

Article

# Prediction Framework with Kalman Filter Algorithm

Janis Peksa 

Institute of Information Technology, Riga Technical University, Kalku Street 1, LV-1658 Riga, Latvia;  
Janis.Peksa@rtu.lv

Received: 25 April 2020; Accepted: 8 July 2020; Published: 10 July 2020



**Abstract:** The article describes the autonomous open data prediction framework, which is in its infancy and is designed to automate predictions with a variety of data sources that are mostly external. The framework has been implemented with the Kalman filter approach, and an experiment with road maintenance weather station data is being performed. The framework was written in Python programming language; the frame is published on GitHub with all currently available results. The experiment is performed with 34 weather station data, which are time-series data, and the specific measurements that are predicted are dew points. The framework is published as a Web service to be able to integrate with ERP systems and be able to be reusable.

**Keywords:** forecasting; framework; Kalman filter; SaaS; Python; prediction framework

## 1. Introduction

Unevenness and data availability is one of the essential things for information systems nowadays. As already known, enterprise resource planning (ERP) systems provide most of the business processes and the success of the business value increase, and it is required to ensure the credibility of data. Data processing logic is deemed as complex decision-making logic if it relies on analytical or managerial models for determining a course of action in business process execution and often requires domain-specific knowledge. Examples of decision-making logics are inventory replenishment, road-network maintenance, and production planning decisions. The demand for accurate decisions has grown, forcing ERP to improve the decision-making process, at both strategic and operational levels, by providing the necessary information, tools, and capabilities essential to enhance the decision-making process. However, many modules contain sophisticated forecasting methods and are part of a decision-making logic. Forecasting is a process for predicting the future based on past data—most often, through trend analyses. Forecasting is one such case that businesses need to gain higher profit and continue business processes successfully. ERP systems have limited forecasting capabilities that are implemented into an outline code. Enterprises spend much money to modify existing methods to satisfy their requirements. Some ERP systems do not have enough forecasting functionality. Forecasting functionality is used as an opportunity that can be enhanced by predictive capabilities in ERP systems. Facts and assumptions arising from various data sources may be identified as useful to the decision-maker. Based on this data, different business decisions can be made that can more effectively provide access to business gain. However, many data are not available in ERP systems to provide sufficient objectivity at the time of decision. Multiple data sources and forecasting algorithms are needed to be placed in ERP systems to increase reliability and make more efficient decisions from numerous references in the same period. It is well known, ERP systems are complex and require the appropriate specialists to perform the necessary programming work to make changes; therefore, outsourcing comes in handy when such specialists are lacking [1–4].

One of the most advanced outsourcing approaches is a software as a service (SaaS), which allows straitening the service for the company, giving several privileges offered by SaaS. The ERP systems

problem that needs to be addressed to allow for increased capabilities from external data sources in ERP systems. These external data will enable us to increase the accuracy of forecasts and enable the decision-maker to make a more valuable and accurate decision in a given situation. Adding different data sources through an integrated approach can ensure an increase in the accuracy outcome. Combining external data into a single mathematical model and then into a unique algorithm under a single framework capable of achieving this goal. This is one of the initial tasks that is accomplished in this article by presenting actual achievements and reflecting a detailed solution with road maintainers case data obtained from metrological stations located within the territory of the Republic of Latvia. The part of the autonomous open data prediction framework (AODPF) will be presented as results gain during the research process [5].

Road maintenances works are very diverse and specific, ranging from laying of the road surface to daily maintenance activities. Many stakeholders, including managers, dispatchers, and maintenance teams, are involved, and maintenance decisions need to be made on time [6]. Proactive maintenance activities are enabled by forecasting. Forecasting provides advance information about the required maintenance activities. The anti-slip maintenance is performed only in winter to provide anti-slip materials at a specific time and place on the road surface. The anti-slip maintenance nowadays uses live contextual data from many different sources, including open data sources and decision-making results significantly depend on data availability [7], including open data sources [8], and decision-making results significantly depend on data availability [9]. A pastime may affect the essential operations of the road-network in total. One of them is driving conditions when traffic speed is rapid. The higher the speed rate, the greater the probability that accidents are possible [10]. As know, the road conditions are subject to rapid changes in surface temperature and precipitation sum. The road conditions most fluctuations are observed during the winter period; however, the road condition is also affected by the snow and icing. The road maintenance is performed for specific Road Sections belonging to a region. The region in research is the Republic of Latvia road network, and the dataset is taken from the State Joint Stock Company "Latvian State Roads". The region consists of 52 road monitoring weather stations that are relatively distant from each other. To be able to respond to changes in the environment, on-road sections near the road surface are located road monitoring weather stations and cameras operated by different entities. The road monitoring weather stations collect raw observations that are processed to be able to make the necessary forecasting for future decisions [11]. For road maintainers, those predictions are crucial to making decisions daily. The road monitoring weather stations and cameras operated by different entities can help for the decision-making. The road maintainers controlling smart road signs are available to give warning messages to drivers on a specific stage of the road network [12]. The missing information in the time-series of weather stations is unavoidable, owing to the full observation of all the continuous processes is almost impossible [13].

The AODPF as a framework for forecasting will be described in detail from the original data called raw data. After receiving the data, the processing is done to make the data more readable in the input state and ready for the next steps. Each step in this forecast framework will display and reflect the programming code that will be published on GitHub. GitHub is a company that provides hosting for software development version control using Git. Git is a distributed version control system for tracking changes in source code during software development. In this case, the demonstration of an example based on the Kalman filter algorithm. Kalman filtering, which is also known as linear quadratic estimation, an algorithm that uses the time to observe the measurement string containing statistical noise and other inaccuracies. The previous results already address the general mathematical approach of the Kalman filter algorithm [14].

The main goal is to reflect the results achieved. Stabilize understanding of the work to be done. Moreover, to present the results in a logically structured way as a proof-of-concept. The subject of the research is the forecasting capabilities of the ERP systems.

The structure of the study reflects the aim mentioned above. The following section starts with a discussion on the materials and methods within the autonomous open data prediction framework.

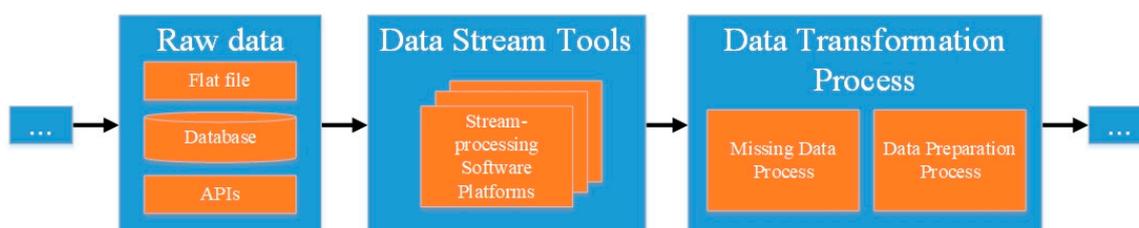
In Section 3, narrow down to the results of the experiment. In Section 4 closes with the conclusion and suggestion for future research.

## 2. Materials and Methods

### 2.1. Autonomous Open Data Prediction Framework (AODPF)

The definition of AODPF is to be able to perform forecasting with open data (OD) from different sources that can provide itself as a framework in the form of autonomous. In other words, a framework that can be driven by managers; who do not have delve into the essence of algorithms and the essence of the forecasting process. AODPF would be able to provide automated algorithm selection and deliver results to ERP systems. One of the most critical privileges is the integration of several data sources into a single framework and the possibility to link the results with ERP systems. These data sources could be configured and modified by managers to improve the decision-making process under specific circumstances. One of the last features would be the standard use in any other business process and with any other data sources as well as with different forecasting methods and approaches. This subsection will outline the components of the AODPF that should be implemented in the future.

Originally hereafter from which the raw data comes. There are three ways to transfer data to AODPF flat file, database, and application programming interface (API). Currently, it is implemented only from the database. API passes data outwards but not inwards. In addition, the flat file is not implanted in AODPF but can be quickly realized as needed. As mentioned above, data processing is a long process when it comes to big data. In this case, the AODPF is built on the principle of solving the big data problem so that when faced with this problem, it would continue to work in the usual rhythm. These data can be processed with data stream tools, which allows data sources to be combined into a single data set. As already known, data may also come off, and that is why there is a need to use different methods to replace the missing data to be able to make a prediction then. The first three processes of AODPF are illustrated in Figure 1.



**Figure 1.** Part of the autonomous open data prediction framework (AODPF)—raw data, data stream tools, and data transformation processes.

After receiving the data and finalizing the processing, it can start the best model selection, which consists of time-series prediction, algorithm knowledgebase, and accuracy summary. Specifically, time-series prediction is a process that repeats itself until it finds the best model to perform full-scale prediction on the entire data set. Time-series prediction looks at and uses the whole algorithm knowledge base that is available and compares with each other to get the best results and use what has gained the best accuracy. All results are recorded in an accuracy summary, which allows the process to determine which is the best algorithm for a particular data source. Currently, in AODPF, this process is ongoing and contains only the Kalman filter approach. The reason is that the purely simple development process is in the initial phase, and to generally understand that forecasting takes place with this method was started. As the algorithm is the only one that is not currently implemented, the autonomous selection of the algorithm from the knowledge base. However, the desired results that will be achieved in AODPF in the future can be seen in Figure 2 below.

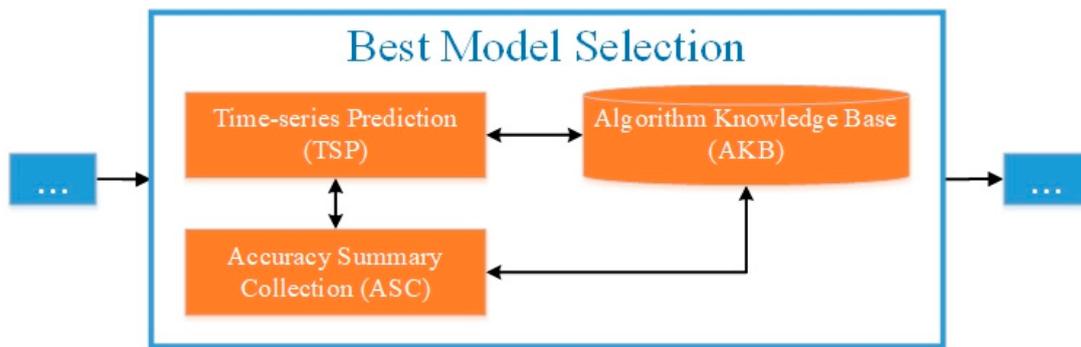


Figure 2. AODPF representation of the best model selection.

ERP systems managers that are also decision support system holder for successful decision-making in AODPF there is a need for integration between AODPF and ERP systems. As well, the framework must be able to provide forecasts regularly and with high-speed accuracy and connectivity. This can very well be provided by business intelligence (BI), which sets rules and specific triggers. At a particular time, one precise data source observation would give off a specific trigger of an event that would be able to provide ERP systems decision-making an additional criterion. Such triggers could be configured as many as needed so that the BI provides all the desires and needs. In order to connect AODPF to ERP systems, there is a need for a data connector. Which nowadays is mostly done with the help of APIs. AODPF has the following API available and can be tested by setting up AODPF and trying out all the instructions available on GitHub, which is listed next to Supplementary Materials. The AODPF connection between ERP systems can be seen in Figure 3 below.

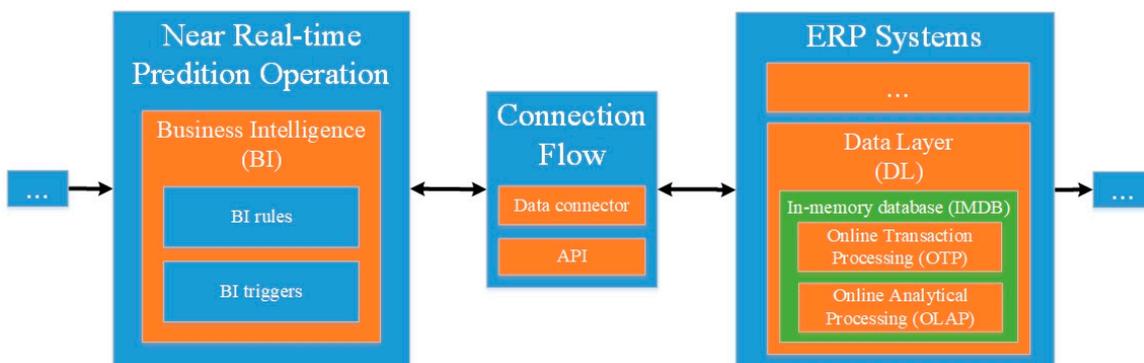


Figure 3. Application programming interface (API) demonstration between enterprise resource planning (ERP) systems and AODPF.

The ERP system itself with the data layer and In-memory database is also slightly illustrated. The experiment with road maintenance using the Kalman filter approach will be discussed below.

### 2.2. Case Study on Road Maintenance Using a Kalman Filter Approach

In order to better understand the theoretical and ADOPF concept, there is a need for a practical example that can be done with the help of ADOPF; the following study presents a case study with road maintenance using the Kalman filter approach.

The Kalman filtering process has two stages: the forecasting step and the update state. The system is predicted by previous measurements and the update phase, where the current state of the system is assessed considering the analysis over a given time period. Actions are translated into equations as follows [15]:

- Prediction:

$$\hat{x}_{k|k-1} = F_k \hat{x}_{k-1|k-1} + B_k u_k \tag{1}$$

$$P_{k|k-1} = F_k P_{k-1|k-1} F_k^T + Q_k \tag{2}$$

- Update step:

$$\tilde{y}_k = z_k - H_k \hat{x}_{k|k-1} \tag{3}$$

$$S_k = H_k P_{k|k-1} H_k^T + R_k \tag{4}$$

$$\hat{x}_{k|k} = \hat{x}_{k|k-1} + K_k \tilde{y}_k \tag{5}$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \tag{6}$$

$$\tilde{y}_{k|k} = z_k - H_k \hat{x}_{k|k} \tag{7}$$

where  $\hat{x}_{k|k-1}$  and  $P_{k|k-1}$  are predicted state mean and covariance, respectively, on the time step  $k$  before seeing the measurement.  $\hat{x}_{k|k-1}$  is vector, where  $k$  is the size of the state.  $B_k$  is the control-input model which controls are applied vector  $u_k$ .  $\tilde{y}_k$  is mean of the value on time step  $k$ .  $z_k$  the observation noise of the remaining calculation over the period  $k$ .  $S_k$  covariance matrices prediction of calculation of the time step covariance  $k$ .  $K_k$  the optimal Kalman gain filter how much the predictions will be corrected in time step  $k$ .  $\hat{x}_{k|k}$  is updated state covariance.  $P_{k|k}$  is updated estimate covariance and lastly  $\tilde{y}_{k|k}$  measurement post-fit residual [15].

The next section summarizes the results of the case study experiment, and the rest of the results are available on GitHub.

### 3. Results

Data for the period from 19 January to 19 February 2020, are used for the experiment. The data consists of a time series and a dew point which is then predicted during the experiment. Each weather station’s dew point is forecasted separately among the region consisting of 52 road monitoring weather stations. Some weather stations do not contain data or do not work during the period. They are automatically taken out and ignored, leaving a total of 34 weather stations. The area is sufficient to perform the initial experiment with the Kalman filter. The previous algorithm that is mentioned is used, and the algorithm can be found on the GitHub.

Weather stations with code “LV01”, “LV09”, “LV36” and “LV64” are shown in Figure 4 as results from the experiment (a) as “LV01”, (b) as “LV09”, (c) as “LV36” and (d) as “LV64”.

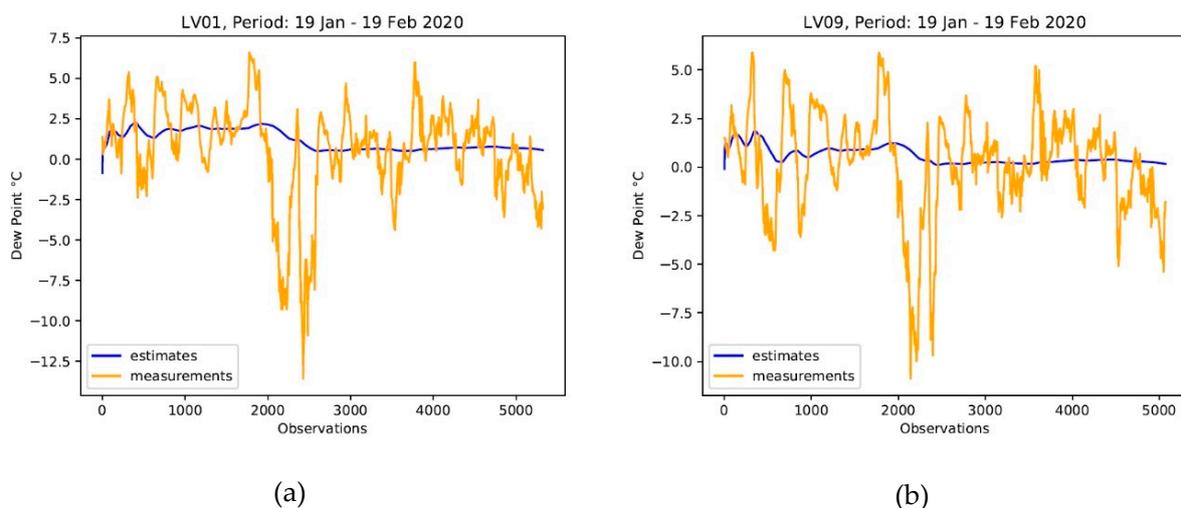
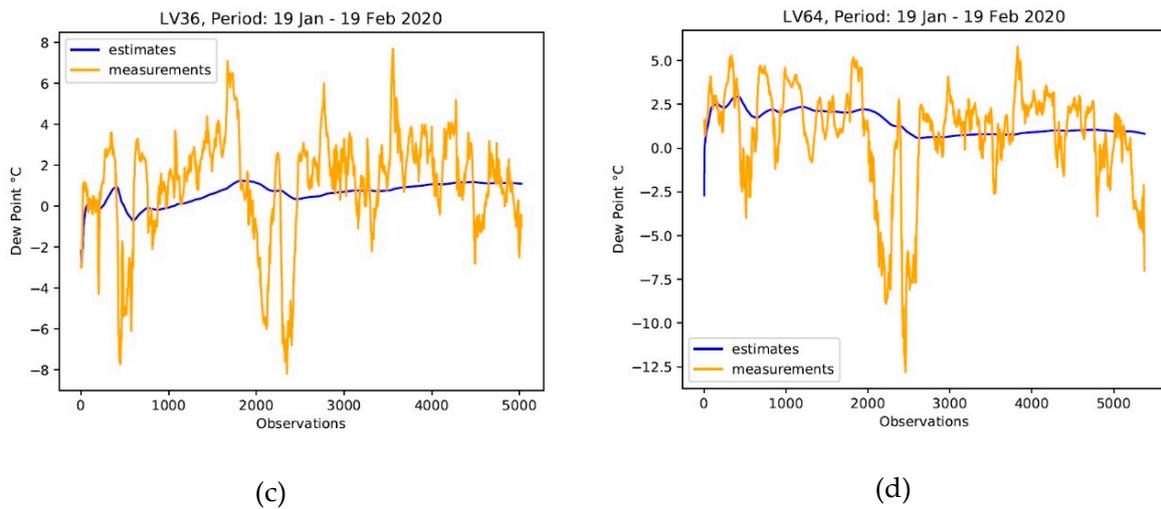
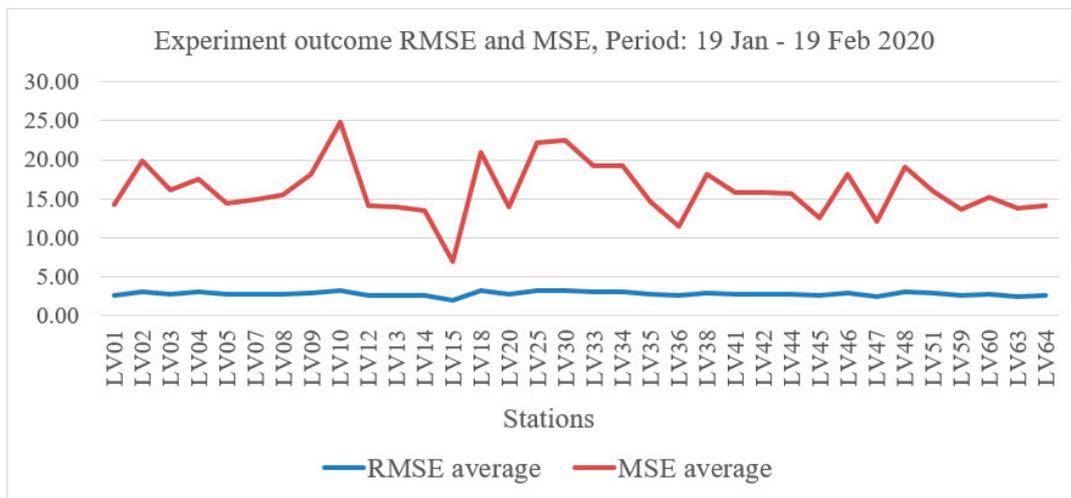


Figure 4. Cont.



**Figure 4.** Kalman filter results of the four weather stations, (a) as “LV01”, (b) as “LV09”, (c) as “LV36” and (d) as “LV64”.

As can be seen in Figure 4, the results are relatively weak. Estimates and measurements vary by several orders of magnitude, which indicates that the Kalman filter accuracy is poor. However, it should be noted that the same velocity is used in the Kalman filter approach, which is not good and would need to be corrected by creating a matrix approach offered by the Kalman filter algorithm. Below are the results of the experiment outcome RMSE and MSE in the period from 19 January to 19 February 2020, see Figure 5.



**Figure 5.** Experiment outcome RMSE and MSE, period: 19 January 2020—19 February 2020.

Dramatic findings show that the outcome needs to be changed so that it can get closer to the actual situation. A more detailed overview can be found in Table 1. Each weather station is listed with its average RMSE and MSE.

The RMSE average for all 34 weather stations was 2.81; the MSE average for all was 16.12. Far from the desired *result*, however, it could be said that the realization of the Kalman filter was done without velocity. In addition, AODPF public access via GitHub was provided, and a web service was tested, which could already be used for integration between ERP systems. The first steps were taken to start the implementation and integration of AODPF between ERP systems in the future.

**Table 1.** Experiment outcome RMSE and MSE, period: 19 January 2020–19 February 2020 in detail.

Stations	RMSE Average	MSE Average
LV01	2.65	14.27
LV02	3.08	19.84
LV03	2.77	16.12
LV04	2.99	17.47
LV05	2.70	14.45
LV07	2.68	14.87
LV08	2.77	15.52
LV09	2.88	18.09
LV10	3.29	24.83
LV12	2.67	14.11
LV13	2.66	13.90
LV14	2.61	13.54
LV15	1.91	6.98
LV18	3.15	20.94
LV20	2.70	13.91
LV25	3.27	22.13
LV30	3.26	22.50
LV33	3.09	19.31
LV34	3.12	19.24
LV35	2.73	14.54
LV36	2.56	11.53
LV38	2.94	18.22
LV41	2.80	15.74
LV42	2.81	15.82
LV44	2.76	15.68
LV45	2.56	12.53
LV46	2.98	18.08
LV47	2.51	12.04
LV48	3.13	19.05
LV51	2.86	16.04
LV59	2.65	13.69
LV60	2.73	15.18
LV63	2.50	13.74
LV64	2.61	14.05

#### 4. Conclusions

AODPF has been presented and also published, and the initial stage is, of course, volatile. There is a need to improve the Kalman filter approach. Repeat the experiment to see if the method gives a better result. Besides, there is a need for other procedures in order to compare and draw conclusions. Much work needs to be done, and there is a need for an automation process and a full-fledged AODPF process. However, it can be emphasized that the Python programming language is advantageous in the given solution and is also able to build the web service that is necessary for future integration with ERP systems. As already mentioned, AODPF is in its infancy. On the other hand, it can be said that the approach works—and the results are marked in a logically structured way as a proof-of-concept.

AODPF is published on GitHub and is available to the general public to use the required implementation with any system in a standard form and, of course, best with ERP systems for tangible results and flexibility. Partial documentation has been created and published alongside the AODPF, which is also available to the general public.

In future work, there will be an over-emphasis on adding other approaches, the choice of automated best model selection, and result representation from AODPF with a user interface implementation by choosing the time-frame and desired measurement type. There is a need for other alternative methods in the future, such as ARIMA and others.

**Supplementary Materials:** The following framework with source code and results are available online at <https://github.com/JanisPeksa/Autonomous-Open-Data-Prediction-Framework>.

**Funding:** This publication was supported by the Riga Technical University's Doctoral Grant program.

**Acknowledgments:** Thanks to SJSC "Latvian State Roads" for providing data availability.

**Conflicts of Interest:** The author declares no conflicts of interest.

## References

1. Holsapple, C.W.; Sena, M.P. ERP plans and decision-support benefits. *Decis. Support Syst.* **2005**, *38*, 575–590. [[CrossRef](#)]
2. Bahrami, B.; Jordan, E. Utilizing Enterprise Resource Planning in Decision-Making Processes. In *Innovation and Future of Enterprise Information Systems*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 153–168.
3. O'Leary, D.E. Supporting decisions in real-time enterprises: Autonomic supply chain systems. *Inf. Syst. e-Bus. Manag.* **2008**, *6*, 239–255. [[CrossRef](#)]
4. Aslan, B.; Stevenson, M.; Hendry, L.C. Enterprise Resource Planning systems: An assessment of applicability to Make-To-Order companies. *Comput. Ind.* **2012**, *63*, 692–705. [[CrossRef](#)]
5. Peksa, J.; Grabis, J. Integration of Decision-Making Components in ERP Systems. In *Proceedings of the 20th International Conference on Enterprise Information Systems*; Scitepress: Funchal, Madeira, Portugal, 2018; Volume 1, pp. 183–189.
6. Grabis, J.; Bondars, Ž.; Kampars, J.; Dobelis, Ē.; Zaharčukovs, A. Context-Aware Customizable Routing Solution for Fleet Management. In *Proceedings of the 19th International Conference on Enterprise Information Systems*; Scitepress: Porto, Portugal, 2017; Volume 1, pp. 638–645.
7. Pindyck, R.S.; Rubinfeld, D.L. *Econometric Models and Economic Forecasts*, 3rd ed.; McGraw-Hill: New York, NY, USA, 1991.
8. Zdravkovic, J.; Kampars, J.; Stirna, J. Using Open Data to Support Organizational Capabilities in Dynamic Business Contexts. In *Lecture Notes in Business Information Processing*; Springer International Publishing: Tallinn, Estonia, 2018; Volume 316, pp. 28–39.
9. Grabis, J.; Minkevica, V. Context-Aware Multi-Objective Vehicle Routing. In *31st Conference on Modelling and Simulation*; European Council for Modelling and Simulation: Budapest, Hungary, 2017; pp. 235–239.
10. Edwards, J.B. Speed adjustment of motorway commuter traffic to inclement weather. *Transp. Res. Part F Traffic Psychol. Behav.* **1999**, *2*, 1–14. [[CrossRef](#)]
11. Peksa, J.; Peka, J. Forecasting using Contextual Data in Road Maintenance Work. In *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*; IEEE: Piscataway, NJ, USA, 2018; pp. 1–6. [[CrossRef](#)]
12. Peksa, J. Autonomous Open Data Prediction Framework. In *2019 IEEE 7th IEEE Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*; IEEE: Piscataway, NJ, USA, 2019; pp. 1–5.
13. Nguwi, Y.-Y.; Kouzani, A.Z. Detection and classification of road signs in natural environments. *Neural Comput. Appl.* **2008**, *17*, 265–289. [[CrossRef](#)]
14. Jeffrey, S.J.; Carter, J.O.; Moodie, K.B.; Beswick, A.R. Using spatial interpolation to construct a comprehensive archive of Australian climate data. *Environ. Model. Softw.* **2001**, *16*, 309–330. [[CrossRef](#)]
15. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 2006; Volume 6, pp. 1–16.



© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).