting point was the observation that raw SPL measurements exhibit a substantial level of missingness (approximately 26% across the fleet), which limits their direct use for reliable noise exposure analysis. The study argued that systematic data reconstruction is necessary and that the choice of imputation method has a measurable impact on the quality of the resulting dataset.

To address this problem, a reproducible imputation pipeline consisting of four methods was developed and implemented. These include a historical median approach based on temporal regularity, a spatial KNN method leveraging geographic proximity, a combined method using inverse-variance weighting, and the TimesFM 2.5 model applied in a zero-shot setting. Each method produces a complete imputed dataset covering all 471 sensors for the September–December 2021 period. The pipeline is deterministic, fully reproducible, and can be re-executed from the original raw data without modification.

The evaluation was conducted using a stratified masking approach across 35 test devices and 17,317 held-out observations. This design allowed for systematic comparison of method performance under different structural conditions rather than only under ideal scenarios. In addition, an interactive web-based dashboard was developed using React to support exploration and analysis of the results. The dashboard provides multiple perspectives on the data, including spatial visualisations, temporal animations, individual time series inspection, and comparative evaluation of imputation methods, all connected through a unified data source selection mechanism.

8.2. Answers to Research Questions

RQ1: The results demonstrate that the hybrid combined method consistently improves accuracy over both the historical median and spatial KNN approaches. With an overall MAE of 2.70 dB, it outperforms the historical method (3.32 dB) and KNN (3.54 dB) across all evaluation groups. This improvement is achieved through adaptive weighting, which dynamically prioritises the more reliable source for each imputed value.

The TimesFM 2.5 model further improves performance, achieving a MAE of 1.18 dB. This represents a substantial reduction in error compared to both the combined method and the individual statistical approaches. The findings therefore confirm that a layered imputation strategy, particularly one incorporating a foundation model, provides the most accurate reconstruction of missing SPL data.

RQ2: The evaluation results show a clear and consistent accuracy advantage for TimesFM. Overall, TimesFM achieves an MAE of 1.18 dB compared to the hybrid Combined method's 2.70 dB — a 2.3× improvement. This gap holds across all three evaluation groups: TimesFM's MAE varies by only 0.06 dB across connected, isolated, and short-history sensors (1.14–1.20 dB), while the hybrid varies by 0.75 dB. The hybrid does outperform its individual components in every group, confirming that inverse-variance weighting adds value, but it remains substantially less accurate than TimesFM under all tested conditions.

The two approaches differ significantly in computational cost. The hybrid statistical methods complete in minutes on standard hardware — Historical Median requires only an in-memory lookup of prior readings, and KNN relies on a one-time precomputed distance matrix. TimesFM, by contrast, runs a 200-million-parameter transformer over approximately 310,000 slots, taking around 13 hours on CPU or under 30 minutes on GPU. For a fixed historical dataset as in this study, this is a one-time cost paid during pipeline setup, after which all queries are instant reads from the pre-computed table.

The appropriate choice therefore depends on the deployment context. For one-time historical reconstruction, TimesFM's accuracy gain justifies the compute investment. For real-time monitoring where new data arrives continuously, the hybrid is the practical choice for immediate gap filling, with TimesFM optionally applied as a nightly batch to retroactively replace statistical estimates with higher-accuracy predictions — the architecture this pipeline is already designed to support.

RQ3: The dashboard demonstrates that interactive visualisation adds analytical value beyond what static tables or aggregate statistics can provide. Three distinct capabilities emerge from the implementation.

First, faulty and unreliable sensors are immediately identifiable through the Devices page, which colour-codes each of the sensors by their missing data rate. This encoding allows

a network operator to assess fleet-wide sensor health at a glance and identify devices that require maintenance or replacement, without inspecting individual records. Sensors that appear red are either persistently disconnected, deployed for only a short portion of the study period, or suffering hardware failures that render their data unusable for analysis.

Second, the animated playback and heatmap reveal spatial and temporal noise patterns that are invisible in static reports. Daytime SPL is consistently higher than nighttime levels across the fleet. Persistently loud zones — including Pallasti, Kalaranna, and Linnamäe tee — stand out clearly in the heatmap animation, while Üliõpilaste tee registers among the quietest locations in the network. The device ranking map shows that western Tallinn is generally quieter than the eastern parts of the city, indicating a geographic structure to noise inequality that would not be apparent from fleet-wide averages. The hourly and day-of-week profiles on the Analysis page further reveal that weekday noise peaks broadly between 08:00 and 18:00, while the weekend peak narrows to around 14:00, with slightly elevated late-night levels between 00:00 and 04:00 reflecting nighttime social activity.

Third, the dashboard makes data quality and imputation provenance transparent at the point of use. Each sensor's imputation source — original measurement, statistical estimate, or TimesFM prediction — is visible in the map popup, allowing the user to judge how much confidence to place in any displayed value.

8.3. Future Work

The reported MAE of 1.18 dB represents a zero-shot baseline, as the model was applied without exposure to Tallinn-specific SPL data during training. Fine-tuning the model on a held-out subset of the dataset would allow it to adapt to the particular characteristics of MEMS acoustic sensors, local traffic dynamics, and seasonal patterns. This is expected to further reduce prediction error and improve performance, especially for sensors located in acoustically atypical environments such as tunnels or parks. Even a relatively small fine-tuning dataset, consisting of a few thousand device-hours, would be sufficient to evaluate the magnitude of this improvement.

Evaluation on summer months. The current analysis is limited to data from September to December. However, summer months introduce substantially different acoustic

conditions, including increased outdoor activity, higher pedestrian density, and longer daylight hours. These factors may influence the temporal structure of noise patterns and, consequently, the performance of imputation methods. Extending the evaluation to a summer dataset would allow assessment of whether the observed method rankings remain consistent under different seasonal conditions.

Evaluation using real missing data. The evaluation framework relies on a synthetic masking approach, where observed values are randomly removed to create ground truth comparisons. While this ensures reproducibility and statistical balance, it does not fully reflect the clustered and correlated nature of real missing data. Future work could involve evaluating imputation performance on genuinely missing segments, using ground truth obtained from reference instruments or an independent data collection period. This would provide a more realistic assessment of method performance in operational settings.

Hybrid real-time deployment architecture. The current pipeline is designed for offline processing of historical data. In a production environment with continuous data ingestion, a hybrid architecture could be implemented. In such a system, statistical methods would provide immediate gap-filling for incoming data, while TimesFM would operate as a scheduled batch process (e.g., nightly) to refine these estimates retrospectively. The existing design — including separate imputation tables and the multi-valued imputation flag — already supports this extension with minimal changes.

The vision of a "smart city" depends on the availability of continuous and reliable environmental data to support informed decision-making. In the context of urban noise monitoring, sensor networks provide the necessary infrastructure, but incomplete data limits their practical utility. This thesis addresses this limitation by developing and evaluating a comprehensive imputation framework capable of reconstructing missing SPL measurements with high accuracy.

The results demonstrate that a foundation model approach can achieve sub-perceptual error levels across a diverse sensor fleet, while the accompanying visualisation dashboard enables intuitive exploration of the reconstructed data. Together, these contributions help bridge the gap between raw sensor infrastructure and actionable datasets, moving Tallinn's acoustic monitoring system closer to supporting evidence-based urban planning and policy decisions.

References

1. Cities Today (2025) *Tallinn unveils noise reduction plan to improve health and housing*. [Online] Available at: https://cities-today.com/tallinn-unveils-noise-reduction-plan/ (Accessed: 16 March 2026).

2. ERR (2025) *Excessive traffic noise holding back Tallinn's real estate development*. [Online] Available at: https://news.err.ee/ (Accessed: 16 March 2026).

3. Green Tallinn (2024) *Three new innovation projects selected for Test in Tallinn program*. [Online] Available at: https://www.greentallinn.ee/ (Accessed: 16 March 2026).

4. Siigur, V. (2025) *Sound Pressure Level Analysis and Visualization in Tallinn Urban Environment Based on Low-Cost IoT Sensor Data*. Master's Thesis. Tallinn University of Technology.

5. Smart Cities World (2025) *Tallinn implements noise reduction action plan*. [Online] Available at: https://www.smartcitiesworld.net/ (Accessed: 16 March 2026).

6. Tallinn City Government (2024) *Tallinn seeks solutions to reduce noise pollution*. [Online] Available at: https://www.tallinn.ee/en/news/tallinn-seeks-solutions-reduce-noise-pollution (Accessed: 16 March 2026).

7. Ahmed, A., Pereira, L. and Jane, K. (2024) *Mixed Methods Research: Combining both qualitative and quantitative approaches*.

8. Kokk, G. and Jönsson, S. (2013) 'Visual research methods and the importance of analytical spaces', *Management & Organizational History*, 8(2), pp. 174–184. https://doi.org/10.1080/17449359.2013.779148

9. European Union. (2002). *Directive 2002/49/EC of the European Parliament and of the Council of 25 June 2002 relating to the assessment and management of environmental noise*. Official Journal of the European Communities.

10. Das, N., et al. (Google Research). (2023). *A Decoder-Only Foundation Model for Time-Series Forecasting (TimesFM)*. arXiv preprint arXiv:2310.10688. [Note: This covers the "Zero-Shot" capabilities of the model you are using].

11. International Electrotechnical Commission. (2013). *IEC 61672-1:2013 Electroacoustics - Sound level meters - Part 1: Specifications*. IEC.

12. ERR News. (2025, February 14). *Tallinn construction permits stalled due to outdated noise simulation data*. Eesti Rahvusringhääling. [Note: This provides the "Motivation" for why your thesis is needed *now*].

13. Tallinn City Government. (2021). *Tallinn 2035 Development Strategy*. Tallinn Strategy Center. https://www.tallinn.ee/en/tallinn2035

14. Murphy, E., & King, E. A. (2022). *Environmental Noise Pollution: Noise Mapping, Public Health, and Policy*. 2nd Edition. Elsevier. [Note: This is the "gold standard" book for modern noise mapping and the shift to IoT].

15. Murphy, E., & King, E. A. (2022). *Environmental Noise Pollution: Noise Mapping, Public Health, and Policy*. 2nd Edition. Elsevier.

16. International Organization for Standardization. (2015). *ISO 1683:2015: Acoustics—Preferred reference values for acoustical and vibratory levels*. https://www.iso.org/standard/64648.html

17. International Electrotechnical Commission. (2013). *IEC 61672-1:2013: Electroacoustics—Sound level meters—Part 1: Specifications*. https://www.iec.ch

18. European Parliament and Council of the European Union. (2002). *Directive 2002/49/EC relating to the assessment and management of environmental noise*. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32002L0049

19. Picaut, J., Can, A., Fortin, N., Ardouin, J., & Lagrange, M. (2020). *Low-cost sensors for urban noise monitoring networks — A literature review*. *Sensors, 20*(8), 2256. https://doi.org/10.3390/s20082256

20. Chai, T., & Draxler, R. R. (2014). *Root mean square error (RMSE) or mean absolute error (MAE)? – Arguments against avoiding RMSE in the literature*. *Geoscientific Model Development, 7*(3), 1247–1250. https://doi.org/10.5194/gmd-7-1247-2014

21. Picaut, J., Can, A., Fortin, N., Ardouin, J., & Lagrange, M. (2020). *Low-cost sensors for urban noise monitoring networks: A literature review*. Sensors, 20(8), 2256. https://doi.org/10.3390/s20082256

22. Oreshkin, B. N., Carpov, D., Chapados, N., & Bengio, Y. (2020). *N-BEATS: Neural basis expansion analysis for interpretable time series forecasting*. arXiv. https://arxiv.org/abs/1905.10437

23. Xie, C., Tank, A., Greaves-Tunnell, A., & Fox, E. (2017). *A unified framework for long range and cold start forecasting of seasonal profiles in time series*. *arXiv*. https://arxiv.org/abs/1710.08473

24. Butler, H., Daly, M., Doyle, A., Gillies, S., Hagen, S., & Schaub, T. (2016). *The GeoJSON format (RFC 7946)*. Internet Engineering Task Force (IETF). https://www.rfc-editor.org/rfc/rfc7946

25. World Health Organization. (2018). *Environmental noise guidelines for the European region*. WHO Regional Office for Europe. https://www.who.int/europe/publications/i/item/9789289053563

26. European Environment Agency. (2020). *Environmental noise in Europe — 2020*. Publications Office of the European Union. https://www.eea.europa.eu/publications/environmental-noise-in-europe

27. Murphy, E., & King, E. A. (2016). *Environmental noise pollution: Noise mapping, public health, and policy*. Elsevier.

28. Noel Cressie, N., & Christopher K. Wikle, C. K. (2011). *Statistics for spatio-temporal data*.

29. Josse Julie, J., & Husson François, F. (2016). *missMDA: A package for handling missing values in multivariate data analysis*. Journal of Statistical Software, 70(1), 1–31.

30. Mike Schuster, M., & Kuldip K. Paliwal, K. K. (1997). *Bidirectional recurrent neural networks*. *IEEE Transactions on Signal Processing, 45*(11), 2673–2681.

31. Sepp Hochreiter, S., & Jürgen Schmidhuber, J. (1997). *Long short-term memory*. *Neural Computation, 9*(8), 1735–1780

32. R. W. Sinnott (1984). *Virtues of the Haversine*. *Sky and Telescope, 68*(2), 159–160.

33. Luca Nencini, L., et al. (2020). *Low-cost sensors for environmental noise monitoring: A review*. *Sensors, 20*(24), 1–25.

34. Hugging Face. (2023). *Hugging Face model hub*. https://huggingface.co

35. Das, A., Kong, W., Sen, R., & Zhou, Y. (2024). *A decoder-only foundation model for time-series forecasting*. arXiv. https://arxiv.org/abs/2310.10688

Appendix 1 - Non-exclusive licence for reproduction and publication of a graduation thesis

I, Khamidjon Khamidov

1. grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Data imputation and Animated visualization of Sound Pressure Level in Tallinn based on IoT Sensor Data", supervised by Jaanus Kaugerand

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2 I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3 I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

Appendix 2 - Source code

The full implementation developed for this thesis is available at: https://github.com/khamidjon-khamidov/spl_analysis