

TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Viktoria Siigur 240681IAPM

**Sound Pressure Level Analysis and
Visualization in Tallinn Urban Environment
Based on Low-Cost IoT Sensor Data**

Master's thesis

Supervisors: Jaanus Kaugerand
PhD

Jeffrey A. Tuhtan
Assoc. Prof. Dr.-Eng.

Tallinn 2025

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Viktoria Siigur 240681IAPM

**Helirõhutaseme analüüs ja visualiseerimine
Tallinna linnakeskkonnas madala
maksumusega IoT-andurite andmete põhjal**

Magistritöö

Juhendajad: Jaanus Kaugerand
PhD

Jeffrey A. Tuhtan
Assoc. Prof. Dr.-Eng.

Tallinn 2025

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 others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Viktoria Siigur

07.05.2025

Abstract

This thesis investigates sound pressure level data collected from the urban environment of Tallinn. The dataset was recorded using a large-scale, low-cost sensor network developed by the TalTech Proactive Technologies research group in collaboration with the Estonian company Thinnect and covers the period from January 2021 to May 2023.

The first major component of the study addresses the persistent issue of missing data, which is common in large-scale asynchronous monitoring systems. Two imputation methods, specifically designed for this dataset, are proposed and evaluated. The first method, self-imputation, relies on a sensor's own historical data to estimate missing values. The second method, nearest-imputation, utilizes data from geographically nearby sensors. Both methods were tested and validated using a dedicated 24-hour test set, allowing for direct comparison of their performance.

The second main contribution of this thesis focuses on the dynamic visualization of the complete dataset, including the imputed values. A full-stack web application was developed for this purpose. The system includes a backend that interfaces with a database containing the cleaned dataset and performs simplified gap imputation, and a frontend that displays an interactive map. On the map, each sensor is represented by a coloured dot indicating sound pressure level, following a standardized colour scheme used by both the Estonian national noise regulation as well as the European Union's Environmental Noise Directive.

This thesis is written in English and is 56 pages long, including 6 chapters, 22 figures and 1 table.

Annotatsioon

Helirõhutaseme analüüs ja visualiseerimine Tallinna linnakeskkonnas madala maksumusega IoT-andurite andmete põhjal

Käesolev magistritöö uurib Tallinna helirõhutaseme andmeid, mis on kogutud madala maksumusega andurite võrgustiku abil. Võrgustik on välja arendatud Proaktiivtehnoloogiate uurimisrühma poolt koostöös Eesti ettevõttega Thinnect. Analüüsitava andmestik hõlmab ajavahemikku 2021. aasta jaanuarist kuni 2023. aasta maini.

Töö esimene põhiosa keskendub andmelünkade käsitlemisele. Selleks on välja töötatud ja kirjeldatud kaks spetsiaalselt antud andmestiku jaoks loodud asendusmeetodit. Esimene neist kasutab puuduolevate väärtuste ennustamiseks anduri enda varasemaid mõõtmisi. Teine meetod, lähimate andurite põhine asendus kasutab andmete täiendamiseks lähedalasuvate andurite andmeid. Mõlemad meetodid testiti ja valideeriti spetsiaalse 24-tunnise testandmestiku abil, võimaldades nende omavahelist võrdlust.

Töö teine põhiosa keskendub andmestiku visualiseerimisele koos täidetud andmelünkadega. Selleks loodi täismahus veebirakendus, mis koosneb andmebaasist koos puhastatud andmetega, *backend*'ist, mis suhtleb andmebaasiga ning rakendab lihtsustatud lünkade täitmist, ning kasutajaliidesest, mis kuvab interaktiivse kaardi. Kaardil on iga andur kuvatud värvilise täpina, mis näitab vastavat helirõhutaseme väärtust, järgides nii Eesti riiklikus müraregulatsioonis kui ka Euroopa Liidu keskkonnamüra direktiivis kasutatavat standardset värviskeemi.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 56 leheküljel, 6 peatükki, 22 joonist, 1 tabelit.

List of abbreviations and terms

IoT	Internet of Things
SPL	Sound Pressure Level
END	European Noise Directive
HCI	Human Computer Interaction
LOFT	Last Observation Carried Forward
STC	Space-Time Cube
ID	Identification
kNN	k Nearest Neighbours
SQL	Structured Query Language
GiST	Generalized Search Tree
ERD	Entity Relationship Diagram
DTO	Data Transfer Object

Table of contents

1 Introduction	13
2 Related Work.....	15
2.1 Global approaches to SPL monitoring.....	15
2.1.1 New York, USA	15
2.1.2 Barcelona, Spain.....	16
2.1.3 Rezé, France	16
2.2 Imputation methods for time series data.....	18
2.2.1 Last Observation Carried Forward (LOCF)	18
2.2.2 Linear interpolation	19
2.2.3 Least Squares based data imputation method.....	20
2.3 Existing visualization methods.....	21
2.3.1 Two-dimensional SPL visualization.....	21
2.3.2 Three-dimensional SPL visualization.....	22
3 Urban Noise Data Collection in Tallinn.....	25
3.1 Sensor network	25
3.2 Research questions	27
4 Data Processing Methods	29
4.1 Understanding the data	29
4.1.1 Find the largest amount of valid data	31
4.1.2 Extra cleanup for the data.....	31
4.2 Filling the gaps	32
4.2.1 Self-imputation method	33
4.2.2 Nearest-imputation method	35
4.3 Developing the web application	37
4.3.1 Server-side application	37
4.3.2 Client-side application.....	38
5 Results	40
5.1 Self-imputation method	40
5.2 Nearest-imputation method	44

5.3 Comparison of the methods	45
5.4 Visualization	46
6 Summary and Conclusions	53
References	54
Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis	56

List of figures

Figure 1. A strategic noise map of Tallinn city for day-evening-night measurements of 2022 from [20]. This map displays the environmental noise levels across Tallinn city during the whole day. Noise intensity is illustrated using a colour gradient from green to purple.	22
Figure 2. A noise contour map showing the impact of a building adjacent to a motorway from [21]. This illustration depicts the levels of environmental noise around a building situated near a motorway. The contours represent noise intensity, starting from lower levels in green to higher levels in red. The building structure influences the propagation and concentration of noise, as indicated by the gradient changes around it. The motorway is labelled to highlight the primary source of noise.	23
Figure 3. Space-time cube visualization for continuous data from [21]. This figure demonstrates an alternative method for time visualization in mapping by utilizing the Z-axis to represent time.	24
Figure 4. Network of low-cost IoT sensors installed on streetlight poles in Tallinn. In 2019, the Institute of Software Science in TalTech and Thinnect OÜ developed technology to monitor urban noise, air quality, and measure traffic flows. Approximately 900 devices equipped with batteries and solar panels were installed on streetlight poles across Tallinn to collect environmental and traffic density data [22]. The locations of these sensors are marked on the map.	26
Figure 5. A single microphone SPL sensor installed on a streetlight pole. The device includes a solar panel for energy harvesting and a SPL sensor to monitor urban noise.	27
Figure 6. Entity Relationship Diagram illustrating the connection between the “device” table, containing device name and location, and the “raw_event” table, storing event records with details such as date, hour, and measured values.	30
Figure 7. Flowchart illustrating the imputation process for missing data by calculating medians of rolling windows of increasing size, comparing the difference of medians against a threshold, and setting the imputation map value based on the findings.	34
Figure 8. New ERD Scheme for the final database. ERD displaying the relationships among the “imputation_map” table, which stores imputation map values configured for	

each device and hour; the “device” table, containing metadata such as device name and location; and the “raw_event” table, recording event values with their respective date, hour, and related device ID.....	34
Figure 9. Flowchart demonstrating the imputation process by fetching previous days count from the imputation map, retrieving values for the current device and hour, checking if at least one value exists, and taking the median of the values as the estimated value, or indicating impossibility of imputation if no values exist.	35
Figure 10. Flowchart illustrating the imputation process by checking for devices within a 110-meter radius, retrieving values from nearby devices, confirming the existence of at least one value, and taking the median of these values as the estimated value, or indicating impossibility of imputation if no values exist.	36
Figure 11. Diagram illustrating the process of handling a GET request for device data on specific date/hour by fetching device information from the device table, real values from the “raw_event” table, the latest existing value for each device, and assembling the response as a list of DeviceDTO (Device Data Transfer Object) objects containing device ID, name, location, real value, and estimated value.....	38
Figure 12. Table showing 25 rows on the imputation map, which consists of device IDs, hours and the previous days count which indicates the number of days needed for the best estimation for missing data for device/hour. The colours give the visual overview of the amount of different number of days needed.....	41
Figure 13. Hourly distribution where previous day value equals 1. This figure shows the hourly distribution of data indicating the percentage of instances where the previous day count equals 1 for each hour. It provides insight into how often a single previous day is used for estimating missing data across different hours.....	42
Figure 14. The imputation map for the device used in the test set, where device_id is 550, showing the number of previous days required for each hour's estimation, with colour coding to indicate different hour counts.....	42
Figure 15. The example of the self-imputation method. The image shows the data from timeframe of 7 th September 2022 at 18 to 8 th of September at 17 on the device 503C with the real values and estimated values based on the self-imputation method, alongside with the error on each timestamp and the number of past values used for calculating the estimation.	43
Figure 16. The example of the nearest-imputation method. The image shows the data from timeframe of 7 th September 2022 at 18 to 8 th of September at 17 on the device	

503C with the real values and estimated values based on the nearest-imputation method, alongside with the error on each timestamp and the number of near values used for calculating the estimation.	44
Figure 17. The earliest possible map of SPL of Tallinn created with the application. It consists of 8 different real values and 0 estimated values, since there is no historical data at that point. On the image, there are 4 green and 4 yellow dots indicating the existing values of the sensors for chosen moment.	47
Figure 18. The latest possible map of SPL of Tallinn created with the application. There are no sensors without values, since all the sensors do have at least 1 historical value to make an estimate with. The colourful dots indicate each sensor values for chosen moment, white small dots in the centre indicate that the indicator colour is based on estimated value, grey dot in the middle shows it is based on the real value.	48
Figure 19. The SPL map of 8 th of September 2022 at 3 o'clock. Most of the sensors are coloured green, which indicates the low overall SPL.	49
Figure 20. The SPL map of 8 th of September 2022 at 15 o'clock. Most of the sensors are yellow/orange, which indicates relatively noisy environment, between 60-69 decibels.	50
Figure 21. The real and estimated values of test device 203C at 8 th of September 2022 for 14 o'clock, which are necessary for the comparison with the previously made self-imputation method calculations.	51
Figure 22. The real and estimated values of test device 203C at 7 th of September 2022 for 18 o'clock, which are necessary for the comparison with the previously made self-imputation method calculations.	52

List of tables

Table 1. The advantages and disadvantages of the self- and nearest-imputation methods.	45
---	----

1 Introduction

Noise is clearly identified as a cause of stress to humans and prolonged, elevated levels of stress are known to cause mental and physical health problems [1]. Environmental noise originates from many sources – road traffic, railway, aircraft noise and leisure activities [2]. The EU Environmental Noise Directive (END) establishes a common approach across EU Member States to avoid, prevent and reduce the harmful effects of environmental noise. The directive requires Member States to produce strategic noise maps every five years for major roads, railways, airports and agglomerations (urban areas with over 100,000 inhabitants), and to develop corresponding action plans to manage noise issues [3]. In Estonia, the requirements of the END are transposed into national legislation through the Estonian government’s regulation “Technical requirements for environmental noise map, strategic noise map and noise reduction action plan and preparation procedure” [4]. While the Estonian regulation closely follows the requirements set by the European Union’s END directive, Estonia applies the same general noise thresholds for strategic mapping as outlined in the END. In Estonia, strategic noise maps must be prepared every five years for Tallinn, which is the only agglomeration over 100,000 inhabitants, as well as for major roads, railways, and Tallinn Airport, where traffic volumes exceed thresholds specified in the directive.

Traffic noise in Tallinn has become a growing concern, with levels exceeding recommended limits, primarily due to heavy traffic [5]. As stated in the Environmental noise guidelines for the European Region created by the World Health Organization, for average noise exposure, it is strongly recommended to reduce noise levels produced by road traffic below 53 dB over a 24-hour period and 45 dB for the night, as road traffic noise above this level is associated with adverse effects for health and sleep [2]. A study conducted by the University of Tartu found that 11.6% of residents are highly annoyed by the traffic noise, while 2.5% have high sleeping disturbances [6]. As car ownership continues to rise, noise pollution is likely increasing further. The issue is particularly severe along major roads across all districts, where traffic is the primary source of noise. The Tallinn city government has halted building permits in several areas due to high noise

levels, leading to a slowdown in new construction projects and keeping real estate prices high. The Health Board has also refused to approve several planning proposals, preventing residential developments along some of the major roads [5].

To address these challenges, the city government has adopted a noise reduction action plan. The plan prioritizes promoting public transport and cycling as alternatives to cars, as well as implementing speed regulations to help manage noise levels. Additionally, new noise measurements will be conducted, and an updated noise map is expected to be released in 2026. This data will provide a basis for future urban planning decisions, allowing policymakers to implement strategies for reducing noise pollution and improving city living conditions [7].

In addition to the planned new measurements, there is already an existing network of sensors across Tallinn that provides real-time sound pressure level (SPL) data, with each sensor collecting data every minute. The END regulations stipulate reporting over daily intervals, which themselves are aggregated into three periods consisting of several hours each. These periods are the day (07:00 – 19:00) evening (19:00 – 23:00) and night (23:00 – 07:00) [3]. While official measurements remain essential for regulatory compliance and standardization, the availability of historical data from the sensor network enables a valuable opportunity for preliminary assessments.

This thesis focuses on analysing data collected from a real sensor network, dynamically estimating missing values and developing meaningful visualizations of SPL at an hourly resolution. To achieve this, data cleaning methods are applied, including removing duplicates, correcting structural errors, eliminating irrelevant observations, and handling outliers. Following the cleaning process, two imputation methods – self-imputation and nearest-imputation – are developed. The final visualizations combine real data with estimates generated using a simplified version of the self-imputation method.

These visual outputs are intended to support faster initial conclusions and contribute to a more responsive and dynamic approach to urban noise management. The goal is to create an easy-to-use and reliable tool that can help the City of Tallinn to rapidly identify, quantify and understand problem areas enabling informed decisions to improve the city's acoustic environment.

2 Related Work

2.1 Global approaches to SPL monitoring

The regulation of environmental noise has deep historical roots, dating back to ancient civilizations. As early as the 6th century BCE, the Greek city of Sybaris banned roosters and noisy trades such as blacksmithing within city walls to preserve public peace. In ancient Rome, Julius Caesar restricted wagon traffic during the day to reduce noise in residential areas. By the 16th century, London implemented bylaws to prevent nighttime disturbances from singing and revelry. The industrial era brought growing concern for occupational noise, with early medical recognition of hearing loss among blacksmiths in the 1830s. In the late 19th and early 20th centuries, organized movements like the Society for the Suppression of Noise emerged in London and New York, advocating against motor horns. These efforts laid the foundation for modern legal frameworks, including the 1957 Chicago Zoning ordinance which is the first noise ordinance in the world to specify maximum noise levels [8].

In 2013, a research paper was published that described the situation in the United States. They estimated that 104 million individuals had an annual continuous average exposure level of >70 dBA over 24 hours in 2013 and were at risk of noise-induced hearing loss and tens of millions more were possibly going to be at risk of heart disease and other noise-related health effects [9]. Since the problem is similar in urban environments worldwide, three distinct approaches around the world will be showcased.

2.1.1 New York, USA

In New York, the city officials came out with the project Sounds of the New York city, shortly SONYC. The project integrates Human Computer Interaction (HCI) research into noise monitoring by deploying sensors and involving municipal agencies and residents. The initial approach was to deploy fixed-position sensors, using municipal buildings or streetlight poles. The project evolved to use portable, domestic sensors deployed in problem areas. To gather data that provides an informative view of the acoustic environment at problem sites, a program was launched in which novel domestic sensors,

designed by the team, were deployed for periods of two to four weeks. Residents were identified by the New York City Department of Environmental Protection via 311 reports of ongoing noise problems. These sensors provide acoustic data shared via a web app, enabling residents to monitor noise levels and report disturbances. Incorporating machine learning, the system aims to automate noise source identification. The project supports community action by leveraging open data for analysis and enforcement [10].

2.1.2 Barcelona, Spain

The Barcelona Noise Monitoring Network is a comprehensive system designed to monitor and manage urban noise pollution across the city. The network comprises two primary components - the main network and the complementary network. The main network utilizes high-precision Class I noise monitors installed at strategic locations, such as municipal and university buildings, as well as urban infrastructure like streetlight poles. These devices provide accurate real-time data on noise levels. To enhance coverage and flexibility, especially in response to residents' concerns in various city areas, a complementary network of low-cost sound sensors was developed. A key objective of this expanded network is to increase the number of measurement points, enabling the detection of real-time changes in noise level trends. Barcelona's city sensor network was planned to enhance data sharing and cross-departmental collaboration. The main network utilized the IBM Cognos platform for data storing and analysis. Providers are required to submit data in a standardized .csv format. For the complementary network, data is managed through the Sentilo platform, designed to serve as a link between sensors and the applications managing urban data [11]. In 2021, the research paper “The Soundscape of the COVID-19 Lockdown: Barcelona Noise Monitoring Network Case Study” [12] utilized the network for analysing the changes in SPL during the pandemic lockdown, where it is revealed that from January 2020 to June 2020, which was the strictest lockdown time for the pandemic, Barcelona experienced significant reductions in noise levels – up to -9 dBA in nightlife areas, -7 dBA in commercial and restaurant zones and -5 dBA in dense traffic areas.

2.1.3 Rezé, France

In Rezé, France, a co-constructed experiment was taken place between researchers and local authorities of an urban noise diagnosis based on the residents and the use of the smartphone application NoiseCapture, which allowed a participative measurement of

sound levels. Data collection lasted 7 months, extending from December 2021 to June 2022 inclusive. The participants were free to take measurements whenever they wished, with the instruction either to take measurements in spots of at least 1 min or to take mobile measurements, taking care not to pollute the measurements with their own steps and to pay attention to the weather conditions (avoid either rain or gusts of wind). The smartphones were properly calibrated to get correct values of sound level and time. They managed to collect a considerable amount of data to analyse it and make informative sound environment maps [13].

2.2 Imputation methods for time series data

The sensor data itself does not provide any meaningful insights in its raw form. To extract valuable information from the collected data, it is essential to clean and impute any missing values. This step is crucial to ensure the continuity of the data before proceeding with the visualization process. When selecting imputation methods for sensor data, it is important to consider the computational constraints of the deployment environment. The chosen methods must be autonomous, requiring no manual intervention once deployed. They should be lightweight as well in terms of memory and processing power, making them suitable for real-time execution. The methods must also be capable of running directly on a microcontroller, which often lacks the resources of a full computer. These requirements ensure that data imputation can be made locally on the sensors in the future. The following paragraphs will describe several lightweight time series imputation methods and explore how they have been applied in real-world scenarios in clinical and environmental research.

2.2.1 Last Observation Carried Forward (LOCF)

The LOCF method is a technique commonly used in longitudinal studies to handle missing data. When a data point is missing at a specific moment, LOCF imputes the missing value by carrying forward the last observed value from the previous time point. This approach assumes that the value at the last observed moment remains unchanged until the next available data point.

In the article "Last observation carry-forward and last observation analysis" by Jun Shao and Bob Zhong [14], the authors discuss how the LOCF method is applied in clinical trials, particularly to handle missing data from participants who drop out of the study. In these trials, participants may leave before the study is completed, resulting in missing data. The LOCF method allows for the retention of this data by filling in the missing values with the last observed data point prior to the drop-out. This method is particularly useful in intention-to-treat analyses, which aim to evaluate the effects of a treatment across all randomized participants, even those who drop out before completing the study enlarging the amount of the population for the research [14].

On the other hand, the LOCF can be considered as a rather conservative method in handling missing data, falsely enhancing the statistical precision, which increases the

likelihood of finding a statistically significant invalid treatment response. Multiple biostatisticians have recommended against the use of LOCF due to the introduction of bias [15].

Despite these concerns, the fundamental idea behind LOCF – relying on the most recently available observation – remains useful in certain contexts. In the current thesis, a modified version of this principle is employed, where missing values are imputed using the last known value or values from a specific sensor-hour group. Unlike traditional LOCF, which carries forward the most recent value regardless of temporal or structural context, the approach proposed in this work leverages temporal and spatial patterns specific to the sensor and time of day, making it more suitable for large datasets with recurring gaps and periodic behaviour. The specifics of this imputation method are detailed further in chapter 4.2.1.

2.2.2 Linear interpolation

Linear interpolation estimates missing values based on the adjacent available values. It is preferred for estimating continuously missing data over a short time interval. For a missing value linear interpolation generates the estimation based on the closest preceding and succeeding available values. Linear imputation is simple, fast, and requires only two available samples to impute each missing data period. On the other hand, the accuracy of linear imputation typically decreases as the length of the missing data period increases. In the paper “Handling missing data in near real-time environmental monitoring: A system and a review of selected methods” by Yifan Zhang and Peter J. Thorburn [16] it is used as one of the methods for imputing the real-time water quality data. While linear interpolation might provide a reasonable estimate for missing water temperature or nitrate concentrations over a few hours, it may fail to accurately capture the true variability of these parameters over longer durations or in highly variable environments. Additionally, linear interpolation does not account for the potential influence of external factors such as weather events or upstream activities that could affect water quality. Despite these limitations, linear interpolation remains a useful initial approach for data imputation, particularly when computational simplicity is a priority and when the data gaps are relatively small [16].

However, in the context of the present research, linear interpolation is not used due to its inability to account for the periodic and structured nature of the sensor data. Many of the

sensor measurements exhibit recurring patterns based on the time of day and sensor-specific behaviour, which linear interpolation does not capture. Because it assumes a linear transition between two points, the method may produce unrealistic values in situations where the underlying data fluctuates in a non-linear or cyclic manner.

2.2.3 Least Squares based data imputation method

The Least Squares data imputation method is a technique used to fill in missing values in datasets, especially those coming from environmental sensors or monitoring systems. As it is described in the paper “Open Environmental Data Assimilation Under Unknown Uncertainty and Multiple Spatio-Temporal Scales” by Lizaveta Miasayedava et al. [17], the main idea behind this method is to combine data from sensor measurements and model simulations to estimate a more accurate value when some data is missing or uncertain. This method works by giving weight to both data sources based on how certain they are. If one source is very accurate, it will have a higher influence on the result. If both sources have some uncertainty, the method balances them according to their estimated error and provides mathematically proven result that has lower uncertainty. However, in many real-life cases, especially with open data or low-cost sensors, the exact uncertainty of the data is not known. To overcome this problem, the method uses a simple prediction model, which is based on past values, to estimate how much error or uncertainty each data source might have. This is done using an autoregressive model, where the next value is predicted based on the previous one. The difference between the predicted value and the actual value gives an idea of how uncertain that source is. The bigger the difference, the less trust is given to that data source when combining it with others. When the data comes from very different types of sources for example, a point measurement from a sensor and a large-scale simulation from a model the method includes a step to calibrate the model data to better match the sensor data. This calibration also has its own uncertainty, which is also considered [17].

While the Least Squares method offers a sophisticated way to combine multiple data sources with consideration of uncertainty, it is not well-suited for the current research. Autoregressive models usually work best when the data is consistent, does not have large gaps and is evenly spaced in time. In the dataset for current thesis, there are often gaps which are large and irregular. These gaps make it harder for autoregressive models to give good predictions, because the method expects a steady flow of data without missing steps.

2.3 Existing visualization methods

Returning to the topic of noise, an effective way to gather information is by mapping it onto an actual visual map, depending on the specific insights we aim to derive from it. The visualization of noise data can be approached in different ways, each suited for distinct purposes and user requirements. This chapter describes two- and three-dimensional mappings of the SPL data.

2.3.1 Two-dimensional SPL visualization

One example of the two-dimensional SPL visualization is an official noise map made in Tallinn in every 5 years. The maps are created according to the Strategical Guide to a Noise Map provided by the Ministry of Climate, which serves as a comprehensive guide for local governments and consultants involved in the creation of strategic noise maps in Estonia. Its primary aim is to provide detailed instructions on the data requirements, quality standards, computational settings and presentation formats. The guide is particularly tailored to ensure compliance with both national and European Union regulations. It also serves as a valuable resource for preparing procurement documents related to noise mapping projects [18].

According to the guide, four different maps are created – for day, which is measured from 07:00 until 19:00, for evening, measured from 19:00 to 23:00 and for the night, from 23:00 to 07:00 and for day-evening-night combined [19]. The example of the day-evening-night map is shown on Figure 1.

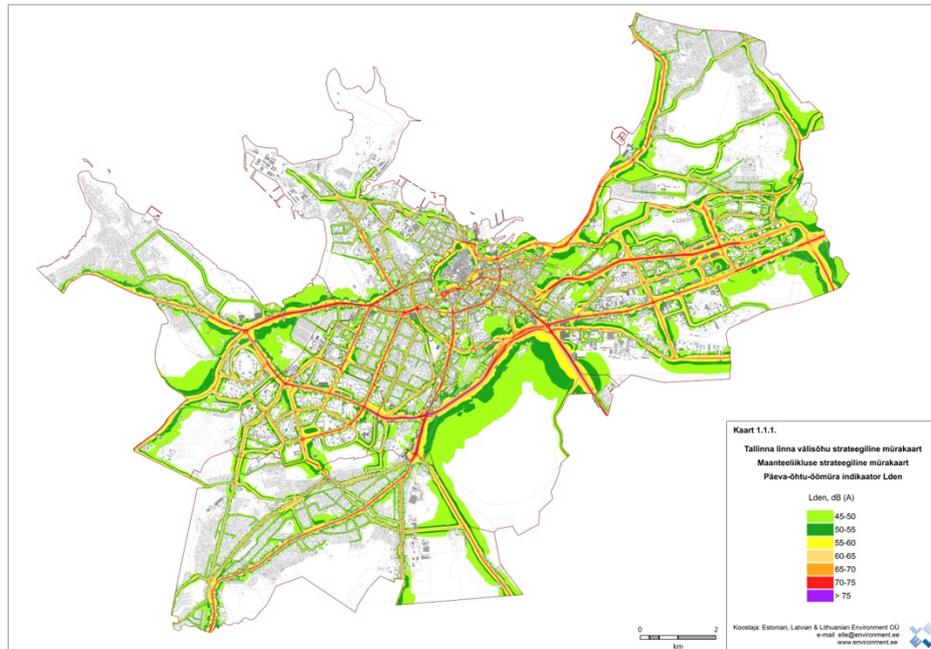


Figure 1. A strategic noise map of Tallinn city for day-evening-night measurements of 2022 from [20]. This map displays the environmental noise levels across Tallinn city during the whole day. Noise intensity is illustrated using a colour gradient from green to purple.

While static noise maps like the one in Figure 1 are detailed and accurate, they have limitations when it comes to analysing how noise levels change over time. For the purposes of this thesis, a more dynamic approach is required, one that allows for observing temporal variations in SPL rather than relying solely on a fixed snapshot. This enables a deeper understanding of how noise behaves throughout the day and across different conditions, which is essential for the goals of this research.

2.3.2 Three-dimensional SPL visualization

For some visualizations, two dimensions may not provide sufficient detail. To achieve a more advanced representation, an additional dimension needs to be incorporated, allowing for a more comprehensive and insightful analysis of the data. In the paper “The Third Dimension in Noise Visualization - a Design of New Methods for Continuous Phenomenon Visualization” by Daniel Beran, Karel Jedlička, Kavisha Kumar, Stanislav Popelka and Jantien Stoter, the researchers collected some existing three-dimensional visualization practices which are worth investigating [21].

One such method is noise facade visualization, which goes beyond displaying noise distribution along streets. It also illustrates how noise spreads across building surfaces

and how buildings influence and disrupt noise propagation. The example of such method is shown on the Figure 2.

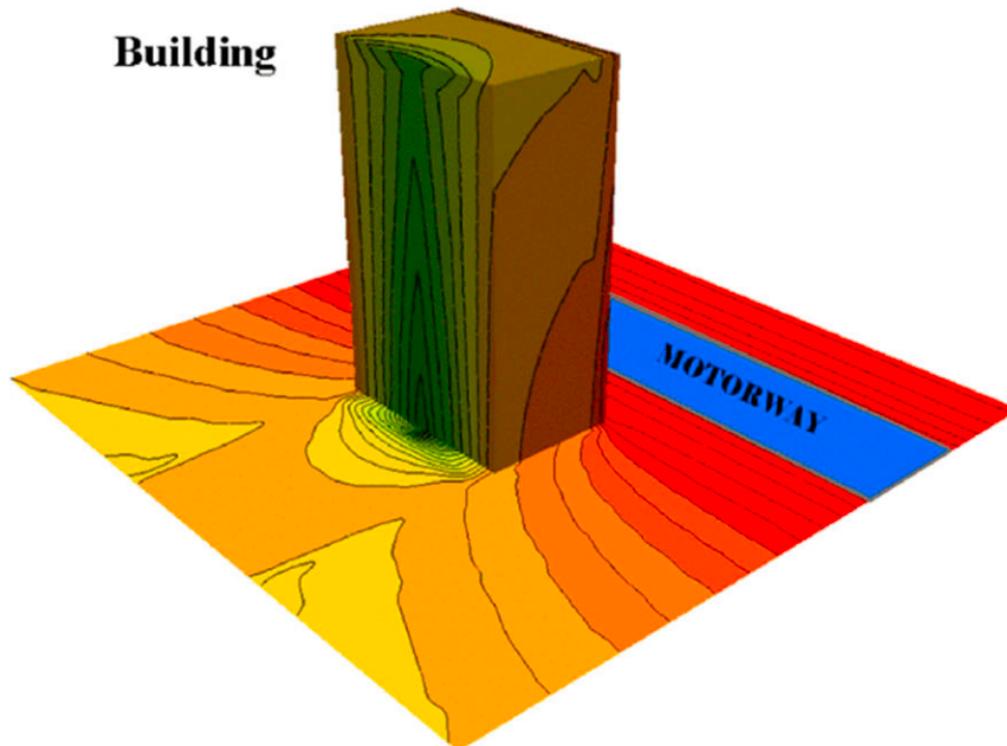


Figure 2. A noise contour map showing the impact of a building adjacent to a motorway from [21]. This illustration depicts the levels of environmental noise around a building situated near a motorway. The contours represent noise intensity, starting from lower levels in green to higher levels in red. The building structure influences the propagation and concentration of noise, as indicated by the gradient changes around it. The motorway is labelled to highlight the primary source of noise.

Another method that uses third dimension not in the classical way, but it utilizes the third dimension of a scene for the representation of time instead of vertical dimension. This approach to time visualization in mapping does not have a universally accepted cartographic term, but in the research paper, it is mentioned as space-time cube (STC). However, there are many examples of such an approach in data visualization. The problem with applying space time cube for continuous data is occlusion of layers as they cover the whole timeframe and thus disallow the user to see changes of phenomena [21]. The example of STC is shown on the Figure 3.

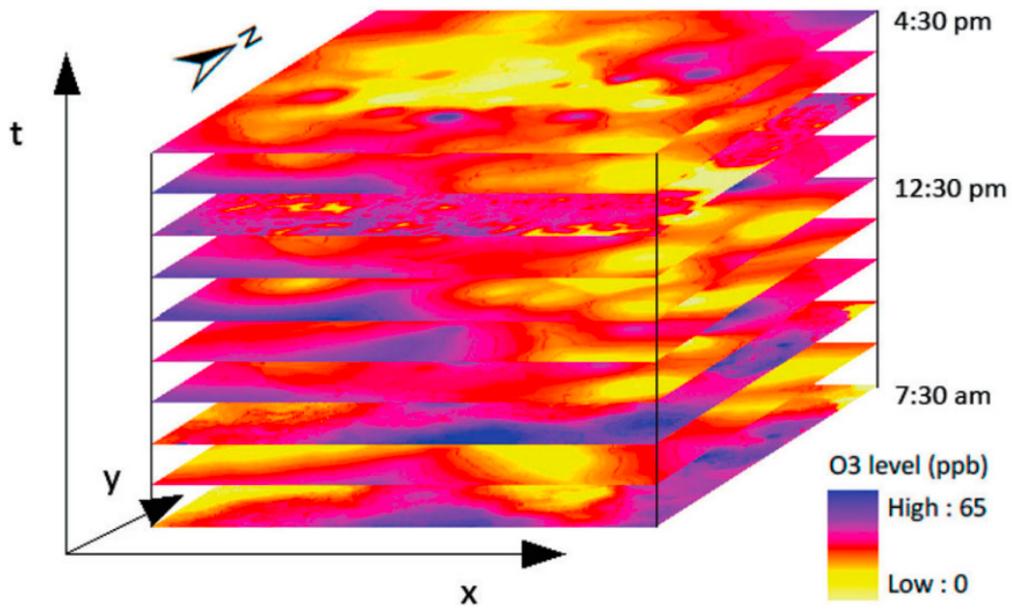


Figure 3. Space-time cube visualization for continuous data from [21]. This figure demonstrates an alternative method for time visualization in mapping by utilizing the Z-axis to represent time.

While three-dimensional visualizations like facade mapping and space-time cubes offer deeper insights into how noise behaves in space and time, they are not ideal for the purposes of this thesis. These methods are often complex, visually dense, and require more detailed input data than is available in this study. Visualizations like the STC can suffer from visual clutter, especially when dealing with continuous data, making it harder to clearly observe changes over time. For this research, a simpler and more focused visualization method is preferred – one that still shows temporal changes but remains easy to interpret and suitable for the available data.

3 Urban Noise Data Collection in Tallinn

The SPL data has been collected from Tallinn sensor network since 2019, but there is not much information generated out of this. The analysed and visualized data would be useful for city planners and residents to have a great overview for problematic areas with too loud environments. Currently the strategic noise maps of Tallinn are created manually every five years for average SPL for 07:00 to 19:00, 19:00 to 23:00, 23:00 to 07:00, and for the whole day [19]. However, having access to noise level data for specific days and hours would significantly enhance usability, allowing for real-time analysis and historical comparisons. This capability would provide valuable insights for city planners by enabling them to assess how urban development projects have influenced noise pollution over time. For instance, if certain streets have been redesigned to reduce traffic, noise maps could effectively illustrate whether the intended noise reduction has been achieved.

3.1 Sensor network

In 2019, the Laboratory for Proactive Technologies at the Institute of Software Science at TalTech and Thinnect OÜ were developing technology to monitor urban noise, air quality and measure traffic flows. A network of smart city sensors was installed on Tallinn's streetlight poles, with ~900 devices equipped with batteries and solar panels to collect environmental and traffic density data [22]. Sensor locations are visible on the Figure 4.

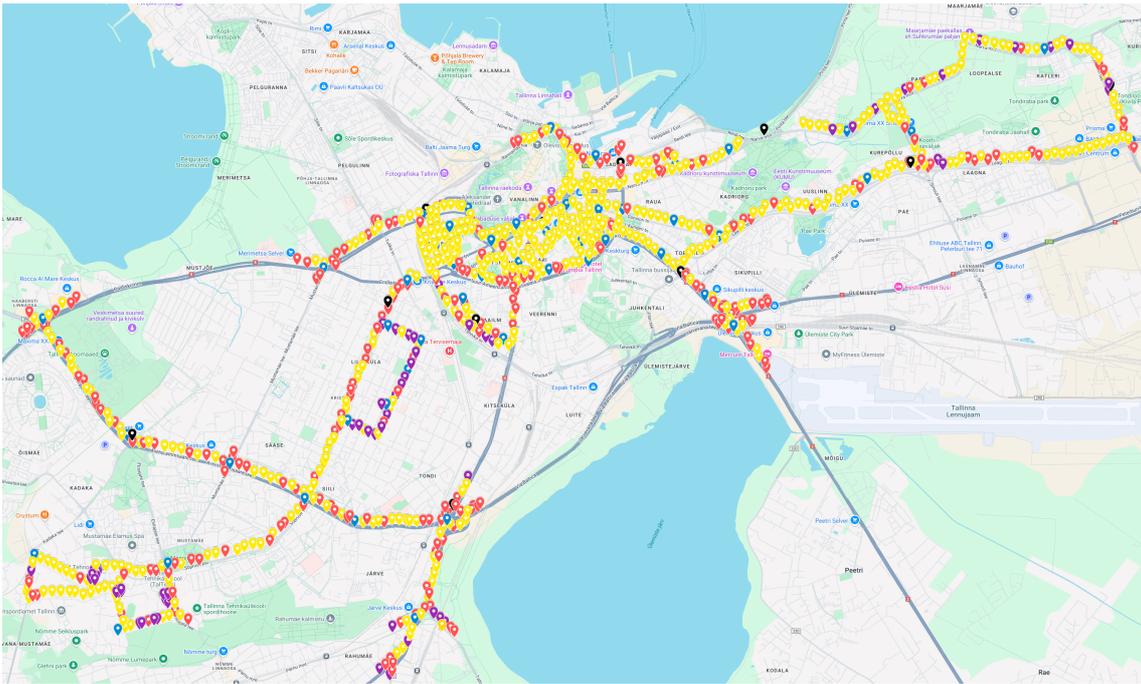


Figure 4. Network of low-cost IoT sensors installed on streetlight poles in Tallinn. In 2019, the Institute of Software Science in TalTech and Thinnect OÜ developed technology to monitor urban noise, air quality, and measure traffic flows. Approximately 900 devices equipped with batteries and solar panels were installed on streetlight poles across Tallinn to collect environmental and traffic density data [22]. The locations of these sensors are marked on the map.

These sensors can create a wireless network without a central control unit. This type of network is called a multi-hop, mesh network [23]. Collecting SPL data is one of the tasks performed by the network. In addition, the weatherproof sensors gather information on air temperature and pollution, as well as detect the presence of pedestrians and monitor vehicle movement and direction [22]. The example of the sensor mounted on the street pole is shown on the Figure 5.



Figure 5. A single microphone SPL sensor installed on a streetlight pole. The device includes a solar panel for energy harvesting and a SPL sensor to monitor urban noise.

3.2 Research questions

The goal of the research is to create a visualization of the SPL in Tallinn based on the collected data. There are several challenges in creating these visualizations. The data provided by sensors can be sometimes intermittent and not fully adequate since they are based on low-cost IoT technology and running on the experimental mesh multi-hop network that is influenced by the surrounding environment. For example, since the sensors are powered with the solar panels, they tend to run out of battery during cloudy days, there can be radio interference and green leaves from trees can sometimes hinder radio communication. As well, since the sensors are small and the hole of the microphone can be filled with different kinds of dust, some values can be misinterpreted.

Question 1: which methods are suitable for sensor data imputation to provide computationally efficient and reliable estimates of the SPL data quality?

The current research approach focuses on identifying patterns across the entire dataset and using these patterns to impute missing data.

Question 2: which methods are the best-suited to generate dynamic maps of the SPL in Tallinn, considering their accuracy and ease-of-interpretation by different types of end-users?

There are numerous frameworks available for visualizing data. Given the large volume of data, it must first be analysed to extract only the relevant information. The visualization then generates dynamic point-based map based on the processed data, ensuring a clear and meaningful visual representation of the sound levels for users.

4 Data Processing Methods

The sensors have been running for several years now, but for the current research, access is limited to approximately two and a half years of data. The dataset chosen for this research is from 1st of January 2021 until 24th of May 2023. Since the sensor network is experimental and operates over a wireless connection, its performance is affected by environmental factors, leading to occasional data corruption or misinterpretation. As mentioned before, since the system is low-cost and lightweight, it is necessary to be prepared for these consequences and obtain the best possible results.

Current research involves deep analysis of the data, going through the raw values and getting familiar with them to find the algorithm to get best estimation of the missing gaps. In the first part of the chapter, there are two hypotheses described, one involves the history of the sensor data, another one deals with the data from neighbouring devices. Second part of the research finds the way to informatively plot the values and uses the lighter version of one of the hypotheses for the estimation. Implementation of both algorithms requires data cleaning, which consists of removing duplicates, correcting structural errors, eliminating irrelevant observations and handling outliers.

The end goal of developing the algorithm is to enable real-time estimation of missing values. This means that if the best estimation is sufficiently accurate, the algorithms could be implemented directly in the sensor's software. In cases where data is missing at a given moment, the system could automatically replace the missing data with the best estimation.

4.1 Understanding the data

The provided database contains large amount of data, as it does not solely consist of SPL data. The main tables relevant to this research are devices and events. The dataset contains 353,570,349 rows of event data and 1015 rows of device metadata, reflecting measurements collected from the sensor network. The devices table includes information on all sensors, such as their names, types, and locations, along with additional details that are not relevant to the current research context. There are approximately 900 sensors in

total out on the streets and only about half of them are sound pressure level sensors, it is necessary to filter out the relevant ones. Among these, sensors with one or two microphones are considered. Sensors equipped with two microphones have been designed for the capability to determine the direction of the sound source. Although this functionality is not required for this research, the data from these sensors remains useful and can be incorporated into the final dataset.

Once the data of correct sensors have been identified, they are transferred to an additional database, which is used to retain only the necessary values for the project and allows for further filtering in the next steps. The additional database is hosted on the Digital Ocean platform [24]. The new table for devices contains only the device ID, name, and location. There is an extension for PostgreSQL database called PostGIS that extends the capabilities of the relational database by adding support for storing, indexing, and querying geospatial data [25]. The GiST index is added to the location column for faster querying of the nearby devices.

All the SPL values are stored in the events table, with the index on device id, date and hour combined. This includes the data for all the other sensors as well, but the SPL values are labelled accordingly. Since the data is very raw and not cleaned in any sense, there are many filtering rules that were developed during the analysing process. Figure 6 shows the Entity Relationship Diagram (ERD) of the end database.

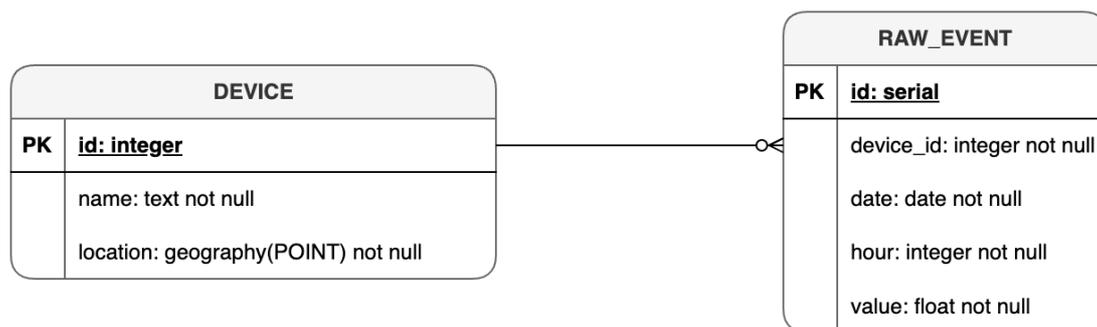


Figure 6. Entity Relationship Diagram illustrating the connection between the “device” table, containing device name and location, and the “raw_event” table, storing event records with details such as date, hour, and measured values.

4.1.1 Find the largest amount of valid data

The first step in transferring the necessary data to the additional database is migrating the device information. As previously mentioned, the list comprises a total of 1015 devices. However, many of these either lack relevant information entirely or contain data solely from sensors other than SPL. Identifying which device types generate SPL values is essential. The main database included an additional table describing all device types, providing valuable insight into how to filter the relevant entries in the devices table. The device types considered in the final table are sensors with one microphone and sensors with two microphones.

Other considerable insight was that some of the sensors had the broken location values. In the database it was stated that the latitude and longitude was valued as zero. Unfortunately, it is not possible to use such devices in the research and are also filtered out.

The SPL values in the main database are recorded with minute-level accuracy. However, for this research, this level of detail is too granular and needs to be aggregated. The approach involves calculating the median for each sensor at every recorded hour. While processing the data, some events contain a flag indicating whether the data is valid. Flagged values often lack either a production timestamp or an SPL value. Due to this, all events where the “is_valid” column does not indicate a true value are filtered out. Once the filtering is complete, the remaining data is grouped by device, date, and hour, and the median is calculated for each group. These aggregated values are then stored in the “raw_event” table in the additional database.

4.1.2 Extra cleanup for the data

Not all the uploaded data turned out to be usable. During the development process, it occurs that some of the median values of some hours were valued as zero. It means that not all the values that had “is_valid” flag stated as true, were valid. Table also included one value, where the median value was 4.5 dB, which is highly unlikely value of decibel levels of SPL on a street condition. All the data that meets these conditions will be removed from the database.

Another requirement for the “raw_event” values is sufficiency. All the data points must have at least 30 days of data for each device and hour groups. Each device must have at

least 30 distinct values of each hour, regardless of whether the values come from consecutive days. It is necessary for the imputation method development. The events that do not fulfil the rules will be excluded from the database, alongside with the other events that belong to the same sensor.

If the unusable events have been sorted out from the table, the sensor table must be updated as well. There are some sensors that currently do not have any events connected to them, all such sensors will also be removed from the table.

The final number of devices for the current project is 481, the number of hour-events is 3,709,363. For the remainder, one event contains device ID, date, hour and dB value. The number of events per device ranges from 995 to 16,084. Grouping these numbers by hours, for each device there is a minimum of 31 and maximum 687 values for one specific hour in a range from 0 to 23. The lowest dB value is 29, highest 100.

4.2 Filling the gaps

The lightweight system in use is prone to errors, leading to inconsistencies in the data. There are several reasons why data may be missing, such as devices running out of battery power, problems with radio communication or overheating, which are often beyond control. This suggests that the gaps in the dataset may be random in nature. Events may contain gaps lasting several days or even months. However, this is not the only challenge. The dataset spans multiple years, but data availability can differ substantially for each sensor. Each sensor has its own independent start and end dates, meaning their recording periods do not necessarily overlap. For example, one sensor may have data from 1st of January 2021 until 3rd of May 2021 while another only starts collecting data on 5th of May 2021. This is an important factor to consider when implementing imputation methods.

The research involves two specific types of imputation methods, one of them considers only neighbouring sensors data and another the history of a specific sensor. Both methods are computationally simple and do not require complex algorithms to calculate the estimated results. However, despite their simplicity, both methods aim to provide effective solutions for dealing with the missing data while maintaining a reasonable level of accuracy.

Both methods are designed to be lightweight in terms of computational resources. They are suitable for real-time applications where quick estimations are needed, and their algorithmic complexity remains manageable for large datasets. Nonetheless, it is important to note that while these methods are efficient, they still carry certain assumptions that may not always hold true. For instance, the nearest-imputation method assumes that neighbouring sensors provide similar data, which may not always be the case in environments with highly variable conditions. Similarly, self-imputation relies on the assumption that past trends of a sensor are reliable indicators for future behaviour.

Therefore, while both methods offer practical solutions, they also require careful evaluation and validation, particularly in terms of how well they preserve data consistency and reflect true sensor behaviour over time.

4.2.1 Self-imputation method

The self-imputation method leverages the historical data from the same sensor to estimate the missing values. This method is inspired by the hot deck imputation, in which each missing value is replaced with an observed response from a similar unit [26]. In self-imputation, the assumption is that the most suitable replacement can be found in the same hour from the previous days. By analysing the sensor's past behaviour, this method calculates a median from a selected number of previous days to fill in the gaps. This allows the system to produce an estimate based on the sensor's own trends.

Before performing self-imputation, it is crucial to determine the appropriate number of previous days needed to calculate a reliable median for estimating missing values. Since sensor data can change due to various factors, using an arbitrary fixed window size may lead to inaccurate imputations. To address this, a rolling window median approach is applied to analyse how the median stabilizes over different window sizes. Specifically, for each sensor and each hour of the day, medians are computed using increasing numbers of past days, starting from a small window and expanding it step by step. By comparing the medians across different window sizes, the point at which the median stabilizes can be identified – meaning additional days no longer significantly change the estimate. This ensures that the chosen window size is both sufficient for capturing patterns in the data and minimal enough to avoid unnecessary computational overhead. Once this analysis is complete, the determined window size can be used consistently for self-imputation,

improving the reliability of the estimated values while maintaining computational efficiency. Figure 7 is the flowchart of how the imputation map is generated.

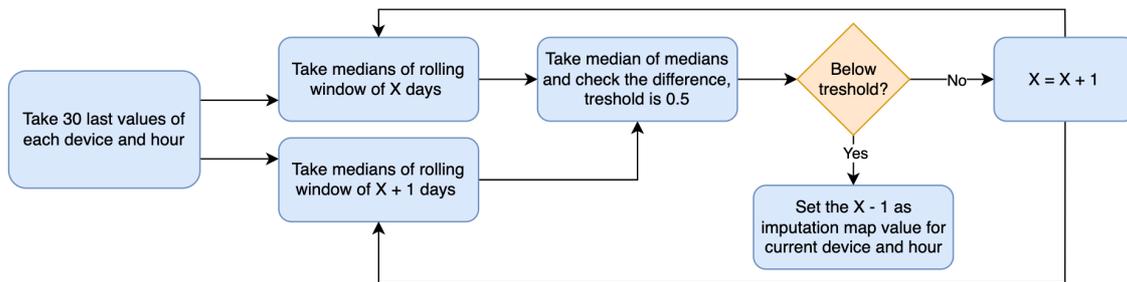


Figure 7. Flowchart illustrating the imputation process for missing data by calculating medians of rolling windows of increasing size, comparing the difference of medians against a threshold, and setting the imputation map value based on the findings.

The generated imputation map gives the overview of each device and hour how many previous days does a sensor need to efficiently estimate the missing value. Once the imputation map is generated, its values remain static and are used over time. The map may need to be regenerated whenever new sensors are added, existing sensors are modified, or there are changes in the overall network or street structure, as these factors can lead to alterations in the SPL across the urban area. The map is stored into .csv file for visual overview and to the database to impute by singular SQL query. Updated database scheme is shown on the Figure 8.

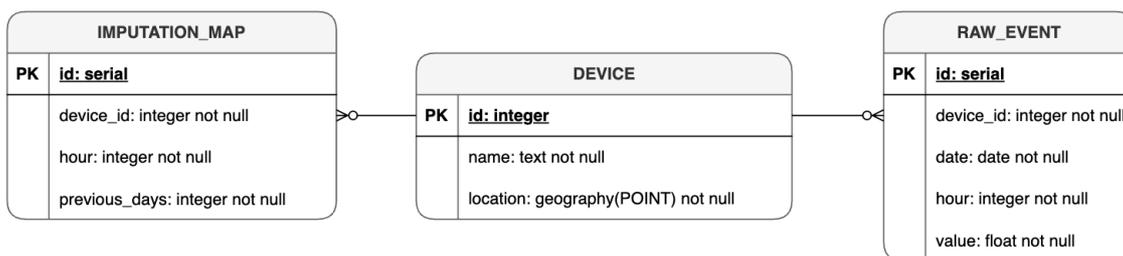


Figure 8. New ERD Scheme for the final database. ERD displaying the relationships among the “imputation_map” table, which stores imputation map values configured for each device and hour; the “device” table, containing metadata such as device name and location; and the “raw_event” table, recording event values with their respective date, hour, and related device ID.

The data imputation process based on the imputation map works by identifying missing values for a sensor at a given hour. The algorithm first retrieves the required number of previous days from the table. Based on this value, it selects the latest available data points

from the same hour across the specified number of previous days and calculates the median of these values to estimate the missing data. If there are not enough values available, the algorithm uses as many data points as it can and calculates the median from those. The flowchart is visible on Figure 9.

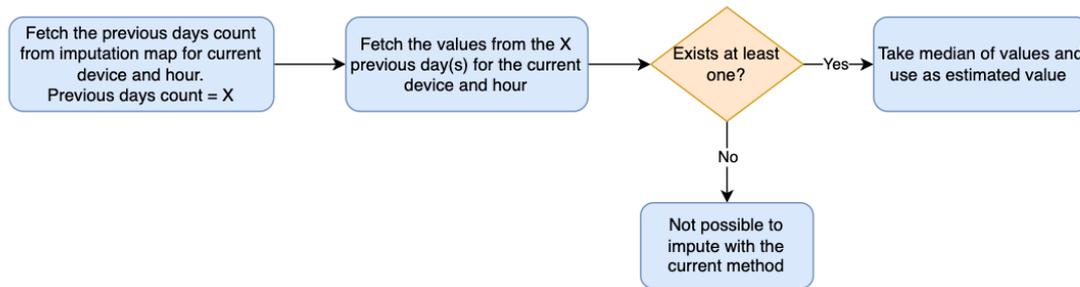


Figure 9. Flowchart demonstrating the imputation process by fetching previous days count from the imputation map, retrieving values for the current device and hour, checking if at least one value exists, and taking the median of the values as the estimated value, or indicating impossibility of imputation if no values exist.

The current method is effective when each device has at least one data point per hour that can be used for imputation. However, problems arise when there is a need to impute values for sensors that lack prior data. This situation can occur, particularly at the beginning of the dataset, where many sensors have no historical data to reference. While this may seem problematic, it does not pose a significant issue in the long run. The goal is for the sensors themselves to handle imputation once enough data has been collected. Generating the imputation map requires at least 30 days of data before any imputation can begin. The period length of 30-days is a practical choice that is necessary for the balance because it is short enough to allow researchers to quickly gather meaningful data from the sensor, but also long enough to capture variations in weather, weekends, and other relevant conditions. Therefore, although some sensors may initially lack data for imputation, this is only a temporary challenge. Once the imputation map is established, the sensors will be capable of imputing values independently, ensuring that the system becomes self-correcting over time.

4.2.2 Nearest-imputation method

Nearest-imputation method utilizes data from neighbouring sensors to estimate missing values. This method assumes that sensors located in proximity should exhibit similar data

patterns. It is inspired by the kNN imputation, which works by identifying the k nearest neighbours of an incomplete instance from among all complete instances in the dataset and using them for filling the missing value [27]. The nearest-imputation method differs from the kNN approach in that it selects neighbours based on their location within a predefined radius, rather than using a fixed number of nearest neighbours. This approach provides an alternative for filling gaps when there is insufficient historical data for a specific sensor.

The method imputes the sensors missing data based on the other sensors that exist in the 110m radius from the main sensor. In case there are sensors in this radius, the algorithm will check the values for the date and hour where in the main sensor the data is missing. If there are values for the specific timestamp for neighbouring sensors, it takes the median of all of those. The calculated value will be the estimated value for the gap. The flowchart is displayed on the Figure 10.

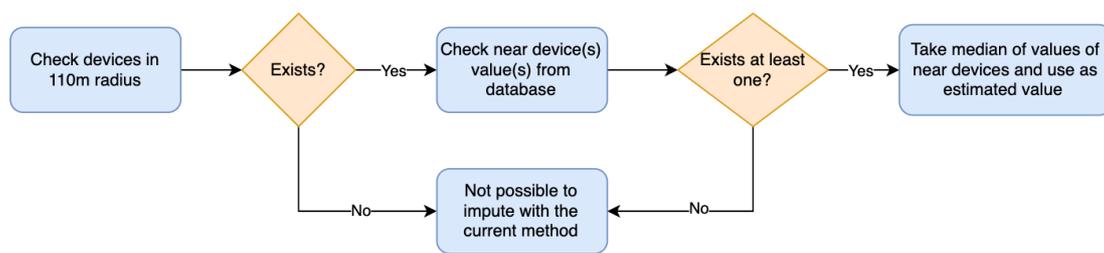


Figure 10. Flowchart illustrating the imputation process by checking for devices within a 110-meter radius, retrieving values from nearby devices, confirming the existence of at least one value, and taking the median of these values as the estimated value, or indicating impossibility of imputation if no values exist.

As mentioned earlier, the current approach is effective when no historical data is available. However, it relies on nearby sensors having the necessary data, and here two conditions must be met – first, the sensors must exist within the specified radius, and secondly, data must be available at the given point in time where imputation is carried out. The radius is set to 110 meters because most sensors are placed approximately 100 meters apart, with an additional 10-meter threshold to account for slight variations. Since many sensors were removed during data cleaning, some remaining sensors do not have neighbours within this radius. To avoid overestimating values, the radius is not increased, as doing so could incorporate too many sensors into the estimation process, potentially leading to less accurate imputation values.

4.3 Developing the web application

For visualization, several ideas were explored regarding the best approach to present the data. Initially, the plan was to impute all the data first and then use the imputed values for visualization purposes. However, this approach would have resulted in an overwhelming amount of data, which could have negatively impacted performance and usability. To mitigate this issue, imputation is instead performed on-demand, only when required, which helps optimize resource usage and ensures that the system remains efficient.

The final project is a full-stack web application with an architecture that consists of a well-defined database layer that includes only the essential “device” and “raw_event” tables, which store the core data. A server handles communication between the front-end and the database, while also managing the imputation process. The client-side application focuses solely on displaying the appropriate imputed values on the map in real-time, ensuring a seamless and responsive user experience. This structure allows for efficient data management and visualization without overburdening the system with unnecessary computations.

4.3.1 Server-side application

The backend of the visualization application has been built using NestJS. It was chosen due to its simplicity and the speed it offers, making it well-suited for the relatively low-complexity requirements of the backend [28]. There are two main responsibilities for the backend – ensuring fast communication between the database and the frontend and performing data imputation. At present, the application utilizes a lightweight version of the self-imputation method, where only the most recent value is used for estimating missing data. This approach helps maintain efficiency while still providing estimates for the missing values in the system. The server-side application also utilizes the TypeORM library to streamline communication with the database [29].

The backend operates as a REST application with a single controller. This controller is responsible for fetching all the devices. Along with retrieving the devices, the controller also performs a join with the events table for the specified date and hour. Additionally, it handles the imputation for each device, also for the devices that already have the real value. This approach allows the application to display both the real and imputed values,

enabling users to compare both. Visual representation of the request is displayed on Figure 11.

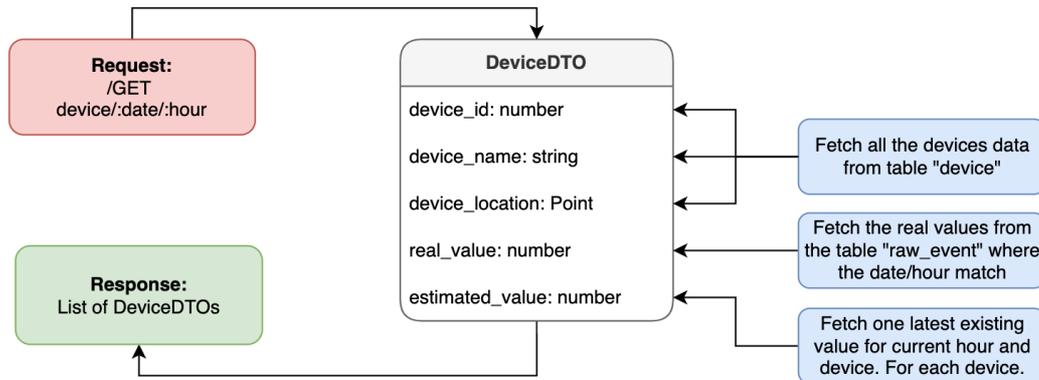


Figure 11. Diagram illustrating the process of handling a GET request for device data on specific date/hour by fetching device information from the device table, real values from the “raw_event” table, the latest existing value for each device, and assembling the response as a list of DeviceDTO (Device Data Transfer Object) objects containing device ID, name, location, real value, and estimated value.

4.3.2 Client-side application

The frontend of the application is developed using TypeScript with the React framework. To fetch data from the backend, the TanStack Query library is utilized, providing an efficient way to handle data fetching and caching [30]. The data fetched from the backend is then cast to a GeoJSON object for compatibility with the “vis.gl/react-maplibre” library, which is used to display the sensor locations and their corresponding values on the map. The current frontend components are inspired by the official VisGL React-Map-Gl heatmap example, which is available in the React-Map-Gl GitHub repository in [31].

The map visualization currently highlights the sensor locations, with colour coding based on the estimated value when the real value is unavailable. However, users also have the option to view both the real and estimated values. The colour codes used in the map visualization are derived from the Strategic Guide to a Noise Map provided by the Ministry of Climate [18]. This guide outlines the colour schemes that correspond to different noise levels, ensuring that the visualization accurately reflects the environmental noise data.

The frontend is designed exclusively for visualizing data based on a selected hour and date. It displays all available sensor data for a specific timestamp simultaneously. Time

selection is facilitated through an interactive slider, enabling users to adjust the timestamp and observe the corresponding data points dynamically. This approach provides an intuitive way to explore temporal variations in noise levels while maintaining a clear and efficient visualization of the dataset.

5 Results

The results analysis focuses on two key aspects – the imputation of missing sensor data and the creation of dynamic point-based map to visualize noise distribution in Tallinn. The goal is to evaluate how well different imputation methods restore missing values and assess the effectiveness of the map in representing SPL across the city.

Since imputation methods can be time-consuming, especially when many values need to be estimated in real-time, this research focuses on a selected test set for evaluating imputation methods while using a simplified self-imputation approach for data visualization. The test set includes data from sensor 203C, positioned on Kaarli Boulevard at coordinates $59^{\circ}25'55.9''\text{N}$ $24^{\circ}44'17.8''\text{E}$. The timeframe of the calculations is 7th September 2022 at 18 to 8th of September at 17. The sensor and timeframe were selected because, during this 24-hour period, the sensor primarily records its own event values, making it suitable for comparison. The sensor provides data for 21 out of the 24 hours, which is sufficient for drawing meaningful conclusions. Additionally, the sensor does have 3 nearby sensors that also mostly have values on that timeframe and the main sensor have enough historical data to successfully apply both imputation methods.

5.1 Self-imputation method

The self-imputation method employed in this study involves two main calculations, with the first being the creation of an imputation map based on the 30 most recent events of each sensor for each hour, chosen for their stability and reliability, regardless of whether they fit within a 30-day timeframe. This approach was developed through trial and error, where it was found that values taken from the beginning of the dataset, certain sensor hours may have needed more than 30 values for stabilizing the medians. Using the values from the end avoids this problem and all the imputation map values which state how many of previous days value does one sensor has to use in certain hour remains less than 30.

Upon deeper investigation of the imputation map, it is observed that values stabilize relatively quickly. As shown in Figure 12, most of the sensor hours stabilize within one

day. This means that for most missing hours, the best estimation can be made by simply using the last existing value, which serves as the most accurate approximation for those hours.

device_id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
679	1	1	1	1	1	1	1	1	4	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
56	1	1	1	1	1	1	1	1	1	1	1	1	1	1	6	1	1	2	1	1	1	1	1	1
96	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	1
243	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	1	1	1	1	1	1	1	1
446	1	1	1	1	1	1	1	1	1	1	1	3	1	1	1	1	1	1	1	1	1	1	1	1
483	1	1	1	3	1	1	1	2	4	1	1	2	1	1	1	2	2	1	1	1	1	1	1	1
671	1	4	1	2	1	1	1	2	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1
437	1	3	1	2	1	7	1	1	1	2	1	2	2	3	2	1	2	1	1	1	2	1	1	1
235	1	1	1	1	1	1	1	1	4	1	1	1	1	1	1	2	4	1	1	1	1	1	1	1
531	1	1	1	1	1	1	1	1	2	3	1	4	1	1	1	2	1	2	1	1	1	1	1	1
504	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
537	1	2	1	1	1	1	1	1	4	2	1	1	1	1	1	1	1	1	4	1	1	1	1	1
312	1	1	1	1	1	2	1	1	12	1	1	1	1	1	2	1	1	1	2	1	1	1	1	1
15	2	1	2	2	2	2	4	2	2	1	1	1	1	3	1	2	1	2	4	6	4	1	2	7
299	7	1	1	2	1	7	1	3	7	1	1	4	1	1	1	1	1	1	1	4	1	1	1	1
252	1	1	1	1	7	2	1	1	1	2	1	1	1	2	1	1	2	1	1	1	1	1	1	1
320	1	2	2	2	1	2	1	1	2	1	1	1	1	1	1	1	1	4	1	1	1	1	2	1
328	1	1	1	1	2	2	1	1	2	1	1	1	1	2	1	1	1	1	1	2	1	3	1	1
577	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
301	1	4	1	1	10	1	1	6	1	6	1	2	2	2	6	1	3	2	2	3	4	1	1	1
242	1	1	1	1	1	2	1	1	4	1	1	4	2	1	1	1	1	1	1	1	1	1	1	1
561	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
316	2	1	2	1	1	1	2	1	3	1	1	1	1	1	5	1	2	1	1	1	1	1	1	2
69	1	1	1	1	1	1	1	1	4	4	1	1	1	1	1	1	1	1	1	1	1	1	1	2

Figure 12. Table showing 25 rows on the imputation map, which consists of device IDs, hours and the previous days count which indicates the number of days needed for the best estimation for missing data for device/hour. The colours give the visual overview of the amount of different number of days needed.

There are 481 devices in total, which means that the imputation map has all the values for each hour for each device. On the Figure 12, it only shows the first 25 rows, but the overall pattern looks similar for all the sensors. The maximum number of previous days to stabilize the value is 22, and it only exists on a singular device. All the values for previous days are 11544 in total. For the validation, 481 devices times 24 hours gives the same number. 9115 device/hour pairs have the value 1 for the previous days. This indicates that for approximately 80% of the device/hour pairs it is possible to get the best estimate by just taking the last existing value prior to the missing data. On Figure 13, it is shown the hourly distribution of the previous days where the value equals 1. They all are around 80% as the total value, which indicates that all the hours are quite equally distributed as well. Only outliers are 7 and 8 o'clock, where the percentage of 1 previous day is around

70%. This may be due to rush hour patterns in traffic, where the weekends and weekdays may have different SPL values at these times.

hour	percent_of_prev_day_1
0	79
1	75
2	78
3	77
4	79
5	78
6	81
7	69
8	68
9	77
10	79
11	80
12	79
13	78
14	77
15	81
16	79
17	78
18	79
19	79
20	81
21	85
22	82
23	84

Figure 13. Hourly distribution where previous day value equals 1. This figure shows the hourly distribution of data indicating the percentage of instances where the previous day count equals 1 for each hour. It provides insight into how often a single previous day is used for estimating missing data across different hours.

On the Figure 14, the test device’s previous days counts for each hour are displayed. The values are used for applying the self-imputation on the test set. The device has relatively stable values, according to the map, having only 1 or 2 previous days needed for the estimation for missing hours.

device_id	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
550	1	1	1	1	1	1	1	1	1	1	1	1	2	1	1	2	1	1	2	2	1	1	1	2

Figure 14. The imputation map for the device used in the test set, where device_id is 550, showing the number of previous days required for each hour's estimation, with colour coding to indicate different hour counts.

Using this imputation map, the self-imputed values are generated. The estimated SPL values compared to the real values are displayed on the Figure 15.

main_id	main_name	current_date	current_hour	real_value	estimated_value	diff	past_values_count
550	203C	2022-09-07	18	57	56.75	0.25	2
550	203C	2022-09-07	19	54.5	55.5	-1	2
550	203C	2022-09-07	20	55	56	-1	1
550	203C	2022-09-07	21	53	56	-3	1
550	203C	2022-09-07	22	53	55	-2	1
550	203C	2022-09-07	23	53.5	52	1.5	2
550	203C	2022-09-08	0	49	49	0	1
550	203C	2022-09-08	1	48	48	0	1
550	203C	2022-09-08	2	48	48	0	1
550	203C	2022-09-08	3	47	48	-1	1
550	203C	2022-09-08	4	47	50	-3	1
550	203C	2022-09-08	5	49	49	0	1
550	203C	2022-09-08	6	51	52	-1	1
550	203C	2022-09-08	7	55	56	-1	1
550	203C	2022-09-08	8	57	59	-2	1
550	203C	2022-09-08	9	59	59	0	1
550	203C	2022-09-08	10	58	59	-1	1
550	203C	2022-09-08	12	57	57.5	-0.5	2
550	203C	2022-09-08	13	57	55	2	1
550	203C	2022-09-08	14	55	58	-3	1
550	203C	2022-09-08	17	60.5	57	3.5	1

Figure 15. The example of the self-imputation method. The image shows the data from timeframe of 7th September 2022 at 18 to 8th of September at 17 on the device 503C with the real values and estimated values based on the self-imputation method, alongside with the error on each timestamp and the number of past values used for calculating the estimation.

Figure 15 shows the self-imputed values compared to the real values found in the test set timeframe for sensor 203C. To compare the Figure 14 and Figure 15, it is possible to see that the past values count corresponds to the imputation map. For the reminder, it may not always be like that – if there are not enough past values for current sensor’s current hour it does take as much values as possible. It leads to the possibility of having less values than required in the map. In current test set, it is not the case. Checking the values that have used 2 past values, on the hours 18, 19, 23 and 12, it is clearly visible, that the values are relatively accurate. The maximum error of the previously mentioned hours is 1.5 dB, minimum 0.25 dB. All the values for the rest also remain in the borders of ± 3.5 and there are 5 hours, where the real and estimated values are the same, 6 if rounding the value of 0.25 dB.

In conclusion, for the current test set, it is possible to say that the imputation method is relevant and does give accurate results. Having the imputation map which is calculated based on the stabilization of the data does give positive outcomes not only for the better estimates, but also for the performance. The less data that needs to be stored in memory and processed, the lower the computational power required to draw conclusions and get best estimates.

5.2 Nearest-imputation method

The nearest-imputation method utilizes the neighbours' data instead of the historical data of the current sensor. It takes all the devices that are in the 110m radius, checks their values for the specific moment and uses the median values of those to get the best estimate. Problems with current approach is that in case there are no values for the sensor that needs to be imputed, it is relatively likely that there are also no values for the nearby sensors as well, which may be because of the solar based battery charging system. Second problem is the possibility of the sensors not having the nearby sensors in the 110m radius at all. As the test set being chosen the way that it does meets these two conditions, it is possible to generate the estimated values based on the nearest imputation methods. On the Figure 16, the estimates and differences with the real values are listed.

main_id	main_name	current_date	current_hour	real_value	estimated_value	diff	near_values_count
550	203C	2022-09-07	18	57	56	1	3
550	203C	2022-09-07	19	54.5	54	0.5	3
550	203C	2022-09-07	20	55	54	1	3
550	203C	2022-09-07	21	53	52	1	3
550	203C	2022-09-07	22	53	52	1	3
550	203C	2022-09-07	23	53.5	52	1.5	3
550	203C	2022-09-08	0	49	49	0	3
550	203C	2022-09-08	1	48	48.5	-0.5	2
550	203C	2022-09-08	2	48	46	2	3
550	203C	2022-09-08	3	47	45.5	1.5	3
550	203C	2022-09-08	4	47	46	1	3
550	203C	2022-09-08	5	49	50	-1	3
550	203C	2022-09-08	6	51	51	0	3
550	203C	2022-09-08	7	55	56.5	-1.5	3
550	203C	2022-09-08	8	57	57	0	3
550	203C	2022-09-08	9	59	58	1	3
550	203C	2022-09-08	10	58	57	1	3
550	203C	2022-09-08	12	57	55.5	1.5	3
550	203C	2022-09-08	13	57	57	0	3
550	203C	2022-09-08	14	55	58.5	-3.5	3
550	203C	2022-09-08	17	60.5	58	2.5	3

Figure 16. The example of the nearest-imputation method. The image shows the data from timeframe of 7th September 2022 at 18 to 8th of September at 17 on the device 503C with the real values and estimated values based on the nearest-imputation method, alongside with the error on each timestamp and the number of near values used for calculating the estimation.

It is visible on the Figure 16 that most of the values of our test set do have the neighbours, and they do have the values on the timeframe. It is worth mentioning that only one hour on the test set, 1 o'clock on the 8th of September 2022, have 2 neighbouring values instead of 3 in total. Investigating and comparing this specific hour and its difference with other values provides no reason to suggest that using the two nearest values and their median, instead of three, would be invalid in this current test set. Overall results are relatively accurate. The maximum error of the estimates is 3.5 dB, existing for only single hour, minimum error is 0, existing for 4 different hours. This means the current test set demonstrates a relatively low level of error, with most estimates being quite accurate.

5.3 Comparison of the methods

The self- and nearest-imputation methods give similar results according to the computations made on the test set. Both have a maximum error of 3.5 dB and contain many estimations that match the real values exactly. It can be concluded that both methods yield informative and accurate estimations, but as mentioned before, there are several conditions under which one or another may work better. For the self-imputation method, it is necessary to have historical values to calculate the estimation, moreover, there must be at least 30 values for each hour creating the imputation map, it is not good method for freshly applied devices that yet have too little number of values. In that case, nearest-imputation method would give the estimates from the first hour, if there are nearest sensors with values. The nearest-imputation method may also correlate better with the rush hour pattern, as the self-imputation method does not account for whether the missing hour falls on a weekend or a weekday. This means that estimates may be invalid in 8-9 and 16-17 o'clock if weekend missing hours are imputed using weekday data, or vice versa. On the other hand, there is a higher chance that using the nearest-imputation method, the estimated value would remain null since there are either no sensors nearby or no values in nearby sensors. For self-imputation method, similar problem only occurs in the earlier stages, if there is at least one existing value for each sensor, all the sensors can have future estimates based on that. Table 1 includes the overview of advantages and disadvantages of both methods.

Table 1. The advantages and disadvantages of the self- and nearest-imputation methods.

Self-imputation method	Nearest-imputation method
– Not suitable for newly added sensors, since it requires at least 30 values for each hour.	+ Suits well for new sensors, does not need historical data for estimations.
– Does not differentiate between weekdays and weekends.	+ May reflect time patterns better (e.g. rush hours).
+ If there is at least one historical value for the missing hour, the estimation is possible.	– If there are no nearby sensors or the values of nearby sensors, the imputation would not be possible.

5.4 Visualization

Current visualization applies the lightweight version of self-imputation method, which means that instead of the values on the imputation map, every sensor only uses the last existing value of the specific sensor's specific hour. It is justified by the analyse of the imputation map in chapter 5.1, that stated that 80% of the device/hour pairs have 1 as the previous days' count. The benefits of using the lightweight version are better speed at processing, getting the estimated values for all the sensors of a specific timestamp in milliseconds. The nearest-imputation method is excluded from the visualization primarily also due to performance concerns, as well as the issue of missing data from nearby sensors previously discussed.

The application itself consists of one user input, which gives user the opportunity to choose the moment when do they want to investigate the results. The user gets the output where it does have the map of Tallinn with all the sensor locations and values of each sensor on a chosen moment. All the values displayed on the map are rounded to the closest integer. On Figure 17, the earliest possible map is shown.

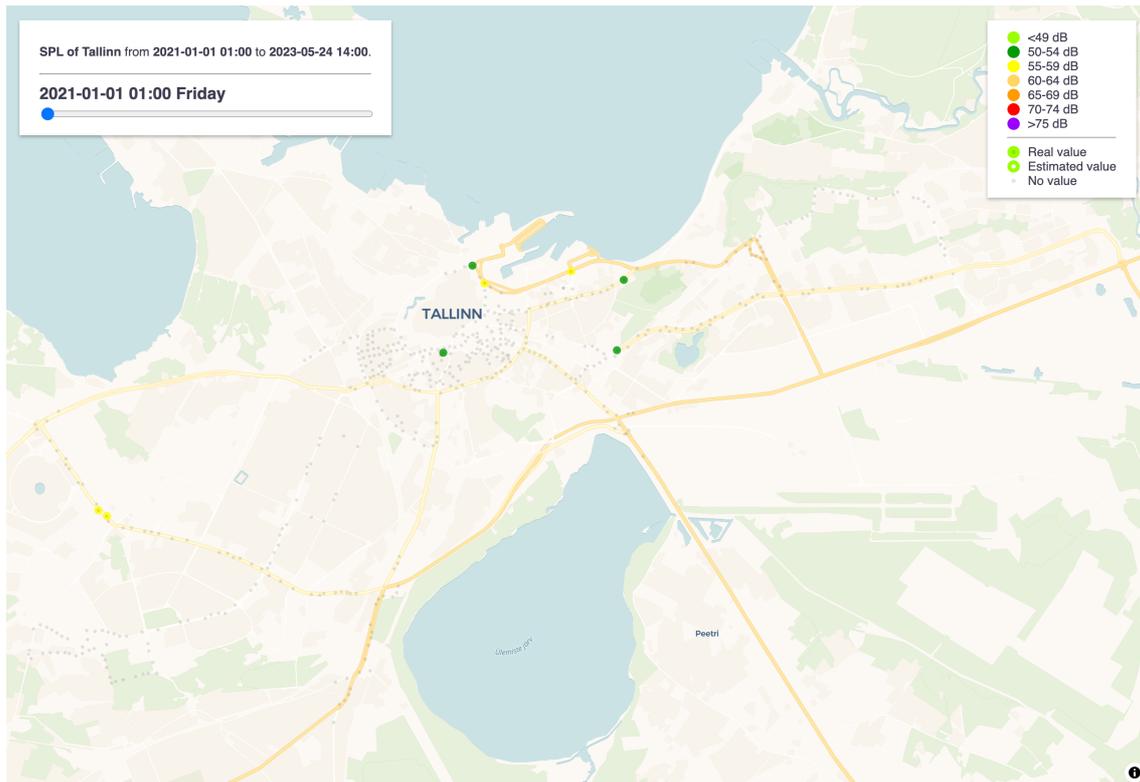


Figure 17. The earliest possible map of SPL of Tallinn created with the application. It consists of 8 different real values and 0 estimated values, since there is no historical data at that point. On the image, there are 4 green and 4 yellow dots indicating the existing values of the sensors for chosen moment.

The map consists of the devices positions as well as the existing values. Since it is the earliest possible visualization, there are no estimations yet and only 8 sensors have values at that moment. Grey dots indicate the sensors that will eventually have values during the full timeframe from 1st of January 2021 at 1 to 24th of May 2023 at 14. Larger dots with the colours listed on the legend show the SPL of the moment. On the Figure 17, there is no big picture formed, but on the Figure 18, where the latest possible map of SPL is displayed, all the sensors have either real or estimated value.

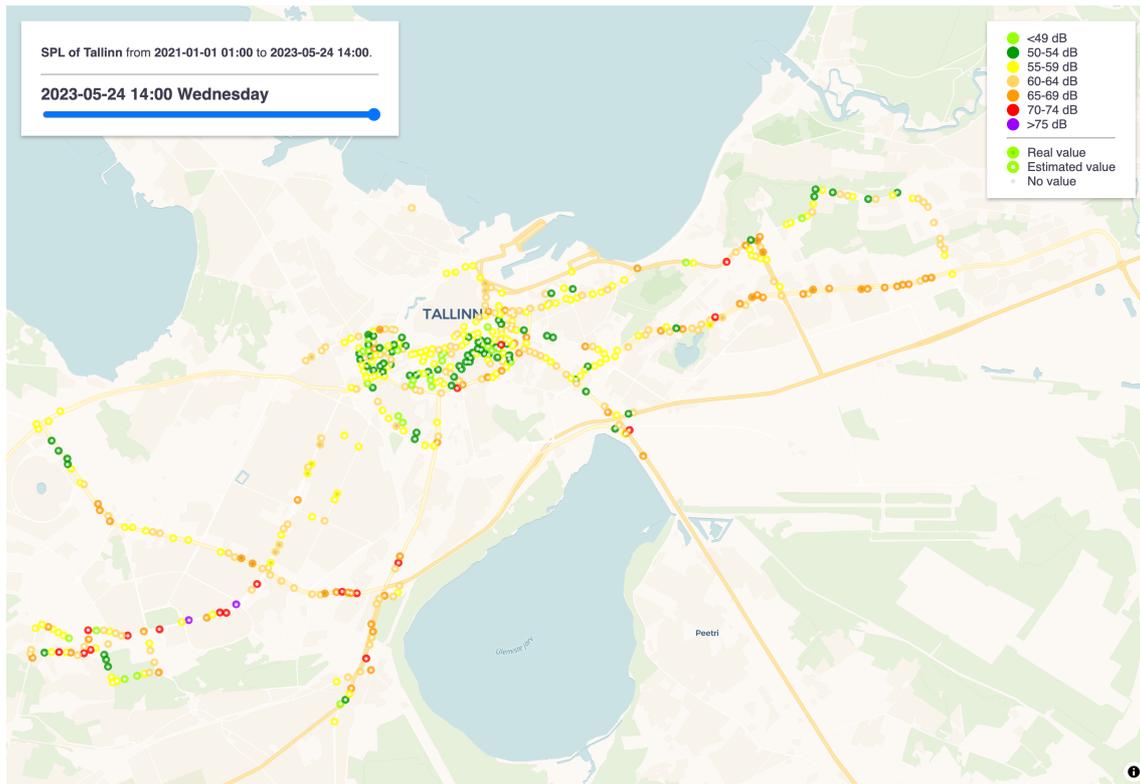


Figure 18. The latest possible map of SPL of Tallinn created with the application. There are no sensors without values, since all the sensors do have at least 1 historical value to make an estimate with. The colourful dots indicate each sensor values for chosen moment, white small dots in the centre indicate that the indicator colour is based on estimated value, grey dot in the middle shows it is based on the real value.

On the latest possible map, there are many sensors indicated the white dot, which means that the value used for the indicator colour is estimated. Where the grey dot remained, there is the real value used for the indicator.

Comparing the values of the day and night, images from the same timeframe are used as comparing the self- and nearest-imputation values, which is 7th September 2022 at 18 to 8th of September at 17. On the Figure 19, the map of 8th of September at 3 o'clock is displayed.

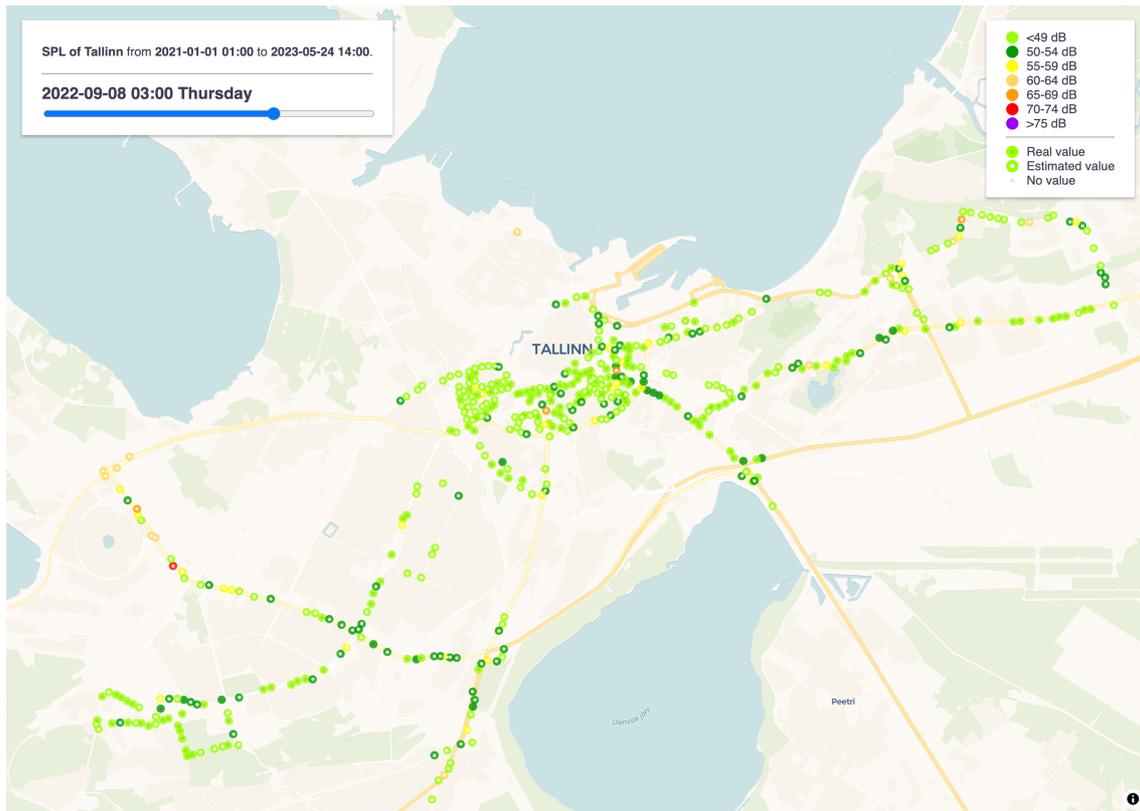


Figure 19. The SPL map of 8th of September 2022 at 3 o'clock. Most of the sensors are coloured green, which indicates the low overall SPL.

It is clearly visible that during the night, SPL values are green, which means they are mostly lower than 49 dB. It justifies the accuracy of the measures, since it is very likely that during the night the traffic is calmer and there are not many noise producers on the streets. On the other hand, passing 12 hours from the moment and observing the 15 o'clock of the same day, the colours on the image look quite different, as seen on the Figure 20.

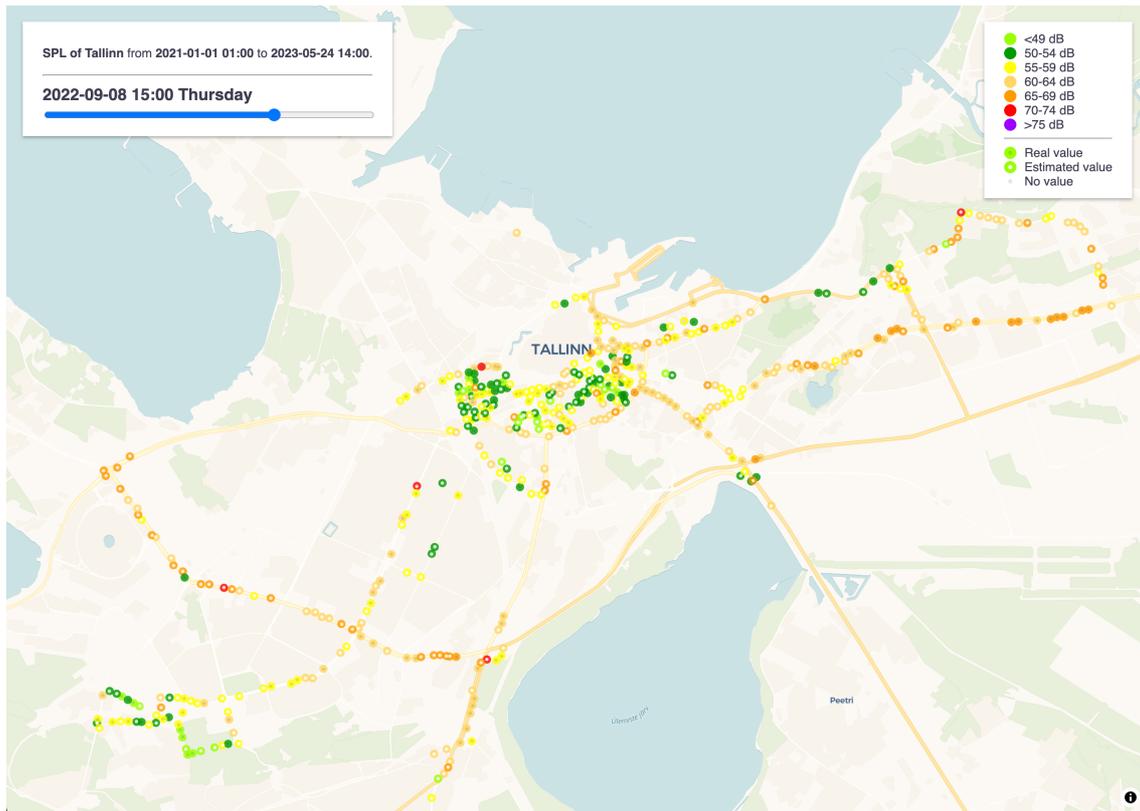


Figure 20. The SPL map of 8th of September 2022 at 15 o'clock. Most of the sensors are yellow/orange, which indicates relatively noisy environment, between 60-69 decibels.

Based on the image, it can be stated that the overall visualization appears predominantly orange, indicating that the SPL mostly range between 65 and 69 dB. While there are some exceptions, the map effectively reflects the general trend of SPL values for the current hour.

There are two moments captured on the Figure 21 and Figure 22. The test sensor 203C is displayed on the 8th of September 2022 at 14 o'clock and 7th of September 2022 at 18 o'clock with the real and estimated values. The same values are visible on the Figure 15 where the self-imputation methods' values are displayed. When analysing the values for 8th of September 2022 at 14, they remain consistent across two different calculation methods – one performed via an SQL query (real values with estimated values based on the self-imputation method) and the other through the application backend (real plus estimated data based on the simplified self-imputation method, using only last existing value) upon request. This consistency supports the correctness of the algorithms and indicates that different calculation methods do not produce different results.

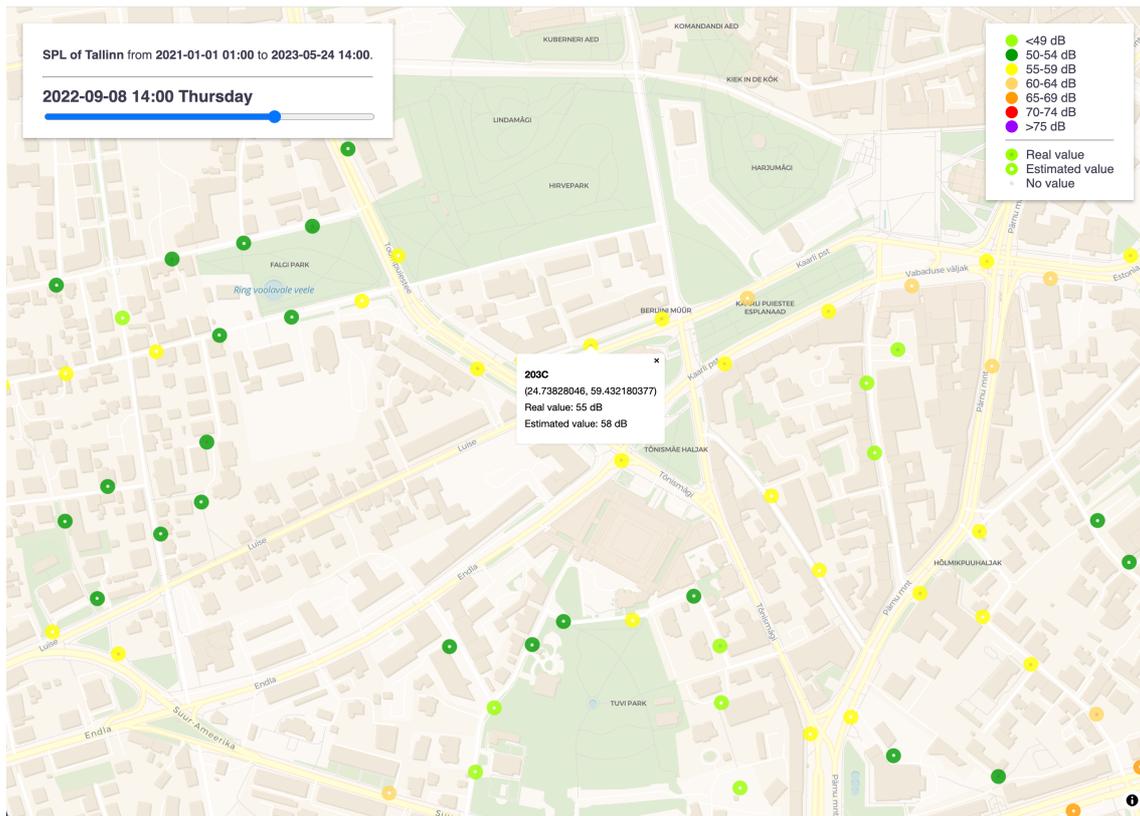


Figure 21. The real and estimated values of test device 203C at 8th of September 2022 for 14 o'clock, which are necessary for the comparison with the previously made self-imputation method calculations.

On the Figure 22, there are values displayed for 7th of September 2022 for 18 o'clock. As it is possible to notice, the estimated values are not really the same with the one stated on the Figure 15.

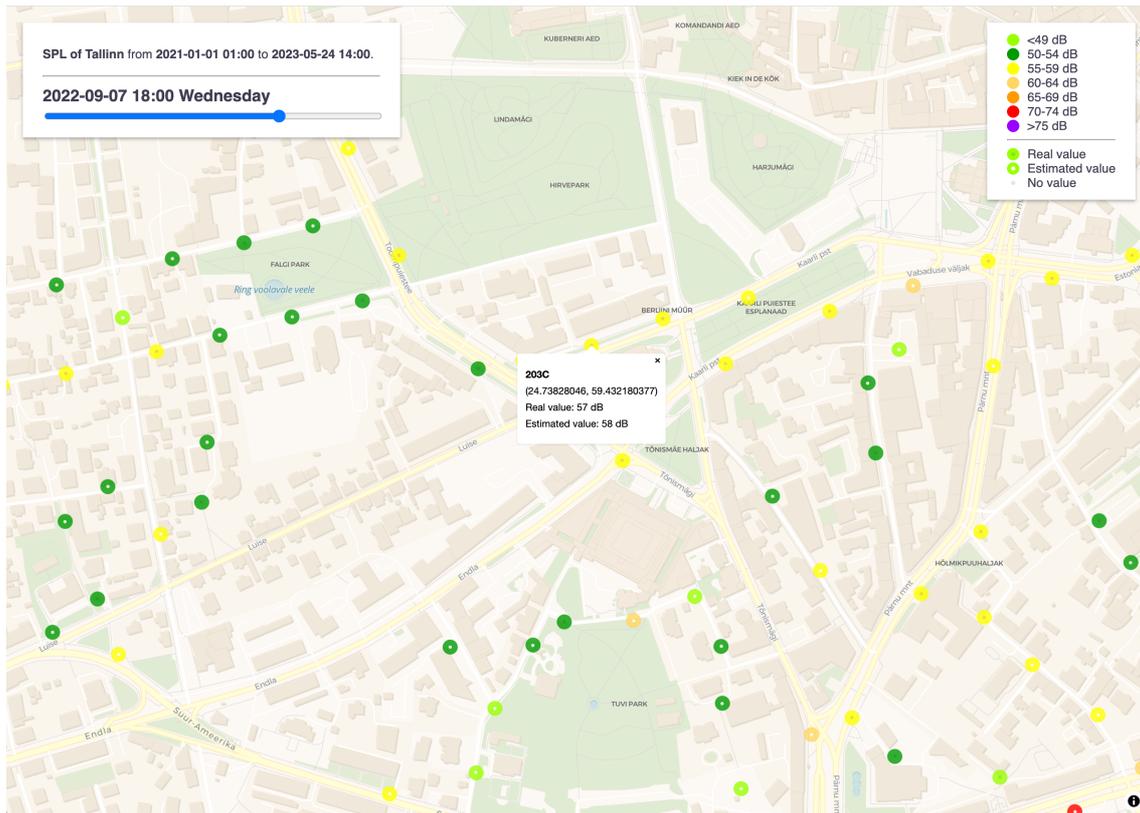


Figure 22. The real and estimated values of test device 203C at 7th of September 2022 for 18 o'clock, which are necessary for the comparison with the previously made self-imputation method calculations.

The value created with the SQL query would round up to the exact same value as the real one, but in the visualization part, it does show that the estimated value is 1 dB higher. The difference comes from the fact that the visualization uses lightweight version of the self-imputation method which makes the estimation on 1 last possible value, but on the Figure 15, the past values count reveals that for the test set calculation, it used 2 last values. The insight tells that the imputation map can help with the accuracy of the imputation.

6 Summary and Conclusions

This thesis focuses on improving the way noise data is collected and managed from sensors placed around Tallinn. The sensors sometimes fail to capture data, creating gaps that need to be filled accurately for effective analysis. To address this, two different methods were developed and tested - self-imputation method, which relies on the sensor's own historical data to fill in the missing information, and nearest-imputation method, which uses data from nearby sensors to estimate and fill the gaps. Both methods were analysed to determine their accuracy and reliability in providing complete datasets.

The thesis also involves creating a visual map to represent noise levels across the city. This visualization is designed to help city planners and authorities quickly identify noisy areas and make decisions to address these issues effectively. By doing so, the goal is to improve the acoustical environment of the city, making it quieter and more pleasant for residents and visitors.

Future work might be to understand how each method performs and when one might be more useful than the other, to find the best possible estimation for each hour that will be missing on the specific sensor on the street. If the best estimation is generated on the fly, the sensors could give out either the real or the estimated data directly and the application for the visualization can only deal with the data from the sensors, without having to create estimates on the request. Ideally the estimates will always be in the initial dataset alongside with the real data.

By continuing to refine these methods and tools, the city's noise management can become more data-driven and responsive. This will help in making informed decisions to reduce noise pollution and enhance the overall quality of life in Tallinn.

References

- [1] A. Muzet, “Environmental noise, sleep and health,” *Sleep Med. Rev.*, vol. 11, no. 2, pp. 135–142, Apr. 2007, doi: 10.1016/j.smr.2006.09.001.
- [2] “Environmental noise guidelines for the European Region.” Accessed: Apr. 13, 2025. [Online]. Available: <https://www.who.int/europe/publications/i/item/9789289053563>
- [3] *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 - Declaration by the Commission in the Conciliation Committee on the Directive relating to the assessment and management of environmental noise*, vol. 189. 2002. Accessed: Apr. 11, 2025. [Online]. Available: <http://data.europa.eu/eli/dir/2002/49/oj/eng>
- [4] “Välisõhu mürakaardi, strateegilise müra ja müravähendamise tegevuskava sisu kohta esitatavad tehnilised nõuded ja koostamise kord–Riigi Teataja.” Accessed: Apr. 11, 2025. [Online]. Available: <https://www.riigiteataja.ee/akt/131122021017>
- [5] I.-M. E. | ERR, “Excessive traffic noise holding back Tallinn’s real estate development,” ERR. Accessed: Apr. 02, 2025. [Online]. Available: <https://news.err.ee/1609602884/excessive-traffic-noise-holding-back-tallinn-s-real-estate-development>
- [6] T. Veber, T. Tamm, M. Ründva, H. K. Kriit, A. Pyko, and H. Orru, “Health impact assessment of transportation noise in two Estonian cities,” *Environ. Res.*, vol. 204, p. 112319, Mar. 2022, doi: 10.1016/j.envres.2021.112319.
- [7] “Tallinn adopts noise reduction action plan | Tallinn.” Accessed: Apr. 02, 2025. [Online]. Available: <https://www.tallinn.ee/en/news/tallinn-adopts-noise-reduction-action-plan>
- [8] “History of noise,” Mike Goldsmith Acoustics, science writing, environment. Accessed: Apr. 13, 2025. [Online]. Available: <https://mikegoldsmith.weebly.com/history-of-noise.html>
- [9] M. S. Hammer, T. K. Swinburn, and R. L. Neitzel, “Environmental Noise Pollution in the United States: Developing an Effective Public Health Response,” *Environ. Health Perspect.*, vol. 122, no. 2, pp. 115–119, Feb. 2014, doi: 10.1289/ehp.1307272.
- [10] G. Dove, C. Mydlarz, J. P. Bello, and O. Nov, “Sounds of New York city,” *Interactions*, vol. 29, no. 3, pp. 32–35, May 2022, doi: 10.1145/3527726.
- [11] J. Camps, “Barcelona noise monitoring network,” 2015.
- [12] D. Bonet-Solà, C. Martínez-Suquía, R. M. Alsina-Pagès, and P. Bergadà, “The Soundscape of the COVID-19 Lockdown: Barcelona Noise Monitoring Network Case Study,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 11, p. 5799, May 2021, doi: 10.3390/ijerph18115799.
- [13] A. Can *et al.*, “Framework for urban sound assessment at the city scale based on citizen action, with the smartphone application NoiseCapture as a lever for participation,” *Noise Mapp.*, vol. 10, no. 1, p. 20220166, Jun. 2023, doi: 10.1515/noise-2022-0166.
- [14] J. Shao and B. Zhong, “Last observation carry-forward and last observation analysis,” *Stat. Med.*, vol. 22, no. 15, pp. 2429–2441, Aug. 2003, doi: 10.1002/sim.1519.
- [15] M. P. Davis, “Missing Data and the Last Observation Carried Forward,” *J. Pain Symptom Manage.*, vol. 67, no. 6, pp. e921–e922, Jun. 2024, doi: 10.1016/j.jpainsymman.2024.02.019.
- [16] Y. Zhang and P. J. Thorburn, “Handling missing data in near real-time environmental monitoring: A system and a review of selected methods,” *Future Gener. Comput. Syst.*, vol. 128, pp. 63–72, Mar. 2022, doi: 10.1016/j.future.2021.09.033.

- [17] L. Miasayedava, “Open Environmental Data Assimilation Under Unknown Uncertainty and Multiple Spatio-Temporal Scales,” TalTech Press, 2024. doi: 10.23658/TALTECH.13/2024.
- [18] M. Ründva, “Strateegilised mürakaardid CNOSSOS-EU arvutusmeetodi juhendmaterjal.” Mar. 2020. [Online]. Available: https://kliimaministeerium.ee/sites/default/files/documents/2021-07/Juhend_%20Strateegilised%20m%C3%BCrakaardid.%20Keskonna%C3%B5iguse%20Keskus.pdf
- [19] “Tallinna linna mürakaart 2022 | Tallinn.” Accessed: Dec. 14, 2024. [Online]. Available: <https://www.tallinn.ee/et/keskkond/tallinna-linna-murakaart-2022>
- [20] “Tallinna linna välisõhu strateegiline mürakaart.” [Online]. Available: <https://www.tallinn.ee/et/media/482178>
- [21] D. Beran, K. Jedlička, K. Kumar, S. Popelka, and J. Stoter, “The Third Dimension in Noise Visualization – a Design of New Methods for Continuous Phenomenon Visualization,” *Cartogr. J.*, vol. 59, no. 1, pp. 1–17, Jan. 2022, doi: 10.1080/00087041.2021.1889450.
- [22] “Tõhus koostöö: 900 anduri võrk kogub andmeid linnaõhu ja liikluse kohta | TalTech.” Accessed: Dec. 17, 2024. [Online]. Available: <https://taltech.ee/uudised/tohus-koostoo-900-anduri-vork-kogub-andmeid-linnaohu-ja-liikluse-kohta>
- [23] J. Kaugerand and J.-S. Preden, *Mediated interactions for collection and exchange of situational information in smart environments =: Vahendatud interaktsioonid olukorrateadlikkuse informatsiooni kogumiseks ja vahetamiseks arukates keskkondades*. in Tallinn University of Technology. Doctoral thesis = Tallinna Tehnikaülikool. Doktoritöö, no. 28/2020. Tallinn: TalTech Press, 2020.
- [24] “DigitalOcean.” Accessed: Mar. 27, 2025. [Online]. Available: <https://www.digitalocean.com/products/managed-databases-postgresql>
- [25] “PostGIS,” PostGIS. Accessed: Mar. 21, 2025. [Online]. Available: <https://postgis.net/>
- [26] R. R. Andridge and R. J. A. Little, “A Review of Hot Deck Imputation for Survey Non-response,” *Int. Stat. Rev.*, vol. 78, no. 1, pp. 40–64, Apr. 2010, doi: 10.1111/j.1751-5823.2010.00103.x.
- [27] S. Zhang, “Nearest neighbor selection for iteratively kNN imputation,” *J. Syst. Softw.*, vol. 85, no. 11, pp. 2541–2552, Nov. 2012, doi: 10.1016/j.jss.2012.05.073.
- [28] “Documentation | NestJS - A progressive Node.js framework,” Documentation | NestJS - A progressive Node.js framework. Accessed: Mar. 27, 2025. [Online]. Available: <https://docs.nestjs.com>
- [29] “TypeORM.” Accessed: Mar. 27, 2025. [Online]. Available: <https://typeorm.io/>
- [30] “Overview | TanStack Query React Docs.” Accessed: Mar. 28, 2025. [Online]. Available: <https://tanstack.com/query/latest/docs/framework/react/overview>
- [31] “react-map-gl/examples/heatmap at 7.0-release · visgl/react-map-gl,” GitHub. Accessed: Mar. 28, 2025. [Online]. Available: <https://github.com/visgl/react-map-gl/tree/7.0-release/examples/heatmap>

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis¹

I, Viktoria Siigur

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis “Sound Pressure Level Analysis and Visualization in Tallinn Urban Environment Based on Low-Cost IoT Sensor Data,” supervised by Jaanus Kaugerand and Jeffrey A. Tuhtan
 - 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.

07.05.2025

¹ The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.