Automated Model Construction for Combined Sewer Overflow Prediction Based on Efficient LASSO Algorithm

Wanqing Zhao, *Member, IEEE*, Thomas H. Beach, and Yacine Rezgui

***Abstract*—The prediction of combined sewer overflow (CSO) operation in urban environments presents a challenging task for water utilities. The operation of CSOs (most often in heavy rainfall conditions) prevents houses and businesses from flooding. However, sometimes, CSOs do not operate as they should, potentially bringing environmental pollution risks. Therefore, CSOs should be appropriately managed by water utilities, highlighting the need for adapted decision support systems. This paper proposes an automated CSO predictive model construction methodology using field monitoring data, as a substitute for the commonly established hydrological-hydraulic modeling approach for time-series prediction of CSO statuses. It is a systematic methodology factoring in all monitored field variables to construct time-series dependencies for CSO statuses. The model construction process is largely automated with little human intervention, and the pertinent variables together with their associated time lags for every CSO are holistically and automatically generated. A fast least absolute shrinkage and selection operator solution generating scheme is proposed to expedite the model construction process, where matrix inversions are effectively eliminated. The whole algorithm works in a stepwise manner, invoking either an incremental or decremental movement for including or excluding one model regressor into, or from, the predictive model at every step. The computational complexity is thereby analyzed with the pseudo code provided. Actual experimental results from both single-step ahead (i.e., 15 min) and multistep ahead predictions are finally produced and analyzed on a U.K. pilot area with various types of monitoring data made available, demonstrating the efficiency and effectiveness of the proposed approach.**

***Index Terms*—Combined sewer overflows (CSOs), efficient model construction, hydraulics, prediction, wastewater.**

I. Introduction

WITHIN the water infrastructure of an urban environment, combined sewer systems (CSSs) are commonly employed to collect and convey both stormwater from precipitation events and sewage/wastewater from domestic, industrial, commercial and municipal release together in the same sewer [1]–[3]. The wastewater gathered in the CSS is then directed to wastewater treatment plants (WWTPs), usually driven by gravity through paved inclined sewers together with a small number of lift pumping stations to assist water transportation between sewers. It should be noted the fact that in dry weather conditions and during light to moderate rainfall, CSSs are usually designed to be capable of conveying all the flows to WWTPs [4]. Besides, a compelling feature of CSSs is that the system is equipped with combined sewer overflow (CSO) structures [5]–[7] to discharge combined untreated wastewater and stormwater runoff to receiving water bodies (via a consent from governing bodies), in order not to overload the maximum capacity of CSSs in case of heavy rainfall (sometimes even moderate rainfall in reality) [4], [8]. The occurrence of CSO spillages, especially unexpected ones can potentially lead to environmental pollution [9]. A variety of research has therefore been devoted to identifying various pollutants, the possible impact on the environment and drinking water quality and correspondingly coping strategies [10], [11]. Since the actual use of CSOs is stringently regulated by environmental agencies, unexpected spillages can also incur fines and damage public relations of water utilities. Whilst the release of diluted wastewater via CSO structures has potential adverse effects, it can, however, help avoid overloading of CSSs and reduce the risk of sewer flooding on properties and streets [8], [12]. Therefore, the real-time CSO status should be adequately monitored and predicted to support the waste network management process.

To gain insight into CSO behaviors, typically, physical hydrological-hydraulic models of the large-scale catchment of interest must be developed and elaborately calibrated, in connection with the actual CSO level formation process (i.e., the hydrological-hydraulic process of forming CSO levels, from rainfall through to runoff and network flows) [13]–[15]. The whole model development process is time-consuming and associated with high costs. Sufficient spatial and temporal resolution of rainfall data [16] monitored from either rain-gauge stations or weather radar and flow survey data are needed to delicately calibrate such physical models. In this regard, Schellart *et al.* [17] discussed different sources of errors that might be presented in rain-gauge (e.g., blockages, wetting, and evaporation) and radar (e.g., spurious echoes and attenuation) approaches. Currently, dedicated commercial products, such as InfoWorks ICM [18], SWMM [19], and MOUSE [20], can be

Manuscript received February 22, 2017; revised April 30, 2017; accepted June 22, 2017. Date of publication August 21, 2017; date of current version May 14, 2019. This work was supported by the EU Seventh Framework Programme under Grant 619795 (WISDOM) involving 11 partners, including universities/institutes, local authorities, water utilities, and ICT companies. This paper was recommended by Associate Editor K. T. Seow. *(Corresponding author: Wanqing Zhao.)*

The authors are with the School of Engineering, Cardiff University, Cardiff CF24 3AA, U.K. (e-mail: zhaow9@cardiff.ac.uk; beachth@cardiff.ac.uk; rezguiy@cardiff.ac.uk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2017.2724440

used to build these physical models. A list of weaknesses faced by this conventional model building approach is summarized in Section II.

Alternatively, the control-oriented modeling approach [21], [22] using the "virtual tank" concept can be adopted to approximate the field model. It is a simplified mathematical processing of a number of subcatchments divided from the whole catchment of interest, where the network topology and some conversion and absorption coefficients need to be specified and estimated. Such modeling process requires the involvement of field experts, which can be viewed as a lighter version of the traditional physical modeling approach. It also includes manual analysis processes, such as model structure determination based on the provision of network topology and division of catchment. With such simplified model, model-based predictive control techniques can then be adopted to control the sewerage network, for example, with the implementation of detention tanks and actuation of retention/diversion gates.

Unlike the above model designed for control problems, recent research has begun to study employing data-driven approaches to tackle the time-series prediction of CSOs for warning of future problems within the sewerage network, such as the use of artificial neural networks [23]–[25]. Their methodologies have been found favorable for modeling the water hydrological-hydraulic behaviors without the need for an in-depth understanding of the underlying sewer system. Of these studies, the cross-correlation between the CSO and rainfall variables, and the serial-correlation within the CSO, are manually analyzed for various time lags in order to find an appropriate range of lags to be considered as data-driven model's inputs. Though promising results have been demonstrated, there is still a lack of systematic work. Some research questions are still open to be addressed for the data-driven approach.

1) The whole model construction process needs a certain degree of human intervention (e.g., the manual correlation analysis and model trial processes vary from site to site) and is not fully automated, limiting the transferability of CSO model construction in different catchments and urban configurations.
2) The quantity of CSOs being analyzed is limited where only one or several CSOs are studied within a catchment for predictive model construction.
3) Only pairwise correlation between a CSO and a variable is considered sequentially with distinct time lags when determining appropriate variable lags, while the global relationship across all field variables is not fully analyzed and utilized.
4) Only CSO and rainfall data are included in the model construction, rarely are other field variables investigated.
5) The interrelationships between different CSOs are not captured in the model construction process.
6) The whole model construction process is still time-consuming.

The research problem in this paper, therefore, is to achieve the real-time prediction of future CSO statuses [single-step (i.e., 15 min) and multistep ahead predictions] using the previously observed statuses of pertinent field variables without the need of human intervention and network topology. The problem is defined on top of the field under investigation and the current practice of network operation (e.g., using predictive control or local control techniques). In light of the aforementioned considerations, this paper proposes a systematic and automated approach for CSO predictive model construction. The whole model construction process is largely automated based on monitored field variables, in the catchment of interest. The least absolute shrinkage and selection operator (LASSO) is employed to perform field variable and the associated time lag selection, as well as model construction in a stepwise manner under the well-known L_1 norm regularization. With the adjustment of the regularization parameter, the overall model size is controllable, enabling the determination of pertinent field variables and lags for a particular CSO. To improve the computational efficiency of model construction, an efficient LASSO solution generating scheme is proposed based on least angle regression (LAR). The matrix inversions are thereby eliminated and different model sizes in the LASSO sense can be produced in sequence. A real catchment is studied as part of this paper with more than 20 CSOs with good data availability. In addition to the CSO and rainfall data, other field variables such as wet well levels and pumped flows are also examined in the model construction. All field variables are dealt with simultaneously in a global manner during the model construction process, set by the criteria of L_1 norm regularization, without considering each correlation separately and independently. Specifically, given the overall methodology, each CSO predictive model also captures its interrelationships with other pertinent CSOs in the field.

This paper is organized as follows. The preliminary relevant domain knowledge and mathematical formulation of CSO predictive models are given in Section II. Section III presents the LASSO concept and its stepwise solutions. The efficient LASSO solution generating scheme for CSO model construction is then given in Section IV. The mathematical derivations, algorithm and computational complexity are all detailed therein. The experimental results from a U.K. pilot area are presented in Section V, where a description of the catchment and the detailed analysis of model construction results are given. Finally, Section VI concludes this paper.

II. Preliminaries and CSO Model Formulation

To predict CSO statuses, the conventional approach uses a first-principle mechanism by constructing a physical model conceptualizing the actual process of CSO level formation. It usually involves the development of two submodels that correspond to the two subprocesses of the CSO level formation, i.e., hydrological process and hydraulic process, each briefly described below.

1) The hydrological process that takes place in the catchment, where rainfall is the input of the process and the runoff hydrograph is the output. It consists of the calculation of effective precipitation and then the calculation of runoff hydrographs, factoring in parameters, such as evaporation, infiltration, wetting, and surface storage [13]. Through this submodel construction, a hydrological rainfall-runoff relationship can be established.

2) The hydraulic process that takes place inside the underground sewer network, where the above runoff hydrograph together with the sewage/wastewater released from residents and businesses is the input of the process, leading to the generation of network flows and CSO levels. The diluted wastewater in the network is conveyed either to WWTPs for treatment before being discharged or to CSO structures for direct discharge into the environment (e.g., following a heavy rainfall). Through this submodel construction, a hydraulic relationship between runoff/wastewater and network flows/CSO statuses can be established.

A conventional modeling approach can therefore be used to predict CSO statuses given the predicted/calibrated meteorology rainfall and wastewater release information. However, the whole model construction process is time-consuming and costly. The developed model also needs to be carefully examined and calibrated. Some key restrictions for use of such conventional approach to develop CSO predictive models are as follows.

1) In-depth expert knowledge in the wastewater domain.
2) Sophisticated modeling skills in terms of using various modeling software packages.
3) Detailed information gathered about the catchment and sewer network.
4) Lack of model transportability (site specific).
5) Low model adaptability to change (e.g., modification and aging of network).
6) Long model development cycle.
7) High model development expenses.
8) Time-consuming model simulation process.
9) Complicated model calibration process.

Although the advantage of using a hydrological-hydraulic approach lies in the ability to provide accurate and reliable CSO time-series predictions, its wide applications are inevitably limited by the above restrictions. Hence, as an alternative to conventional models, in the sense of providing time-series predictions for a number of time-steps, the data-driven approach relying on the monitoring data from the field is to be investigated in this paper. The intention is not to completely replace the often-needed hydrological-hydraulic model, as the latter is essential to support a number of stormwater management functionalities, such as the modeling of drainage networks for real-time control and water behavior analysis. It is worth noting that the aim of this paper is to predict future CSO statuses (e.g., the future statuses of the next 15–60 min) in real-time for the purpose of daily network management under given catchment, network and operational configurations, rather than performing predictions between different constructions of network and catchment infrastructure. The low model adaptability comment above, therefore, refers to the fact that, in order to provide such CSO time-series predictions after the change of catchment and network, the physical network modeling approach would need a high level of human intervention and calibration work. In contrast, the data-driven approach just requires an execution of the automated model reconstruction based on new monitoring data. However, sufficient quantity of new data, say one year, should be collected to reflect the full

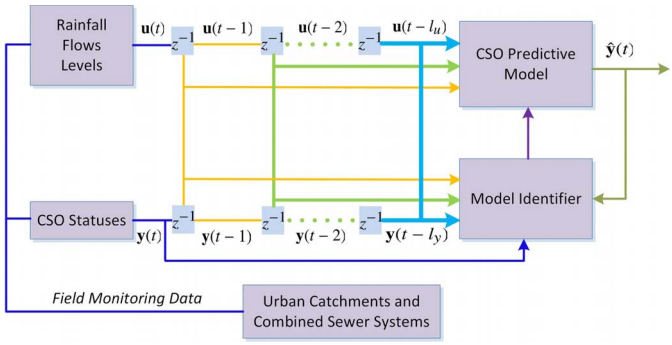


Fig. 1. CSO data-driven model prediction framework.

complexity of the CSO behavior (e.g., seasonal variations). Practically, the model can be reconstructed every month to improve accuracy before the required historical data has been made available. Within the data-driven approach, the nonlinear dynamic time-series relationships between input/process variables and the CSO status are then artificially established exploring the measured data. Considering the general availability of data, the field variables presented in the problem can be categorized as system input variables, system process variables, and CSO outputs, which are in turn defined as follows.

1) *System Input Variables:* Specifying the external information that can be generally considered as the inputs to the sewer system.
   a) *Rainfall Data:* This involves real-time rainfall information which is envisaged to have large impact on CSO level formation. Rain-gauge and radar measurements are both acceptable with sufficient spatial (e.g., 1–5 km) and temporal (e.g., 5–15 min) resolutions [16], [17], [26], [27] for the catchments of interest.
   b) *Consumer's Discharge:* It provides another channel of input to the sewer system from the domestic release of sewage as well as other forms of discharge such as industrial wastewater. This is deemed to have considerably smaller impact to the occurrence of CSO spillage.
2) *System Process Variables:* Specifying the internal information originating from the operation of sewer systems.
   a) Flow variables, such as those monitored from pumps, sewers, treatment works, and inlets and outlets of temporary storage tanks.
   b) Level variables, such as those monitored from wet wells and manholes.
   c) Pump running statuses.
3) *CSO Statuses:* Specifying the actual levels/flows in CSO structures. As CSO statuses are of particular interest to the underlying research question, these are treated differently to other system process variables described above and thus taken as the output variable from the sewer system.

Given the above discussion, the following well-known time-series prediction model (as depicted in Fig. 1), the nonlinear autoregressive model with exogenous inputs [28]–[30], is

employed to cope with the data-driven prediction of CSOs in the waste sector

$$y_i(t) = f_i\big(\mathbf{u}(t-1), \ldots, \mathbf{u}(t-l_u), \mathbf{y}(t-1), \ldots, \mathbf{y}\big(t-l_y\big)\big) + e_i(t), i = 1, \ldots, N_{\mathrm{CSO}} \quad (1)$$

where $\mathbf{u}(t) = [u_1(t), \ldots, u_{N_{\mathrm{IP}}}(t)]$ denotes the actual sewer system's input and process variables (with the total number being N_{IP}) at time sequence t with a maximal time lag of l_u, $\mathbf{y}(t) = [y_1(t), \ldots, y_{N_{\mathrm{CSO}}}(t)]$ defines the CSO output variables with a maximal time lag of l_y, and N_{CSO} is the total number of CSO structures, $f_i(\cdot)$ and $e_i(t)$ are, respectively, the predictive model and model residual to be identified for the ith CSO. As shown in Fig. 1, the CSO prediction problem lies in the identification of a data-driven model that relates the future CSO status to the past status of field variables. It should be noted that though the next step (the next 15 min for the pilot area to be given in Section V-A) prediction is formulated in (1) and primarily pursued in this paper, multistep predictions can simply be achieved by replacing $y_i(t)$ and $e_i(t)$ with $y_i(t+t_m)$ and $e_i(t+t_m)$ therein (where $t_m + 1 \geq 2$ is the number of multiple steps predicted ahead). For simplicity, the subscript i in $y_i(t)$, $f_i(\cdot)$ and $e_i(t)$ will be omitted from now on to generally refer to any CSO quantity.

The task then involves extracting a proper set of field variables associated with certain time lags to construct the following model for each CSO identity:

$$y(t) = \sum_{i=1}^{m} p_i(t)\Theta_{i,m} + e(t) \quad (2)$$

where $p_i(t)$ $(i = 1, \ldots, m)$ is the ith model regressor selected from the set $\boldsymbol{\phi}(t) = [\varphi_1(t), \ldots, \varphi_M(t)] = [\mathbf{u}(t-1), \ldots, \mathbf{u}(t-l_u), \mathbf{y}(t-1), \ldots, \mathbf{y}(t-l_y)] \in \Re^M$ $(M = N_{\mathrm{IP}}l_u + N_{\mathrm{CSO}}l_y)$, $\Theta_{i,m}$ is the corresponding model coefficient for the ith regressor, and m is the number of selected model regressors. Assuming that the full set of $\boldsymbol{\phi}(t)$ is employed at the beginning of model construction while N training samples are provided, (2) can be reformulated as the following generic matrix form:

$$\mathbf{y} = \boldsymbol{\Phi}\boldsymbol{\Theta} + \mathbf{e} \quad (3)$$

where $\mathbf{y} = [y(1), \ldots, y(N)]^T \in \Re^N$ and $\mathbf{e} = [e(1), \ldots, e(N)]^T \in \Re^N$ are, respectively, the actual CSO output and model error vectors, $\boldsymbol{\Phi} = [\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M] \in \Re^{N\times M}$ formulates the initial CSO regression matrix $(\boldsymbol{\varphi}_i = [\varphi_i(1), \ldots, \varphi_i(N)]^T$, $1 \leq i \leq M)$, and $\boldsymbol{\Theta} = [\Theta_{1,M}, \ldots, \Theta_{M,M}]^T$ is the model parameter vector.

III. Lasso and Its Stepwise Solutions

Given (3), a relevant set of field variables together with appropriate time lags must be identified in order to construct the predictive model of a particular CSO. In this paper, LASSO [31]–[34] is adopted to perform such variable selection. Compared with other model selection methods, such as forward/backward stepwise selection and ridge regression [28], LASSO is able to perform both variable selection and regularization with enhanced generalization ability, while also possessing geometric and Bayesian interpretations [35]. Essentially, the objective function of LASSO is to minimize

$$J(\lambda, \boldsymbol{\Theta}) = \frac{1}{2}\mathbf{e}^T\mathbf{e} + \lambda||\boldsymbol{\Theta}||_1 \quad (4)$$

where λ is a tradeoff parameter controlling the degree of L_1 regularization. Noting that the L_1 regularization possesses better shrinking properties compared to the well-known L_2 ridge regularization $(||\boldsymbol{\Theta}||_2^2)$, in terms of being able to force part model coefficients exactly to zeros [36]. However, the objective function is no longer quadratic though still convex; the corresponding solutions become nonlinear and no closed form expression is thus available. As λ varies from 0 to larger values, the resultant coefficients generally move from the least-squares estimate to partial zeros, until complete zeros (meaning that different sizes of optimal model regressors can be selected in the LASSO sense).

Through deducing the gradient and subgradient of the objective function (4) with respect to the model coefficient vector $\boldsymbol{\Theta}$, the following Karush–Kuhn–Tucker optimal conditions [34] can be obtained for deriving an LASSO solution for a given λ:

$$\boldsymbol{\Phi}^T\mathbf{e} = \lambda\mathbf{s} \quad (5)$$

where $\mathbf{s} = [s_1, \ldots, s_M]^T$ and

$$s_i \in \begin{cases} \{1\}, & \hat{\Theta}_{i,M} > 0 \quad (6) \\ \{-1\}, & \hat{\Theta}_{i,M} < 0 \quad (7) \\ [-1, 1], & \hat{\Theta}_{i,M} = 0. \quad (8) \end{cases}$$

Therefore, an LASSO solution $(\hat{\Theta}_{i,M},\ i = 1, \ldots, M)$ is considered to satisfy (5)–(8), simultaneously. This leads to a quadratic programming problem and there is no general analytical solution available [37].

To efficiently solve the LASSO problem, Efron *et al.* [38] proposed a novel approach based on LAR. Interestingly, the LAR approach operates in a stepwise selection manner. This means that it is able to locate the global optimum model regressors in different sizes in a stepwise manner, in correspondence to different values of λ in the LASSO sense. This is an important property as the traditional forward/backward stepwise selection has historically been only able to search for suboptimal subsets of regressors in the least-squares sense. In detail, every step, say at the kth step, a submodel $\boldsymbol{\Phi}_k\hat{\boldsymbol{\theta}}_k$ is introduced to explain the remaining model error $\mathbf{e}_{k-1}$ resulting from the previous step, where $\hat{\boldsymbol{\theta}}_k = \gamma_k(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k)^{-1}\boldsymbol{\Phi}_k^T\mathbf{e}_{k-1}$, $\boldsymbol{\Phi}_k = [\mathbf{p}_1, \ldots, \mathbf{p}_k]$ and γ_k are, respectively, the submodel coefficients, selected model regressors and step size. As γ_k increases from zero, the next model regressor $(\mathbf{p}_{k+1})$ to be selected is determined such that the largest absolute correlation between those unselected regressors (say $\boldsymbol{\varphi}_i$, $i = k+1, \ldots, M$) and the resulting model error $\mathbf{e}_k$ is first found just equal to the absolute correlation incurred by any selected regressors. This will make the absolute correlations exhibited for selected regressors always equal to one another and no smaller than those for unselected ones. Based on this property and to find successive sets of LASSO solutions, the following two quantities of γ_k $(k = 1, \ldots, M-1)$ need to be computed:

$$\begin{cases} \gamma_k^{\star} = \min\limits_{i=k+1}^{M}\left[\dfrac{\pm\boldsymbol{\varphi}_i^T\mathbf{e}_{k-1} - \left|\mathbf{p}_k^T\mathbf{e}_{k-1}\right|}{\pm\boldsymbol{\varphi}_i^T\boldsymbol{\Phi}_k(\boldsymbol{\Phi}_k^T\boldsymbol{\Phi}_k)^{-1}\boldsymbol{\Phi}_k^T\mathbf{e}_{k-1} - \left|\mathbf{p}_k^T\mathbf{e}_{k-1}\right|}\right]_+ & (9) \\ \gamma_k^{\circ} = \min\limits_{i=1}^{k-1}[-\hat{\Theta}_{i,k-1}/\hat{\theta}_{i,k}]_+ & (10) \end{cases}$$

where "$[]_+$" denotes retrieving the minimum but positive value for the above two quantities of γ_k, "$\pm$" means that the corresponding values in the denominator and numerator of (9) should be chosen as either positive or negative simultaneously, $\hat{\Theta}_{i,k-1}$ is the accumulated coefficient for the ith model regressor until the previous step with $k-1$ regressors selected, and $[\hat{\theta}_{1,k}, \ldots, \hat{\theta}_{k,k}]^T = (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T \mathbf{e}_{k-1}$.

Here, on the one hand, if $0 < \gamma_k^\circ \leq \gamma_k^\star$ is not met, then $\gamma_k = \gamma_k^\star$ and the $(k+1)$th regressor $\mathbf{p}_{k+1} = \arg \gamma_k^\star$ is selected to form the new regression matrix, i.e., $\mathbf{\Phi}_{k+1} = [\mathbf{\Phi}_k, \mathbf{p}_{k+1}]$. The LASSO conditions (5)–(8) are all satisfied simply because the absolute correlations as stated in (5)–(7) for the selected regressors are the same (being the value of the tradeoff parameter λ) and greater than that for any candidate regressors as indicated in (5) and (8). On the other hand, if $0 < \gamma_k^\circ \leq \gamma_k^\star$ is met, then $\gamma_k = \gamma_k^\circ$ and the rth regressor $\mathbf{p}_r = \arg \gamma_k^\circ$ is to be removed from the current regression matrix, resulting in the new one $\tilde{\mathbf{\Phi}}_{k-1} = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_{k-1}]$. This is because if the selection proceeds using $\gamma_k^\star$ and thus the model size increases, the model coefficient sign for regressor $\mathbf{p}_r$ is going to be changed and thus different to the sign of this term's correlation, breaking the LASSO sign condition (6) or (7). To continue meeting the LASSO conditions, $\mathbf{p}_r$ is required to be removed from the model, through adjusting the step size (i.e., $\gamma_k = \gamma_k^\circ$) accordingly to just make the corresponding model coefficient equal to zero. Therefore, as the controlling parameter λ (equally the absolute correlation for the selected regressors) decreases, it can be found that though the number of nonzero model coefficients overall increases it does not increase in a monotonic fashion.

IV. Efficient Lasso Algorithm for Automated CSO Model Construction

As presented in the previous section, the key task now is to determine the size γ_k successively in order to derive different sets of LASSO solutions. This in turn lies in how to efficiently compute $\gamma_k^\star$ and γ_k° based on (9) and (10). An efficient LASSO solution generating scheme is proposed in this section to relax the heavy computation requirements of performing matrix inversions and vector correlations. As mentioned previously, the model construction process guided by the LASSO criterion involves the bidirectional movement of model regressors for either including or excluding a regressor at every step, i.e., incremental movement and decremental movement. The incremental movement performs the same as in the original LAR and its efficient solution was recently introduced by Zhao *et al.* [39], where a regression framework was given for forward selection in LAR. To enable the efficient derivation of LASSO solutions, the incremental movement is briefly introduced first, followed by the decremental movement.

A. Incremental Movement

To perform efficient computations in the case that only the increase of model size is allowed, a new efficient LAR algorithm has recently been presented in [39]. In detail, a so-called residue matrix is defined as $\mathbf{R}_k = \mathbf{I} - \mathbf{\Phi}_k (\mathbf{\Phi}_k^T \mathbf{\Phi}_k)^{-1} \mathbf{\Phi}_k^T$, with the following main properties [39], [40]:

$$\begin{cases} \mathbf{R}_k = \mathbf{R}_{k-1} - \dfrac{\mathbf{R}_{k-1}\mathbf{p}_k\mathbf{p}_k^T\mathbf{R}_{k-1}^T}{\mathbf{p}_k^T\mathbf{R}_{k-1}\mathbf{p}_k}, & k = 1, \ldots, M \quad (11) \\ \mathbf{R}_k\mathbf{p}_i = \mathbf{0};\ \mathbf{R}_k\mathbf{e}_i = \mathbf{R}_k\mathbf{y}, & i = 1, \ldots, k \quad (12) \\ \mathbf{p}_i^T\mathbf{R}_{i-1}\mathbf{e}_j = \mathbf{p}_i^T\mathbf{R}_{i-1}\mathbf{y}\displaystyle\prod_{l=i}^{j}(1-\gamma_l) & \\ \qquad i \leq j \leq k-1,\ 1 \leq i \leq k-1. & \quad (13) \end{cases}$$

To facilitate continuous computation of (9), a set of variables including scalars, vectors and matrices have been used and updated throughout the model construction process. Assuming that a total of k regressors have just been added into the model while deciding the size of $\gamma_k^\star$, those variables are expressed as follows ($\rho_k \in \Re$, $\mathbf{c}^{k)} \in \Re^{M-k}$, $\mathbf{d}^{k)} \in \Re^{M-k}$, $\mathbf{A}^{k)} \in \Re^{k \times M}$, and $\mathbf{b}^{k)} \in \Re^k$), which can later be extended in the decremental movement where removal of model regressor occurs:

$$\begin{cases} \rho_k = \left|\mathbf{p}_i^T\mathbf{e}_{k-1}\right|, & i = 1, \ldots, k \quad (14) \\ c_i^{k)} = \boldsymbol{\varphi}_i^T\mathbf{e}_k, k = 0, \ldots, M-1; i = k+1, \ldots, M & \quad (15) \\ d_i^{k)} = \boldsymbol{\varphi}_i^T\mathbf{\Phi}_k(\mathbf{\Phi}_k^T\mathbf{\Phi}_k)^{-1}\mathbf{\Phi}_k^T\mathbf{e}_{k-1} & \\ \qquad k = 1, \ldots, M-1; i = k+1, \ldots, M & \quad (16) \\ a_{k,i} = \mathbf{p}_k^T\mathbf{R}_{k-1}\boldsymbol{\varphi}_i, \quad k = 1, \ldots, M; i = k, \ldots, M & \quad (17) \\ b_k = \mathbf{p}_k^T\mathbf{R}_{k-1}\mathbf{y}, & k = 1, \ldots, M. \quad (18) \end{cases}$$

The step size in (9) is now computed as

$$\gamma_k^\star = \min_{i=k+1}^{M} \left[\frac{\pm c_i^{k-1)} - \rho_k}{\pm d_i^{k)} - \rho_k} \right]_+, \quad k = 1, \ldots, M-1. \quad (19)$$

Therefore, whilst only the increase of model size is considered, the aforementioned (11)–(19) give the rationales for fast computation of its step size. The full algorithm, computational complexity and relevant derivations can be referred to [39]. Based on these, the efficient removal of model regressors to obtain LASSO solutions when triggering condition (10) will be given based on the further adjustment of (14)–(18).

B. Decremental Movement

Assuming that now we are going to determine the step size γ_k (where an error vector $\mathbf{e}_{k-1}$ and a total of k selected model regressors are given) and $\gamma_k^\star$ is computed as in the previous section as if the $(k+1)$th regressor is to be added into the predictive model, the value of γ_k° is taken as the smallest positive one that drives an existing model coefficient to zero, given by (10). First of all, $\hat{\theta}_{i,k}$ can be computed as

$$\begin{aligned} &\left[\hat{\theta}_{1,k}, \ldots, \hat{\theta}_{k,k}\right]^T \\ &\quad = \left(\mathbf{\Phi}_k^T\mathbf{\Phi}_k\right)^{-1}\mathbf{\Phi}_k^T\mathbf{e}_{k-1} \\ &\quad = \left(\mathbf{\Phi}_k^T\mathbf{\Phi}_k\right)^{-1}\mathbf{\Phi}_k^T\left(\mathbf{y} - \mathbf{\Phi}_{k-1}\hat{\mathbf{\Theta}}_{k-1}\right) \\ &\quad = \left(\mathbf{\Phi}_k^T\mathbf{\Phi}_k\right)^{-1}\mathbf{\Phi}_k^T\mathbf{y} - \left[\hat{\mathbf{\Theta}}_{k-1}^T, 0\right]^T \\ &\quad = \left[\hat{\vartheta}_{1,k} - \hat{\Theta}_{1,k-1}, \ldots, \hat{\vartheta}_{k-1,k} - \hat{\Theta}_{k-1,k-1}, \hat{\vartheta}_{k,k}\right]^T \quad (20) \end{aligned}$$

where $\mathbf{R}_k\mathbf{y} = \mathbf{y} - \mathbf{\Phi}_k(\mathbf{\Phi}_k^T\mathbf{\Phi}_k)^{-1}\mathbf{\Phi}_k^T\mathbf{y} = \mathbf{y} - (\mathbf{p}_1\hat{\vartheta}_{1,k} + \cdots + \mathbf{p}_k\hat{\vartheta}_{k,k})$ and the following can be obtained for $i = k, \ldots, 1$:

$$\hat{\vartheta}_{i,k} = \frac{\mathbf{p}_i^T\mathbf{R}_{i-1}\mathbf{y} - \sum_{l=i+1}^{k}\mathbf{p}_i^T\mathbf{R}_{i-1}\mathbf{p}_l\hat{\vartheta}_{l,k}}{\mathbf{p}_i^T\mathbf{R}_{i-1}\mathbf{p}_i} = \frac{b_i - \sum_{l=i+1}^{k}a_{i,l}\hat{\vartheta}_{l,k}}{a_{i,i}}. \tag{21}$$

Given that $\hat{\mathbf{\Theta}}_{k-1}$ is already known from the previous process, the corresponding model regressor $\mathbf{p}_r = \arg\min_{i=1}^{k-1}[-\hat{\Theta}_{i,k-1}/\hat{\theta}_{i,k}]_+$ will be removed from the selected regression matrix $\mathbf{\Phi}_k$ in the case that $0 < \gamma_k^\circ \le \gamma_k^\star$. Otherwise, the selection procedure proceeds as described in the previous section, where the regression matrix expands by adding one more regressor.

Now, consider that the removal of some selected regressor $\mathbf{p}_r$ from the regression matrix is required at the kth step, first of all, the entries of the correlation vector $\mathbf{c}^{k-1)} \in \Re^{M-k+1}$ for the remaining regressors are updated as

$$c_i^{k-1)} = \begin{cases} \mathbf{p}_r^T\mathbf{e}_k = (1-\gamma_k^\circ)\rho_k s_r, & i = k \\ \boldsymbol{\varphi}_i^T\mathbf{e}_k = c_i^{k-1)} - \gamma_k^\circ d_i^{k)}, & i = k+1, \ldots, M \end{cases} \tag{22}$$

where s_r denotes the sign of the correlation for $\mathbf{p}_r$. Moreover, the absolute correlation for selected regressors is simply updated as $\rho_{k-1} = (1-\gamma_k^\circ)\rho_k$. The resulted model coefficients from adding the current submodel can be computed as

$$\hat{\Theta}_{i,k} = \hat{\Theta}_{i,k-1} + \gamma_k^\circ\hat{\theta}_{i,k}, \quad i = 1, \ldots, k-1; i \ne r \tag{23}$$

and $\hat{\Theta}_{k,k} = \gamma_k^\circ\hat{\theta}_{k,k}$, where $\hat{\Theta}_{r,k} = 0$ together with the corresponding regressor $\mathbf{p}_r$ is going to be removed from the coefficient vector and the regression matrix. The size of the resulted overall model coefficient vector remains unchanged, i.e., $\hat{\mathbf{\Theta}}_{k-1} \in \Re^{k-1}$.

At the next step, the model error resulted by adding a new submodel to the overall model can be written as

$$\tilde{\mathbf{e}}_{k-1} = \mathbf{e}_k - \gamma_{k-1}\tilde{\mathbf{\Phi}}_{k-1}\left(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1}\right)^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k \tag{24}$$

where $\tilde{\mathbf{\Phi}}_{k-1}$ is the reduced set of selected regressors by removing $\mathbf{p}_r$, i.e., $\tilde{\mathbf{\Phi}}_{k-1} = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_{k-1}]$. On the one hand, it can be seen that the new correlation for the remaining regressors after introducing this submodel becomes

$$\tilde{c}_i^{k-1)} = \begin{cases} \mathbf{p}_r^T\mathbf{e}_k - \gamma_{k-1}\mathbf{p}_r^T\tilde{\mathbf{\Phi}}_{k-1}\left(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1}\right)^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k \\ \qquad\qquad i = k \\ \boldsymbol{\varphi}_i^T\mathbf{e}_k - \gamma_{k-1}\boldsymbol{\varphi}_i^T\tilde{\mathbf{\Phi}}_{k-1}\left(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1}\right)^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k \\ \qquad\qquad i = k+1, \ldots, M. \end{cases} \tag{25}$$

Combining (22), this can be equivalently formulated as

$$\tilde{c}_i^{k-1)} = c_i^{k-1)} - \gamma_{k-1}\tilde{d}_i^{k-1)}, \quad i = k, \ldots, M \tag{26}$$

where

$$\tilde{d}_i^{k-1)} = \begin{cases} \mathbf{p}_r^T\tilde{\mathbf{\Phi}}_{k-1}\left(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1}\right)^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k, & i = k \\ \boldsymbol{\varphi}_i^T\tilde{\mathbf{\Phi}}_{k-1}\left(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1}\right)^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k \\ \qquad\qquad i = k+1, \ldots, M. \end{cases} \tag{27}$$

On the other hand, the associated parameters $\hat{\theta}_{i,k-1}$ $(i = 1, \ldots, k-1)$ for the newly added submodel are given by $[\hat{\theta}_{1,k-1}, \ldots, \hat{\theta}_{k-1,k-1}]^T = (\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1})^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k$. As described at the beginning of this section, in order to proceed with the algorithm and determine the value of γ_{k-1}° and $\gamma_{k-1}^\star$ based on the reduced set of selected regressors, the term $\tilde{\mathbf{d}}^{k-1)} \in \Re^{M-k+1}$ in (27) and the term $\hat{\theta}_{i,k-1}$ $(i = 1, \ldots, k-1)$ need computed. By using (11), the direction vector $\tilde{d}_i^{k-1)}$ $(i = k, \ldots, M)$ can be calculated as

$$\begin{aligned} \tilde{d}_k^{k-1)} &= c_k^{k-1)} - \mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{e}_k \\ &= c_k^{k-1)} - \mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\left(\mathbf{y} - \tilde{\mathbf{\Phi}}_{k-1}\hat{\mathbf{\Theta}}_{k-1}\right) \\ &= c_k^{k-1)} - \mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{y} \end{aligned} \tag{28}$$

and for $i = k+1, \ldots, M$

$$\begin{aligned} \tilde{d}_i^{k-1)} &= (1-\gamma_k^\circ)d_i^{k)} - \frac{\boldsymbol{\varphi}_i^T\tilde{\mathbf{R}}_{k-1}\mathbf{p}_r\mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{e}_k}{\mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{p}_r} \\ &= (1-\gamma_k^\circ)d_i^{k)} - \frac{\boldsymbol{\varphi}_i^T\tilde{\mathbf{R}}_{k-1}\mathbf{p}_r\mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{y}}{\mathbf{p}_r^T\tilde{\mathbf{R}}_{k-1}\mathbf{p}_r} \end{aligned} \tag{29}$$

where $\tilde{\mathbf{R}}_{k-1} = \mathbf{I} - \tilde{\mathbf{\Phi}}_{k-1}(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1})^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T$ denotes the residue matrix resulted from excluding $\mathbf{p}_r$ from the regression matrix $\mathbf{\Phi}_k$. Similar as in (20), the following can also be easily obtained:

$$\begin{aligned} &[\hat{\theta}_{1,k-1}, \ldots, \hat{\theta}_{k-1,k-1}]^T \\ &\quad = (\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1})^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{e}_k \\ &\quad = (\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1})^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{y} - \hat{\mathbf{\Theta}}_{k-1} \\ &\quad = [\hat{\vartheta}_{1,k-1} - \hat{\Theta}_{1,k-1}, \ldots, \hat{\vartheta}_{k-1,k-1} - \hat{\Theta}_{k-1,k-1}]^T \end{aligned} \tag{30}$$

where $\tilde{\mathbf{R}}_{k-1}\mathbf{y} = \mathbf{y} - \tilde{\mathbf{\Phi}}_{k-1}(\tilde{\mathbf{\Phi}}_{k-1}^T\tilde{\mathbf{\Phi}}_{k-1})^{-1}\tilde{\mathbf{\Phi}}_{k-1}^T\mathbf{y} = \mathbf{y} - (\tilde{\mathbf{p}}_1\hat{\vartheta}_{1,k-1} + \cdots + \tilde{\mathbf{p}}_{k-1}\hat{\vartheta}_{k-1,k-1})$ and the following holds for $i = k-1, \ldots, 1$:

$$\hat{\vartheta}_{i,k-1} = \frac{\tilde{\mathbf{p}}_i^T\tilde{\mathbf{R}}_{i-1}\mathbf{y} - \sum_{l=i+1}^{k-1}\tilde{\mathbf{p}}_i^T\tilde{\mathbf{R}}_{i-1}\tilde{\mathbf{p}}_l\hat{\vartheta}_{l,k-1}}{\tilde{\mathbf{p}}_i^T\tilde{\mathbf{R}}_{i-1}\tilde{\mathbf{p}}_i}. \tag{31}$$

Here, it can be found that if $\mathbf{p}_r$ was the last selected regressor in the regression matrix, i.e., $\tilde{\mathbf{\Phi}}_k = [\tilde{\mathbf{\Phi}}_{k-1}, \mathbf{p}_r]$, then the corresponding matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{b}}$ can be used to solve (28)–(31). This can be achieved by restarting the process with the newly selected sequence of regressors, though obviously it would be computationally expensive. Instead, a computationally friendly solution [41] can be readily employed here, by each time swapping two neighboring regressors in $\mathbf{\Phi}_k$ (starting at which $\mathbf{p}_r$ is located) and updating the corresponding matrix and vector, for a number of times until $\mathbf{p}_r$ has been shifted to the last position of the regression matrix. Therefore, after a total of $k-r$ swaps, the selected regression matrix will become $\tilde{\mathbf{\Phi}}_k = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_{k-1}, \tilde{\mathbf{p}}_k]$, in which $\mathbf{p}_r$ is moved to the last position of $\tilde{\mathbf{\Phi}}_k$, i.e., $\mathbf{p}_r = \tilde{\mathbf{p}}_k$. The intermediate regression matrix $\tilde{\mathbf{A}}$ and vector $\tilde{\mathbf{b}}$ are thus obtained. Given this, the direction vector $\tilde{d}_i^{k-1)}$ and coefficients $\hat{\theta}_{i,k-1}$ can now be computed as

$$\tilde{d}_i^{k-1)} = \begin{cases} c_k^{k-1)} - \tilde{b}_k, & i = k \\ (1-\gamma_k^\circ)d_i^{k)} - \tilde{a}_{k,i}\tilde{b}_k/\tilde{a}_{k,k}, & i = k+1, \ldots, M \end{cases} \tag{32}$$

and $\hat{\theta}_{i,k-1} = \hat{\vartheta}_{i,k-1} - \hat{\Theta}_{i,k-1}$ $(i = 1, \ldots, k-1)$, where

$$\hat{\vartheta}_{i,k-1} = \frac{\tilde{b}_i - \sum_{l=i+1}^{k-1} \tilde{a}_{i,l} \hat{\vartheta}_{l,k-1}}{\tilde{a}_{i,i}}. \tag{33}$$

The model learning process is then repeated by determining whether an incremental or decremental movement is required at each step to search for successive LASSO solutions until some stopping criterion is satisfied.

C. Algorithm

The efficient successive LASSO solution generating scheme for CSO model construction is now summarized, the pseudo code being given in Algorithm 1. In consideration of all the field monitoring variables with the designated maximum time lags, a number of candidate regressors $\boldsymbol{\varphi}_i$, say $i = 1, \ldots, M$, can be first obtained. For each CSO identity, a predictive model can then be automatically constructed based on such candidate regressors and the proposed efficient LASSO algorithm. To get a predictive model with a size of one, the correlations between these candidate regressors and the CSO output are used to initialize the two vectors $\mathbf{c}^{0)}$ and $\mathbf{b}^{1)}$; consequently, the regressor leading to the largest absolute correlation is selected and added into the predictive model. Correspondingly, the variables ρ_1, $\boldsymbol{\Phi}_1$, $\mathbf{A}^{1)}$, $\mathbf{b}^{1)}$, $\mathbf{d}^{1)}$, k, $\hat{\theta}_{1,1}$, and $\gamma_1^{\star}$ are computed in sequence (where γ_1° is assigned with zero in order to initiate the model learning process) to prepare the computing framework for locating the next LASSO solution.

As the regularization parameter decreases, the following procedure is then performed efficiently to find the corresponding LASSO solutions. In the case that $0 < \gamma_k^{\circ} \leq \gamma_k^{\star}$ is not met, the next regressor $\mathbf{p}_{k+1} = \arg \gamma_k^{\star}$ to be added into the predictive model is determined together with variables $\hat{\Theta}_{i,k}$ $(i = 1, \ldots, k)$, $c_i^{k)}$ $(i = k+2, \ldots, M)$, ρ_{k+1} and $\boldsymbol{\Phi}_{k+1}$. The model size k will then increase by one and variables $\mathbf{A}^{k)}$, $\mathbf{b}^{k)}$, $\mathbf{d}^{k)}$, and $\hat{\theta}_{i,k}$ $(i = 1, \ldots, k)$ are updated subsequently to compute γ_k° and $\gamma_k^{\star}$ for use in pursuing the next LASSO solution. On the contrary, if $0 < \gamma_k^{\circ} \leq \gamma_k^{\star}$ is met, the term $\mathbf{p}_r = \arg \gamma_k^{\circ}$ is removed from the current regression matrix, the value of $\hat{\Theta}_{i,k}$ $(i = 1, \ldots, k)$ and $c_i^{k-1)}$ $(i = k, \ldots, M)$ being, respectively, updated according to (23) and (22) together with $\rho_{k-1} = (1 - \gamma_k^{\circ})\rho_k$. Then, through consecutively swapping a series of two neighboring selected regressors in $\boldsymbol{\Phi}_k$ (starting with $\mathbf{p}_r$) to update items $\tilde{\boldsymbol{\Phi}}_k = [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_{k-1}, \mathbf{p}_r]$, $\tilde{\mathbf{A}}^{k)}$ and $\tilde{\mathbf{b}}^{k)}$, upon which the value of $\tilde{d}_i^{k-1)}$ $(i = k, \ldots, M)$ is thus obtained by using (32). As a result, the size of the regression matrix is reduced by one $(k = k - 1)$ together with $\boldsymbol{\Phi}_k = \tilde{\boldsymbol{\Phi}}_k$, $\mathbf{A}^{k)} = \tilde{\mathbf{A}}^{k)}$, $\mathbf{b}^{k)} = \tilde{\mathbf{b}}^{k)}$, and $\mathbf{d}^{k)} = \tilde{\mathbf{d}}^{k)}$ updated. The parameter $\hat{\theta}_{i,k}$ $(i = 1, \ldots, k)$, $\gamma_k^{\circ} = \min_{i=1}^{k} [-\hat{\Theta}_{i,k}/\hat{\theta}_{i,k}]_+$, and $\gamma_k^{\star} = \min_{i=k+1}^{M} [(\pm c_i^{k)} - \rho_k)/(\pm d_i^{k)} - \rho_k)]_+$, are thereby calculated and ready for use in searching for the next LASSO solution. The whole algorithm can be terminated by designating a specified number of model regressors first reached during the model learning process or using other criteria such as Akaike information criterion (AIC); thereby the selected model regressors and associated coefficients are retrieved.

Algorithm 1 Pseudo Code for CSO Predictive Model Construction

1: Generate candidate CSO predictive model regressors $\boldsymbol{\varphi}_1, \ldots, \boldsymbol{\varphi}_M$ from field variables associated with time lags.
2: Initialize items $\{\mathbf{c}^{0)}, \mathbf{b}^{1)}\} \leftarrow [\boldsymbol{\varphi}_1^{\mathrm{T}}\mathbf{y}, \ldots, \boldsymbol{\varphi}_M^{\mathrm{T}}\mathbf{y}]$, $\rho_1 \leftarrow \max_{i=1}^{M} |c_i^{0)}|$, $\mathbf{p}_1 \leftarrow \arg \rho_1$, $\boldsymbol{\Phi}_1 \leftarrow \mathbf{p}_1$, $\mathbf{A}^{1)}$, $\mathbf{b}^{1)}$, $\mathbf{d}^{1)}$, and $k \leftarrow 1$ in sequence.
3: Compute $\hat{\theta}_{1,1} \leftarrow b_1/a_{1,1}$.
4: Find $\gamma_1^{\star} \leftarrow \min_{i=2}^{M} [(\pm c_i^{0)} - \rho_1)/(\pm d_i^{1)} - \rho_1)]_+$ and $\gamma_1^{\circ} \leftarrow 0$.
5: **while** $k \leq m$ **do**
6: **if** $0 < \gamma_k^{\circ} \leq \gamma_k^{\star}$ **then**
7: Assign $\mathbf{p}_r \leftarrow \arg \gamma_k^{\circ}$.
8: Update $\hat{\Theta}_{i,k}$ $(i = 1, \ldots, k)$ and $c_i^{k-1)}$ $(i = k, \ldots, M)$.
9: Update $\rho_{k-1} \leftarrow (1 - \gamma_k^{\circ})\rho_k$.
10: Compute $\tilde{\boldsymbol{\Phi}}_k \leftarrow [\tilde{\mathbf{p}}_1, \ldots, \tilde{\mathbf{p}}_{k-1}, \mathbf{p}_r]$, $\tilde{\mathbf{A}}^{k)}$, and $\tilde{\mathbf{b}}^{k)}$.
11: Update $\tilde{d}_i^{k-1)}$ $(i = k, \ldots, M)$.
12: Update $k \leftarrow k - 1$.
13: Assign $\boldsymbol{\Phi}_k \leftarrow \tilde{\boldsymbol{\Phi}}_k$, $\mathbf{A}^{k)} \leftarrow \tilde{\mathbf{A}}^{k)}$, $\mathbf{b}^{k)} \leftarrow \tilde{\mathbf{b}}^{k)}$, and $\mathbf{d}^{k)} \leftarrow \tilde{\mathbf{d}}^{k)}$.
14: Compute $\hat{\theta}_{i,k}$ $(i = 1, \ldots, k)$.
15: Find $\gamma_k^{\circ} \leftarrow \min_{i=1}^{k} [-\hat{\Theta}_{i,k}/\hat{\theta}_{i,k}]_+$.
16: Find $\gamma_k^{\star} \leftarrow \min_{i=k+1}^{M} [(\pm c_i^{k)} - \rho_k)/(\pm d_i^{k)} - \rho_k)]_+$.
17: **else**
18: Assign $\mathbf{p}_{k+1} \leftarrow \arg \gamma_k^{\star}$.
19: Update $\hat{\Theta}_{i,k}$ $(i = 1, \ldots, k)$ and $c_i^{k)}$ $(i = k+2, \ldots, M)$.
20: Assign $\rho_{k+1} \leftarrow (1 - \gamma_k^{\star})\rho_k$ and $\boldsymbol{\Phi}_{k+1} \leftarrow [\boldsymbol{\Phi}_k, \ \mathbf{p}_{k+1}]$.
21: Update $k \leftarrow k + 1$.
22: Update $\mathbf{A}^{k)}$, $\mathbf{b}^{k)}$, and $\mathbf{d}^{k)}$.
23: Compute $\hat{\theta}_{i,k}$ $(i = 1, \ldots, k)$.
24: Find $\gamma_k^{\circ} \leftarrow \min_{i=1}^{k-1} [-\hat{\Theta}_{i,k-1}/\hat{\theta}_{i,k}]_+$.
25: Find $\gamma_k^{\star} \leftarrow \min_{i=k+1}^{M} [(\pm c_i^{k-1)} - \rho_k)/(\pm d_i^{k)} - \rho_k)]_+$.
26: **end if**
27: **end while**
28: Assign $k \leftarrow m$.
29: Output $\boldsymbol{\Phi}_k$ and $\hat{\boldsymbol{\Theta}}_k$.

As for using the AIC criterion, it is expressed as [42]

$$\mathrm{AIC} = N \log(\mathrm{SSE}/N) + 2k \tag{34}$$

where N, k, and SSE refer to the sample number, model size, and sum of squared errors. In correspondence to the incremental and decremental movements of the proposed algorithm, SSE can be recursively computed as:

$$\begin{cases} \mathbf{e}_k^T \mathbf{e}_k \\ \quad = (1 - \gamma_k)^2 \mathbf{e}_{k-1}^T \mathbf{e}_{k-1} + \gamma_k (2 - \gamma_k) \mathbf{e}_{k-1}^T \mathbf{R}_k \mathbf{e}_{k-1} \end{cases} \tag{35}$$

$$\begin{cases} \tilde{\mathbf{e}}_{k-1}^T \tilde{\mathbf{e}}_{k-1} \\ \quad = (1 - \gamma_{k-1})^2 \mathbf{e}_k^T \mathbf{e}_k + \gamma_{k-1} (2 - \gamma_{k-1}) \mathbf{e}_k^T \tilde{\mathbf{R}}_{k-1} \mathbf{e}_k \end{cases} \tag{36}$$

where

$$\begin{cases} \mathbf{e}_{k-1}^T \mathbf{R}_k \mathbf{e}_{k-1} = \mathbf{e}_{k-2}^T \mathbf{R}_{k-1} \mathbf{e}_{k-2} - b_k^2/a_{k,k} \end{cases} \tag{37}$$

$$\begin{cases} \mathbf{e}_k^T \tilde{\mathbf{R}}_{k-1} \mathbf{e}_k = \mathbf{e}_{k-1}^T \mathbf{R}_k \mathbf{e}_{k-1} + \tilde{b}_k^2/\tilde{a}_{k,k}. \end{cases} \tag{38}$$

D. Computational Complexity

Based on the proposed algorithm described in the previous section, the computational complexity comprises a fixed element arising from successive inclusion of model regressors without removal, and a varied element taking into account the removal plus again inclusion of new regressors. Given that N data samples and M candidate model regressors are made available at the beginning of the training process, the fixed amount of computational complexity resulted from m model regressors being first generated is

$$C_{\text{fixed}} = mN(2M - m + 1) + (2N - 1)M + mM(m + 9) - m(2m^2 - 3m + 49)/6. \quad (39)$$

On the other hand, the computational complexity for removing a selected model regressor and adding another varies with the underlying model size being considered (say k) and the position of the regressor being removed from the regression matrix (say δ_k), yielding

$$C_{\text{varied}} = 2(k - \delta_k)(2M - k - \delta_k + 6) + (2N + 2k + 19) \times (M - k + 1) + 2(k + 2)^2 - 28. \quad (40)$$

Given the CSO predictive problem, it usually holds that $\delta_k < k < m \ll M \ll N$. The computational complexity involved in (39) and (40) then mainly relies on terms $2mMN$ and $2MN$, respectively. The actual computational burden will therefore depend on the number of decremental movements during the LASSO solution searching process, and when and where they occur. In addition, if AIC is applied to terminate the training process, a fixed additional amount of $2N+13m+1$ complexity will be incurred, together with an extra of 18 computations added to the varied complexity C_{varied}.

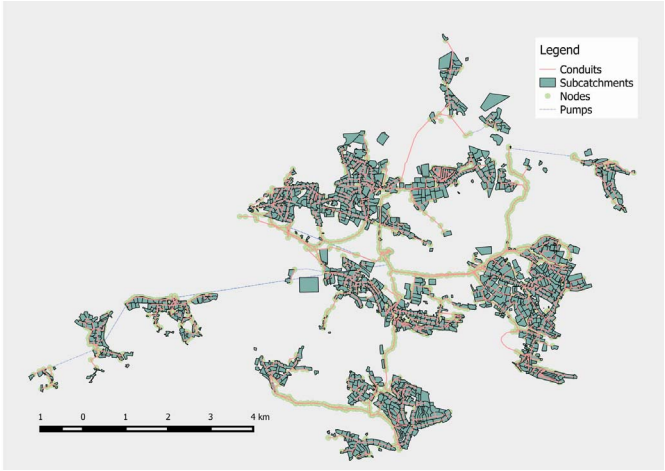
V. Experiments

In this section, the effectiveness of the proposed methodology to automate the CSO predictive model construction and the efficiency of the proposed algorithm are demonstrated. As part of an ongoing research, a pilot area was chosen with the presence of multiple CSOs and various field monitoring variables already in existence. The experiments were all performed on a Intel Core2 Duo Processor P8600 2.40 GHz, with programs executed by MATLAB. A range of model performance and results are provided.

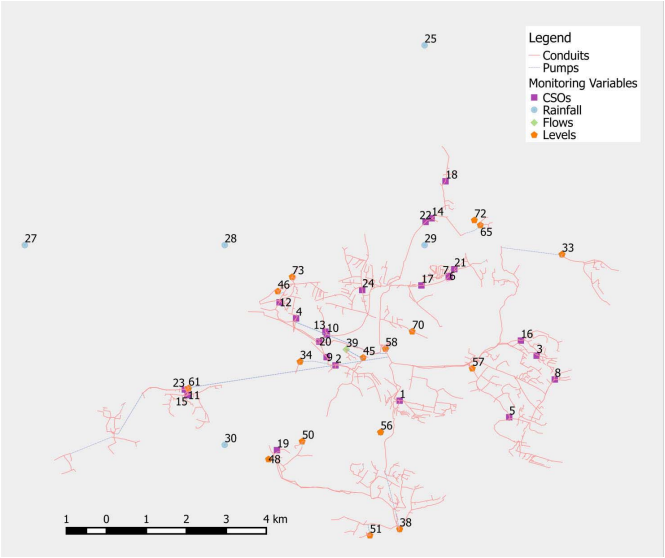
1) The obtained model structure, and training/test time, error and accuracy for CSO predictive model construction terminated by both designated number of model regressors and AIC criterion [see (34)–(38)].
2) The consideration of modeling results for imbalanced dataset.
3) The comparisons of the proposed algorithm with neural network and fuzzy approaches.
4) The integration of multistep ahead predictions.

A. Pilot Area Description and Data Gathering

A U.K. pilot area depicted in Fig. 2(a) is employed to represent various features related to wastewater collection networks and the corresponding catchments. The associated sensing variables are shown in Fig. 2(b), with the entire area serving around 52 000 residents. The network assets within this pilot area are solely owned by Welsh Water (also known as DCWW or Dŵr Cymru Cyfyngedig). The pilot area has been chosen due to the fact that it contains a typical CSS with certain complexities of catchment and topography, and the network operator already has the network closely monitored which would provide the opportunity of applying a data-driven approach.



(a)



(b)

Fig. 2. Schematic of the pilot area. (a) Field map. (b) Monitoring variables.

After an in-depth analysis of the quality of the data collected by the existing sensing infrastructure, a total of 73 monitoring variables were considered, including 24 CSO level variables (percent), six rainfall variables (mm/h), nine flow variables (l/s, e.g., pumped flows, treatment flows, and storm flows), and 34 other level variables (m or percent, e.g., wet well levels, sump levels, and screen levels). A CSO level with a percent value larger than 100% indicates the occurrence of CSO spillages. These monitoring variables can be seen from Fig. 2(b), where one of the weather stations for monitoring rainfall information is located to the west just outside the trial area. The field variables are monitored via Welsh Water's proprietary systems.

The monitoring data from various field variables was collected from April 1, 2014 to March 1, 2016 at a time resolution of 15 min (thus also being considered as the prediction time-step). In principle, the total amount of samples for each variable would be 67 205. In the case where missing values exist, they were simply estimated using linear interpolation methods. However, given that the proposed predictive methodology is independent of the data preprocessing, it would not hinder the use of other (complex) data imputation techniques though there is no consensus on the best approach to do this. As a matter of fact, having reliable monitoring system is required in order to avoid the long periods of missing data as it would be very difficult to accurately infer the missing values using imputation techniques. The negative and hugely positive values together with other abnormal values resulting from sensor reading errors were also regarded as missing values and processed in the same manner. It was found that an average of 2.28% missing data existed over the various field variables, where for some individual variables this can reach as high as 20%. Specifically, for the variable with a high level of missing data, if the missing data comprises a small number of long periods, it makes little difference whether considering such a variable at the beginning of model construction as the methodology can hardly relate this variable's behavior (resulting from the interpolated low quality of data) to the CSO behavior of interest. Therefore, this variable would not be selected in the resulting CSO predictive model. Whilst in the case that the missing data is continuously accumulated (say one missed in every five measurements), the behavior exhibited by this variable can still be somewhat recovered by interpolation and potentially considered for being related to the CSO behavior. Finally, a data partition of 60% (from April 1, 2014 to May 26, 2015) of the entire collection period was used to train the CSO predictive models, while the remaining 40% (from May 26, 2015 to March 1, 2016) was employed for model testing.

B. Results and Analysis

The maximum time lag of the field variables was assigned with a value of 10 (ten time-steps, i.e., 150 min of prior data) followed by a preliminary site trial (determined by trial-and-error as usual in time-series prediction), amounting to a total of 730 potentially available model regressors at the beginning of the model construction. It is worth mentioning that, as the maximum lag increases, more time is needed to construct the CSO model as the number of initial model regressors gets larger for the model learning. In this case, it was found that a value larger than 10 would not help improve the model performance significantly. The original algorithm as well as its realization [43] (followed by the modification of LAR) for deriving LASSO solutions as described in Section III, was also examined to verify the computational advantage of our algorithm. To evaluate the efficiency of the proposed algorithm as well as the effectiveness of the overall methodology of CSO model construction, Table I lists the average modeling results over all the CSOs produced by both original and proposed algorithms with varying number of selected model regressors. The model structure is expressed in terms of the number of selected weather stations, CSOs and field variables and the model size (similarly for other tables in their respective settings as presented in this paper). It is apparent that the proposed algorithm possesses significant computational advantage over the original algorithm. The elapsed time for constructing CSO predictive models using the proposed algorithm compared with the original one is expected to decrease by around two times, especially so when more model regressors are included. It should be noted that in case of large-scale sewer networks and associated datasets, the time needed for each CSO model construction can increase significantly due to the increased number of field variables (also see Section IV-D for computational complexity analysis), so does the average model construction time using AIC criterion to be given in Table II. As model size (the number of model regressors, i.e., the number of field variables associated with time lags) increases, the training root mean squared error (RMSE) is consistently decreased as expected, by optimizing the updated LASSO objective function where the weighting of the L_1 norm of model parameters decreases and correspondingly the importance of L_2 norm of model errors increases. On the other hand, the test RMSE generally decreases first, then stabilizes and increases again (where overfitting appears). Reflecting on this, the test $R2$ generally increases first, then stabilizing and decreasing again as the number of model regressors increases.

TABLE I
AVERAGE MODEL CONSTRUCTION RESULTS OVER ALL THE CSOS PRODUCED BY THE ORIGINAL AND PROPOSED ALGORITHMS WITH VARYING NUMBER OF MODEL REGRESSORS

Model Size	Training Time (s)		Training RMSE	Test RMSE	Training R2	Test R2	# Weather Stations	# Related CSOs	# Total Vars.
	Orig.	Prop.							
5	3.76	2.16	5.9056	7.9812	0.7169	0.7517	0.3750	2.2917	2.6667
10	6.62	3.55	5.1770	6.5696	0.7726	0.8216	1.0833	3.7917	5.0417
20	12.71	6.39	4.9104	6.1661	0.7908	0.8410	2.3333	7.0833	10.7500
30	19.15	9.26	4.8230	6.0532	0.7956	0.8449	2.5833	10.1667	15.5000
40	25.54	12.08	4.7838	6.0314	0.7981	0.8459	3.0833	11.8750	19.0000
50	31.97	14.88	4.7583	6.0274	0.7998	0.8459	3.6667	13.1667	23.0833
60	38.59	17.66	4.7409	6.0217	0.8011	0.8461	4.0417	14.3750	26.9583
70	44.96	20.41	4.7263	6.0226	0.8021	0.8463	4.3750	15.7083	30.7083
80	51.40	23.06	4.7149	6.0230	0.8029	0.8464	4.5833	16.6667	34.5833
90	57.77	25.62	4.7037	6.0211	0.8037	0.8465	4.7083	17.5833	37.7083
100	64.20	28.15	4.6925	6.0201	0.8044	0.8466	4.9583	18.2500	41.0000
120	77.76	33.15	4.6784	6.0237	0.8054	0.8466	5.1667	20.0417	47.6250
140	91.15	38.00	4.6670	6.0326	0.8036	0.8464	5.3750	21.1250	52.5000
160	104.97	42.76	4.6571	6.0419	0.8070	0.8461	5.5833	21.6667	56.6250
180	117.79	47.19	4.6482	6.0536	0.8076	0.8457	5.7500	22.4167	60.3333
200	132.03	51.63	4.6397	6.0667	0.8082	0.8453	5.9583	23.0417	63.1250

To examine the importance of weather rain-gauge stations and the existence of interrelationships amongst different CSOs, the average number of weather stations and CSOs contained over all constructed CSO models for each subset of model regressors are listed in the eighth and ninth columns of Table I, while the last column gives the average amount of total selected field variables also including other measures such as pumped flows and wet well levels. Overall, the selected field variables play an important role in constructing the CSO predictive models. It can be found that the CSO plus rainfall variables account for the major contributing factors when a small number of model regressors are required to construct the predictive models (where the summation of numbers of selected weather stations and CSOs, over the total number of selected variables is very high). Notably, in addition to the involvement of weather stations, as many CSO variables are also picked up for model construction, a clear interrelationship between distinct CSOs is thus evident. With the model

TABLE II
MODEL CONSTRUCTION RESULTS BY THE ORIGINAL AND PROPOSED ALGORITHMS USING AIC FOR ALL THE CSOS

CSO	Training Time (s)		Training	Test	Training	Test	# Weather	# Related	# Total	Model
ID	Orig.	Prop.	RMSE	RMSE	R2	R2	Stations	CSOs	Vars.	Size
1	35.52	16.39	4.6770	5.8749	0.6691	0.7683	5.00	14.00	27.00	55.00
2	31.83	15.37	3.0088	4.0244	0.8341	0.8219	5.00	12.00	20.00	53.00
3	20.90	10.41	3.1577	3.7432	0.7097	0.7939	3.00	13.00	19.00	34.00
4	41.09	18.46	6.1707	7.2116	0.8470	0.8976	5.00	16.00	29.00	61.00
5	23.08	10.37	10.1949	11.9765	0.8345	0.8866	3.00	14.00	19.00	30.00
6	49.62	22.29	2.5486	3.0383	0.9127	0.9365	3.00	19.00	30.00	77.00
7	19.19	9.57	2.9167	3.0529	0.7748	0.8696	2.00	12.00	16.00	31.00
8	11.13	5.71	2.9831	2.7691	0.8610	0.8777	2.00	4.00	6.00	17.00
9	12.98	6.56	6.3748	6.4174	0.9291	0.9392	2.00	7.00	12.00	20.00
10	28.38	13.03	2.2408	3.4044	0.9912	0.9822	3.00	10.00	13.00	41.00
11	15.67	7.94	1.5349	3.4617	0.8178	0.7424	4.00	6.00	10.00	25.00
12	15.68	7.36	3.8753	5.3628	0.8925	0.9178	4.00	9.00	14.00	21.00
13	17.99	9.03	3.9765	6.2173	0.9382	0.9785	2.00	7.00	12.00	29.00
14	32.49	14.85	13.9395	18.2809	0.7631	0.7625	2.00	13.00	20.00	48.00
15	23.76	11.41	4.3517	3.2469	0.9531	0.9854	2.00	12.00	18.00	37.00
16	14.09	6.86	3.4326	4.0446	0.6513	0.7327	5.00	8.00	13.00	20.00
17	11.13	5.47	2.0279	2.8000	0.9581	0.9706	1.00	9.00	12.00	15.00
18	14.57	7.41	3.7099	4.4563	0.9303	0.7537	2.00	4.00	6.00	23.00
19	10.01	5.20	7.6395	10.0413	0.5896	0.6312	1.00	3.00	5.00	15.00
20	26.69	12.74	2.9376	5.2857	0.7524	0.8656	3.00	10.00	18.00	42.00
21	22.04	10.38	2.3775	4.3419	0.4302	0.7657	2.00	16.00	19.00	32.00
22	32.14	14.59	6.1534	7.5369	0.8741	0.9147	3.00	14.00	23.00	47.00
23	20.28	9.57	12.0035	14.7200	0.4318	0.6980	0.00	5.00	11.00	29.00
24	19.18	9.41	2.6571	3.9023	0.7940	0.7958	2.00	11.00	16.00	29.00
Avg.	22.89	10.85	4.7871	6.0505	0.7975	0.8453	2.75	10.33	16.17	34.63

size increasing, other pertinent field variables would then have more chance of getting selected to further improve model performance. After a certain number of model regressors being included (can be otherwise determined using the AIC criterion), say around 30, the test performance becomes steady, whereby the expansion of model size does not improve much (or maybe reduce) the model performance. In this situation, adding more field variables into predictive models would not help improve prediction accuracy. In addition, from the model complexity point of view, a simpler model in small size is usually preferred.

Alternatively, through using the AIC criterion, the model construction results for all the CSOs produced by the original and proposed algorithms are given in Table II. It is again shown that the proposed algorithm reduces dramatically the computational time compared to the original for all the CSO models; in this case, roughly 50% reductions are achieved. The difference between the training and test RMSEs is acceptable and indicates well-trained models with good generalization ability. To facilitate direct comparisons across different CSO models, the training and test $R2$ values (the percentage/proportion of the CSO behavior/variation that is predicted/explained by the model) are given for every model, where those exhibiting a larger value represent a better constructed predictive model (a maximum value of 1.0 indicates that the underlying CSO dynamic behavior is completely explained and predicted by the model). Due to the distinct data quality of each CSO and field constraints (e.g., some CSOs might lack monitoring of close neighbor or correlated field variables), it shows that different levels of model goodness-of-fit are presented amongst these CSO models. Specifically, it can be seen from Table II, there is a clear relationship between the CSO and rainfall, whilst particularly, no correspondingly existing rainfall stations were found important for explaining the behavior of CSO #23, which potentially led to the less accurate predictions. In this regard, improved spatial resolution of rainfall data can be approached to enhance the model accuracy. Overall, the upper middle part of the pilot area received comparatively accurate predictions, as more field monitoring variables are distributed therein. Whilst some CSO models obtained relatively low level of accuracy, others can achieve extremely high accuracy with a test $R2$ value larger than 0.90. This demonstrates the effectiveness of the proposed data-driven methodology for tackling the CSO prediction problem, provided that high quality and resolution field data is made available.

While the averaged test RMSE (6.0505) is a bit larger than the training one (4.7871), notably, the averaged test $R2$ (0.8453) over all the CSO models is better than the training one (0.7975), demonstrating the good generalization ability of the constructed models. Though it maybe often seen that the model generalization performance is worse than the training one, this is not always the case as it is highly dependent on the performance measure, the type of model and training algorithm (e.g., using regularization, subset selection and local learning techniques) as well as the differing data quality between the training and test dataset. Looking through the eighth to tenth columns, each CSO tends to exhibit a relationship with a relatively higher proportion of the weather stations and the CSOs than with the other 43 field variables.

It may be interesting to see the selected field variables with time lags for the constructed models. Due to the large number of CSOs involved, here, CSO model #9 is simply taken as an example to illustrate the resultant model structure. It is found that the following variables are presented in the predictive model: $y_2(t-1)$, $y_3(t-1)$, $y_4(t-1)$, $y_9(t-1)$, $y_9(t-4)$, $y_9(t-5)$, $y_9(t-7)$, $y_9(t-10)$, $y_{11}(t-1)$, $y_{15}(t-1)$, $y_{18}(t-1)$, $w_{29}(t-1)$, $w_{30}(t-1)$, $w_{30}(t-2)$, $l_{48}(t-5)$, $l_{48}(t-9)$, $l_{51}(t-1)$, $l_{54}(t-1)$, $l_{54}(t-9)$, $l_{54}(t-10)$, where y, w, and l denote the corresponding CSO, rainfall, and level variables, respectively. In order to predict future statuses for CSO #9, it can be seen that the model captures the previous statuses of seven CSO variables (including itself #9), two rainfall variables (#29 and #30), and three level variables (#48, #51, and #54). The variable #54 is monitored at the same location as #34. These variables are considered as the most important determined by the LASSO criterion and the time-series pattern exhibited in the monitoring data. Under the current catchment/network configuration and given the monitored data, the surrounding critical CSOs, weather stations and level variables together with their time lags are thus identified, leading to a total of 12 field variables included in this model and a model size of 20 including the various lags.

To visualize the model performance, Fig. 3 depicts one of the best obtained CSO models (model #10) with a training and test $R2$ of 0.9912 and 0.9822, respectively. It can be found that the majority of CSO levels were distributed in the range between 20 and 60, resulting in the dense plot of data in the bottom part of Fig. 3(a). Compared with the measured values, both trained and predicted CSO levels are well modeled for the whole period starting from April 1, 2014 to March 1, 2016. It is worth noting that the missing data for this CSO variable is mainly between October 10, 2014 and November 12, 2014 (accounting for less than 5% of the entire data collection period), and linear interpolation was applied as indicated in Section V-A. Considering the large number of monitoring variables and that different variables can have

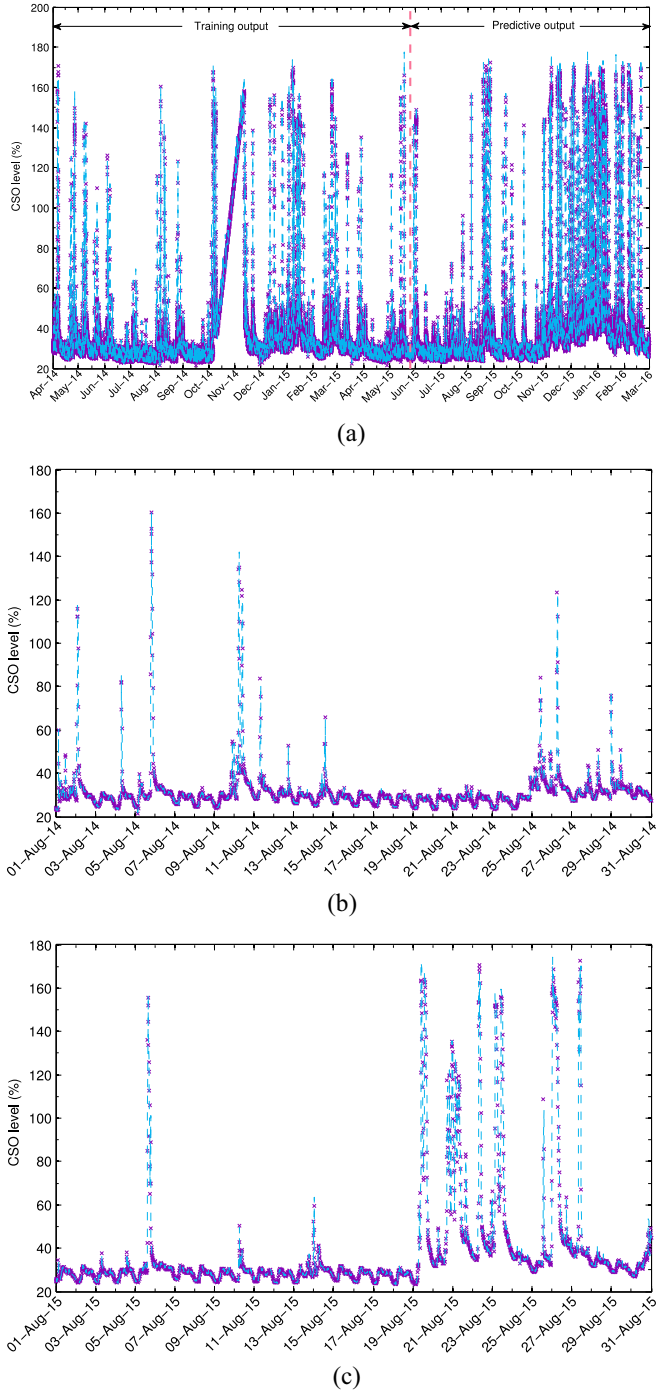


Fig. 3. One of the best constructed CSO predictive models (#10). (a) Measured and modeled CSO levels for the whole period between April 1, 2014 and March 1, 2016. Zoomed-in view of measured and (b) trained CSO levels in August 2014 and (c) predicted CSO levels in August 2015. (The red sign "x" denotes the sensor reading and the blue dashed line depicts the model output.)

different periods of missing data, the overall usable amount of data can be dramatically reduced if such durations of missing data are removed directly (every removal of a sensing period due to one field variable can cause a correspondingly overall size reduction of useful dataset given its a time-series problem). On the other hand, the model will also need to have certain level of anti-noise ability regarding data quality (as well illustrated here) where the missing values could be due to accumulation from many small periods or one/several long periods. Notwithstanding, given the focus of this paper (i.e., the proposed methodology) is independent of the handling of missing values, other imputation techniques can also be used. Here, the historical data from three weather stations and ten CSO variables were found to be relevant to construct the underlying model, while a total of 41 model regressors were selected also including various degrees of time lags from these variables. In order to see a more detailed comparison between the model output and the system output, the modeled and measured CSO levels in August 2014 (training dataset) and August 2015 (test dataset), are illustrated in Fig. 3(b) and (c), respectively. Out of these, the underlying complexity of the CSO behavior including large and small instances has been clearly learned and predicted by the developed model. In this case, though the instances monitored are imbalanced in terms of huge amount of small levels of CSOs and considerably less quantity of large measurements, the learning algorithm was still able to cope with it appropriately.

It should be mentioned that the mechanism for the CSO time-series prediction is to predict future CSO statuses based on the input of a number of previously observed statuses of field monitoring variables. During the model training period, the modeled CSO levels try to fit all those contained in the training dataset including both monitored and interpolated (where missing values occur) data. However, due to the input of less accurate (or even incorrect) interpolated values for the previous network statuses, the model can thereby infer wrong predictions at that specific moment (the period in which missing values occur). This would not be a problem for model training as it is just a way of maximizing the overall length of training dataset. In short, though the model is constructed to explain the CSO behavior with the selected field variables and associated time lags, however, the actual prediction is also dependent on the quality of the monitoring data that feeds into the model. Once the model is constructed, in the worst case scenario, the model will not be able to produce any predictions if there are persistent missing values from the field monitoring data. In the case of missing data (which can be detected directly from the sensing system, not from the model), both predicted and monitored values do not exist for comparison. The comparison between the predicted and monitored CSO statuses can therefore be made under the normal running of sensing system to indicate if there is a malfunction with the sewer network.

On a different note, it was found that for some CSOs, e.g., #24, the imbalanced data issue can be extremely serious, i.e., the number of monitored large instances (e.g., larger than 90%) of CSO level readings is dramatically less than the number of monitored small instances (e.g., less than 90%) of CSO level readings. As shown in Fig. 4(a), though the model was trained well to predict the majority of small levels of CSOs, the large levels were not fully modeled. Given the imbalanced dataset, the training procedure tended to learn dynamics more exhibited by the small instances. This issue can be simply addressed using the common upsampling technique (i.e., replicating instances from the minority) [44] to increase the amount of instances from those that are under-represented. In detail,

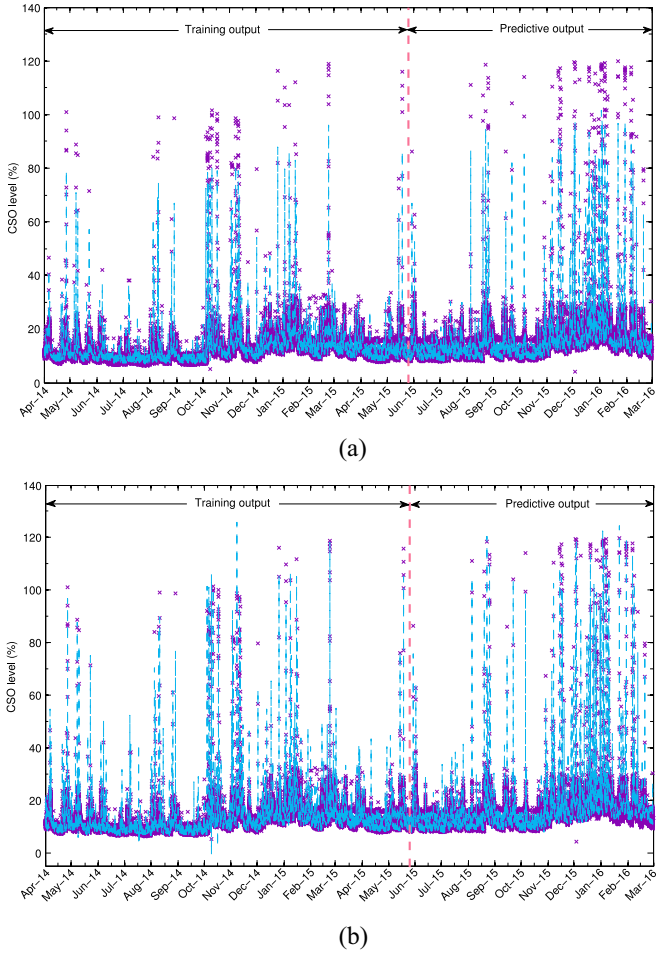


Fig. 4. CSO data is seriously imbalanced in terms of relatively few large CSO levels measured. (a) Predictive models constructed without upsampling (large CSO levels not well fitted). (b) Predictive models constructed with upsampling (large CSO levels well fitted). (The red sign x denotes the sensor reading and the blue dashed line depicts the model output.)

for each CSO model construction, as long as the total number of large instances for the CSO of interest is less than a particular proportion (say a threshold of 2%) amongst the training dataset, these large instances together with the corresponding instances of input variables will be replicated up to this proportion for model training. By using the upsampling techniques, the average model construction results under varying number of model regressors and the updated model construction results for these CSOs seriously suffered from this imbalance issue are, respectively, shown in Tables III and IV. This again confirms the computational superiority of the proposed algorithm in comparison with the original one by looking at the second and third columns of both tables. The overall training RMSE/$R2$ is slightly reduced as the training process was forced to fit more onto the rarely occurring large CSO values at the expense of partially sacrificing fitting the absolute majority of low CSO values. Moreover, as expected, the involvement of rainfall information in the constructed models is seen generally enhanced, as these large CSO values are intrinsically more driven by heavy rainfall. Specifically, it is worth noting that the training and test RMSEs for CSO #21 as well as the small difference between them indicate the model is

TABLE III
AVERAGE MODEL CONSTRUCTION RESULTS OVER ALL THE CSOS PRODUCED BY THE ORIGINAL AND PROPOSED ALGORITHMS WITH VARYING NUMBER OF MODEL REGRESSORS (UPSAMPLING CASE)

Model Size	Training Time (s) Orig.	Time (s) Prop.	Training RMSE	Test RMSE	Training R2	Test R2	# Weather Stations	# Related CSOs	# Total Vars.
5	3.88	2.26	5.9416	7.6045	0.7039	0.7771	0.5417	2.6250	3.2500
10	6.70	3.66	5.3294	6.5646	0.7379	0.8146	1.4583	4.3333	6.2083
20	12.92	6.56	5.1312	6.2960	0.7458	0.8208	2.5417	6.8333	11.4583
30	19.40	9.46	5.0470	6.2073	0.7508	0.8230	2.9167	9.7917	16.5000
40	25.70	12.27	5.0134	6.2074	0.7527	0.8221	3.6250	11.1667	20.5833
50	32.02	15.02	4.9925	6.2184	0.7533	0.8208	4.2500	12.5417	24.4167
60	38.50	17.74	4.9821	6.2337	0.7524	0.8190	4.4583	13.8333	28.1667
70	44.62	20.47	4.9736	6.2534	0.7518	0.8172	4.7083	15.1667	32.2917
80	51.76	23.15	4.9649	6.2646	0.7516	0.8162	4.9167	16.2917	36.1250
90	58.10	25.74	4.9581	6.2755	0.7508	0.8149	5.0000	17.1667	39.1667
100	64.35	28.35	4.9511	6.2872	0.7499	0.8136	5.1667	17.7500	41.9583
120	77.94	33.28	4.9457	6.3102	0.7477	0.8113	5.2500	19.2083	48.1667
140	91.15	39.88	4.9427	6.3346	0.7455	0.8093	5.4167	20.2917	52.5417
160	104.50	42.87	4.9404	6.3604	0.7436	0.8072	5.6250	21.3333	56.7083
180	118.49	47.54	4.9373	6.3853	0.7419	0.8055	5.7500	21.9583	59.6667
200	131.76	53.10	4.9344	6.4102	0.7401	0.8038	5.8333	22.6250	62.5417

TABLE IV
MODEL CONSTRUCTION RESULTS BY THE ORIGINAL AND PROPOSED ALGORITHMS USING AIC FOR THOSE SERIOUSLY IMBALANCED CSOS (UPSAMPLING CASE)

CSO ID	Training Time (s) Orig.	Time (s) Prop.	Training RMSE	Test RMSE	Training R2	Test R2	# Weather Stations	# Related CSOs	# Total Vars.	Model Size
1	21.91	11.14	5.1786	6.0560	0.5943	0.7538	4.00	12.00	21.00	35.00
2	56.81	27.75	3.7491	4.7180	0.7425	0.7552	6.00	18.00	40.00	90.00
3	73.91	36.50	3.9404	5.3203	0.5480	0.5837	6.00	19.00	48.00	116.00
6	69.29	29.62	2.5928	3.6263	0.9096	0.9095	6.00	17.00	43.00	102.00
8	49.19	22.82	4.1144	4.7125	0.7356	0.6458	4.00	17.00	42.00	81.00
11	40.26	18.54	1.8847	3.6939	0.7252	0.7066	6.00	16.00	32.00	63.00
13	29.66	13.90	3.9857	6.1726	0.9379	0.9788	4.00	11.00	17.00	45.00
16	40.13	18.52	3.9328	5.3432	0.5422	0.5335	5.00	12.00	25.00	62.00
17	29.36	13.99	1.9949	2.7835	0.9595	0.9709	3.00	14.00	29.00	46.00
19	20.05	9.98	7.7874	10.2493	0.5736	0.6157	3.00	6.00	20.00	32.00
20	46.34	20.93	3.0590	5.1154	0.7315	0.8741	6.00	11.00	33.00	70.00
21	20.09	9.96	2.8639	2.4559	0.1731	0.9250	1.00	7.00	20.00	32.00
24	23.57	11.09	3.0800	4.1074	0.7232	0.7738	6.00	8.00	22.00	34.00

acceptable. The large difference between the training and test $R2$ values is because the CSO levels in the training dataset are very closely distributed around its mean value (giving the low training $R2$), whereas it is not the case in the test dataset. Finally, Fig. 4(b) illustrates a better predictive model in the sense of well-fitted large CSO levels compared with Fig. 4(a).

To continue examining the performance of the constructed models, neural networks and fuzzy systems were also employed to learn the CSO behavior based on the selected field variables and associated time lags produced by our methodology, envisaging the potential of improving model predictive accuracy. Here, the well-known feed-forward backpropagation network (optimized by Levenberg–Marquardt method) and Sugeno-type fuzzy system (optimized by the hybrid of least-squares and gradient descent methods) were used employing MATLAB neural network (*feedforwardnet* and *train*) and fuzzy logic (*genfis3* and *anfis*) toolboxes, respectively. During the model training process, 20% of the training data was used for validation purposes in order to mitigate overfitting. The training and test time/RMSE/$R2$ are shown in Fig. 5 based on the nonupsampled data. It once again shows that our approach required significantly less training and test times as in Fig. 5(a) and (b). Though the training RMSE and $R2$ [Fig. 5(c) and (e)] of the proposed models were slightly increased and decreased, compared to that of the neural and fuzzy models, more importantly, our model's generalization performance indicated by test RMSE and $R2$ [Fig. 5(d) and (f)]

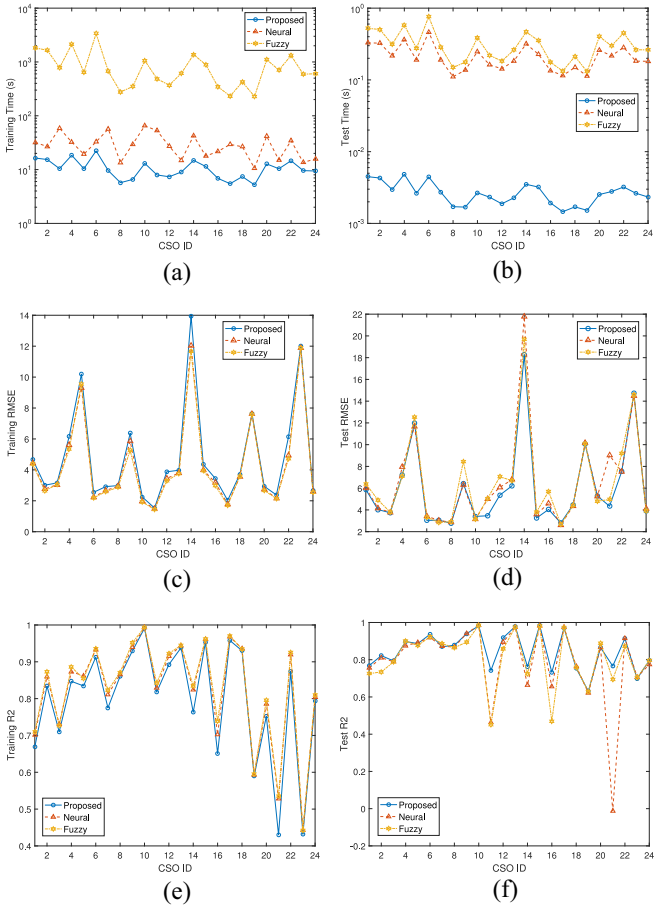


Fig. 5. Comparisons between the proposed, neural, and fuzzy models for all the CSOs. (a) Training time. (b) Test time. (c) Training RMSE. (d) Test RMSE. (e) Training $R2$. (f) Test $R2$.

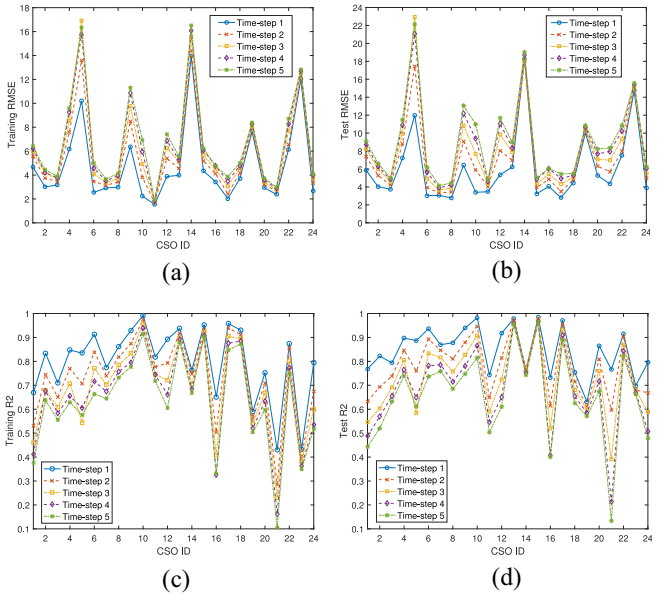


Fig. 6. Multistep ahead predictions for all the CSOs. (a) Training RMSE. (b) Test RMSE. (c) Training $R2$. (d) Test $R2$.

was even better as fundamentally desired. Some CSO models (e.g., #11 and #21) produced by neural networks and fuzzy systems give very poor test $R2$ values. Therefore, the outstanding performance of our algorithm, in terms of both model accuracy and computational time, has been demonstrated owing to the regularization and fast training.

Furthermore, given the proposed methodology, it is also straightforward to develop multistep CSO predictive models where needed, by using the CSO status at the required prediction step as the model output. The training and test RMSE/$R2$ across the 24 CSOs for five prediction steps are illustrated in Fig. 6. As expected, with the increase of prediction steps, the prediction accuracy generally reduces due to less recent information about the system being available (uncertainty increases) and considered by the model. However, as for the CSOs with high accuracy at single-step prediction, they still possess very good performance at multistep prediction where large accuracy reduction was not seen.

Based on the aforementioned facts including the obtained performance in a variety of settings and comparisons with other approaches, in conclusion, the proposed methodology is confirmed capable of quickly and effectively automating the entire CSO predictive model construction process. It is worth mentioning that other field variables such as flows and levels (other than that in CSOs) can be predicted in the same way as CSO statuses using the proposed approach; this, however, is out of the scope of this paper.

C. Discussion

The main aim of this paper is to propose an automated predictive model construction methodology to address the future CSO status prediction problem. The requirements for developing CSO predictive models were elicited as part of an EU FP7 water project (WISDOM) involving a multi-disciplinary consortium from the water value chain across Europe, and mainly attributed to our water utility partner (DCWW). It is an important research topic that has attracted interest of a number of previous studies [23]–[25]. In general, the predictive model can provide data for the construction of an online decision support tool that can be used to consistently.

1) Predict future CSO statuses (especially those indicating a spillage event) in advance (using either single or multiple step predictions), thus enabling network operators to take corrective actions (e.g., getting the excess CSO spillages properly treated/processed) as early as possible in order to mitigate the potential adverse effects, or alerting customers/authorities;
2) As a secondary benefit, inform network operators about abnormal CSO performance by comparing the predicted with monitored statuses and detecting if there is a significant discrepancy between them, thus allowing timely CSO performance restoration from asset malfunctions such as that originated from failed pumping and sewer blockages.

In the latter case, if at some point, it is found that there is a significant discrepancy starting to appear between the predicted and actual monitored CSO statuses, this can potentially indicate part of malfunctions (e.g., blockages) that have occurred within the sewer network if such malfunctions can

lead to the CSO behavior change. In this case, the predicted CSO statuses can (dramatically) either rise above or drop below the monitored statuses depending on the actual malfunction, as the predictive model no longer represents the CSO behavior given the malfunction. The CSO predictive model for normal conditions in the sewer network therefore gives predicted CSO statuses different from the monitored ones resulting from the network with malfunctions. It should be noted that rather than prediction of malfunctions, here it considers to inform network operators timely whenever such a malfunction has occurred. Therefore, our model can help improve the management of CSOs and network assets, for example, in the development of an online warning system underpinned by some rules that can be triggered based on the real-time predicted and monitored values of CSOs, to alert water utilities enabling them to react with remedial actions proactively or timely, thus reducing the volume or quantity of unexpected CSO spillages. More specifically, both single-step and multistep ahead predictions can be generated in real-time but with different level of accuracies (the prediction accuracy improves with the decrease in number of forward prediction steps), as more recent field information (leading to less uncertainty) is collected and processed by the model for fewer step ahead predictions. Thus, the multistep ahead prediction can be used in the control room for the preliminary/coarse decision making (e.g., to put field teams and resources on standby), while more recent predictions (especially next step predictions) can be adopted for more precise decision making (e.g., to examine specific CSO structures and determining solutions to fix issues). Moreover, the proposed data-driven approach for CSO time-series predictions (e.g., statuses of the next 15–60 min) also helps reduce the cost and time associated with model development and calibration in comparison with the hydrological-hydraulic modeling approaches, while meeting regulatory obligations imposed by environment agencies and/or local authorities.

As the aim of the research is to develop a predictive model for real-time prediction of the future state of CSOs, based on the current composition and operation of the sewer network and catchment, therefore, similar to using the hydrological-hydraulic model, the data-driven model usually does not change once constructed. However, the developed model does need the continuous provision (update) of new field monitoring data to produce consecutive predictions as time moves forward. If there is a significant change in the field network or the catchment, then the data-driven model can be reconstructed automatically using the new sensing data after the change, a simpler process compared with utilizing and updating the hydrological-hydraulic model (where a tedious manual process is involved to modify, test and calibrate the model). In order to acquire an accurate data-driven model, the field data collected for model learning should be representative and of sufficient quantity to reflect the full complexity of the CSO behavior. Roughly, a year's worth of data was utilized in this paper to cover any seasonal effects on the CSO behavior. The requirement of the relatively long time-series data somewhat constitutes a drawback of the data-driven approach if the model needs to be reconstructed. In practice, to improve model accuracy while also providing predictions after a change to the network, the model can be reconstructed say every month before the whole year data is made available. In addition, though independent of the proposed methodology, techniques of data validation and reconciliation can be employed to improve the quality (i.e., accuracy and reliability) of field measurements. Notably, the model is not able to respond to the actual intervention, but to alert when a remedial intervention is required; however, whenever such an intervention occasionally alters the sewer network an update of the model is needed as indicated above.

Moreover, the time saving achieved for model construction can have both practical and methodological meanings. The practical implications lie in the algorithmic ability to not only accurately but also quickly (re)construct the CSO predictive models for their subsequent use and integration in the water utility's network management process. This is especially important when dealing with large-scale networks and data where substantial model construction time can be experienced. On the other hand, the methodological meanings from a wider research community perspective provide contributions to an important research topic: development of low-complexity machine learning algorithms for fast model construction. The efficient model selection algorithm developed in this paper can be employed or integrated in various model training and structure determination tasks including that for polynomial models, artificial neural networks, support vector machines and fuzzy systems, where high computational burden and model complexity are usually a concern as experienced in model learning, understanding and reasoning processes.

Finally, the proposed data-driven approach does not need to know the network topology or detailed information about the drainage network, thus to reduce the effort and time spent in collecting and analyzing the corresponding information related to a particular network and also to improve the general applicability of the algorithm across different networks. However, given that the aim of the data-driven model is to predict CSO statuses in a number of future steps, it cannot be used to analyze the actual propagation of a malfunction within the network. Our methodology is systematically based on the global optimization of LASSO criterion further enhanced by computational advantage, without testing all the possible combinations (i.e., exhaustive approach) of model regressors resulting from the monitoring variables and their associated lags. It is widely recognized that an exhaustive approach guarantees the optimality of the solution, but it can take years or even be practically impossible to complete. Other approaches such as relying on expert knowledge (perhaps together with simplified mathematical processing) with the aid of network topology information, and performing forward/backward stepwise selection can be employed to reduce the model development time but at the expense of reduced solution quality. In this regard, our approach enjoys both global optimality in the LASSO sense as well as computational efficiency.

VI. Conclusion

This paper has proposed a systematic and automated data-driven methodology to construct CSO predictive models.

Various field monitored variables can be holistically taken into account by the proposed approach. Little human involvement is needed given the fact that the proposed approach is able to collectively search for the relevant field variables and their time lags associated to a particular CSO model. The model training solutions provided are directly under the global optimization of L_1 regularization, making it more convenient and effective than performing the pairwise correlation analysis for a CSO as previously used. Computational efficiency was also achieved by the proposal of a successive LASSO solution generating scheme without matrix inversions. Given the field investigation where many CSOs were involved, it is evident that most CSOs exhibited a clear interrelationship with other CSOs and field variables, in addition to the relationship with the rainfall data as previously studied. Experimental results showed that the proposed approach was able to automatically construct CSO predictive models with good generalization capability. For some CSOs with good spatial availability and quality of surrounding data, the prediction accuracy can be extraordinarily high, where more than 90% ($R2$ value) of the underlying CSO behavior (variation) has been predicted by the model. The superiority of the proposed approach in terms of computational efficiency and model generalization performance was also illustrated in comparison with neural networks and fuzzy models. Furthermore, in addition to single-step ahead predictions (i.e., 15 min), multistep ahead predictions were examined to demonstrate the promising potential of the proposed methodology though the accuracy decreases as the number of prediction steps increases. Such CSO predictive models are easily constructed and run online, by simply connecting a computing platform to the network's existing sensing framework. It can then be used to provide decision support to network operators as discussed in Section V-C, in order to alleviate the impact of unexpected CSO spillages.

Given this is a data-driven approach, it is worth highlighting that the model construction requires availability and access to reliable field monitoring data which can present a substantial effort and challenge to network operators. Despite the regulatory and operational requirements, it is also recognized that the quality and availability of sensing data in drainage networks can vary with different sites and network operators. With the overall improvement of the situation in the future, the chance of providing better CSO predictions using the proposed data-driven methodology can be correspondingly increased. Moreover, it is known that the use of gray or green infrastructure (e.g., storage and attenuation) can reduce the amount of CSO spills through moving the stormwater runoff outside the catchment or absorbing/leveraging the runoff across the catchment (utilizing natural cycles and ecological systems). The data-driven model is able to predict CSO statuses in catchments with existing gray or green infrastructure as the model is constructed to learn such particular catchment and network behaviors. However, on the other hand, it cannot be used to predict the potential benefit (effect) of using an envisaged gray/green infrastructure though this particular problem is outside the scope of this paper. The same conclusion can also be drawn on other options of stormwater management, e.g., the model can be trained to predict the CSO behavior in the near future but cannot deduce the effect in the planning phase. In that respect, our proposed model addresses a specific use case aiming at the prediction of future outcomes, as opposed to the testing/assessing of hypotheses. Finally, future work within this paper will involve increasing prediction time-steps and conducting online model learning while also considering acceptable model accuracies by leveraging the advanced model and algorithm development, in order to allow more response time for network operators to react with remedial actions.

ACKNOWLEDGMENT

The authors would like to thank the editors, the reviewers, and Welsh Water for their constructive comments for improving the quality of this paper.

REFERENCES

[1] A. Montserrat, L. Bosch, M. A. Kiser, M. Poch, and L. Corominas, "Using data from monitoring combined sewer overflows to assess, improve, and maintain combined sewer systems," *Sci. Total Environ.*, vol. 505, pp. 1053–1061, Feb. 2015.

[2] B. Joseph-Duran, C. Ocampo-Martinez, and G. Cembrano, "Output-feedback control of combined sewer networks through receding horizon control with moving horizon estimation," *Water Resources Res.*, vol. 51, no. 10, pp. 8129–8145, Oct. 2015.

[3] W. Zhao, T. H. Beach, and Y. Rezgui, "Optimization of potable water distribution and wastewater collection networks: A systematic review and future research directions," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 5, pp. 659–681, May 2016.

[4] *Combined Sewer Overflow Management Fact Sheet (Sewer Separation)*, document EPA 832-F-99-041, United States Environ. Protect. Agency, Washington, DC, USA, Sep. 1999.

[5] L. García *et al.*, "Modeling and real-time control of urban drainage systems: A review," *Adv. Water Resources*, vol. 85, pp. 120–132, Nov. 2015.

[6] H. Baek, J. Ryu, J. Oh, and T.-H. Kim, "Optimal design of multi-storage network for combined sewer overflow management using a diversity-guided, cyclic-networking particle swarm optimizer—A case study in the Gunja subcatchment area, Korea," *Expert Syst. Appl.*, vol. 42, no. 20, pp. 6966–6975, Nov. 2015.

[7] A.-S. Madoux-Humery *et al.*, "The effects of combined sewer overflow events on riverine sources of drinking water," *Water Res.*, vol. 92, pp. 218–227, Apr. 2016.

[8] Dŵr Cymru Welsh Water (DCWW). *What is a Combined Sewer Overflow?* Accessed on Aug. 9, 2017. [Online]. Available: https://www.dwrcymru.com/_library/leaflets_publications_english/combined_sewer_overflow.pdf

[9] C. Henriques *et al.*, "The future water environment—Using scenarios to explore the significant water management challenges in England and Wales to 2050," *Sci. Total Environ.*, vols. 512–513, pp. 381–396, Apr. 2015.

[10] J. Gasperi *et al.*, "Priority pollutants in urban stormwater: Part 2—Case of combined sewers," *Water Res.*, vol. 46, no. 20, pp. 6693–6703, Dec. 2012.

[11] I. Jalliffier-Verne *et al.*, "Cumulative effects of fecal contamination from combined sewer overflows: Management for source water protection," *J. Environ. Manag.*, vol. 174, pp. 62–70, Jun. 2016.

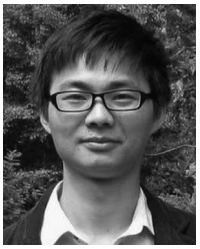
[12] U.K. Environmental Law Association. *Types of Flooding*. Accessed on Aug. 9, 2017. [Online]. Available: http://www.environmentlaw.org.uk/rte.asp?id=100

[13] D. Sempere-Torres, C. Corral, J. Raso, and P. Malgrat, "Use of weather radar for combined sewer overflows monitoring and control," *J. Environ. Eng.*, vol. 125, no. 4, pp. 372–380, Apr. 1999.

[14] L. S. Nanía, A. S. León, and M. H. García, "Hydrologic-hydraulic model for simulating dual drainage and flooding in urban areas: Application to a catchment in the metropolitan area of Chicago," *J. Hydrol. Eng.*, vol. 20, no. 5, May 2015, Art. no. 04014071.

[15] V. M. Morales, J. M. Mier, and M. H. Garcia, "Innovative modeling framework for combined sewer overflows prediction," *Urban Water J.*, vol. 14, no. 1, pp. 97–111, 2017.

[16] N. A. Mancipe-Munoz, S. G. Buchberger, M. T. Suidan, and T. Lu, "Calibration of rainfall-runoff model in urban watersheds for stormwater management assessment," *J. Water Resources Plan. Manag.*, vol. 140, no. 6, Jun. 2014, Art. no. 05014001.
[17] A. N. A. Schellart, W. J. Shepherd, and A. J. Saul, "Influence of rainfall estimation error and spatial variability on sewer flow prediction at a small urban scale," *Adv. Water Resources*, vol. 45, pp. 65–75, Sep. 2012.
[18] Innovyze. *InfoWorks ICM*. Accessed on Aug. 9, 2017. [Online]. Available: http://www.innovyze.com/products/infoworks_icm/
[19] U.S. Environmental Protection Agency. *Storm Water Management Model (SWMM)*. Accessed on Aug. 9, 2017. [Online]. Available: https://www.epa.gov/water-research/storm-water-management-model-swmm
[20] Danish Hydraulic Institute. *Modelling Urban Drainage Systems With MOUSE*. Accessed on Aug. 9, 2017. [Online]. Available: http://www.crwr.utexas.edu/gis/gishyd98/dhi/mouse/mousmain.htm
[21] V. Puig *et al.*, "Predictive optimal control of sewer networks using CORAL tool: Application to Riera Blanca catchment in Barcelona," *Water Sci. Technol.*, vol. 60, no. 4, pp. 869–878, Apr. 2009.
[22] C. Ocampo-Martinez, V. Puig, G. Cembrano, and J. Quevedo, "Application of predictive control strategies to the management of complex networks in the urban water cycle," *IEEE Control Syst.*, vol. 33, no. 1, pp. 15–41, Feb. 2013.
[23] A. K. Fernando, X. Zhang, and P. F. Kinley, "Combined sewer overflow forecasting with feed-forward back-propagation artificial neural network," *Enformatika Trans. Eng. Comput. Technol.*, vol. 12, pp. 58–64, Mar. 2006.
[24] A. Kurth, A. Saul, S. Mounce, W. Shepherd, and D. Hanson, "Application of artificial neural networks (ANNs) for the prediction of CSO discharges," in *Proc. 11th Int. Conf. Urban Drain.*, Edinburgh, U.K., 2008.
[25] S. R. Mounce, W. Shepherd, G. Sailor, J. Shucksmith, and A. J. Saul, "Predicting combined sewer overflows chamber depth using artificial neural networks with rainfall radar data," *Water Sci. Technol.*, vol. 69, no. 6, pp. 1326–1333, Mar. 2014.
[26] A. Berne, G. Delrieu, J.-D. Creutin, and C. Obled, "Temporal and spatial resolution of rainfall measurements required for urban hydrology," *J. Hydrol.*, vol. 299, nos. 3–4, pp. 166–179, Dec. 2004.
[27] C. Charlton-Perez, H. L. Cloke, and A. Ghelli, "Rainfall: High-resolution observation and prediction," *Meteorol. Appl.*, vol. 22, no. 1, pp. 1–127, Jan. 2015.
[28] E. M. A. M. Mendes and S. A. Billings, "An alternative solution to the model structure selection problem," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 31, no. 6, pp. 597–608, Nov. 2001.
[29] S. A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*. Chichester, U.K.: Wiley, 2013.
[30] A. A. Adeniran and S. E. Ferik, "Modeling and identification of nonlinear systems: A review of the multimodel approach—Part 1," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 7, pp. 1149–1159, Jul. 2017.
[31] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Soc. B*, vol. 58, no. 1, pp. 267–288, 1996.
[32] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *J. Stat. Softw.*, vol. 33, no. 1, pp. 1–22, Jan. 2010.
[33] M. McClelland and M. Campbell, "Probabilistic modeling of anticipation in human controllers," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 4, pp. 886–900, Jul. 2013.
[34] J. D. Lee, D. L. Sun, Y. Sun, and J. E. Taylor, "Exact post-selection inference, with application to the lasso," *Ann. Stat.*, vol. 44, no. 3, pp. 907–927, 2016.
[35] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer-Verlag, 2009.
[36] C. Xiao and W. A. Chaovalitwongse, "Optimization models for feature selection of decomposed nearest neighbor," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 2, pp. 177–184, Feb. 2016.
[37] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2009.
[38] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Ann. Stat.*, vol. 32, no. 2, pp. 407–451, 2004.
[39] W. Zhao, T. H. Beach, and Y. Rezgui, "Efficient least angle regression for identification of linear-in-the-parameters models," *Proc. Roy. Soc. A*, vol. 473, no. 2198, 2017, Art. no. 20160775.
[40] K. Li, J.-X. Peng, and G. W. Irwin, "A fast nonlinear model identification method," *IEEE Trans. Autom. Control*, vol. 50, no. 8, pp. 1211–1216, Aug. 2005.
[41] K. Li, J.-X. Peng, and E.-W. Bai, "A two-stage algorithm for identification of nonlinear dynamic systems," *Automatica*, vol. 42, no. 7, pp. 1189–1197, Jul. 2006.
[42] K. Aho, D. Derryberry, and T. Peterson, "Model selection for ecologists: The worldviews of AIC and BIC," *Ecology*, vol. 95, no. 3, pp. 631–636, Mar. 2014.
[43] K. Sjöstrand, *MATLAB Implementation of LASSO, LARS, the Elastic Net and SPCA*. Lyngby, Denmark: Informat. Math. Model., Jun. 2005. [Online]. Available: http://www2.imm.dtu.dk/pubdb/p.php?3897
[44] B. Krawczyk, "Learning from imbalanced data: Open challenges and future directions," *Progress Artif. Intell.*, vol. 5, no. 4, pp. 221–232, Nov. 2016.



Wanqing Zhao (M'13) received the B.Eng. degree in automation from Anhui Polytechnic University, Anhui, China, in 2006, the M.Eng. degree in control theory and control engineering from Shanghai University, Shanghai, China, in 2009, and the Ph.D. degree from the Intelligent Systems and Control Group, Queen's University Belfast, Belfast, U.K., in 2012.

He is currently a Research Fellow with the School of Engineering, Cardiff University, Cardiff, U.K. He was a Research Associate with the Department of Computer Science, Loughborough University, Loughborough, U.K. His current research interests include machine learning, computational intelligence, automatic control, autonomous systems, water resource management, and built environment resilience.

Thomas H. Beach received the Ph.D. degree in computer science from Cardiff University, Cardiff, U.K., in 2011.

He is currently a Lecturer of construction informatics with Cardiff University. His current research interests include application of computing technologies to the built environment, specification and implementation of building/district/city data storage, the Internet of Things and its application to the monitoring and control of the built environment, data analytics covering machine learning and artificial intelligence, application of cloud/distributed computing to data storage and processing for built environment applications, and the semantics of data within the built environment.

Yacine Rezgui received the M.Sc. degree from University Paris 6, Paris, France, in 1991, and the Ph.D. degree from Ecole Nationale des Ponts et Chaussees, Paris, in 1994, both in construction informatics.

He is a Professor of building systems and informatics with Cardiff University, Cardiff, U.K., and the Founding Director of the Building Research Establishment (BRE) Centre for Sustainable Engineering, Watford, U.K., sponsored by the BRE. In 1995, he joined Salford University, Lancashire, U.K., as a Research Fellow, where he was a Lecturer in 1996, a Senior Lecturer in 1998, a Professor in 2001, and the Founding Director of the 5* (RAE 2001) rated Informatics Research Institute, from 2003 to 2007. He has successfully completed over 40 research and development projects at national (U.K. Engineering and Physical Sciences Research Council and Technology Strategy Board) and international (European Framework Programs 5–7) levels. His current research interests include informatics, covering ontology engineering and artificial intelligence applied to the built environment. He has over 150 refereed publications in the above areas, which appeared in international journals, such as the IEEE TRANSACTIONS ON SERVICES COMPUTING, *Information Sciences*, *Data and Knowledge Engineering*, and *Computer-Aided Design*.