Code will be available on https://github.com/SunnyBingoMe/sun2017overview-github-public

See also: https://github.com/SunnyBingoMe/sun2018shortterm-github

See also: https://github.com/SunnyBingoMe/sun2017flowaware-github

An Overview of Parameter and Data Strategies for *k*-Nearest Neighbours Based Short-Term Traffic Prediction

Bin Sun
Blekinge Institute of Technology
Karlskrona, Sweden
bin.sun@bth.se

Wei Cheng
Kunming University of Science and Technology
Kunming, China. Corresponding Author.
wei.cheng@bth.se

Prashant Goswami
Blekinge Institute of Technology
Karlskrona, Sweden
prashant.goswami@bth.se

Guohua Bai
Blekinge Institute of Technology
Karlskrona, Sweden
guohua.bai@bth.se

ABSTRACT

Modern intelligent transportation systems (ITS) requires reliable and accurate short-term traffic prediction. One widely used method to predict traffic is k-nearest neighbours (kNN). Though many studies have tried to improve kNN with parameter strategies and data strategies, there is no comprehensive analysis of those strategies. This paper aims to analyse kNN strategies and guide future work to select the right strategy to improve prediction accuracy. Firstly, we examine the relations among three kNN parameters, which are: number of nearest neighbours (k), search step length (d) and window size (v). We also analysed predict step ahead (m) which is not a parameter but a user requirement and configuration. The analyses indicate that the relations among parameters are compound especially when traffic flow states are considered. The results show that strategy of using v leads to outstanding accuracy improvement. Later, we compare different data strategies such as flow-aware and time-aware ones together with ensemble strategies. The experiments show that the flow-aware strategy performs better than the time-aware one. Thus, we suggest considering all parameter strategies simultaneously as ensemble strategies especially by including v in flow-aware strategies.

CCS CONCEPTS

• **Information systems → Information systems applications → Data mining → Nearest-neighbor search**

KEYWORDS

Short-Term Traffic Prediction, k-Nearest Neighbours Regression, Parameter and Data Strategies

1 INTRODUCTION

Reliable and accurate short-term traffic prediction is a key requirement in modern intelligent transportation systems (ITS) [29]. Short-term traffic prediction is essential for efficient traffic management and incident detection [5]. However, it is a complex task and has been a research subject for the past few decades [14]. The difficulty is due to the fact that traffic flow is influenced by many factors including people, vehicles, roads, environment and information [17].

There are two categories of methods for short-term traffic prediction: parametric and nonparametric [11, 14, 22]. Within the nonparametric methods, one widely used algorithm is k-Nearest Neighbours (kNN) [24–26]. With the substantial increment of available data [14], kNN is gaining attention due to its flexibility in solving nonlinear problems [1]. Besides, it is easy to understand and implement [10].

Strategies for improving kNN can be divided into two categories, parameter strategies and data strategies. Considering the parameter strategies, three parameters of kNN are critical: the number of nearest neighbours (k), the search step length (also known as lag d), and the window size (also known as lag constraint) (v) [1, 28]. Although the way to measure the distance of neighbours is also important, it is beyond this paper's scope when considering parameters. One problem in basic kNN is that it has fixed parameters which do not consider time-varying and nonstable statistical characteristics of traffic flow. Thus, kNN needs different parameter values for its three parameters under different flow situation. That is why data strategies matter. [11] compared kNN performance under four datasets. Their results show that datasets with different traffic characteristics require diverse parameter configurations to perform well and the parameters of kNN should be chosen carefully.

To the best knowledge of the authors, there is no analysis considering all strategies simultaneously. Two possible reasons are: much computation is required due to the compound parameter relations and there was not enough data to test data strategies. This paper analyses and compares strategies for all parameters at the same time to provide a comprehensive understanding. Also, we provide a guideline to choose strategies with limited and fixed amount of data for short-term traffic prediction while taking data strategies and other settings into consideration.

ICSET 2017, October 12–14, 2017, Chengdu, China

ACM ISBN 978-1-4503-5376-2/17/10...$15.00
https://doi.org/10.1145/3157737.3157749

2 STRATEGIES FOR kNN

This section introduces the basic kNN algorithm then the parameter and data strategies that are used to improve kNN.

2.1 The kNN Algorithm

kNN predicts the future using history traffic data which are similar to current one. Recent traffic is described by a traffic state vector using both flow rate and speed:

$$S_{[t]} = \begin{bmatrix} r_{t-1} & r_{t-2} & \cdots & r_{t-d} \\ s_{t-1} & s_{t-2} & \cdots & s_{t-d} \end{bmatrix} \tag{1}$$

where $t-1$ is the time point of the last received data, r is flow rate and s is speed. S is then compared with each day's data in database and the most similar k neighbours are selected. The dissimilarity is measured using Euclidean distance. While previous work considers d search steps as d dimensions [2], we consider all search steps as one dimension to avoid the curse of dimension problem. The following equation is used to calculate distance for d search steps:

$$\frac{\sum_{i_d=1}^{d} \sqrt{(r_{t-i_d} - r'_{t-i_d})^2 + (s_{t-i_d} - s'_{t-i_d})^2}}{d} \tag{2}$$

where r' and s' are flow rate and speed of an arbitrary day in the history database. If we take window size into consideration (suppose $v = 1$), while fixing S, the flow rate vector can be $[r'_{t-2} \cdots r'_{t-d-1}]$ and $[r'_{t} \cdots r'_{t-d+1}]$ in addition to $[r'_{t-1} \cdots r'_{t-d}]$. The speed vector s' changes together with r'. That is, there are not one but three potential neighbours in one history day when $v = 1$.

While taking all three parameters into consideration, kNN becomes complicated since there are many possible tuples when assigning values to parameters. Ensemble method can be used to solve this complicated problem as introduced in [20]. The basic idea is to use weighted average of predictions of parameter tuples as final prediction. The weights of tuples are generated using training dataset.

2.2 Parameters Strategies

Although there is no rule for the choice of kNN parameter values, some values are frequently used, and this configuration is used as baseline A during comparison. To have a good value assigned to k, [27] trained artificial neural network to map characteristics of a dataset to a good k value. [18] updates k continuously considering piecewise linear nature in time series. For search step length (d), [25] suggests it to be bigger than $2D + 1$ according to Takens theorem, where D is the number of features. However, some research [12, 15] shows opposite results. Though v is also an important parameter, only a few studies considered it. Using v was not showing promising improvement in [28], the reason can be that they determined v separately from k and d. In some studies, both k and d get optimised [2, 12]. [25] shows we should optimise k and d at the same time. [12] uses part of the data (like training data) to find best parameters tuple and then the tuple is used for prediction. We modified this method to include v and use it as baseline B for comparison in our study.

Some research shows improved performance using ensemble kNN [6, 7]. While they are only considering k, [20] uses ensemble strategy for all kNN parameters.

2.3 Data Strategies

A data strategy is a way to separate training dataset to suitable sub-datasets according to different characteristics of instances. Data strategies are usually domain specific.

A frequently used data strategy is time-aware separation (TA) as the traffic at similar time usually has similar patterns. Previous studies have pointed out the traffic has different patterns at different hour of the day, workday vs. holiday [3, 4, 8, 9]. Considering hour of the day is a typical time-aware strategy and considering workday vs. holiday is a workday-aware strategy. The literature has used the time-variant characteristics to improve traffic prediction [23], where one-day time is separated into four stages. We compare data segmentation strategies of four stages (TA4) and ten stages (TA10)

However, if records are separated by hour of the day, it is not guaranteed that the flow situation is the same, neither day of week, etc. We propose to use another data strategy which is flow-aware separation (FA). The kNN algorithm is adapting to different flow rate levels. We compare data segmentation strategies of four levels (FA4) and ten levels (FA10). The flow rate should be determined by averaging the last 15 minutes traffic which is the minimal averaging time to get stable traffic flow data [21].

For flow-aware strategies, traffic flow can have two trends which are increasing flow and decreasing flow. One way is to consider those two trends separately (up vs. down), noted as UvD in this work. Another way is to not distinguish those two trends, noted as U+D.

3 EXPERIMENTS

The traffic data are from devices along a highway named Kunshi in China. The flow on the road is usually under-saturated. Based on a five-minute interval, each device sends a record to the central database. Every record contains statistical values such as flow rate and speed. Data from one device are cleaned using the method from [19] and then used in this paper. The time range of data are between early-April 2013 and mid-May 2014.

The collected data are divided into 80% and 20% for training and testing respectively, separated on date 20th February 2014. If more than 10% searching steps or prediction steps are influenced by any incident, the record will be discarded. Exponential incremental values are assigned to k, d and v. As a nonstrategy setting, predict step ahead (m) also impacts experiment results and also uses exponential incremental values.

The experiments are conducted with CUDA [13] 8.0 on GeForce GTX 690 graphics card. The analyses are conducted with R language [16] version 3.3.

For the measurement of prediction performance, mean absolute error (MAE) is used. Mean absolute percentage error (MAPE) is not suitable as the flow can be zero during night.

Three terms are used to distinguish different types of relations within and out of strategies as well as nonstrategy settings. *Influence*: Given a time point, all three parameters contribute to the prediction performance, i.e. MAE $= g(k, d, v)$. The influences g_k, g_d, g_v of three parameters are presented separately to make results clear. *Effect*: The influence (g) of parameters on the prediction is also affected by the other two parameters. For instance, the influence of k on MAE (g_k) is under the effect of d and v, i.e. g_k changes when d

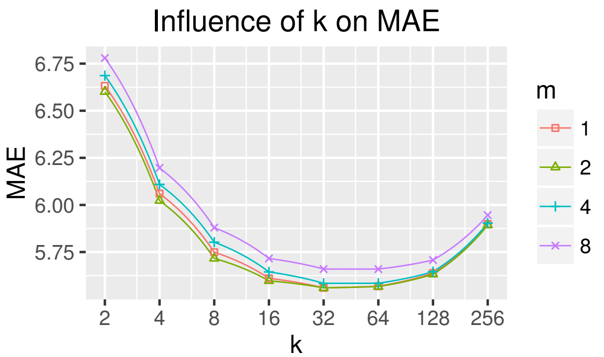

Figure 1. Influence of k on MAE (g_k) when $d = 4, v = 0$. Results of predictions with different parameter values construct four curves with turning points between $k = 32$ and $k = 64$.

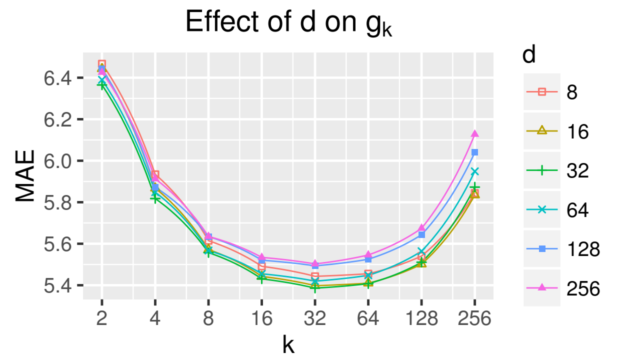

Figure 2. Effect of d on g_k when $v = 0$. The value of d affects the shape and turning point of g_k curves. The gradients after turning points increase if d becomes bigger. The turning point is moving from range $[32, 64]$ to range $[16, 32]$.

or v change.Those effects are presented after the influences. *Impact*: For different flow-/time-aware index and predict step ahead (m) etc., the best values of parameters are different. Those factors are nonstrategy settings, but still impact factors to kNN parameters.

4 RESULTS

This section provides comprehensive and detailed results regarding influences, effects and impacts of different strategies in the accuracy aspect.

As there are hundreds of thousands of parameter tuples (p), only some results are plotted. Most patterns are general and can hold for values that are not plotted unless otherwise mentioned and explained. For instance, only flow rate prediction is shown because speed prediction results have similar trends. The parameter values for plots are set to $k = 8$, $d = 4$, $v = 0$, unless otherwise mentioned. Those values are used as they are similar to default or well-performed values in previous studied [25, 26]. This can be considered as slicing of the multi-dimension results from this study. The differences of patterns introduced by m within different strategies are ignorable unless otherwise mentioned.

4.1 Influence of k

If the results are sliced by $d = 4, v = 0$ (i.e. search time is 20 minutes, no window), the remaining is g_k, i.e. the influence of k on kNN performance (MAE) as shown in Fig. 1. The result is showing comparisons between parameter predictions and ensemble strategies, each with four different m values. The plot also shows that when k increases, MAE decreases first (when $k < 32$) and then increases (when $k > 64$) on g_k curves. Thus, the influences g_k contain turning points (the k value for lowest MAE on one curve) between $k = 32$ and $k = 64$. Ensemble strategies are giving the lowest MAE when being compared with any parameter tuple p for any m. This conclusion holds for any v and d values when considering the influence from k on performance.

The **effect of** d **on** g_k is shown in Fig. 2, all curves have turning points. The effect of d is, if d is big, MAE increases fast when k grows. Besides, the turning points occur at smaller k values, moving

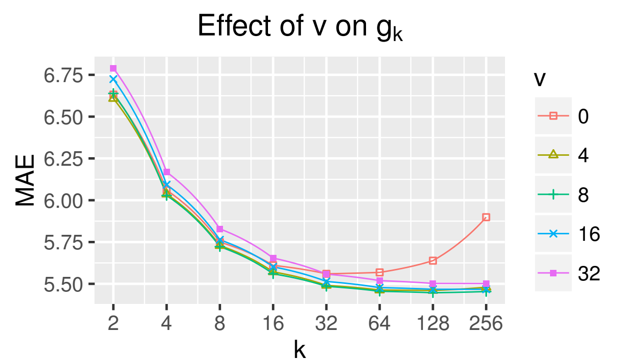

Figure 3. Effect of v on g_k when $d = 4$. As v becomes bigger, turning points are moving from around $k = 32$ to higher k values and has potential and trend to go beyond 256 if $v > 32$. This plot is when $d = 4$.

from $k > 32$ to $k < 32$. The **effect of** v **on** g_k is shown in Fig. 3, most curves have turning points around $k = 128$. However, when $v = 0$, the turning point is around $k = 32$. If v is small, say $v < 8$, there are turning points on g_k curves $k < 128$. If v is bigger than 8, the turning points of g_k move to higher k values ($k > 128$). The above conclusion is much clearer when d is big.

4.2 Influence of d

If the results are sliced by $k = 8, v = 0$ (i.e. 8 nearest neighbours, no window), the remaining is g_d, i.e. the influence of d on MAE as shown in Fig. 4. The plot also shows that when d increases, MAE decreases first (when $d < 32$) then increase (when $d > 32$) for g_d curves. Thus, the influences (g_k) contain turning points around $d = 32$. A special pattern is that there are peaks when $d = 128$.

The **effect of** k **on** g_d is shown in Fig. 5, there are always turning points around $d = 16$ or 32 on all curves. If k increases, turning points are clearer and occur at a smaller d value, moving from around $d = 32$ to near $d = 16$. If k is small, the curves are steeper.

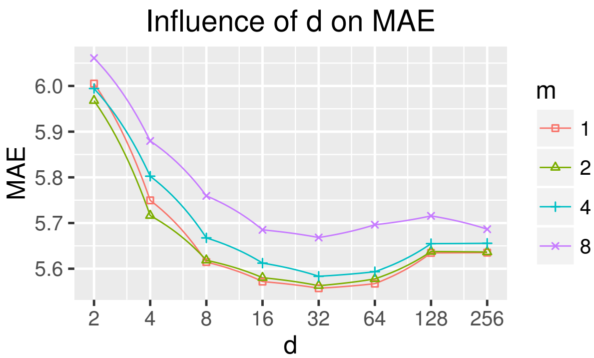

Figure 4. Influence of d on MAE (g_d) when $k = 8, v = 0$. Results of parameter predictions construct four curves with turning points around $k = 32$ and. Ensemble strategies are giving the lowest MAE compared with any parameter tuple p for any m.

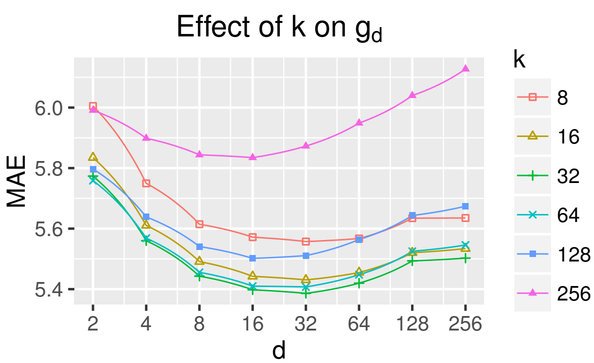

Figure 5. Effect of k on g_d when $v = 0$. There are always turning points for all curves. If k increases, turning points will occur at a smaller d value and the curves becomes flatter before turning points.

For big values of k, the curves are flatter before turning points. The **effect of v on g_d** is shown in Fig. 6, if v increases, g_d becomes steeper both before turning points and after turning points, which makes the turning points clearer. If v is big enough ($v \geq 16$), there are some peaks at $d = 128$ (only when k is too small). When d's value is close to the curve's turning point (around $d = 32$), v's effect is smaller. There are turning points on all curves and are in the range of $d \in [16,64]$, around 32. For the **effect of both k and v on g_d**, if window size (v) becomes big enough (e.g. 8), the turning points of g_d occur at higher values ($d = 32$) and not changing.

4.3 Influence of v

If the results are sliced by $k = 8, d = 4$, the remaining is g_v, i.e. the influence of v on MAE as shown in Fig. 7. For any m there are turning points around $v = 4$. When m increases, v values of turning points are slightly decreasing.

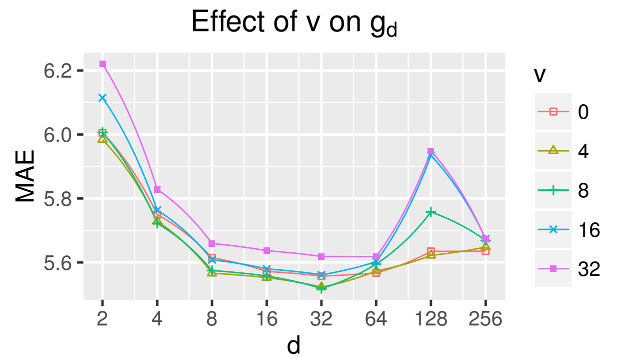

Figure 6. Effect of v on g_d when $k = 8$. If v increases, g_d becomes steeper and turning points are clearer (around $d = 32$). When d value is close to the curve's turning point, v's effect is smaller.

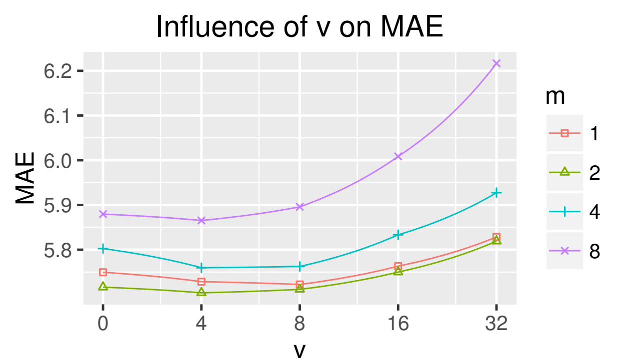

Figure 7. Influence of v on MAE (g_v) when $k = 8, d = 4$. Results of parameter predictions construct four curves. For some m values, there are turning points at $v = 4$.

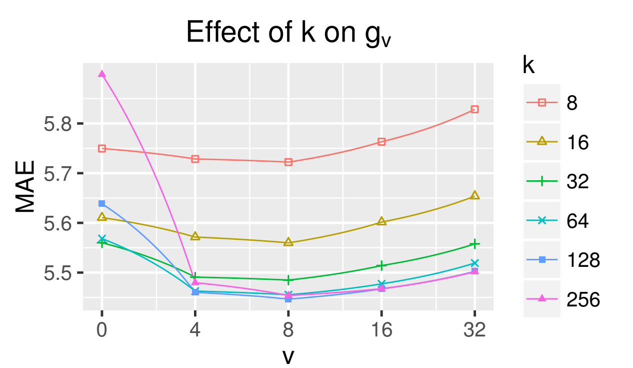

Figure 8. Effect of k on g_v when $d = 4$. The g_v curves have turning points around $v = 8$. If k becomes bigger, the slopes before the turning points become steeper.

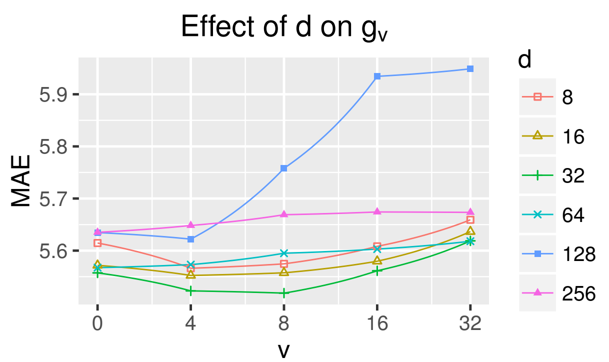

Figure 9. Effect of d on g_v when $k = 8$. As shown in Fig. 9, when $k = 8$ there are usually turning points around $v = 4$ or $v = 8$ except for $d = 256$. When $d = 128$ (and k is too small), there is a MAE peak for big window sizes ($v > 4$). If d grows continuously after the peak, the turning points disappear.

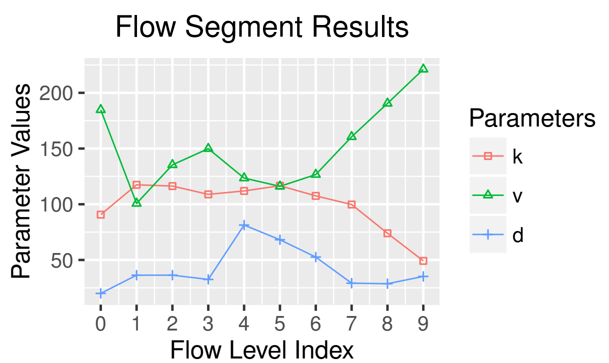

Figure 10. Parameters impacted by flow rate when flow rate is decreasing and increasing (U+D). Higher index means higher flow rate. Values of v are squared to ease display on the same scale.

The **effect of** k **on** g_v is shown in Fig. 8, when search step length is small ($d = 4$), if k is small, there are always turning points around at $v = 8$. If k becomes big, the effect of k on g_v is stable, the pattern of the curves are not changing, but just become more obvious. The **effect of** d **on** g_v is shown in Fig. 9, there are usually turning points around $v = 4$ or $v = 8$ except for $d = 256$. If k increases to 128 (not small any longer), the results indicate that v can reduce around 5% error. However, previous work reported that v is not so useful as it reduced the error by about only 0.05% as shown in the second figure of [28].

4.4 Impact of Flow Rate

To have a robust pattern analysis, the ensemble weights are used to calculate weighted parameter values as shown in Fig. 10. The results show that optimal parameter values are dependent on flow levels. Three parameters have different changing patterns when

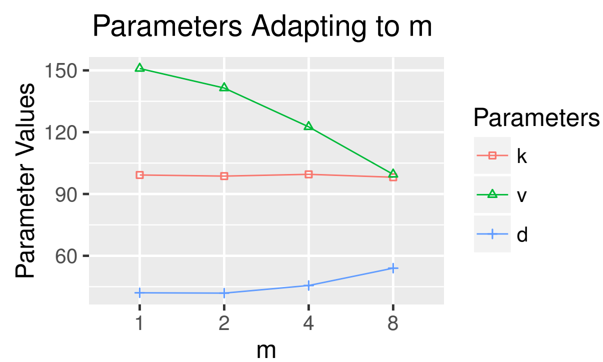

Figure 11. Parameters impacted by m (1 step is 5 minutes). When m increases, k is constant, d is increasing slightly, v is decreasing. Values of v are squared to ease display on the same scale.

the flow changes. Besides, the pattern of v values under increasing flow is different from the pattern under decreasing flow.

4.5 Impact of Predict Step Ahead

As shown in Fig. 11, when m increases, d is increasing slightly. Similar trend has been seen in previous literature [2], but it was not obvious. The result also shows that v is decreasing and k is constant, while [2] reported their k values were going up and down. It might be due to the limited data that had been used.

4.6 Impact of Data Strategies

Results of data strategies are shown in Fig. 12. All strategies have the same parameter options as in parameter strategies except baselines. Baseline A is commonly used (default) parameter tuple values: $k = 8, d = 4, v = 0$, which give MAE 6.2429. It is too high and not shown here. Baseline B is the best single tuple according to training data, potentially with the problem of overfitting. Workday-aware strategy has no segmentation situation, so it is shown as a dashed black line. The results show that all strategies that are considering window size parameter are better than the baseline A. Some flow-aware fixed segmentation strategies are better than baseline B. Ensemble strategies are always better than both baselines, and better than other nonensemble strategies.

5 ANALYSIS OF RESULTS

This section summarizes and interprets the results, and analyses the influence, effects and impacts of strategies and settings.

Analysis of Parameter Strategies. A general pattern is that the error goes down first and then goes up when one parameter increase, i.e., when increasing the values, turning points will occur. This pattern is clearer especially near the optimal values. Small values were used for parameters in some literature (e.g. $k = 8$ or $10, d = 3$ or 4 and $v = 0$) which leads to lack of accuracy. To get higher accuracy, the values need to be increased to turning points. Although the turning points are the best choices for high accuracy, the weighted results from ensemble strategy are more accurate than any specific tuple in the experiments. Even

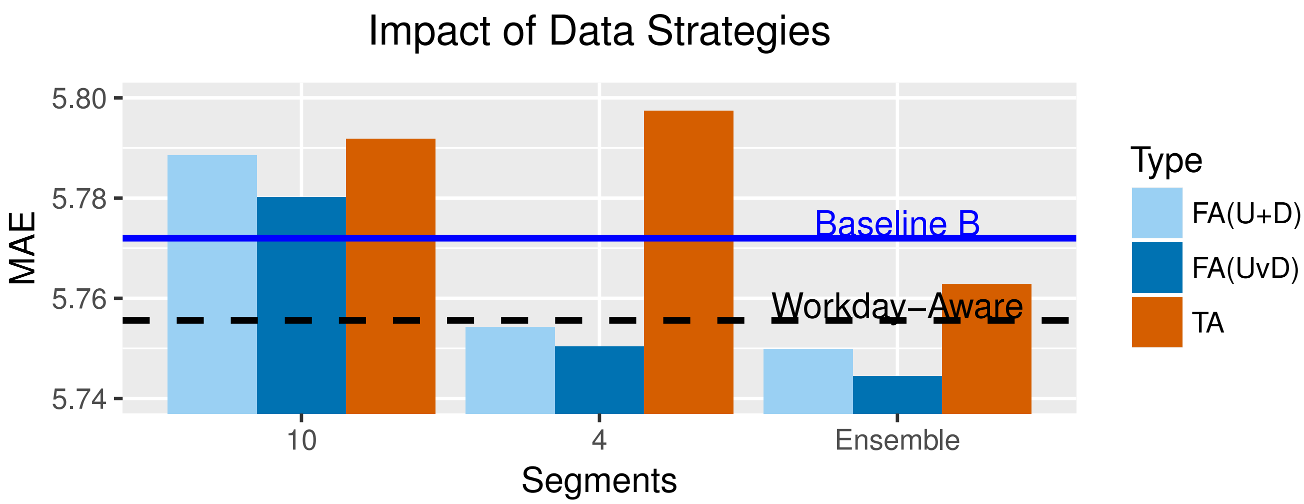

Figure 12. MAE of data strategies, including flow-aware(FA), time-aware(TA) and workday-aware strategy. Three segmentation situations: 10 segments, 4 segments and ensemble(E). Baseline A: common values (not shown). Baseline B (solid blue line): best values from training data.

if the peaks are not influencing final results a lot, they make the influence functions (g) not purely convex. Thus, it is necessary to consider all parameter strategies at the same time instead of using separately. It is also not possible to use simple gradient descent without step size tuned.

Analysis of Predict Step Ahead. As shown in Fig. 11, longer predict time (bigger m) needs longer search steps (bigger d). Besides, window size should be smaller. However, the number of actual selected nearest neighbours (k) is stable. Previous work indicates big m contributes to prediction error a lot, such as the ninth figure in [25] shows around 20% more error when m increases from 1 to 6. However, using our ensemble kNN, when m increases from 1 to 8, prediction error increases only 2%.

Analysis of Data Strategies. It is necessary to separate incremental and decremental flow data. Window size v gives different patterns for increasing and decreasing without doubt, which can be the actual reason that UvD strategies are better than U+D. The TA4 strategy here is not giving a good result. A previous study showed good results with TA4 [23], the reason could be that the segmentation is suitable for their data. One need to be more cautious if a nonensemble method is used. Although we can see clear difference between workday and holiday traffic patterns, the workday-aware strategy is giving less improvement comparing with FA strategies. Thus, FA strategies are preferred, if possible. When the flow rate is low, kNN can benefit from more nearest neighbours. The reason is that low flow occurs in workdays and the number of workdays are much more than holidays. The patterns of d and v are not monotonous. All patterns of the three parameters are nonlinear. Thus, the algorithm should consider flow rate when applying parameter strategies.

6 CONCLUSION

This paper firstly analysed the compound relations when applying parameter strategies. The performance of ensemble kNN cannot be achieved using manual adjustment due to the compound relations, which make it important to consider parameter strategies of all parameters at the same time. It is important to investigate parameters in detail before applying any strategy. Our work is general and it covers previous work in literature where only part of parameter strategies has been considered. For instance, the prediction error decreases continuously when d increases in [25], but we found that there is turning point when d is big enough.

When using data strategies, it is better to separate increasing traffic from decreasing one. Too detailed separation (e.g. 10 segments instead of 4) is usually not good, because each segment has less training data. However, even the number of segments is good, the separation should also be optimised, otherwise different types of instances are in the same segment, TA4 is a negative example here.

In one sentence, we suggest considering all parameter strategies simultaneously as ensemble strategies especially by including v together with k and d in flow-aware strategies.

During the data pre-processing stage, we found the real world data are hard to use directly. Cleaning transportation data requires correct data imputation. We plan to investigate this problem in our future work.

7 ACKNOWLEDGEMENTS

This work was supported by National Natural Science Foundation of China (NSFC), under grant agreement 61364019.

REFERENCES

[1] Marcin Bernas, Bartlomiej Placzek, Piotr Porwik, and Teresa Pamula. 2015. Segmentation of Vehicle Detector Data for Improved K-Nearest Neighbours-Based

Traffic Flow Prediction. *IET Intelligent Transport Systems* 9, 3 (April 2015), 264–274. https://doi.org/10.1049/iet-its.2013.0164

[2] H. Chang, Y. Lee, B. Yoon, and S. Baek. 2012. Dynamic Near-Term Traffic Flow Prediction: System-Oriented Approach Based on Past Experiences. *IET Intelligent Transport Systems* 6, 3 (Sept. 2012), 292–305. https://doi.org/10.1049/iet-its.2011.0123

[3] Edward Chung. 2003. Classification of Traffic Pattern. In *10th World Congress on Intelligent Transport Systems*, Vol. 10. Brussels, 11.

[4] Pietro Dell'Acqua, Francesco Bellotti, Riccardo Berta, and Alessandro De Gloria. 2015. Time-Aware Multivariate Nearest Neighbor Regression Methods for Traffic Flow Prediction. *IEEE Transactions on Intelligent Transportation Systems* 16, 6 (2015), 3393–3402.

[5] Jianhua Guo, Wei Huang, and Billy M. Williams. 2015. Real Time Traffic Flow Outlier Detection Using Short-Term Traffic Conditional Variance Prediction. *Transportation Research Part C: Emerging Technologies* 50 (Jan. 2015), 160–172. https://doi.org/10.1016/j.trc.2014.07.005

[6] Ahmad Basheer Hassanat, Mohammad Ali Abbadi, Ghada Awad Altarawneh, and Ahmad Ali Alhasanat. 2014. Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach. *International Journal of Computer Science and Information Security* 12, 8 (Sept. 2014), 33–39. arXiv:1409.0919

[7] Haikun Hong, Xiabing Zhou, Wenhao Huang, Xingxing Xing, Fei Chen, Yu Lei, Kaigui Bian, and Kunqing Xie. 2015. Learning Common Metrics for Homogenous Tasks in Traffic Flow Prediction. In *IEEE 14th International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 1007–1012. https://doi.org/10.1109/ICMLA.2015.188

[8] S. Innamaa. 2009. Self-Adapting Traffic Flow Status Forecasts Using Clustering. *IET Intelligent Transport Systems* 3, 1 (March 2009), 67–76. https://doi.org/10.1049/iet-its:20070048

[9] Yiannis Kamarianakis, H. Oliver Gao, and Poulicos Prastacos. 2010. Characterizing Regimes in Daily Cycles of Urban Traffic Using Smooth-Transition Regressions. *Transportation Research Part C-Emerging Technologies* 18, 5 (Oct. 2010), 821–840. https://doi.org/10.1016/j.trc.2009.11.001

[10] Elia Liitiainen, Michel Verleysen, Francesco Corona, and Amaury Lendasse. 2009. Residual Variance Estimation in Machine Learning. *Neurocomputing* 72, 16-18 (Oct. 2009), 3692–3703. https://doi.org/10.1016/j.neucom.2009.07.004

[11] Lei Lin, Qian Wang, and Adel W. Sadek. 2013. Short-Term Forecasting of Traffic Volume - Evaluating Models Based on Multiple Data Sets and Data Diagnosis Measures. *Transportation Research Record* 2392 (2013), 40–47. https://doi.org/10.3141/2392-05

[12] Hrvoje Markovic, Bojana Dalbelo Basic, Hrvoje Gold, Fang Dong, and Kaoru Hirota. 2010. GPS Data-Based Non-Parametric Regression for Predicting Travel Times in Urban Traffic Networks. *PROMET-Traffic & Transportation* 22, 1 (2010).

[13] Nvidia. 2016. *Cuda Programming Guide 8.0*. Nvidia.

[14] Simon Oh, Young-Ji Byon, Kitae Jang, and Hwasoo Yeo. 2015. Short-Term Travel-Time Prediction on Highway: A Review of the Data-Driven Approach. *Transport Reviews* 35, 1 (Jan. 2015), 4–32. https://doi.org/10.1080/01441647.2014.992496

[15] N. H. Packard, J. P. Crutchfield, J. D. Farmer, and R. S. Shaw. 1980. Geometry from a Time Series. *Physical Review Letters* 45, 9 (Sept. 1980), 712–716. https://doi.org/10.1103/PhysRevLett.45.712

[16] R Core Team. 2016. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

[17] Roger P. Roess, Elena S. Prassas, and William R. McShane. 2010. *Traffic Engineering* (4 ed.). Prentice Hall, Upper Saddle River, NJ.

[18] Himanshu Singh, Aditya Desai, and Vikram Pudi. 2010. PAGER: Parameter Less, Accurate, Generic, Efficient kNN-Based Regression. In *Database and Expert Systems Applications, Pt 2*, P. G. Bringas, A. Hameurlain, and G. Quirchmayr (Eds.). Vol. 6262. Springer-Verlag Berlin, Berlin, 168–176.

[19] Bin Sun, Wei Cheng, Guohua Bai, and Prashant Goswami. 2017, in press. Correcting and Complementing Freeway Traffic Accident Data Using Mahalanobis Distance Based Outlier Detection. *Tehnicki Vjesnik-Technical Gazette* 24, 5 (Oct. 2017, in press). https://doi.org/10.17559/TV-20150616163905

[20] Bin Sun, Wei Cheng, Prashant Goswami, and Guohua Bai. 2017. Flow-Aware WPT k-Nearest Neighbours Regression for Short-Term Traffic Prediction. In *2017 IEEE Symposium on Computers and Communication (ISCC)*.

[21] TRB. 2010. *Highway Capacity Manual (HCM)*. Transportation Research Board, Washington, DC.

[22] Eleni I. Vlahogianni, Matthew G. Karlaftis, and John C. Golias. 2014. Short-Term Traffic Forecasting: Where We Are and Where We're Going. *Transportation Research Part C: Emerging Technologies* 43, Part 1 (June 2014), 3–19. https://doi.org/10.1016/j.trc.2014.01.005

[23] D. Wang, Q. Zhang, S. Wu, X. Li, and R. Wang. 2016. Traffic Flow Forecast with Urban Transport Network. In *2016 IEEE International Conference on Intelligent Transportation Engineering (ICITE)*. 139–143. https://doi.org/10.1109/ICITE.2016.7581322

[24] Shanhua Wu, Zhongzhen Yang, Xiaocong Zhu, and Bin Yu. 2014. Improved K-Nn for Short-Term Traffic Forecasting Using Temporal and Spatial Information. *Journal of Transportation Engineering* 140, 7 (July 2014), 04014026. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000672

[25] Byoungjo Yoon and Hyunho Chang. 2014. Potentialities of Data-Driven Nonparametric Regression in Urban Signalized Traffic Flow Forecasting. *Journal of Transportation Engineering* 140, 7 (July 2014), 04014027. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000662

[26] Bin Yu, Xiaolin Song, Feng Guan, Zhiming Yang, and Baozhen Yao. 2016. K-Nearest Neighbor Model for Multiple-Time-Step Prediction of Short-Term Traffic Condition. *Journal of Transportation Engineering* 142, 6 (2016), 04016018. https://doi.org/10.1061/(ASCE)TE.1943-5436.0000816

[27] Xueying Zhang and Qinbao Song. 2014. Predicting the Number of Nearest Neighbors for the K-NN Classification Algorithm. *Intelligent Data Analysis* 18, 3 (2014), 449–464. https://doi.org/10.3233/IDA-140650

[28] Zuduo Zheng and Dongcai Su. 2014. Short-Term Traffic Volume Forecasting: A k-Nearest Neighbor Approach Enhanced by Constrained Linearly Sewing Principle Component Algorithm. *Transportation Research Part C-Emerging Technologies* 43 (June 2014), 143–157. https://doi.org/10.1016/j.trc.2014.02.009

[29] Yuexian Zou, Guangyi Shi, Hang Shi, and He Zhao. 2011. Traffic Incident Classification at Intersections Based on Image Sequences by HMM/SVM Classifiers. *Multimedia Tools and Applications* 52, 1 (March 2011), 133–145. https://doi.org/10.1007/s11042-010-0466-6