TALTECH UNIVERSITY
School of Information Technology
Department of Software Science

Andres Suislepp 204042IAPM

MULTI-MODAL DATA FUSION FOR SHORT-TERM URBAN NOISE PREDICTION IN INTELLIGENT TRANSPORTATION SYSTEMS

Master's thesis

Supervisor
Sadok Ben Yahia
Professor
Co-supervisor
Chahinez Ounoughi
PhD student

Tallinn 2023

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Andres Suislepp 204042IAPM

MULTIMODAALSETE ANDMETE ÜHITAMINE LINNAMÜRA LÜHIAJALISEKS ENNUSTAMISEKS INTELLIGENTSETES TRANSPORDISÜSTEEMIDES

Magistritöö

Juhendaja
Sadok Ben Yahia
Professor
Kaasjuhendaja
Chahinez Ounoughi
PhD student

Tallinn 2023

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:	Andres Suislepp
Date:	02.01.2023

Annotatsioon

Intelligentsed transpordisüsteemid on kriitilise tähtsusega tänapäeva linnades. Tarkade rakenduste olemasolu linnades on oluline, et pakkuda inimestele kaasaegset, turvalist ja mugavat linnaruumi. Meie liikumise ja teguviiside analüüsimiseks kogutakse massiivseid andmehulkasid, mille protsessimiseks kulub rohkesti aega ja arvutijõudlust. Selle probleemi lahendamiseks on välja töötatud mitmeid andmeühitamise tehnoloogiaid, mille tulemusena andmete hulk ja modaalsus väheneb, kuid nendes sisalduv informatsioonihulk jääb samaks või suureneb.

Töö eesmärk on välja pakkuda andmeühitamise metoodika, mille kaasabil on võimalik pakkuda usaldusväärseid, kiireid, töökindlaid ja täpseid mürataseme ennustusi linnades.

Välja töötatud lahendus on hübriidne andmeühitamine, milles on kasutatud andmeühitusmetoodikat nii tunnuse- kui ka otsustuse tasemel. Tunnuse taseme ühitamiseks on kasutatatud *Smoothed Kalman Filter* lähenemist, mis töötab hästi mitte-täielike andmehulkade peal. Lisaks vähendab see andmete modaalsust, mille tulemusena ka mudeli keerukus väheneb. Lõplik väljatöötatud mudel koosneb lisaks eelnevale ka otsustustasemel ühitusest, kuhu on lisatud Tallinna avakaameratel põhineva mudeli tulemused. See saavutas veelgi väiksema ennustusvea tulemuse. Lõplik hübriidne andmeühitusmeetod põhineb tunnustasemel *Smoothed Kalman Filter* tehnoloogial ja otsustustasemel *Support Vector Regression* tehnoloogial.

Andmete ühitamise meetodi tulemuste valideerimiseks loodi ennustusmudel, mis koosneb konvolutsioonilisest ja rekurrentsest närvivõrgust. Eksperimendid viidi läbi Tallinna linnast ühe kuu vältel kogutud andmehulga pealt. Tulemuste efektiivsust hinnati neid mitmete üldtuntud andmete ühitamise meetoditega võrreldes. Lisaks sellele kõrvutati tulemusi statistiliste aegridade ennustamise meetoditega. Võrdluseks kasutati kahte karakteristikut: täpsus ja efektiivsus. Täpsus mõõdab ennustuse vea suurust ja efektiivsus mudeli treenimiseks kulunud energiat ja aega. Välja pakutud ühitamisstrateegia saavutas parima tulemuse kõigi võrreldavatega, olles kõige väiksema vea väärtusega. Tulemustest sai lisaks välja lugeda, et mudelile piltide ja otsustusühituse lisamine tõstis väga vähesel määral mudeli täpsust, kuid sellega kaasnes väga suur efektiivsuse langus.

Töö tulemusena valmis täpne ja efektiivne linnamüra ennustusmudel, mis põhineb hübriidsel andmete ühitamise meetodil. Ennustusmudeli tulemusena on võimalik ehitada tarku aplikatsioone, mis tõstaksid intelligentsete transpordisüsteemide kasutajakogemust ja usaldusväärsust. Pakutud andmeühituslahenduse adapteerimine teistesse valdkondadesse on üks võimalikest töö edasiarendustest.

Lõputöö on kirjutatud inglise keeles ning sisaldab teksti 46 leheküljel, 8 peatükki, 14 joonist, 9 tabelit.

Abstract

The amount of data being collected each second is enormous. It takes loads of time and computational power to extract valuable information to process and analyze this data. Data fusion methodologies have been implemented to tackle these problems and reduce complexity while maintaining or improving the information content.

The primary goal of this thesis is to propose a data fusion strategy to provide reliable, accurate, and efficient predictions for urban noise levels in Intelligent Transportation Systems.

The proposed approach to data fusion is a hybrid data fusion, using the feature and decision-level fusions in parallel. For feature fusion, a statistical method, Smoothed Kalman Filter, was used to deal with the data unreliability and simultaneously reduce the complexity of the model. For the model that includes images from Tallinn open cameras, a decision fusion based on a Support Vector Regression was applied to further improve the final prediction's accuracy.

A deep learning network was built to evaluate the impact of the data fusion strategy. Experiments were carried out from the multi-modal data set acquired from Tallinn over the period of 1 month. The results were evaluated against multiple data fusion algorithms and statistical time series baselines based on accuracy and complexity. The proposed model was able to outperform all the other baselines on average. Adding the decision fusion with images to our model had a small improvement in accuracy. However, the increased complexity was immense. The model outperformed baselines by a high margin when predicting 5 or 15 minutes into the future. Regarding 30 or 60-minute predictions, two baselines, namely *Univariate, no fusion* and *Moving average* were able to produce better results due to the simplistic approach of filling the missing target variable values.

As a result of the proposed data fusion strategy, a performant and accurate prediction model was built. This enables building smart applications for Intelligent Transportation Systems on top of urban noise predictions. Generalization to different contexts could be researched for further improvement to the data fusion model.

The thesis is in English and contains 46 pages of text, 8 chapters, 14 figures, 9 tables.

List of abbreviations and terms

API	Application Programming Interface
ARIMA	Autoregressive Integrated Moving Average
AVG	Average
CNN	Convolutional Neural Network
CSV	Comma Separated Values
DF	Data Fusion
GPS	Global Positioning System
ISO	International standard for date and time presentation
ITS	Intelligent Transportation System
KF	Kalman Filter
KNN	K-Nearest Neighbors
LSTM	Long short-term memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MLR	Multinomial Logistic Regression
MSE	Mean Squared Error
P1	Road section 1
P2	Road section 2
px	Pixel, the smallest addressable element in a raster image
RMSE	Root Mean Squared Error
RNN	Recurrent neural network
SKF	Smoothed Kalman Filter
SVM	Support Vector Machine
SVR	Support Vector Regression
UKF	Uncented Kalman Filter
relu	Rectified linear unit

Table of Contents

List of Figures **11**

List of Tables **12**

1 Introduction **13**

1.1 Motivation . . . 13
1.2 Problem statement . . . 14
1.3 Structure . . . 14

2 Background and related work **15**

2.1 Intelligent Transportation Systems . . . 15
2.2 Data fusion . . . 15
2.2.1 Data-level fusion . . . 16
2.2.2 Feature-level fusion . . . 16
2.2.3 Decision-level fusion . . . 16
2.3 Data fusion for time series prediction . . . 17
2.4 Multivariate time series prediction . . . 18
2.5 Long short-term memory . . . 18

3 Data acquisition **20**

3.1 Data sources . . . 20
3.2 Data processing . . . 20
3.2.1 Combining data from multiple sources . . . 23
3.3 Exploratory analysis . . . 23
3.3.1 Handling of missing data . . . 28
3.3.2 Privacy preserving . . . 28

4 Methodology **29**

4.1 Model building . . . 30
4.1.1 Fusion approach . . . 31
4.1.2 Prediction approach . . . 32

5 Experimental evaluation **33**

5.1 Experimental setup . . . 33
5.2 Evaluation . . . 34
5.2.1 Walk-forward validation . . . 36

5.3 Targeted baselines 36
5.3.1 Data fusion methods 36
5.3.2 Statistical time series prediction techniques 37
5.4 Results and discussion 38
5.4.1 Comparison of data fusion methods 38
5.4.2 Comparison of time series methods 41
5.4.3 Sequence length impact on model performance 42

6 Future work and applications **44**

7 Summary **45**

8 Acknowledgements **46**

Bibliography **47**

Appendix 1 - Non-exclusive licence for reproduction and publication of a graduation thesis **49**

Appendix 2 - Results **50**

Appendix 3 - Dataset **96**

Appendix 4 - Code **97**

List of Figures

1 LSTM architecture for a supervised model in time series prediction context. Figure referenced from [13] 19

2 TomTom road sections 21
3 Tallinn Live Cameras: Concatenated 22
4 Noise series 24
5 Noise histogram 25
6 Correlation matrix 26
7 Rain histogram 27

8 High-level model architecture 29
9 Final model architecture 30
10 Final model architecture without images and decision fusion 30

11 Walk-forward validation 36
12 Example proposed model training 38
13 Proposed model predictions for different output steps 39
14 Comparison of time series input sequence length 43

List of Tables

1 Data sources 20
2 Noise characteristics 24
3 Rain characteristics 26
4 Categorical value counts 27

5 Prediction model architecture 34
6 Data fusion accuracy comparison 40
7 Data fusion performance comparison 41
8 Proposed approach comparison with statistical time series methods 42
9 Comparison of input sequence length on training the proposed model . . . 43

1. Introduction

Developing Intelligent Transportation Systems (ITS) is crucial to improve people's mobility in densely populated cities. Evolution in the field has led to a high demand for smart applications that provide useful information as an input for reliable and smart transportation networks [1].

The amount of information we have today is enormous, and the challenge relies on extracting the useful and informative parts of the collected data and providing meaningful analysis on top of it. In the context of ITS, the biggest generators of data are the deployed sensors, including data from GPS, video cameras, LIDAR, RADAR, and loop detectors, to name a few. This data is often supported by other sources such as social media, weather data, public transportation data, etc. The biggest challenges in the field can be identified as *(i)* analyzing real-time heterogeneous big data and *(ii)* data reliability [1].

Data Fusion (DF) is considered an elegant and efficient way to tackle the problems related to multi-modal big data. Studies on DF have delivered significant enhancements in ITS and demonstrated a vital impact on its evolution [1].

The proportion of the world population living in urban areas is expected to grow rapidly in the following decades [2], which indicates the actuality of the problem. In addition, high urban noise levels are known to be a source of many illnesses, starting from constant stress and sleep issues to more severe problems like cardiovascular diseases [3].

This thesis proposes a novel data fusion method to improve and simplify deep-learning prediction model outcomes. The model is developed using data from the city of Tallinn.

1.1 Motivation

Traffic congestion in big cities is a huge cost for city stakeholders. Accurately predicting traffic characteristics, including urban noise, can reduce congestion and the overall CO_2 emissions, fuel consumption, and travel times. This will lead to safer, more modern urban environments and a healthier planet.

1.2 Problem statement

Urbanization increases the need for smart cities to manage people's mobility efficiently. Therefore, to overcome this issue, a myriad of researchers have conducted combined data fusion techniques with traffic prediction approaches by processing the vast amount of collected heterogeneous traffic data from different sources. Better prediction results allow ITS stakeholders, managers, and applications to reduce congestion, travel time, CO_2 emissions, allocate resources, and increase safety.

My thesis aims to propose a method for traffic data fusion that improves the performance and accuracy of predicting urban noise levels in Intelligent Transportation Systems. The hypothesis to be proven is that data fusion methods mixed with hybrid deep learning methods can yield highly accurate and performant results for short-term urban noise prediction. The problem is interpreted as a forecasting problem.

1.3 Structure

The rest of the work is structured as follows. Chapter 2 introduces the general methodology and summarizes the state-of-the-art data fusion methods, emphasizing feature and decision fusions. Additionally, a comprehensive introduction to the techniques used in work is given. Chapter 3 gives an overview and initial analysis of the available dataset and the pre-processing strategies used. The proposed architecture is provided in Chapter 4, where the underlying technology choices are justified. Chapter 5 thoroughly analyzes the experimental evaluation, where the results are discussed and compared to the relevant baselines. Possible applications for the final implementation are given in Chapter 6, together with ideas for further improvement of the models.

2. Background and related work

This chapter provides the necessary context needed to understand further work. State-of-the-art data fusion methods are introduced, and a high-level overview of the used technologies is given.

2.1 Intelligent Transportation Systems

European Union directive 2010/40/EU states Intelligent Transportation Systems as a group of advanced applications that aim to provide helpful and innovative solutions to traffic management and different modes of transportation [4]. It integrates telecommunications, electronics, and information technologies with transport engineering to enable various stakeholders to be better informed for making safer and more coordinated decisions. A critical factor in deploying these systems is preserving individual consumers' privacy. The directive also suggests increasing the number of deployments of intelligent applications, which has accelerated the demand and interest in the field.

2.2 Data fusion

Data Fusion is an advanced technique to combine information coming from several sources to get more accurate results in an execution of an application in a way that would be performed by the use of individual sources separately [5]. The expectation is that fused data is more informative and synthetic than the original inputs. Another significant factor of DF is dimensionality reduction. The ability to simplify models both algorithmically and computationally is a precious aspect when dealing with significant amounts of multi-modal data. In a recent study, [1] has classified the current directions of DF as hybrid data fusion, explainable deep neural network data fusion, adaptive sensor selection, privacy-preserving, and real-time data acquisition and processing.

Data fusion can be categorized into three main categories based on when the fusion takes place: data-level fusion, feature-level fusion, and decision-level fusion.[6]

2.2.1 Data-level fusion

Data-level fusion, also recognized as low-level fusion, is most widely used when collecting data from the environment. Suppose multiple homogeneous sensors collect the exact measurement. In that case, these inputs from the sensors can be fused directly to improve sensor reliability, reduce noise and achieve more accurate and informative data than the sources. This also decreases the network bandwidth used, making it less expensive to handle big amounts of data [6].

2.2.2 Feature-level fusion

Feature-level fusion, also known as intermediate-level fusion, merges multiple data sources into a new high-dimensional dataset. Since high-dimensional datasets are computationally and algorithmically expensive, simple concatenation of feature sets is usually not good enough. Feature-level DF algorithms and feature engineering can be beneficial when dealing with high-modality datasets. However, in some cases, simple concatenation has shown to be a viable option with the popularization of deep learning [6].

Authors in [7] built a deep-learning model to predict the occupancy of electric vehicle charging stations. Their proposed fusion component integrates information from the dynamic encoder and static feature component. It uses concatenation to feed the encoded feature vectors to the fully-connected layer. The prediction model is evaluated concerning multiple metrics: precision, recall, and F1-score. The proposed model shows the best performance regarding the Recall and the F1-score compared to various statistical, machine learning, and deep learning baselines. The paper shows promising results with a simplistic fusion approach.

2.2.3 Decision-level fusion

Decision-level fusion, also recognized as high-level fusion, is used to fuse multiple independent, often weaker decisions to a final unique decision [6]. This is especially useful for capturing the different characteristics of the dataset by using specialized models and fusing the outputs instead of trying to build a generalized model for the whole dataset.

Paper [8] is introducing a decision-level data fusion framework based on homogeneous (machine-learning) and heterogeneous (Extended Kalman Filter) data for traffic congestion prediction. Decisions from three data mining algorithms (deep belief network, k nearest neighbors, random forest) are fused using Extended Kalman Filtering. The three models' fused output shows a significant performance increase over any models independently. The dataset of the evaluation is based on daily motorway traffic in Montreal, obtained from Genetec blufaxcloud travel-time system engine. The model validation is done by measuring the predicted travel time and comparing it against the estimates obtained from Genetec blufaxcloud travel-time system engine. The model outperforms Genetec estimations 12 times out of 23. Authors in [9] use feature- and decision-level fusion to assess tea quality based on the tea's image and scent. Paper is extracting features from images and e-nose sensors and fusing the data to classify the quality of the tea batch. K-nearest neighbors (KNN), support vector machine (SVM), and multinomial logistic regression (MLR) were applied for classification modeling. Both studies (feature and decision) show better classification results than those based on independent inputs. For this paper, decision-level fusion is the most effective approach.

2.3 Data fusion for time series prediction

Traffic prediction has been a hot research topic for many years. Traffic's complex, non-linear and stochastic characteristics are the main problems of making accurate predictions [10]. In recent years, traffic data fusion has been used to improve traffic characteristic predictions in cities [8]. Kalman Filter and its variations have shown high usage in forecasting and estimating short-term traffic characteristics. The authors in [11] use Kalman Filter to fuse spatial and location-based data to estimate traffic density. Subsequently, the estimated data are utilized for predicting density to future time intervals using a time series regression model. The experiment was carried over in Chennai, India, which adds a significant complexity due to the unique nature of traffic that poses both technological and algorithmic challenges. The density prediction model performed relatively well despite the challenges arising from India's heterogeneous traffic flow conditions. A more recent study [12] has proposed a hybrid model for regional traffic flow prediction based on the convolutional neural network (CNN) and long short-term memory (LSTM). The data being fused is both spatial and temporal. As CNN is generally more suitable for spatial data and LSTM models are appropriate for processing sequential types of data (temporal), a hybrid approach combining CNN and LSTM is introduced. The model shows the highest accuracy compared to the targeted baselines for multi-step forecasting. Another paper [13] with a similar approach proposes a multivariate CNN-LSTM model to predict stock market prices. The stock market is a noisy, stochastic environment identical to traffic in big cities. The proposed model used multiple stock market indices by considering the state of correlation between them in the forecasting process. The multivariate CNN-LSTM model outperformed standalone CNN and LSTM models by a relatively high margin.

Besides traffic predictions, other commonly used applications for DF in the context of

ITS include autonomous vehicles, travel time estimation, traffic prediction, congestion prediction, incident detection, vehicle communication, and different management systems [1].

2.4 Multivariate time series prediction

Multiple time-dependent variables in a multivariate time series prediction depend on their previous value and other variables' previous values. This allows the model to capture the trend of observed variables' relationships when forecasting the future value [13]. The input to a multivariate time series prediction is a uniformly distributed time-dependent sequence of prior values. Formula 2.1 represents the prediction as a function *fn* where *X* represents the variable set, *y* as the target variable, $\hat{y}$ as the predicted value, *l* is the input sequence time length in time steps, and *o* represents the output time steps (how many steps in the future to predict).

$$\hat{y}_{t+o} = fn(\{X_{t+(-l+1)}, X_{t+(-l+2)}, ..., X_t\}) \tag{2.1}$$

2.5 Long short-term memory

To evaluate the fusion methodologies, a multivariate prediction methodology is needed. Multi-modal non-linear urban noise data is known to be too complex and noisy for traditional time series prediction methods to handle. The advancements in machine learning research are providing viable options to overcome these limitations. Neural networks can learn the complex relationships between data features in big datasets without relying on previous information [13].

Due to its popularity in late time series prediction publications, the chosen prediction methodology was long short-term memory (LSTM). LSTM is a popular recurrent neural network (RNN). Recurrent neural network (RNN) is a deep network architecture where the connections between hidden units form a directed cycle [14]. The LSTM network can capture long-term dependencies by using internal memory that keeps the previous information from the last hidden states, as illustrated in Figure 1.

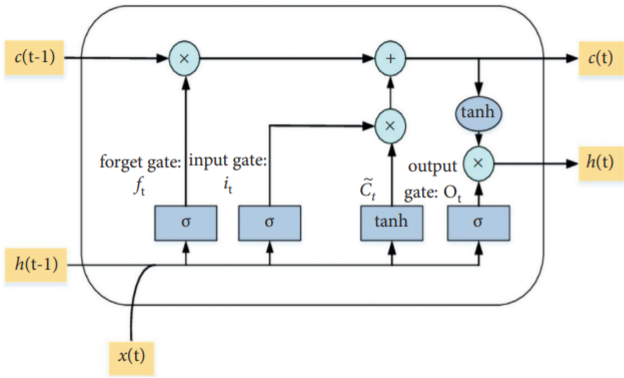


Figure 1. LSTM architecture for a supervised model in time series prediction context. Figure referenced from [13]

.

Traditional LSTM unit comprises forget, input, output gates, and a memory cell. The architecture assumes uniformly distributed elapsed time between the elements of a sequence [14]. LSTM has been widely used and proven to be very capable of forecasting time series data [13].

3. Data acquisition

This chapter aims to give a good understanding of the underlying dataset. This includes the data source, acquisition information, and a thorough analysis of the available features. A thorough exploration of the available dataset is needed to propose a suitable data fusion strategy for an accurate prediction.

3.1 Data sources

The intersection of interest for the model building is in Tallinn, between Sõpruse Puiestee and Tammsaare Tee. The main reason for this decision was the availability of the urban noise sensor data.

Different traffic characteristics data were acquired from multiple sources. Data sources, methods of acquisition, and processing strategies are described in Table 1. The interval for data acquisition was 5 minutes. Data were acquired from 10 February 2022 until 6 March 2022.

Features	**Source**	**Acquisition method**
Noise level	Thinnect	Export
Traffic characteristics	TomTom Developer Portal	API Scraping
Weather	ilm.ee	Website Scraping
Road condition, weather	Tallinn Weather Portal	Website Scraping
Camera images	Tallinn Live Cameras	Scraping
Datetime features	-	Computed

Table 1. Data sources

3.2 Data processing

Noise data

Noise data was exported in CSV format from Thinnect portal and required no preprocessing. The used sensor is situated at the intersection of Tammsaare - Sõpruse and sends the average noise level to the server every minute.

TomTom

Data from TomTom was acquired by accessing the TomTom Maps API [15]. TomTom provides data about different road sections. The two road sections used are denoted as *P1* and *P2* and depicted in Figure 2.

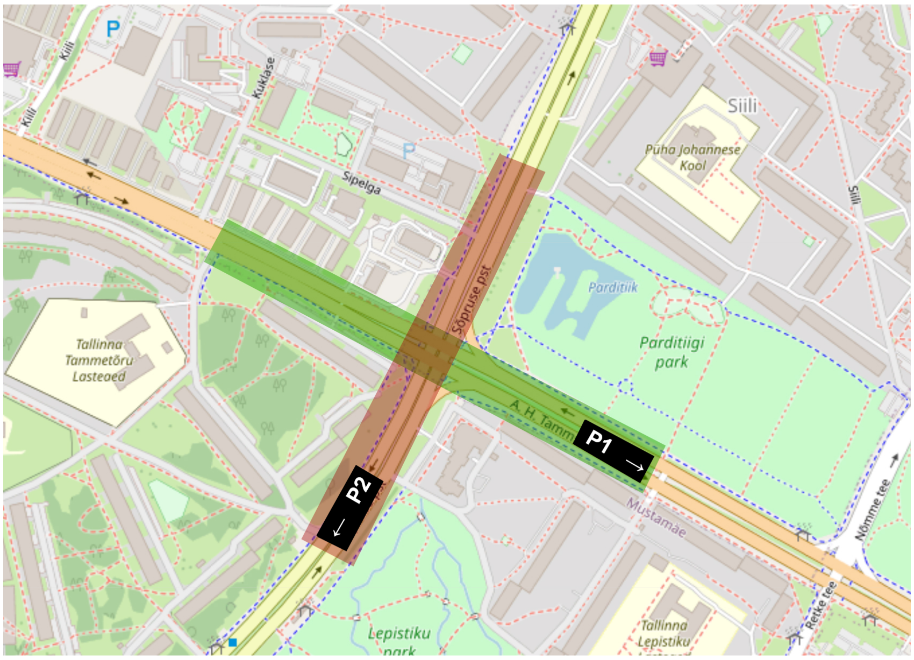


Figure 2. TomTom road sections

TomTom provides average car speed and travel-time information about each road section for two cases: the current and free flow states. Free flow state describes the situation for a case where there is no extensive amount of traffic. This allows us to calculate the differences between the free flow state and the current state, using the formula 3.1. In addition, TomTom provides data about road closures and road types in real time.

$$t_{diff} = t_{current} - t_{freeflow} \tag{3.1}$$

```
df['TT_P1_Travel_Time_Diff_Sec'] = df.apply(
        lambda row: row['TT_P1_Current_Travel_Time_Sec'] -
            row['TT_P1_Freeflow_Travel_Time_Sec'], axis=1)
df['TT_P1_Speed_Diff_Kmh'] = df.apply(
        lambda row: row['TT_P1_Freeflow_Speed_Kmh'] -
            row['TT_P1_Current_Speed_Kmh']), axis=1)
```

Ilm.ee

Scraping *ilm.ee* website provides data points about the current weather and air conditions. The available features are temperature, wind temperature, air pressure, air humidity, wind speed and direction, rainfall, sunset and sunrise times, cloudiness, and coldness class.

Tallinn Weather Portal

Scraping *Tallinn Weather portal* website provides data points about the current weather and road conditions. The available features are temperature, air humidity, and road temperature.

Tallinn Live Cameras

There are three live cameras for the intersection of interest. The images are scraped from the Tallinn Live Cameras website, resized into a standardized size of 100px x 100px, and concatenated into a single 300px x 100px picture collage depicted in Figure 3. Concatenation is needed to allow the deep learning model to learn about all the driving directions simultaneously. The small size for the images had to be chosen to optimize the further processing and training procedures since dealing with high volumes of image data is computationally expensive.

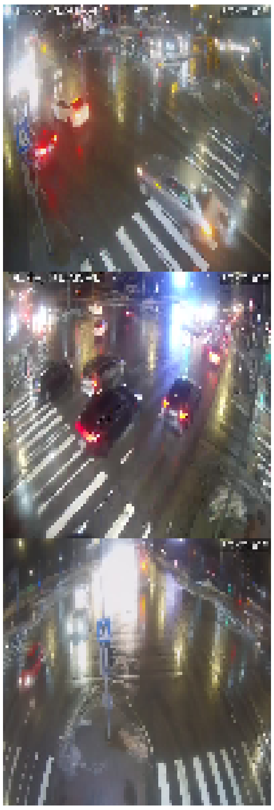


Figure 3. Tallinn Live Cameras: Concatenated

DateTime features

DateTime features are computed from the ISO timestamp. The following features are computed: date, hour, minute, minute of the day, day of the month, day of the week, and is-weekend.

```
def compute_datetime_features(df):
    df['Datetime'] = df.apply(lambda row: datetime
            .fromisoformat(row['Timestamp']), axis=1)
    df['Date'] = df.apply(lambda row: row['Datetime']
            .strftime("%Y-%m-%d"), axis=1)
    df['Hour'] = df.apply(lambda row: row['Datetime']
            .hour, axis=1)
    df['Minute'] = df.apply(lambda row: row['Datetime']
            .minute, axis=1)
    df['Minute_Of_Day'] = df.apply(lambda row: (row['Hour'] *
            60) + row['Minute'], axis=1)
    df['Day_Of_Month'] = df.apply(lambda row: row['Datetime']
            .day, axis=1)
    df['Day_Of_Week'] = df.apply(lambda row: row['Datetime']
            .weekday(), axis=1)'
    df['Is_Weekend'] = df.apply(lambda row: row['Day_Of_Week']
            == 5 or row['Day_Of_Week'] == 6, axis=1)

    return df
```

3.2.1 Combining data from multiple sources

For further processing of the dataset, data from multiple sources are combined using the Pandas DataFrame merge functionality. Merging is based on computed DateTime features.

```
df = pd.merge(df_noise, df_ilmee_weather,
  how='left',
  on=['Date', 'Hour', 'Minute'])
```

3.3 Exploratory analysis

The initial dataset contains $7,252$ data points. Exploration of the target variable, noise, shows the first immediate problem. As depicted in Figure 4, noise value is only present for 26.54% of the dataset. Another feature visible from Figure 4 is the temporal characteristic of the urban noise. Further exploration of noise characteristics are shown in Table 2 and a histogram in Figure 5. The target variable is a numeric value between the range of 45-80 dB with a mean of 58.931 dB.

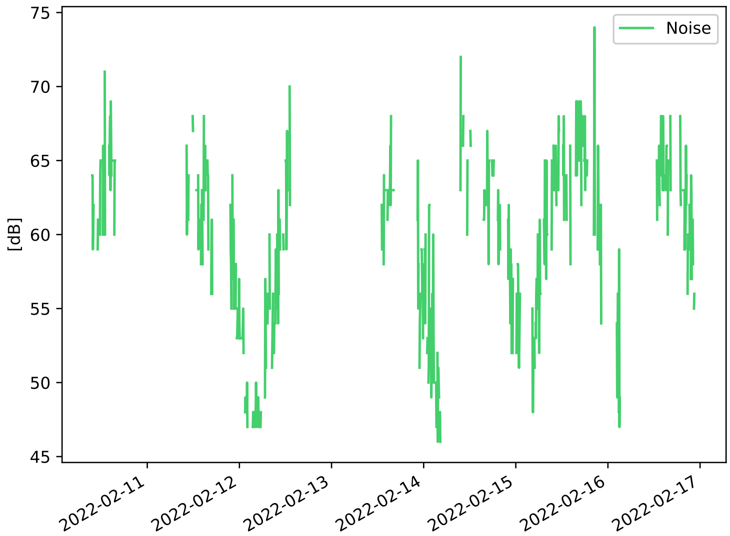


Figure 4. Noise series

Characteristic	Value
Count	1925
Mean	58.931
Standard deviation	5.954
Minimum	45
25%	54
50%	59
75%	63
Maximum	80

Table 2. Noise characteristics

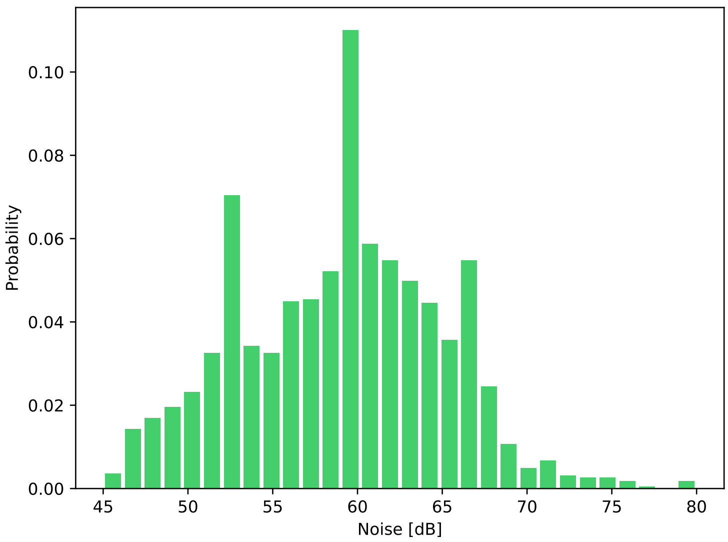


Figure 5. Noise histogram

Exploring the correlations between continuous features shows a very low correlation for rainfall. The correlation matrix in Figure 6 shows less than 0.1 correlation between any other feature and rain. Further exploration of the rain feature in Table 3 and Figure 7 shows that there was minimum rain detected during our interest of time. The outcome of the exploration is removing the rain feature from further model development. The biggest correlation with the target variable is the road, air temperature, and wind speed.

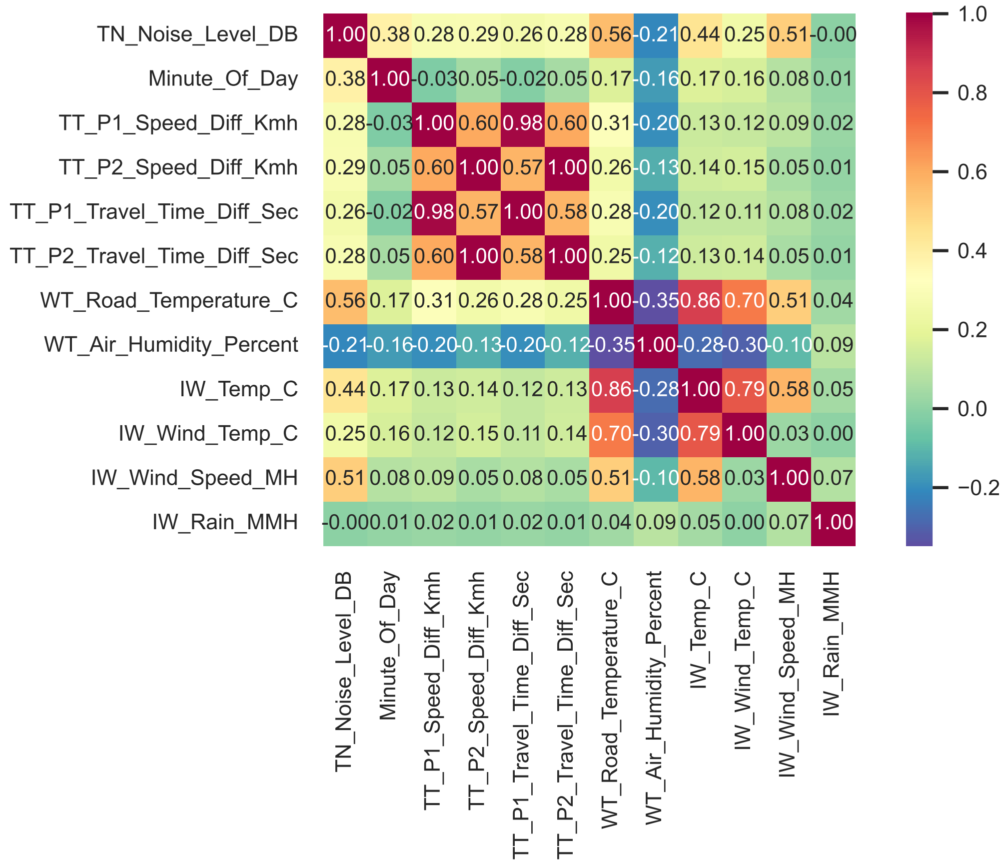


Figure 6. Correlation matrix

Characteristic	Value
Count	7177
Mean	0.008
Standard deviation	0.089
Minimum	0
25%	0
50%	0
75%	0
Maximum	1

Table 3. Rain characteristics

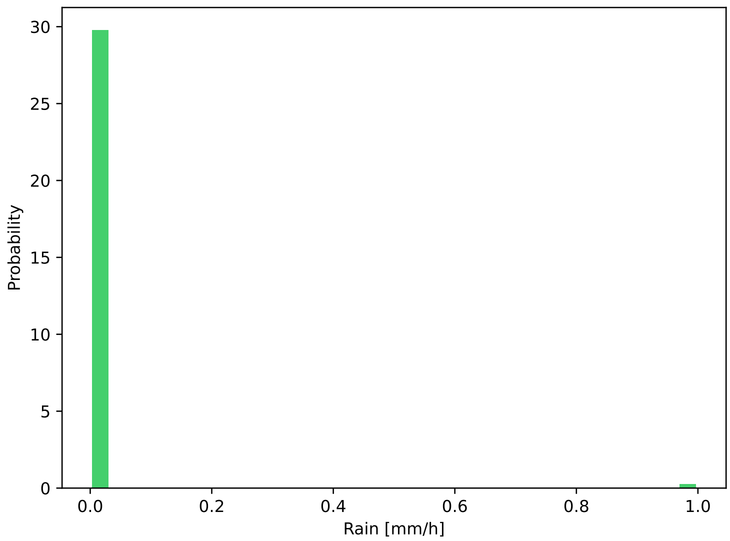


Figure 7. Rain histogram

Categorical features value count exploration shows us four variables with static values as described in Table 4. These values are not providing additional information to the model and, therefore, can be excluded from further model development. Because wind speed has a significant correlation, we can expect the wind direction to be very influential.

Value	**Count**
FRC3	7252

P1 road type

Value	**Count**
FRC2	7252

P2 road type

Value	**Count**
False	7252

P1 is closed

Value	**Count**
False	7252

P2 is closed

Value	**Count**
SW	2728
S	1509
W	1380
SE	508
N	462
NW	457
NE	114
E	96

Wind direction

Value	**Count**
cloud_norain	3490
cloud	2089
cloud_lightsnow	653
cloud_lightrain	634
cloud_modrain	202
cloud_lightrainsnow	90
cloud_modsnow	78
cloud_modrainsnow	18

Cloudiness class

Value	**Count**
cold	4158
hot	3096

Coldness class

Table 4. Categorical value counts

The final dataset contains 11 continuous features, three categorical features, and one image feature combined with three images. The immediate problems regarding data preprocessing that arose from the exploratory analysis are the following:

1. Missing values for the target variable.
2. Privacy-preserving for images.
3. Capturing periodic temporal characteristics of the target variable.

3.3.1 Handling of missing data

The requirement for training accurate deep learning models is a sufficient amount of training data. To increase the size of the training dataset, noise values are imputed using a method of backward fill, where the last known valid value is carried backward in the dataset, with a limit of 6. This ensures that short-term missing values are not affecting the training process. Still, the limit is set not to fill vast gaps of missing data in training, which can cause the model to learn the backward fill implementation instead of the actual dependencies and variable movements. In addition, as described in Chapter 2.4, the input to a time series prediction is a sequence of datasets where elements are expected to be uniformly distributed. The sequences that violate that assumption are removed from the training set as described in Formula 3.2, where the sequence of variables *S* including *l* variable sets *X* with a parameter timestamp in minutes *X(t)*. The sequence *S* is included in the training set when function *fn* evaluates to *true* with a time step *ts* between each item.

$$fn(S) = (X_t(t) - X_{t+(-l+1)}(t)) <= l * ts \tag{3.2}$$

3.3.2 Privacy preserving

Data privacy is an important matter to discuss when dealing with image processing. The collected images used in this work are processed only to predict future noise values. The persons and the personal cars visible in the images are not processed in a standalone approach.

4. Methodology

The proposed architecture is described in detail in this chapter. The distinct strategies are justified, and an overview of underlying logic is provided.

Considering the late success shown in recently published papers that proposed deep learning models which can recognize complicated and unknown patterns in large varying data sets, the proposed methodology chosen for the prediction model is deep learning. The idea is to use a combination of data fusion techniques before feeding it into the LSTM-based model that predicts short-term urban noise levels. Deep learning models greatly succeed when dealing with nonlinear multivariate time series data. However, the limitation is the need for a vast amount of data and computationally expensive training times [16]. A combination of DF strategies improves the prediction model's accuracy and performance. The proposed approach is a hybrid DF method combining feature and decision-level fusions. Figure 8 shows the high-level architecture to be built.

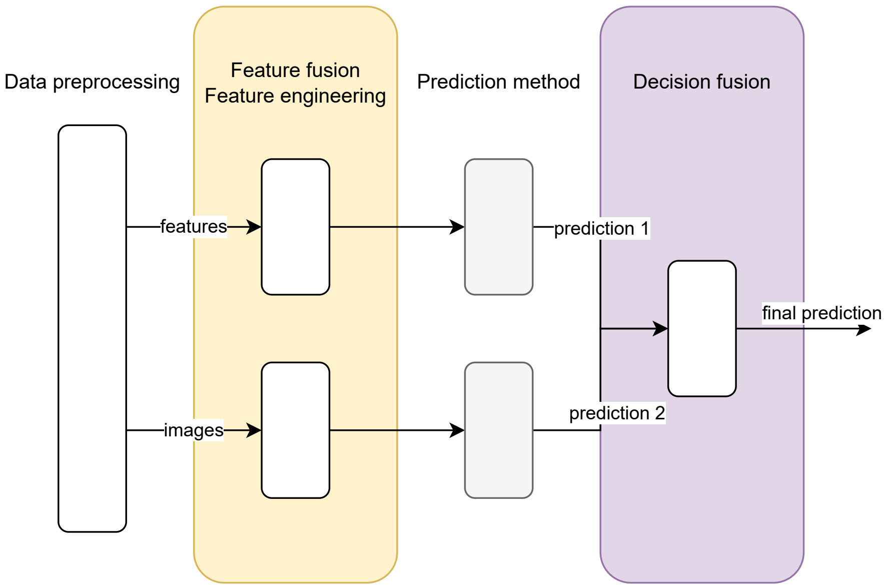


Figure 8. High-level model architecture

4.1 Model building

The proposed architecture depicted in Figure 9 aims to overcome the issues described earlier and provide an accurate forecast.

The proposed model is a combination of fusions. It uses the traditional DF method Smoothed Kalman Filter (SKF) for feature fusion to reduce the dimensionality and complexity of the model to allow for faster training times while maintaining or even improving the accuracy. This is combined with a CNN architecture to extract unknown features and patterns from the already fused data by introducing a multi-fusion strategy. The image input is processed using another independent CNN. The prediction outputs are fused using a Support Vector Regression decision fusion with a *rbf* kernel to improve the model's accuracy further. Both independent models use a deep learning prediction network based on CNN and LSTM. The final architecture of the model is described in Figure 4.1. The architecture of the same but more straightforward approach without images and decision fusion is shown in Figure 10.

The final architecture is a hybrid data fusion model, where multiple types of fusions are used to achieve an accurate prediction.

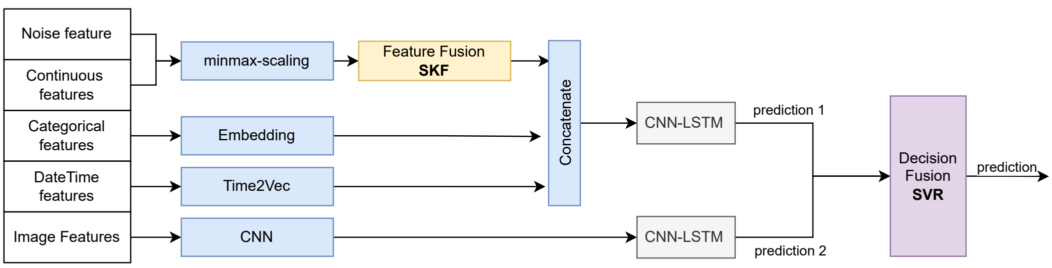


Figure 9. Final model architecture

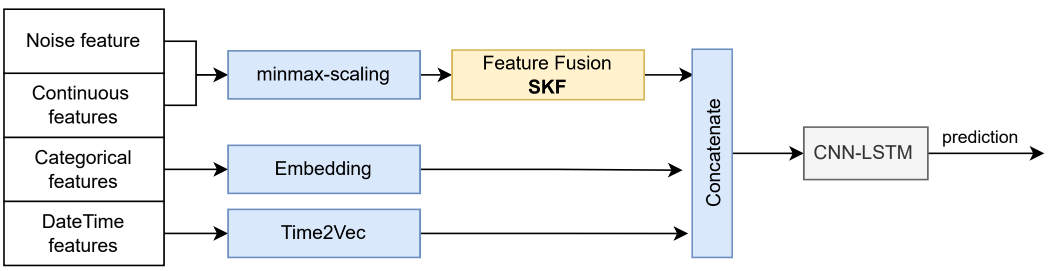


Figure 10. Final model architecture without images and decision fusion

4.1.1 Fusion approach

Continuous features. A Kalman Filter with Smoothing, Smoothed Kalman Filter (SKF) is proposed to implement as a feature fusion method. It is an extended version of one of the oldest state estimation methods, the Kalman Filter(KF). It is simple and effective to use and helps to reduce the observation noises. The most beneficial outcome of the added smoothing usually becomes apparent when there is a more complex multivariate problem. The smoothed estimates of component values like the trend, cycles, and regressor effects can improve the forecasting target series [17].

Categorical features. Deep learning models require all inputs and outputs to be numeric. A learned embedding is an excellent way to overcome this and allow the network to learn the dependencies of categorical values. The implementation of embedding maps each categorical value to a vector, which allows the network to learn the categorical parameters when training.

Datetime features. Time is an important feature when building the prediction model. For the neural network to understand the properties of time, such as periodicity and invariance to time scaling, a Time2Vec implementation proposed in [18] is implemented. Time2Vec is mainly implemented to capture the periodicity characteristic of the target variable described in Chapter 3.3.

Images. To extract unknown features from image sequences, a CNN is used. CNN has been shown to learn accurate patterns and insights from images. Its built-in convolutional layer reduces the high dimensionality of images without losing its information [19]. It is one of the most popular choices when dealing with image data in a deep-learning context. The biggest disadvantage of using images and CNN is the computational expense. Image sequences take a long time to process and train the network.

Decision fusion, Support Vector Regression. Support Vector Machines (SVM) have been studied, generalized, and applied to several problems, including time series predictions. Support Vector Regression (SVR) shares the same advantages as SVMs [20]. They are efficient and work well in cases when there are not many outliers, making them ideal for decision fusions with an assumption that our models are generally accurate independently.

4.1.2 Prediction approach

The prediction approach is an ensemble of CNN and LSTM layers heavily influenced by a similar approach proposed in [13]. This combination is referred to as CNN-LSTM. It uses CNN to extract complex hidden patterns in the dataset and feeds its output to the LSTM layer input for time series prediction. This allows taking advantage of the powers from both independent layers to allow for accurate predictions. CNN extracts the hidden relationships between multi-modal data features, and LSTM is learning the time sequence relationships.

This must be noted that the prediction approach is not the main contribution of this work. The prediction approach must be in place to evaluate the fusion approaches.

5. Experimental evaluation

In this chapter, the results of the performed experiments are given. A description of the configurations and parameters used to run the experiments is provided. The evaluation metrics and baselines for comparisons are introduced, and the performance of the proposed approach is compared to the baselines in detail.

5.1 Experimental setup

The data preprocessing, fusion techniques, and prediction models were all implemented in Python (version 3.8.10) programming language. Many standard Python libraries were used for data processing, evaluation, and visualization, such as *matplotlib*, *numpy*, *matplotlib*, *scikit-learn* and *pandas*. Keras (version 2.7.0), the Python deep learning framework, was used with the Tensorflow backend to implement deep learning models for predictions.

To measure the performance of the proposed approach, a prediction for future urban noise levels for the next 5 minutes, 15 minutes, 30 minutes, and 60 minutes is computed. For all experiments, a *min-max* normalization technique between the range 0 to 1 is performed on all the continuous feature values, including the target feature, before applying the proposed fusion strategy. Embeddings are extracted for categorical variables. As described in Chapter 2.4, the values are aggregated into fixed-length sequences of 12-time steps that result in 60 minutes of look-back time. When choosing the sequence length, two aspects were considered carefully. The sequence must be long enough to learn the models' complex patterns. However, too-long sequences are computationally much more expensive and rely too much on perfect data quality.

The CNN for models where images were included were composed of three convolutional layers, with 100, 200, and 300 units, respectively, followed by a dense layer with 1024 units. All mentioned layers are using *relu* activation. The final layer of the CNN for extracting image features is a dense layer with one unit and a *linear* activation function. Pooling and dropout were added to reduce the complexity of the network and prevent overfitting.

The deep learning prediction model described in Chapter 4.1.2 was tuned to fit during the implementation and kept static during all experiments to give a fair evaluation of the data fusion approaches. The CNN-LSTM architecture contains three convolutional layers with 64, 64, and 64 units, respectively, together with *relu* activation. Dropout with a rate of 0.2 is added between each convolutional layer to prevent overfitting. Pooling is added to reduce the complexity of the network. The output of the final convolution layer is fed into an LSTM layer with ten units and *relu* activation function. The final layer of the prediction network is a dense layer with linear activation and units equal to the prediction length. Table 5 gives an overview of all the parameters used.

All the experiments were run in the TalTech AI-Lab environment. 80% of the data is used for training purposes and 20% for validation of the results. The models were trained for 100 epochs with the *Adam* optimizer and a mean average error (MAE) loss function. The learning rate.

Type	**Value**
Sequence length (look back)	12
CNN layers	3
CNN filters	64, 64, 64
LSTM layers	1
LSTM units	10
Epochs	100
Optimizer	Adam
Loss function	MAE
Learning rate	0.001

Table 5. Prediction model architecture

5.2 Evaluation

As described in Chapter 2, data fusion aims to solve two problems: improve the accuracy of the models and reduce the computational and algorithmic complexity by reducing the dimensionality. Based on this assumption, the evaluation of the proposed approach is also grouped into two segments: the model's accuracy and performance. Model accuracy evaluates the difference between the predicted noise level with the actual noise level. Model performance shows the computing resources and time used to train the model.

To evaluate the accuracy of the model, four different metrics are used, where y represents the actual value, $\hat{y}$ the predicted value, and n the size of the dataset.

Mean Squared Error (MSE) - Popular metric to evaluate the errors for the models. [21] Calculated with formula 5.1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{5.1}$$

Root Mean Squared Error (RMSE) - Similar metric to MSE, but giving more weight to big outliers. [21]. Calculated with formula 5.2.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5.2}$$

Mean Absolute Error (MAE) - A scale-dependent metric over the whole dataset. Calculated with formula 5.3.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{5.3}$$

Mean Absolute Percentage Error (MAPE) - Percentage error that is easy to interpret without knowing the context of the data. Calculated with formula 5.4. [22]

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|\frac{y_i - \hat{y}_i}{y_i}| \tag{5.4}$$

RMSE and MSE have seen high usage in evaluating forecast models due to their theoretical relevance in statistical modeling. On the other hand, MAE and MAPE are less sensitive to outliers [21].

To evaluate the performance of the model, two different metrics are used:

Time to train (t) - Time elapsed to train the model. Less time spent on training the model means this model is computationally more performant as it spends fewer resources.

Model trainable parameter count (n_{param}) - Number of parameters model has to train. This is another way to evaluate the number of computational resources spent.

5.2.1 Walk-forward validation

Walk-forward validation is a testing approach designed to test models in a realistic scenario, imitating what would happen in a real-life setting. It provides a testing framework for evaluating the predictive power of a model on the data not used to train it [23]. A regular cross-validation strategy is not optimal for time series data because of temporal characteristics like seasonality, unexpected pulses, or trends. Using future observations to predict past values does not fairly indicate the actual model performance. This is why walk-forward validation is also an excellent method to avoid overfitting for time series models [24]. Figure 11 illustrates the walk-forward procedure used to evaluate the prediction outputs.

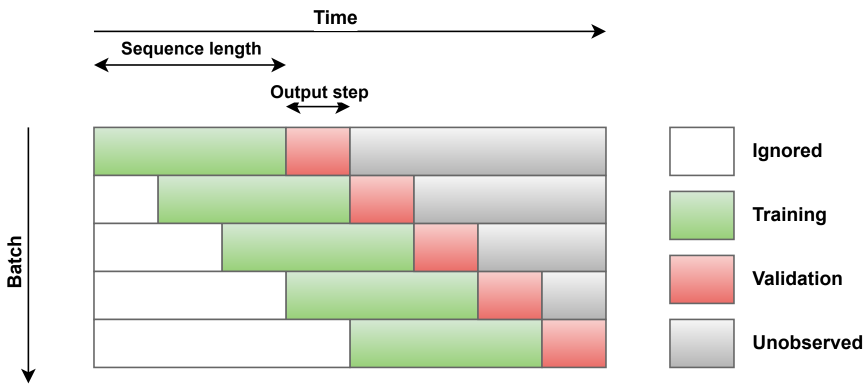


Figure 11. Walk-forward validation

5.3 Targeted baselines

5.3.1 Data fusion methods

No fusion, univariate. No fusion algorithm. Prediction is only based on the previous sequence of noise values.

No fusion. No feature fusion algorithm, all the parameters from feature engineering are fed into the deep learning prediction model. This baseline is a good comparison to evaluate if, using the same deep learning model, can the proposed feature fusion methodology improve the results.

No fusion + images. Similar to the previous, except for the outputs of the CNN are concatenated into the input of the final model. This is an excellent baseline to show that the decision fusion with an independent image prediction model should improve upon the model where images are concatenated into a single model together with other features.

Kalman Filter. Kalman Filter is one the oldest state estimators for linear systems. It has been widely used to improve observation errors. It is simple to use and computationally very inexpensive. However, applying the KF to nonlinear systems can be difficult [25].

Uncented Kalman Filter. Unscented Kalman Filter (UKF) is an extension to the regular Kalman Filter. The internal method called unscented transformation allows UKF to calculate the statistics and state estimation of a random nonlinear state [25].

Average. An simple average over multiple decisions or predictions can be used. The main positive effect of using average is that it helps balance the outliers in the model outputs. In addition, it is straightforward to implement. This only applies to decision fusion since the variables must be of the same type and scale.

5.3.2 Statistical time series prediction techniques

Statistical time series baselines are added as a comparison to validate the problem-solving approach using deep learning. Statistical methods have been widely used to solve time series prediction problems, and they tend to be much more efficient and easier to build and understand than deep learning models. To justify the deep learning approach, statistical methods are added as baselines.

Naive. Naive is the most simplistic forecasting method, where the last observation is carried over as a prediction. This can yield surprisingly good results for many economic and financial time series [26]. It is an excellent first baseline to improve upon.

Moving average. Moving average is a classical time series forecast algorithm. Observations near each other in time are likely to be proximate in value. In case of outliers, the moving average smoothes the output and therefore gives a smooth trend-line prediction [26].

Linear regression. As shown in 6, there is a correlation between temperature and the time of day. The univariate linear regression model is built upon the assumption that there is a linear relationship between the target variable and a predictor [26].

ARIMA. Autoregressive integrated moving average (ARIMA) model is one of the most widely used time series forecasting models. It aims to find autocorrelations in the dataset. [26]

5.4 Results and discussion

All experiment results are shown in Appendix 2. The dataset used is provided in Appendix 3 and code together with instructions for running the experiments is given in Appendix 4.

Figure 12 shows the loss of the training and validation set during the training process. The figure shows that training and validation losses do not diverge significantly. This explains that our model generalizes well instead of overfitting by remembering the input data [19]. Figure 13 illustrates the model predictions against the validation data set.

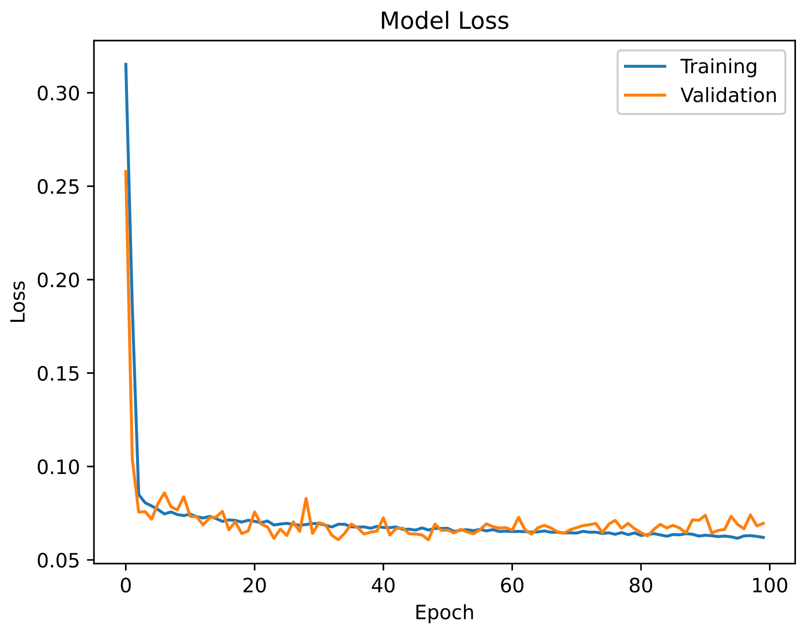
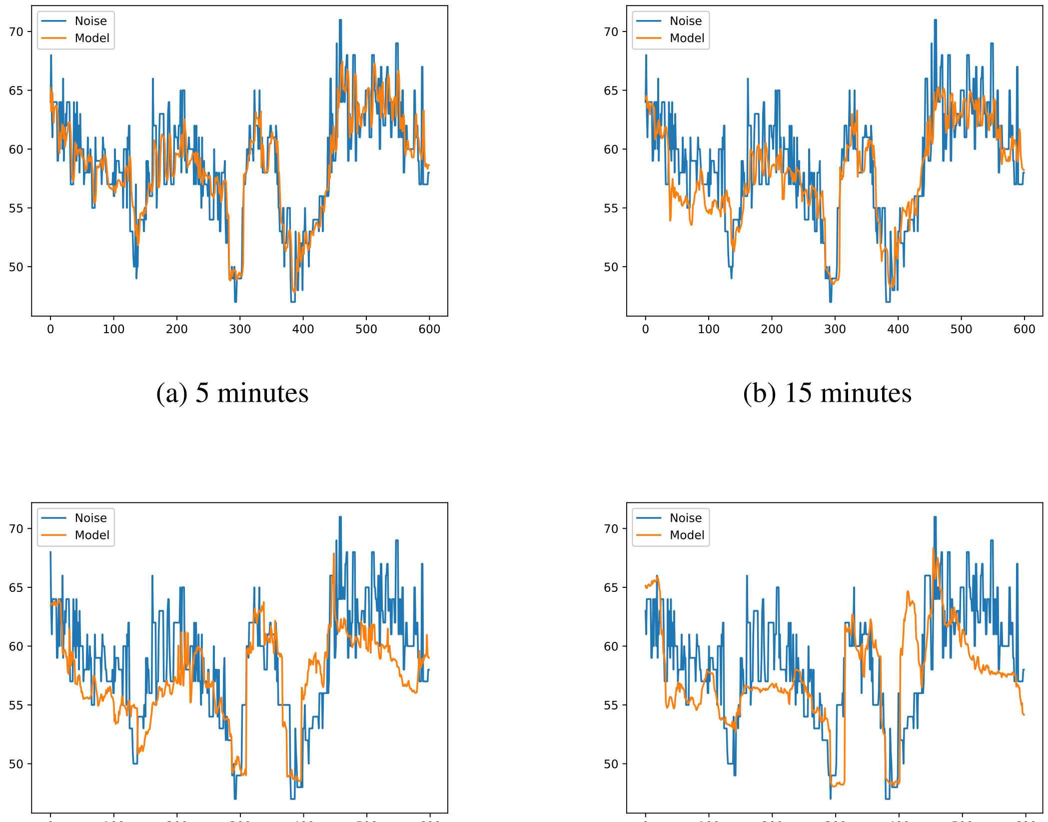


Figure 12. Example proposed model training

5.4.1 Comparison of data fusion methods

The proposed model, both with and without images as inputs, is compared to the relevant baselines to evaluate the impact of the data fusion strategy. Table 6 combines the accuracy results for the experiments. Only the best-performing model of each type is given. Overall, on average the proposed model can outperform all the other baselines. As expected, the model works best when predicting just one-time steps into the future. The proposed model achieves the best average RMSE (3.113), MSE (11.000), MAE (2.456), and MAPE (0.042). When predicting 5 minutes or 15 minutes into the future, the proposed model achieves low error rates with RMSE values of 2.220 and 2.788 respectively. These values outperform other baselines by a huge margin. When predicting 30 minutes or 60 minutes into the future, the simplest *Univariate, no fusion* manages to outperform the proposed



(a) 5 minutes (b) 15 minutes

(c) 30 minutes (d) 60 minutes

Figure 13. Proposed model predictions for different output steps

model by a relatively low margin. The proposed model without images is also performing exceptionally well overall. On average, it outperforms all the other models, excluding the proposed model with images and the univariate model, while. This indicates that our feature fusion approach is the most impactful part of our model.

When comparing the proposed approach without decision fusion to the no-fusion approach, we see that our data fusion strategy has improved the accuracy by a high margin. On average, the proposed model RMSE is lower by 0.381 than the no-fusion approach. This indicates that our data fusion strategy is giving the expected results and significantly impacting the predictions.

Method Feature fusion × Decision fusion	Prediction length	RMSE	MSE	MAE	MAPE
Univariate, no fusion	5min	3.038	9.23	2.322	0.039
	15min	3.624	13.136	2.875	0.048
	30min	**3.343**	**11.177**	**2.668**	**0.046**
NONE × NONE	60min	**3.647**	**13.303**	**2.894**	**0.050**
	AVG	3.413	11.712	2.690	0.046
No fusion	5min	3.100	9.611	2.434	0.041
	15min	4.403	19.385	3.524	0.059
	30min	4.320	18.666	3.362	0.056
NONE × NONE	60min	3.691	13.622	2.976	**0.050**
	AVG	3.879	15.321	3.074	0.052
Combined model with images	5min	38.033	1446.497	9.431	0.157
	15min	50.747	2575.249	7.522	7.522
	30min	13.430	180.354	4.824	0.082
SKF × NONE	60min	6.335	40.133	4.527	0.078
	AVG	27.136	1060.558	6.576	1.960
Only images	5min	6.147	37.783	4.946	0.083
	15min	7.411	54.927	6.115	0.102
	30min	6.946	48.252	5.633	0.094
	60min	6.886	52.982	5.677	0.094
	AVG	6.848	48.486	5.593	0.093
Proposed model	5min	**2.220**	**4.928**	**1.677**	**0.029**
	15min	**2.788**	**7.773**	**2.228**	**0.038**
	30min	3.520	12.390	2.876	0.049
SKF × SVR	60min	3.923	15.390	3.043	0.052
	AVG	**3.113**	**10.120**	**2.456**	**0.042**
Proposed model, no decision fusion	5min	2.598	6.752	1.970	0.033
	15min	3.083	9.504	2.451	0.041
	30min	3.979	15.836	3.246	0.054
SKF × NONE	60min	4.332	18.768	3.419	0.057
	AVG	3.498	12.715	2.772	0.046
UKF × UKF	5min	4.410	19.452	3.467	0.057
	15min	5.473	29.951	4.444	0.074
	30min	4.807	23.104	3.887	0.065
	60min	5.285	27.929	4.173	0.069
	AVG	4.994	25.109	3.993	0.066
KF × SVR	5min	2.875	8.268	2.203	0.037
	15min	3.136	9.836	2.506	0.042
	30min	4.104	16.846	3.358	0.057
	60min	4.443	19.740	3.500	0.059
	AVG	3.640	13.673	2.892	0.049

Table 6. Data fusion accuracy comparison

When evaluating the performance and computational expensiveness of the models in Table 7, we immediately see that models with image inputs have many more trainable parameters and, therefore, longer training times. However, when comparing the *No-fusion* approach with the proposed model with no decision fusion, it is clear that our fusion strategy is not only improving the accuracy of the predictions but also making the model computationally less expensive. The data fusion strategy reduced the time to train in our experiments from 169 seconds to 57 seconds and reduced the number of trainable parameters by 112. Yet, here we see that the most simple model that is based only on the target variable, *Univariate, no fusion*, is computationally the most performing.

Method	No. trainable params	Time to train (s)
Univariate, no fusion	11 459	28
No fusion	91 915	169
Only images	137 128 974	9498
Proposed model	137 220 249	9555
Proposed model, no decision fusion	91 275	57

Table 7. Data fusion performance comparison

5.4.2 Comparison of time series methods

When comparing the proposed approach to statistical time series methods in Table 8, it is shown that the proposed approach outperforms all the other methods. When comparing to *Moving average*, the margin of outperformance is slightly small (RMSE 3.113 vs. RMSE 3.260). When predicting 30 or 60 minutes ahead, the *Moving average* can beat our approach. The most significant factor of the high performance of these straightforward methods like *Naive* and *Moving average* is the handling of missing values described in Chapter 3.3.1. The simplistic backfill approach with a limit of 6 heavily favors these methods. *Linear regression* shows the poorest results, with an RMSE value of 12.30.

Method	Prediction length	RMSE	MSE	MAE	MAPE
Naive	5min	3.007	9.045	1.955	0.033
	15min	3.516	12.362	2.582	0.044
	30min	3.824	14.624	3.025	0.052
	60min	4.053	16.423	3.169	0.054
	AVG	3.6	13.114	2.683	0.046
Moving average	5min	2.783	7.744	2.202	0.038
	15min	3.045	9.270	2.429	0.042
	30min	**3.379**	**11.420**	**2.719**	**0.047**
	60min	**3.832**	**14.683**	3.059	0.053
	AVG	3.260	10.779	2.602	0.045
ARIMA	5min	3.047	9.286	2.164	0.037
	15min	3.399	11.550	2.584	0.044
	30min	3.765	14.177	2.962	0.051
	60min	4.031	16.247	3.168	0.055
	AVG	3.561	12.815	2.720	0.047
Linear regression	5min	3.169	10.044	2.360	0.040
	15min	5.344	28.560	3.203	0.055
	30min	16.779	281.548	5.027	0.090
	60min	23.906	571.497	9.072	0.164
	AVG	12.300	222.912	4.916	0.087
Proposed model SKF $\times$ SVR	5min	**2.220**	**4.928**	**1.677**	**0.029**
	15min	**2.788**	**7.773**	**2.228**	**0.038**
	30min	3.520	12.390	2.876	0.049
	60min	3.923	15.390	**3.043**	**0.052**
	AVG	**3.113**	**10.120**	**2.456**	**0.042**

Table 8. Proposed approach comparison with statistical time series methods

5.4.3 Sequence length impact on model performance

To measure the models' ability to generalize to the time series data, experiments with different input sequence lengths were carried out. Longer training sequences make models computationally more expensive and more reliant on data quality. Three input sequences were tested with the time horizon of $6, 12$, and 24 steps that represent 30-minute, 60-minute, and 120-minute look-back times respectively. The accuracy metrics are presented in Table 9. On average the model trained on an input sequence of 12 shows the smallest RMSE value of 3.113 when compared to others. In Figure 14 the models' performance is compared with all the output time perspectives. When predicting 6 steps or 30 minutes into the future, the model with a time series input sequence of 6 manages to slightly improve upon the 12-input model, with an RMSE improvement of 0.012. Input sequence 12 shows the lowest error value for all other output horizons.

Input sequence length	Prediction length	RMSE	MSE	MAE	MAPE
6	5min	2.362	5.579	1.700	**0.029**
	15min	3.166	10.024	2.458	0.042
	30min	**3.508**	**12.306**	**2.771**	**0.047**
	60min	4.656	21.678	3.551	0.060
	AVG	3.423	12.397	2.620	0.045
12	5min	**2.220**	**4.928**	**1.677**	**0.029**
	15min	**2.788**	**7.773**	**2.228**	**0.038**
	30min	3.520	12.390	2.876	0.049
	60min	**3.923**	**15.390**	**3.043**	**0.052**
	AVG	**3.113**	**10.120**	**2.456**	**0.042**
24	5min	2.730	7.454	2.077	0.035
	15min	3.472	12.055	2.929	0.049
	30min	4.659	21.707	3.896	0.066
	60min	4.153	17.246	3.317	0.056
	AVG	3.754	14.616	3.055	0.052

Table 9. Comparison of input sequence length on training the proposed model

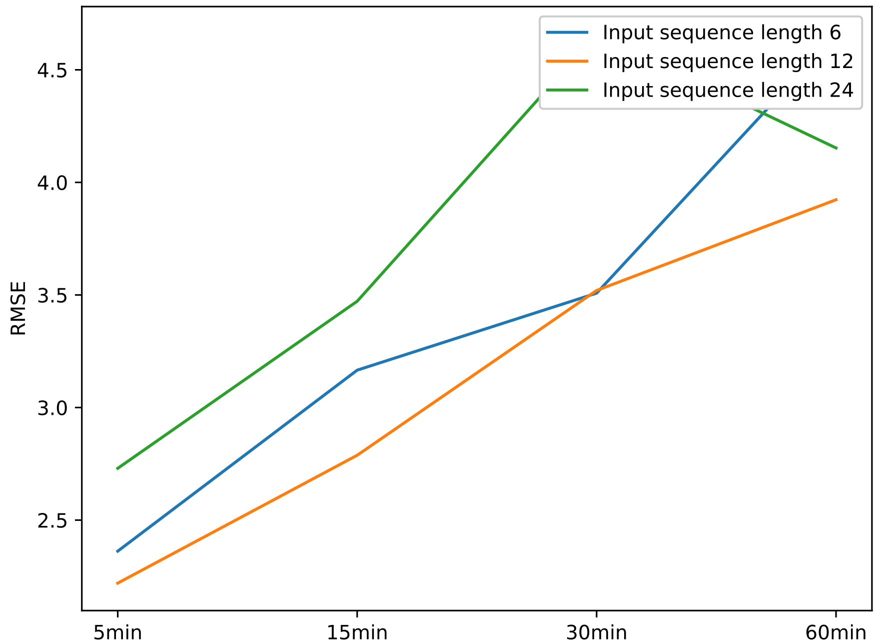


Figure 14. Comparison of time series input sequence length

Among all the input sequences and baseline methods, the proposed approach with an input sequence of 12 shows the best accuracy with a low average RMSE value of 3.113.

6. Future work and applications

The proposed data fusion strategy greatly improved the accuracy and performance of the final prediction model when compared to the one without fusion. However, there are many aspects that could be implemented to further improve. The deep learning prediction model could be further fine-tuned and tested to further improve the performance and have a greater advantage over the statistical time series prediction models, especially when predicting 30 or 60 minutes ahead. Experiments with smaller sequence lengths can improve the model used in the real-world setting, allowing us to build a more robust model.

The proposed data fusion strategies should be tested upon other datasets, in the context of ITS and outside it. The results of this could be a generalized data fusion technique that works across many problem domains.

One of the applications of this work is an input to a full-scale application for city stakeholders called Urban Mobility Hub. The proposed prediction model is integrated into the dashboard that supports city stakeholders to make further business decisions. Moreover, it is possible to build preventive applications that react to the predictions of urban noise increases.

7. Summary

The thesis aimed to analyze and propose an efficient traffic data fusion strategy with a prediction model to present accurate short-term urban noise predictions. An extensive data acquisition was carried out over a period of 1 month. The biggest challenge from the acquired data set was the missing values of the target variable, urban noise level.

The data fusion strategy was implemented using a hybrid approach containing a mixture of feature fusion and decision fusion algorithms. For feature fusion, a strategy implementing a Smoothed Kalman Filter was used to deal with the data unreliability and simultaneously reduce the model's complexity. For models that include images from cameras, a decision fusion based on a Support Vector Regression was applied to improve the accuracy of the final prediction further.

A CNN-LSTM deep learning network was used to evaluate the proposed fusion strategies. An extensive amount of data fusion and statistical time series methods were evaluated as baselines to confirm the proposed approach's validity. Evaluations were based on two criteria: the predictions' accuracy and the model's complexity.

The proposed model achieved the best accuracy among the baselines irrelevant to the sequence length of the experiment. The proposed approach without images showed the great aspect of DF, where experiment training times were reduced by three times and, on the other hand, significantly improved accuracy of the results for more than 10% when comparing against no fusion baseline. The proposed model with images and decision fusion outperformed the one without images by a relatively small margin; on average RMSE decreased by 0.385. However, adding images added a lot of complexity, and training time increased significantly. The proposed approach showed the best performance when predicting a one-time step 5 minutes ahead. On average, a simple statistical time series prediction method *Moving average* outperformed the proposed model when predicting 30 or 60 minutes ahead due to the simplistic approach of filling in missing values. This opens up opportunities for future improvements like CNN-LSTM network fine-tuning or reducing the sequence length to increase the reliability against missing values.

8. Acknowledgements

I would like to sincerely thank my supervisor and co-supervisor for extensive support, feedback, guidance, and motivation throughout the whole process. In addition, I would like to thank Jaanus Kaugerand and Thinnect for providing access to their sensor data deployed in the city of Tallinn.

I would like to acknowledge TalTech AI-Lab for the extensive use of computational resources for running experiments.

Bibliography

[1] C. Ounoughi and S. B. Yahia. “Data fusion for ITS: A Systematic Literature Review”. In: *Submitted to Information fusion journal* (2021).

[2] United Nations. “68% of the world population projected to live in urban areas by 2050, says UN”. In: (2018).

[3] N. Mucci et al. “Urban Noise and Psychological Distress: A Systematic Review”. In: *Int J Environ Res Public Health* (2020).

[4] *DIRECTIVE 2010/40/EU OF THE EUROPEAN PARLIAMENT AND OF THE COUNCIL of 7 July 2010*. `https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A02010L0040-20180109`. 2010.

[5] M. M. Alyannezhadi, A. A. Pouyan, and V. Abolghasemi. “An efficient algorithm for multisensory data fusion under uncertainty condition”. In: *Journal of Electrical Systems and Information Technology* (2016).

[6] R. Gravina et al. “Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges”. In: *Information Fusion* (2016).

[7] A. Sao, N. Tempelmeier, and E. Demidova. “Deep Information Fusion for Electric Vehicle Charging Station Occupancy Forecasting”. In: (2021).

[8] T. Adetiloye and A. Awasthi. “Multimodal Big Data Fusion for Traffic Congestion Prediction”. In: (2019).

[9] M. Xu, J. Wang, and S. Gu. “Rapid identification of tea quality by E-nose and computer vision combining with a synergetic data fusion strategy”. In: *Journal of Food Engineering* (2019).

[10] L. Ghouti. “Mobility prediction in mobile ad hoc networks using neural learning machines”. In: *Simulation Modelling Practice and Theory* (2016).

[11] A. Anand, G. Ramadurai, and L. Vanajakshi. “Data Fusion-Based Traffic Density Estimation and Prediction”. In: *Journal of Intelligent Transportation Systems* (2014).

[12] S. Khan and S. Nazir. “Deep learning-based urban big data fusion in smart cities: Towards traffic monitoring and flow-preserving fusion”. In: *Computers and Electrical Engineering* (2021).

[13] H.Widiputra, A. Mailangkay, and E.Gautama. “Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction”. In: *Hindawi* (2021).

[14] I. M. Baytas et al. “Patient Subtyping via Time-Aware LSTM Networks”. In: *Proceedings of KDD ’17* (2017).

[15] *TomTom Developer API*. `https://developer.tomtom.com/`.

[16] A. Mouakher et al. “EXPECT: EXplainable Prediction Model for Energy ConsumpTion”. In: *Mathematics* (2022).

[17] A. Harway. *Handbook of Economic Forecasting, Volume 1, (Pages 327-412)*. 2006. (accessed: 02.11.2022).

[18] S. M. Kazemi et al. “Time2Vec: Learning a Vector Representation of Time”. In: *arXiv:1907.05321* (2019).

[19] D. Osinga. *Deep Learning Cookbook, Volume 1*. O’Reilly Media, Inc., 2018. ISBN: 9781491995846.

[20] P. Rivas-Perea et al. “Support Vector Machines for Regression: A Succinct Review of Large-Scale and Linear Programming Formulations”. In: *International Journal of Intelligence Science, 2013, 3, 5-14* (2012).

[21] R. J. Hyndman and A. B. Koehler. “Another look at measures of forecast accuracy”. In: *International Journal of Forecasting* (2006).

[22] A. Myttenaere, B. Golden, and B. Le Grand. “Mean Absolute Percentage Error for regression models”. In: *Neurocomputing* (2016).

[23] R. M. Stein. “Benchmarking default prediction models: pitfalls and remedies in model validation”. In: *Journal of Risk Model Validation* (2007).

[24] L. Börjesson and M. Singull. “Forecasting Financial Time Series through Causal and Dilated Convolutional Neural Networks”. In: *Entropy (Basel)* (2020).

[25] S. J. Julier and J. K. Uhlmann. “A New Extension of the Kalman Filter to Nonlinear Systems”. In: *Proceedings Volume 3068, Signal Processing, Sensor Fusion, and Target Recognition VI* (1997).

[26] R.J. Hyndman and G. Athanasopoulos. *Forecasting: principles and practice, 2nd edition*. OTexts: Melbourne, Australia, 2018. (accessed: 25.10.2022).

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

I Andres Suislepp

1. Grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis "Multi-Modal Data Fusion for Short-Term Urban Noise Prediction in Intelligent Transportation Systems" , supervised by Sadok Ben Yahia
   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.

02.01.2023

Appendix 2 - Results

Method **Feature fusion × Decision fusion**	**Input seq.**	**Output step**	**RMSE** **MSE** **MAE** **MAPE**	**Time** **Trainable params**
Combined model SKF × NONE	12	1	38.033 2575.249 9.431 0.157	3075 29,205,074
Images only -	12	1	6.147 37.783 4.946 0.083	7043 137,125,726
SKF × NONE	12	1	2.598 6.752 1.970 0.033	57 91,275
Univariate NONE × NONE	12	1	3.038 9.230 2.322 0.039	28 o,459
KF × NONE	12	1	2.970 8.819 2.275 0.038	34 91,275
Images only -	12	1	6.147 37.783 4.946 0.083	14086 137,125,726
SKF × NONE	12	1	2.598 6.752 1.970 0.033	57 91,275

Univariate NONE × NONE	12	1	3.038 9.230 2.322 0.039	28 11,459
KF × NONE	12	1	2.970 8.819 2.275 0.038	34 91,275
UKF × NONE	12	1	4.186 17.525 3.332 0.056	36 91,275
No fusion NONE × NONE	12	1	3.100 9.611 2.434 0.041	169 91,915
Naive -	12	1	3.007 9.045 1.955 0.033	0 -
Moving average -	12	1	2.783 7.744 2.202 0.038	0 -
Linear regression -	12	1	3.169 10.044 2.360 0.040	0 -
ARIMA -	12	1	3.047 9.281 2.164 0.037	0 -
Univariate + DF NONE × AVG	12	1	4.143 17.166 3.256 0.054	7877 -
Univariate + DF NONE × KF	12	1	6.019 36.230	7877 -

			4.819 0.081	
Univariate + DF NONE × SKF	12	1	6.071 36.854 4.874 0.081	7880 -
Univariate + DF NONE × UKF	12	1	4.126 17.023 3.236 0.054	7877 -
Univariate + DF NONE × SVR	12	1	2.996 8.974 2.388 0.041	7877 -
No fusion + DF NONE × AVG	12	1	4.092 16.741 3.233 0.054	8018 -
No fusion + DF NONE × KF	12	1	6.019 36.230 4.819 0.081	8018 -
No fusion + DF NONE × SKF	12	1	6.071 36.856 4.874 0.081	8021 -
No fusion + DF NONE × UKF	12	1	4.065 16.521 3.206 0.053	8019 -
No fusion + DF NONE × SVM	12	1	2.867 8.218 2.229 0.038	8019 -
SKF × AVG	12	1	3.919 15.357 3.104 0.052	7907 -
	12	1	6.019	7907

SKF × KF			36.230 4.819 0.081	-
SKF × SKF	12	1	6.070 36.847 4.874 0.081	7909 -
SKF × UKF	12	1	3.908 15.275 3.086 0.051	7907 -
SKF × SVM	12	1	2.220 4.928 1.677 0.029	7907 -
UKF × AVG	12	1	4.443 19.736 3.502 0.058	7885 -
UKF × KF	12	1	6.019 36.230 4.819 0.081	7885 -
UKF × SKF	12	1	6.070 36.848 4.874 0.081	7887 -
UKF × UKF	12	1	4.410 19.452 3.467 0.057	7885 -
UKF × SVM	12	1	3.600 12.957 2.831 0.048	7885 -
KF × AVG	12	1	4.037 16.299 3.176	7883 -

			0.053	
KF × KF	12	1	6.019 36.230 4.819 0.081	7883 -
KF × SKF	12	1	6.071 36.859 4.874 0.081	7886 -
KF × UKF	12	1	4.015 16.123 3.153 0.052	7883 -
KF × SVM	12	1	2.875 8.268 2.203 0.037	7883 -
Combined model SKF × NONE	12	3	50.747 2575.249 7.522 0.127	3083 29,205,092
Images only -	12	3	7.411 54.927 6.115 0.102	7848 137,125,744
SKF × NONE	12	3	3.083 9.504 2.451 0.041	60 91,297
Univariate NONE × NONE	12	3	3.624 13.136 2.875 0.048	27 11,481
KF × NONE	12	3	3.312 10.970 2.639 0.044	38 91,297
UKF × NONE	12	3	4.841 23.433	39 91,297

			3.948 0.066	
No fusion NONE × NONE	12	3	4.403 19.385 3.524 0.059	186 91,937
Naive -	12	3	3.516 12.362 2.582 0.044	0 -
Moving average -	12	3	3.045 9.270 2.429 0.042	0 -
Linear regression -	12	3	5.344 28.560 3.203 0.055	0 -
ARIMA -	12	3	3.398 11.545 2.583 0.044	0 -
Univariate + DF NONE × AVG	12	3	4.897 23.976 3.910 0.065	7875 -
Univariate + DF NONE × KF	12	3	7.278 52.966 5.974 0.100	7875 -
Univariate + DF NONE × SKF	12	3	7.341 53.894 6.050 0.101	7878 -
Univariate + DF NONE × UKF	12	3	4.861 23.634 3.888 0.065	7875 -
Univariate + DF	12	3	3.262	7875

NONE × SVR			10.639 2.592 0.044	-
No fusion + DF NONE × AVG	12	3	5.355 28.681 4.335 0.072	8034 -
No fusion + DF NONE × KF	12	3	7.278 52.966 5.974 0.100	8034 -
No fusion + DF NONE × SKF	12	3	7.342 53.905 6.050 0.101	8037 -
No fusion + DF NONE × UKF	12	3	5.324 28.341 4.298 0.071	8034 -
No fusion + DF NONE × SVM	12	3	4.208 17.707 3.346 0.056	8034 -
SKF × AVG	12	3	4.632 21.457 3.749 0.062	7909 -
SKF × KF	12	3	7.278 52.966 5.974 0.100	7909 -
SKF × SKF	12	3	7.342 53.904 6.050 0.101	7911 -
SKF × UKF	12	3	4.601 21.167 3.723	7909 -

			0.062	
SKF × SVM	12	3	2.788 7.773 2.228 0.038	7909 -
UKF × AVG	12	3	5.504 30.297 4.471 0.075	7887 -
UKF × KF	12	3	7.278 52.966 5.974 0.100	7887 -
UKF × SKF	12	3	7.338 53.851 6.048 0.101	7890 -
UKF × UKF	12	3	5.473 29.951 4.444 0.074	7887 -
UKF × SVM	12	3	4.692 22.013 3.751 0.063	7887 -
KF × AVG	12	3	4.808 23.115 3.890 0.065	7886 -
KF × KF	12	3	7.278 52.966 5.974 0.100	7886 -
KF × SKF	12	3	7.342 53.911 6.050 0.101	7889 -
KF × UKF	12	3	4.774 22.791	7886 -

			3.866 0.064	
KF × SVM	12	3	3.136 9.836 2.506 0.042	7886 -
Combined model SKF × NONE	12	6	13.430 180.354 4.824 0.082	3096 29,205,119
Images only -	12	6	6.946 48.252 5.633 0.094	7849 137,125,771
SKF × NONE	12	6	3.979 15.836 3.246 0.054	61 91,330
Univariate NONE × NONE	12	6	3.343 11.177 2.668 0.046	26 11,514
KF × NONE	12	6	4.275 18.274 3.473 0.059	37 91,330
UKF × NONE	12	6	4.939 24.395 3.905 0.065	39 91,330
No fusion NONE × NONE	12	6	4.320 18.666 3.362 0.056	160 91,970
Naive -	12	6	3.824 14.624 3.025 0.052	0 -
Moving average	12	6	3.379	0

-			11.420 2.719 0.047	-
Linear regression -	12	6	16.779 281.548 5.027 0.090	0 -
ARIMA -	12	6	3.769 14.207 2.968 0.051	0 -
Univariate + DF NONE × AVG	12	6	4.200 17.643 3.339 0.056	7875 -
Univariate + DF NONE × KF	12	6	6.652 44.247 5.445 0.090	7875 -
Univariate + DF NONE × SKF	12	6	6.600 43.564 5.413 0.090	7878 -
Univariate + DF NONE × UKF	12	6	4.117 16.949 3.295 0.055	7875 -
Univariate + DF NONE × SVR	12	6	3.465 12.008 2.754 0.047	7875 -
No fusion + DF NONE × AVG	12	6	4.673 21.837 3.744 0.062	8009 -
No fusion + DF NONE × KF	12	6	6.652 44.247 5.445	8009 -

			0.090	
No fusion + DF NONE × SKF	12	6	6.601 43.574 5.414 0.090	8012 -
No fusion + DF NONE × UKF	12	6	4.590 21.066 3.668 0.061	8009 -
No fusion + DF NONE × SVM	12	6	4.121 16.981 3.206 0.054	8009 -
SKF × AVG	12	6	4.614 21.291 3.760 0.063	7909 -
SKF × KF	12	6	6.652 44.247 5.445 0.090	7909 -
SKF × SKF	12	6	6.604 43.609 5.417 0.090	7912 -
SKF × UKF	12	6	4.532 20.535 3.694 0.062	7910 -
SKF × SVM	12	6	3.520 12.390 2.876 0.049	7910 -
UKF × AVG	12	6	4.889 23.898 3.963 0.066	7888 -
UKF × KF	12	6	6.652 44.247	7888 -

			5.445 0.090	
UKF × SKF	12	6	6.604 43.611 5.417 0.090	7891 -
UKF × UKF	12	6	4.807 23.104 3.887 0.065	7888 -
UKF × SVM	12	6	4.587 21.043 3.600 0.060	7888 -
KF × AVG	12	6	4.816 23.195 3.967 0.066	7886 -
KF × KF	12	6	6.652 44.247 5.445 0.090	7886 -
KF × SKF	12	6	6.604 43.607 5.417 0.090	7889 -
KF × UKF	12	6	4.732 22.391 3.909 0.065	7887 -
KF × SVM	12	6	4.104 16.846 3.358 0.057	7887 -
Combined model SKF × NONE	12	12	6.335 40.133 4.527 0.078	3060 29,205,173
Images only	12	12	7.279	7845

-			52.982 5.677 0.094	137,125,825
SKF × NONE	12	12	4.332 18.768 3.419 0.057	61 91,396
Univariate NONE × NONE	12	12	3.647 13.303 2.894 0.050	27 11,580
KF × NONE	12	12	4.521 20.436 3.578 0.060	38 91,396
UKF × NONE	12	12	4.359 19.004 3.447 0.058	40 91,396
No fusion NONE × NONE	12	12	3.691 13.622 2.976 0.050	185 92,036
Naive -	12	12	4.053 16.423 3.169 0.054	0 -
Moving average -	12	12	3.832 14.683 3.059 0.053	0 -
Linear regression -	12	12	23.906 571.497 9.072 0.164	0 -
ARIMA -	12	12	4.035 16.284 3.169	0 -

			0.055	
Univariate + DF NONE × AVG	12	12	4.766 22.714 3.715 0.062	7872 -
Univariate + DF NONE × KF	12	12	7.104 50.463 5.541 0.091	7872 -
Univariate + DF NONE × SKF	12	12	7.210 51.988 5.621 0.093	7875 -
Univariate + DF NONE × UKF	12	12	4.733 22.402 3.691 0.061	7872 -
Univariate + DF NONE × SVR	12	12	3.586 12.857 2.873 0.049	7872 -
No fusion + DF NONE × AVG	12	12	4.898 23.991 3.937 0.065	8030 -
No fusion + DF NONE × KF	12	12	7.104 50.463 5.541 0.091	8030 -
No fusion + DF NONE × SKF	12	12	7.214 52.038 5.623 0.093	8033 -
No fusion + DF NONE × UKF	12	12	4.853 23.553 3.904 0.065	8030 -
No fusion + DF NONE × SVM	12	12	3.686 13.587	8030 -

			2.974 0.050	
SKF × AVG	12	12	5.253 27.595 4.163 0.069	7906 -
SKF × KF	12	12	7.104 50.463 5.541 0.091	7906 -
SKF × SKF	12	12	7.214 52.042 5.624 0.093	7908 -
SKF × UKF	12	12	5.212 27.165 4.138 0.068	7906 -
SKF × SVM	12	12	3.923 15.390 3.043 0.052	7906 -
UKF × AVG	12	12	5.322 28.329 4.206 0.069	7885 -
UKF × KF	12	12	7.104 50.463 5.541 0.091	7885 -
UKF × SKF	12	12	7.214 52.036 5.623 0.093	7887 -
UKF × UKF	12	12	5.285 27.929 4.173 0.069	7885 -
	12	12	4.052	7885

UKF × SVM			16.422 3.093 0.052	-
KF × AVG	12	12	5.384 28.990 4.279 0.071	7883 -
KF × KF	12	12	7.104 50.463 5.541 0.091	7883 -
KF × SKF	12	12	7.213 52.032 5.623 0.093	7885 -
KF × UKF	12	12	5.344 28.557 4.248 0.070	7883 -
KF × SVM	12	12	4.443 19.740 3.500 0.059	7883 -
Combined model SKF × NONE	6	1	31.683 1003.796 12.469 0.209	2569 29,204,756
Images only -	6	1	306.261 93795.544 28.870 0.471	7559 137,125,726
SKF × NONE	6	1	2.373 5.629 1.730 0.029	59 90,765
Univariate NONE × NONE	6	1	4.081 16.653 3.357	25 11,459

			0.056	
KF × NONE	6	1	2.674 7.150 2.017 0.034	37 90,765
UKF × NONE	6	1	3.421 11.705 2.688 0.046	39 90,765
No fusion NONE × NONE	6	1	2.889 8.346 2.232 0.038	200 91,405
Naive -	6	1	2.941 8.648 1.911 0.032	0 -
Moving average -	6	1	2.815 7.925 2.123 0.036	0 -
Linear regression -	6	1	3.315 10.989 2.351 0.040	0 -
ARIMA -	6	1	3.040 9.243 2.116 0.036	0 -
Univariate + DF NONE × AVG	6	1	152.948 23393.109 15.043 0.246	7584 -
Univariate + DF NONE × KF	6	1	218.690 47825.252 28.434 0.464	7584 -
Univariate + DF NONE × SKF	6	1	129.615 16800.019	7587 -

			28.363 0.463	
Univariate + DF NONE × UKF	6	1	119.881 14371.547 14.779 0.241	7584 -
Univariate + DF NONE × SVR	6	1	2.897 8.395 2.240 0.039	7584 -
No fusion + DF NONE × AVG	6	1	153.097 23438.658 14.996 0.245	7758 -
No fusion + DF NONE × KF	6	1	218.690 47825.252 28.434 0.464	7759 -
No fusion + DF NONE × SKF	6	1	129.264 16709.101 27.476 0.447	7762 -
No fusion + DF NONE × UKF	6	1	120.066 14415.884 14.797 0.242	7759 -
No fusion + DF NONE × SVM	6	1	2.761 7.625 2.115 0.036	7759 -
SKF × AVG	6	1	153.119 23445.566 14.882 0.243	7618 -
SKF × KF	6	1	218.690 47825.252 28.434 0.464	7618 -
	6	1	129.260	7621

SKF × SKF			16708.248 27.471 0.447	-
SKF × UKF	6	1	120.095 14422.732 14.721 0.241	7618 -
SKF × SVM	6	1	2.362 5.579 1.700 0.029	7618 -
UKF × AVG	6	1	153.059 23427.083 15.150 0.248	7597 -
UKF × KF	6	1	218.690 47825.252 28.434 0.464	7597 -
UKF × SKF	6	1	129.420 16749.448 27.826 0.453	7601 -
UKF × UKF	6	1	120.021 14404.954 14.938 0.244	7598 -
UKF × SVM	6	1	3.312 10.971 2.570 0.044	7598 -
KF × AVG	6	1	153.075 23431.838 14.990 0.245	7596 -
KF × KF	6	1	218.690 47825.252 28.434	7596 -

			0.464	
KF × SKF	6	1	129.263 16708.833 27.470 0.447	7599 -
KF × UKF	6	1	120.037 14408.938 14.802 0.242	7596 -
KF × SVM	6	1	2.668 7.120 1.997 0.034	7596 -
Combined model SKF × NONE	6	3	8.158 66.549 6.451 0.107	2598 29,204,774
Images only -	6	3	5.929 35.155 4.759 0.080	7560 137,125,744
SKF × NONE	6	3	3.198 10.230 2.467 0.042	60 90,787
Univariate NONE × NONE	6	3	3.341 11.161 2.635 0.044	26 11,481
KF × NONE	6	3	3.413 11.652 2.696 0.045	38 90,787
UKF × NONE	6	3	4.182 17.489 3.302 0.055	39 90,787
No fusion NONE × NONE	6	3	3.481 12.117	181 91,427

			2.789 0.047	
Naive -	6	3	3.465 12.007 2.576 0.044	0 -
Moving average -	6	3	3.141 9.867 2.470 0.042	0 -
Linear regression -	6	3	12.176 148.266 4.211 0.073	0 -
ARIMA -	6	3	3.585 12.851 2.698 0.046	0 -
Univariate + DF NONE × AVG	6	3	4.223 17.830 3.408 0.057	7586 -
Univariate + DF NONE × KF	6	3	5.731 32.845 4.611 0.077	7586 -
Univariate + DF NONE × SKF	6	3	5.800 33.635 4.657 0.078	7589 -
Univariate + DF NONE × UKF	6	3	4.176 17.439 3.358 0.056	7586 -
Univariate + DF NONE × SVR	6	3	3.113 9.690 2.468 0.042	7586 -
No fusion + DF	6	3	4.206	7741

NONE $\times$ AVG			17.693 3.387 0.057	-
No fusion + DF NONE $\times$ KF	6	3	5.731 32.845 4.611 0.077	7741 -
No fusion + DF NONE $\times$ SKF	6	3	5.800 33.644 4.658 0.078	7745 -
No fusion + DF NONE $\times$ UKF	6	3	4.153 17.248 3.335 0.056	7742 -
No fusion + DF NONE $\times$ SVM	6	3	3.396 11.536 2.740 0.046	7742 -
SKF $\times$ AVG	6	3	4.140 17.139 3.332 0.056	7619 -
SKF $\times$ KF	6	3	5.731 32.845 4.611 0.077	7620 -
SKF $\times$ SKF	6	3	5.800 33.642 4.657 0.078	7623 -
SKF $\times$ UKF	6	3	4.089 16.723 3.282 0.055	7620 -
SKF $\times$ SVM	6	3	3.166 10.024 2.458	7620 -

			0.042	
UKF × AVG	6	3	4.606 21.216 3.697 0.062	7599 -
UKF × KF	6	3	5.731 32.845 4.611 0.077	7599 -
UKF × SKF	6	3	5.801 33.654 4.658 0.078	7603 -
UKF × UKF	6	3	4.555 20.748 3.647 0.061	7600 -
UKF × SVM	6	3	3.823 14.617 2.980 0.050	7599 -
KF × AVG	6	3	4.261 18.155 3.422 0.057	7598 -
KF × KF	6	3	5.731 32.845 4.611 0.077	7598 -
KF × SKF	6	3	5.801 33.647 4.658 0.078	7601 -
KF × UKF	6	3	4.209 17.713 3.371 0.056	7598 -
KF × SVM	6	3	3.210 10.302	7598 -

			2.537 0.043	
Combined model SKF × NONE	6	6	42.475 1804.108 9.881 0.166	2505 29,204,801
Images only -	6	6	8.738 76.350 7.244 0.120	7556 137,125,771
SKF × NONE	6	6	3.545 12.565 2.814 0.047	60 90,820
Univariate NONE × NONE	6	6	3.539 12.526 2.811 0.047	26 11,514
KF × NONE	6	6	4.050 16.400 3.264 0.055	37 90,820
UKF × NONE	6	6	4.113 16.916 3.157 0.054	39 90,820
No fusion NONE × NONE	6	6	3.766 14.180 2.968 0.050	202 91,460
Naive -	6	6	3.720 13.836 2.947 0.051	0 -
Moving average -	6	6	3.304 10.914 2.605 0.045	0 -
Linear regression	6	6	34.023	0

-			1157.550 8.256 0.148	-
ARIMA -	6	6	3.883 15.076 2.985 0.051	0 -
Univariate + DF NONE $\times$ AVG	6	6	5.635 31.758 4.572 0.076	7583 -
Univariate + DF NONE $\times$ KF	6	6	8.548 73.065 7.069 0.117	7583 -
Univariate + DF NONE $\times$ SKF	6	6	8.645 74.742 7.160 0.119	7586 -
Univariate + DF NONE $\times$ UKF	6	6	5.579 31.129 4.520 0.075	7583 -
Univariate + DF NONE $\times$ SVR	6	6	3.370 11.360 2.709 0.046	7583 -
No fusion + DF NONE $\times$ AVG	6	6	5.634 31.743 4.631 0.077	7758 -
No fusion + DF NONE $\times$ KF	6	6	8.548 73.065 7.069 0.117	7758 -
No fusion + DF NONE $\times$ SKF	6	6	8.648 74.789 7.162	7762 -

			0.119	
No fusion + DF NONE × UKF	6	6	5.578 31.109 4.571 0.076	7759 -
No fusion + DF NONE × SVM	6	6	3.637 13.226 2.888 0.049	7759 -
SKF × AVG	6	6	5.653 31.954 4.668 0.077	7616 -
SKF × KF	6	6	8.548 73.065 7.069 0.117	7617 -
SKF × SKF	6	6	8.649 74.801 7.163 0.119	7620 -
SKF × UKF	6	6	5.592 31.276 4.613 0.076	7617 -
SKF × SVM	6	6	3.508 12.306 2.771 0.047	7617 -
UKF × AVG	6	6	5.794 33.572 4.807 0.080	7596 -
UKF × KF	6	6	8.548 73.065 7.069 0.117	7596 -
UKF × SKF	6	6	8.646 74.750	7599 -

			7.160 0.119	
UKF × UKF	6	6	5.736 32.905 4.753 0.079	7596 -
UKF × SVM	6	6	4.224 17.842 3.211 0.055	7596 -
KF × AVG	6	6	5.929 35.148 4.891 0.081	7594 -
KF × KF	6	6	8.548 73.065 7.069 0.117	7594 -
KF × SKF	6	6	8.648 74.783 7.162 0.119	7597 -
KF × UKF	6	6	5.871 34.470 4.834 0.080	7594 -
KF × SVM	6	6	3.921 15.375 3.152 0.053	7594 -
Combined model SKF × NONE	6	12	7.685 59.063 5.712 0.095	2572 29,204,855
Images only -	6	12	6.796 46.182 5.570 0.093	7555 137,125,825
	6	12	4.783	61

SKF × NONE			22.877 3.713 0.063	90,886
Univariate NONE × NONE	6	12	3.690 13.619 2.959 0.051	26 11,580
KF × NONE	6	12	5.286 27.945 4.106 0.069	38 90,886
UKF × NONE	6	12	4.521 20.444 3.560 0.060	39 90,886
No fusion NONE × NONE	6	12	4.102 16.825 3.234 0.054	200 91,526
Naive -	6	12	4.003 16.023 3.149 0.054	0 -
Moving average -	6	12	3.885 15.093 3.131 0.054	0 -
Linear regression -	6	12	50.296 2529.661 14.990 0.272	0 -
ARIMA -	6	12	4.271 18.245 3.246 0.056	0 -
Univariate + DF NONE × AVG	6	12	4.703 22.116 3.819	7582 -

			0.064	
Univariate + DF NONE × KF	6	12	6.658 44.327 5.462 0.091	7582 -
Univariate + DF NONE × SKF	6	12	6.707 44.986 5.495 0.092	7585 -
Univariate + DF NONE × UKF	6	12	4.678 21.884 3.798 0.064	7582 -
Univariate + DF NONE × SVR	6	12	3.708 13.748 3.001 0.052	7582 -
No fusion + DF NONE × AVG	6	12	4.978 24.776 3.997 0.066	7756 -
No fusion + DF NONE × KF	6	12	6.658 44.327 5.462 0.091	7756 -
No fusion + DF NONE × SKF	6	12	6.707 44.989 5.495 0.092	7759 -
No fusion + DF NONE × UKF	6	12	4.942 24.421 3.970 0.066	7756 -
No fusion + DF NONE × SVM	6	12	4.101 16.818 3.215 0.054	7756 -
SKF × AVG	6	12	5.134 26.361	7616 -

			4.163 0.069	
SKF × KF	6	12	6.658 44.327 5.462 0.091	7616 -
SKF × SKF	6	12	6.704 44.950 5.492 0.092	7619 -
SKF × UKF	6	12	5.099 25.999 4.145 0.069	7616 -
SKF × SVM	6	12	4.656 21.678 3.551 0.060	7616 -
UKF × AVG	6	12	5.044 25.438 4.127 0.069	7594 -
UKF × KF	6	12	6.658 44.327 5.462 0.091	7594 -
UKF × SKF	6	12	6.706 44.964 5.493 0.092	7598 -
UKF × UKF	6	12	5.005 25.050 4.094 0.068	7595 -
UKF × SVM	6	12	4.663 21.747 3.509 0.060	7595 -
	6	12	5.347	7593

KF × AVG			28.588 4.426 0.074	-
KF × KF	6	12	6.658 44.327 5.462 0.091	7593 -
KF × SKF	6	12	6.704 44.948 5.492 0.092	7597 -
KF × UKF	6	12	5.308 28.172 4.402 0.074	7593 -
KF × SVM	6	12	5.023 25.233 3.821 0.065	7594 -
Combined model SKF × NONE	24	1	4.345 18.878 2.760 0.046	3187 29,205,926
Images only -	24	1	6.886 47.419 5.493 0.091	8048 137,125,726
SKF × NONE	24	1	2.859 8.173 2.231 0.037	54 92,511
Univariate NONE × NONE	24	1	3.164 10.013 2.412 0.040	25 11,459
KF × NONE	24	1	3.147 9.902 2.448	31 92,511

			0.041	
UKF × NONE	24	1	4.562 20.808 3.699 0.061	33 92,511
No fusion NONE × NONE	24	1	3.623 13.126 2.760 0.046	121 93,151
Naive -	24	1	3.163 10.008 2.059 0.034	0 -
Moving average -	24	1	3.088 9.535 2.489 0.042	0 -
Linear regression -	24	1	3.095 9.581 2.386 0.040	0 -
ARIMA -	24	1	3.036 9.218 2.186 0.037	0 -
Univariate + DF NONE × AVG	24	1	4.563 20.822 3.580 0.059	8073 -
Univariate + DF NONE × KF	24	1	6.704 44.946 5.389 0.089	8073 -
Univariate + DF NONE × SKF	24	1	6.643 44.133 5.365 0.089	8075 -
Univariate + DF NONE × UKF	24	1	4.527 20.493	8073 -

			3.562 0.059	
Univariate + DF NONE × SVR	24	1	2.904 8.433 2.319 0.039	8073 -
No fusion + DF NONE × AVG	24	1	4.687 21.964 3.726 0.061	8169 -
No fusion + DF NONE × KF	24	1	6.704 44.946 5.389 0.089	8169 -
No fusion + DF NONE × SKF	24	1	6.660 44.355 5.376 0.089	8171 -
No fusion + DF NONE × UKF	24	1	4.643 21.556 3.691 0.061	8169 -
No fusion + DF NONE × SVM	24	1	3.580 12.817 2.787 0.046	8169 -
SKF × AVG	24	1	4.247 18.038 3.302 0.055	8102 -
SKF × KF	24	1	6.704 44.946 5.389 0.089	8103 -
SKF × SKF	24	1	6.651 44.236 5.371 0.089	8104 -
	24	1	4.216	8103

SKF × UKF			17.776 3.287 0.054	-
SKF × SVM	24	1	2.730 7.453 2.077 0.035	8103 -
UKF × AVG	24	1	5.243 27.492 4.233 0.070	8081 -
UKF × KF	24	1	6.704 44.946 5.389 0.089	8081 -
UKF × SKF	24	1	6.664 44.413 5.379 0.089	8083 -
UKF × UKF	24	1	5.199 27.027 4.211 0.069	8081 -
UKF × SVM	24	1	4.025 16.199 3.171 0.053	8081 -
KF × AVG	24	1	4.259 18.135 3.314 0.055	8080 -
KF × KF	24	1	6.704 44.946 5.389 0.089	8080 -
KF × SKF	24	1	6.656 44.303 5.374	8081 -

			0.089	
KF × UKF	24	1	4.219 17.802 3.286 0.054	8080 -
KF × SVM	24	1	3.059 9.359 2.378 0.040	8080 -
Combined model SKF × NONE	24	3	55.081 3033.900 7.045 0.117	3215 29,205,944
Images only -	24	3	5.311 28.210 4.170 0.070	8048 137,125,744
SKF × NONE	24	3	3.481 12.118 2.886 0.049	57 92,533
Univariate NONE × NONE	24	3	3.938 15.505 3.100 0.051	24 11,481
KF × NONE	24	3	3.501 12.255 2.856 0.048	34 92,533
UKF × NONE	24	3	4.731 22.378 3.783 0.063	36 92,533
No fusion NONE × NONE	24	3	3.808 14.501 3.066 0.051	138 93,173
Naive -	24	3	3.703 13.709	0 -

			2.719 0.045	
Moving average -	24	3	3.335 11.119 2.690 0.046	0 -
Linear regression -	24	3	3.934 15.475 2.807 0.048	0 -
ARIMA -	24	3	3.383 11.446 2.655 0.045	0 -
Univariate + DF NONE × AVG	24	3	4.125 17.015 3.221 0.053	8073 -
Univariate + DF NONE × KF	24	3	5.314 28.243 4.193 0.070	8073 -
Univariate + DF NONE × SKF	24	3	5.285 27.934 4.149 0.069	8075 -
Univariate + DF NONE × UKF	24	3	4.148 17.208 3.257 0.054	8073 -
Univariate + DF NONE × SVR	24	3	3.151 9.928 2.551 0.043	8073 -
No fusion + DF NONE × AVG	24	3	4.098 16.796 3.313 0.055	8186 -
No fusion + DF	24	3	5.314	8186

NONE × KF			28.243 4.193 0.070	-
No fusion + DF NONE × SKF	24	3	5.288 27.964 4.150 0.069	8188 -
No fusion + DF NONE × UKF	24	3	4.117 16.951 3.333 0.056	8186 -
No fusion + DF NONE × SVM	24	3	3.806 14.485 3.066 0.051	8186 -
SKF × AVG	24	3	3.704 13.719 2.994 0.050	8106 -
SKF × KF	24	3	5.314 28.243 4.193 0.070	8106 -
SKF × SKF	24	3	5.288 27.962 4.150 0.069	8108 -
SKF × UKF	24	3	3.728 13.900 3.023 0.051	8106 -
SKF × SVM	24	3	3.472 12.055 2.929 0.049	8106 -
UKF × AVG	24	3	4.480 20.070 3.642	8085 -

			0.061	
UKF × KF	24	3	5.314 28.243 4.193 0.070	8085 -
UKF × SKF	24	3	5.288 27.960 4.150 0.069	8086 -
UKF × UKF	24	3	4.495 20.205 3.663 0.061	8085 -
UKF × SVM	24	3	4.264 18.181 3.312 0.056	8085 -
KF × AVG	24	3	3.749 14.059 3.066 0.051	8083 -
KF × KF	24	3	5.314 28.243 4.193 0.070	8083 -
KF × SKF	24	3	5.288 27.964 4.150 0.069	8085 -
KF × UKF	24	3	3.767 14.189 3.088 0.052	8083 -
KF × SVM	24	3	3.621 13.112 3.046 0.052	8083 -
Combined model SKF × NONE	24	6	8.999 80.977	3183 29,205,971

			4.741 0.078	
Images only -	24	6	4.497 20.220 3.746 0.064	8047 137,125,771
SKF × NONE	24	6	4.477 20.044 3.721 0.063	57 92,566
Univariate NONE × NONE	24	6	3.503 12.269 2.747 0.046	25 11,514
KF × NONE	24	6	4.024 16.191 3.283 0.056	37 92,566
UKF × NONE	24	6	4.196 17.605 3.228 0.055	38 92,566
No fusion NONE × NONE	24	6	4.698 22.068 3.857 0.064	122 93,206
Naive -	24	6	3.972 15.773 3.120 0.053	0 -
Moving average -	24	6	3.656 13.363 2.943 0.050	0 -
Linear regression -	24	6	6.439 41.455 3.443 0.060	0 -
ARIMA	24	6	3.865	0

-			14.935 2.977 0.051	-
Univariate + DF NONE × AVG	24	6	3.461 11.979 2.890 0.049	8072 -
Univariate + DF NONE × KF	24	6	4.372 19.112 3.649 0.063	8072 -
Univariate + DF NONE × SKF	24	6	4.397 19.335 3.668 0.063	8074 -
Univariate + DF NONE × UKF	24	6	3.440 11.834 2.861 0.049	8072 -
Univariate + DF NONE × SVR	24	6	3.577 12.795 3.024 0.052	8072 -
No fusion + DF NONE × AVG	24	6	3.826 14.637 3.124 0.053	8169 -
No fusion + DF NONE × KF	24	6	4.372 19.112 3.649 0.063	8169 -
No fusion + DF NONE × SKF	24	6	4.392 19.288 3.663 0.063	8170 -
No fusion + DF NONE × UKF	24	6	3.778 14.273 3.084	8169 -

			0.052	
No fusion + DF NONE × SVM	24	6	4.676 21.862 3.844 0.064	8169 -
SKF × AVG	24	6	3.902 15.222 3.127 0.053	8104 -
SKF × KF	24	6	4.372 19.112 3.649 0.063	8104 -
SKF × SKF	24	6	4.392 19.292 3.663 0.063	8105 -
SKF × UKF	24	6	3.856 14.865 3.091 0.053	8104 -
SKF × SVM	24	6	4.659 21.707 3.896 0.066	8104 -
UKF × AVG	24	6	3.743 14.007 3.011 0.051	8085 -
UKF × KF	24	6	4.372 19.112 3.649 0.063	8085 -
UKF × SKF	24	6	4.392 19.291 3.663 0.063	8087 -
UKF × UKF	24	6	3.698 13.677	8085 -

			2.977 0.051	
UKF × SVM	24	6	4.137 17.115 3.247 0.055	8085 -
KF × AVG	24	6	3.680 13.544 2.960 0.050	8084 -
KF × KF	24	6	4.372 19.112 3.649 0.063	8084 -
KF × SKF	24	6	4.392 19.292 3.663 0.063	8086 -
KF × UKF	24	6	3.635 13.214 2.926 0.050	8084 -
KF × SVM	24	6	4.194 17.589 3.405 0.058	8084 -
Combined model SKF × NONE	24	12	16.039 257.251 7.152 0.116	3152 29,206,025
Images only -	24	12	7.283 53.047 5.826 0.096	8051 137,125,825
SKF × NONE	24	12	4.386 19.238 3.473 0.058	60 92,632
Univariate	24	12	4.009	26

NONE × NONE			16.074 3.136 0.052	11,580
KF × NONE	24	12	4.433 19.650 3.462 0.058	35 92,632
UKF × NONE	24	12	4.570 20.884 3.686 0.062	36 92,632
No fusion NONE × NONE	24	12	4.161 17.315 3.304 0.056	120 93,272
Naive -	24	12	4.201 17.651 3.231 0.055	0 -
Moving average -	24	12	4.179 17.463 3.321 0.057	0 -
Linear regression -	24	12	7.308 53.410 4.509 0.079	0 -
ARIMA -	24	12	4.166 17.356 3.163 0.054	0 -
Univariate + DF NONE × AVG	24	12	5.254 27.600 4.250 0.070	8076 -
Univariate + DF NONE × KF	24	12	7.135 50.904 5.732	8076 -

			0.095	
Univariate + DF NONE × SKF	24	12	7.209 51.974 5.766 0.095	8078 -
Univariate + DF NONE × UKF	24	12	5.224 27.295 4.222 0.070	8076 -
Univariate + DF NONE × SVR	24	12	4.123 16.997 3.328 0.056	8076 -
No fusion + DF NONE × AVG	24	12	4.893 23.939 4.022 0.067	8171 -
No fusion + DF NONE × KF	24	12	7.135 50.904 5.732 0.095	8171 -
No fusion + DF NONE × SKF	24	12	7.210 51.988 5.766 0.095	8173 -
No fusion + DF NONE × UKF	24	12	4.853 23.548 3.986 0.066	8171 -
No fusion + DF NONE × SVM	24	12	4.267 18.207 3.401 0.058	8171 -
SKF × AVG	24	12	5.068 25.688 4.151 0.069	8110 -
SKF × KF	24	12	7.135 50.904	8110 -

			5.732 0.095	
SKF × SKF	24	12	7.216 52.073 5.771 0.095	8112 -
SKF × UKF	24	12	5.034 25.345 4.123 0.068	8110 -
SKF × SVM	24	12	4.153 17.246 3.317 0.056	8110 -
UKF × AVG	24	12	5.413 29.304 4.414 0.073	8087 -
UKF × KF	24	12	7.135 50.904 5.732 0.095	8087 -
UKF × SKF	24	12	7.216 52.072 5.771 0.095	8088 -
UKF × UKF	24	12	5.375 28.888 4.381 0.072	8087 -
UKF × SVM	24	12	4.091 16.737 3.196 0.054	8087 -
KF × AVG	24	12	5.127 26.283 4.205 0.070	8086 -
	24	12	7.135	8086

KF × KF			50.904 5.732 0.095	-
KF × SKF	24	12	7.222 52.152 5.775 0.095	8088 -
KF × UKF	24	12	5.084 25.849 4.166 0.069	8086 -
KF × SVM	24	12	4.062 16.501 3.242 0.055	8086 -

Appendix 3 - Dataset

Datasetd used for experiments is available in pickle format at:

`https://bitbucket.org/andressuislepp/magistritoo/src/master/app/10feb-06mar-cameras-sm.pkl`

Appendix 4 - Code

Source code for the development of models is available as a Git Repository:

`https://bitbucket.org/andressuislepp/magistritoo`